

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК 004.6

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«__» _____ 2025 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою
«Інтегровані інформаційні системи»**

зі спеціальності 126 «Інформаційні системи та технології»

**на тему: «Інформаційна система збору, аналізу та прогнозування
даних про витрати місцевих бюджетів»**

Виконала:

студентка 2 курсу, групи ІА-42мп
Лемешова Софія Сергіївна _____

Керівник:

доцент каф. ІСТ, к.т.н., доц.
Ульяницька Ксенія Олександрівна _____

Рецензент:

доцент каф. ОТ, к.т.н., доц.
Волокита Артем Миколайович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.
Студентка _____

Київ — 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти — другий (магістерський)

Спеціальність — 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Лемешовій Софії Сергіївні

1. Тема дисертації «Інформаційна система збору, аналізу та прогнозування даних про витрати місцевих бюджетів», науковий керівник дисертації Ульяницька Ксенія Олександрівна, к.т.н., доц., затверджені наказом по університету від «06» 11 2025 р. № 4841-с
2. Термін подання студентом дисертації «15» 12 2025 р.
3. Об'єкт дослідження: процеси формування, використання та аналітичного оцінювання місцевих бюджетів, які мають важливе значення для розвитку територіальних громад, прозорості управління фінансами та підвищення ефективності державного управління.
4. Вихідні дані: зведені таблиці з основними фінансовими показниками за вибраний період; інтерактивні графіки, гістограми, кругові діаграми розподілу витрат; порівняльні звіти за роками; результати прогнозів на наступні бюджетні періоди; аналітичні звіти, доступні для завантаження.
5. Перелік завдань, які потрібно розробити: дослідити предметну область, провести аналіз існуючих рішень, визначити вимоги до системи (функціональні та технічні), обрати методики прогнозування, створити інфологічну та датологічну модель, обрати технології для розробки, розробити інформаційну систему, провести тестування системи.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: діаграма прецедентів, структурна схема, діаграма компонентів, діаграма розгортання, інфологічна модель, діаграма класів, діаграма послідовності, діаграма діяльності.

7. Орієнтовний перелік публікацій: не заплановано

8. Дата видачі завдання 01.09.2025 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Дослідження предметної області	01.09.2025-07.09.2025	
2.	Огляд та аналіз існуючих рішень	01.09.2025-07.09.2025	
3.	Визначення цілей та задач розробки, вхідних та вихідних даних, структури масивів інформації	01.09.2025-14.09.2025	
4.	Визначення вимог до системи	08.09.2025-14.09.2025	
5.	Дослідження та вибір методик прогнозування даних	08.09.2025-21.09.2025	
6.	Створення сценаріїв використання системи та структурної схеми системи	15.09.2025-21.09.2025	
7.	Вибір технологій для розробки	15.09.2025-21.09.2025	
8.	Створення ER-діаграми	22.09.2025-28.09.2025	
9.	Опис структури бази даних	22.09.2025-28.09.2025	
10.	Проектування елементів системи	22.09.2025-12.10.2025	
11.	Розробка системи	12.10.2025-01.12.2025	
12.	Тестування системи	01.11.2025-01.12.2025	
13.	Підготовка пояснювальної записки	10.11.2025-07.12.2025	

Студентка

Софія ЛЕМЄШОВА

Науковий керівник

Ксенія УЛЬЯНИЦЬКА

РЕФЕРАТ

Інформаційна система збору, аналізу та прогнозування даних про витрати місцевих бюджетів: 140 с., 29 табл., 27 рис., 9 дод., 19 джерел.

АНАЛІТИЧНА СИСТЕМА, ПРОГНОЗУВАННЯ, ВЕБ-ЗАСТОСУНОК, МІСЦЕВИЙ БЮДЖЕТ, МАТЕМАТИЧНІ МОДЕЛІ, ФІНАНСОВА АНАЛІТИКА, OPENBUDGET API.

Актуальність теми зумовлена критичною потребою у підвищенні прозорості, структурованості та доступності публічної фінансової інформації в Україні, особливо в контексті децентралізації та розвитку цифрового врядування. Відсутність сучасних інструментів для обробки та прогнозування бюджетних показників створює практичну проблему, вирішення якої є важливим для розвитку сфери аналітики публічних фінансів та цифрової демократії.

Метою роботи є створення веб-орієнтованої інформаційної системи, здатної забезпечувати автоматизований збір, обробку, аналіз та прогнозування витрат місцевих бюджетів України на основі відкритих джерел даних.

Об'єктом дослідження є процеси формування, використання та аналітичного опрацювання даних місцевих бюджетів. Предметом дослідження є методи організації та реалізації інформаційної системи збору, аналізу й прогнозування бюджетних даних, включно з методами математичної обробки та засобами візуалізації. Методами дослідження є аналіз відкритих джерел даних, методи об'єктно-орієнтованого та структурного проєктування, використання математичних моделей прогнозування часових рядів.

Практичне значення результатів полягає у можливості застосування створеної системи органами місцевого самоврядування, громадськими організаціями, журналістами, дослідниками та громадянами для моніторингу витрат бюджетних коштів, оцінювання ефективності управління фінансами, формування аналітичних та прогнозних звітів. Система може бути основою для подальшого розширення сервісів відкритих даних та інструментів підтримки управлінських рішень на місцевому рівні.

ABSTRACT

Information system for collecting, analyzing, and forecasting data on local budget expenditures: 140 p., 29 tab., 27 draw., 9 app., 19 sources.

ANALYTICAL SYSTEM, FORECASTING, WEB APPLICATION, LOCAL BUDGET, MATHEMATICAL MODELS, FINANCIAL ANALYTICS, OPENBUDGET API.

The relevance of the topic is determined by the critical need to enhance the transparency, structure, and accessibility of public financial information in Ukraine, especially in the context of decentralization and the development of digital governance. The lack of modern tools for integrating, processing, and forecasting budget indicators creates both a scientific and practical problem, the solution of which is essential for the advancement of information systems, public finance analytics, and digital democracy.

The aim of the work is to develop a web-oriented information system capable of ensuring automated collection, processing, analysis, and forecasting of local budget expenditures of Ukraine based on open data sources.

The object of the study is the processes of formation, utilization, and analytical processing of local budget data. The subject of the study is the methods of organizing and implementing an information system for collecting, analyzing, and forecasting budget data, including mathematical processing methods and means of visualization. The research methods include the analysis of open data sources, methods of object-oriented and structural design, as well as the use of mathematical models for time-series forecasting.

The practical significance of the results lies in the possibility of applying the developed system by local self-government bodies, civil society organizations, journalists, researchers, and citizens for monitoring budget expenditures, evaluating the efficiency of financial management, and generating analytical and forecast reports. The system can serve as a foundation for further expansion of open data services and decision-support tools at the local level.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП.....	11
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	13
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	15
2.1 Портал відкритих даних data.gov.ua	15
2.2 Єдиний вебпортал використання публічних коштів spending.gov.ua	16
2.3 Інформаційна система “Open Budget”	17
2.4 Громадські ініціативи та незалежні аналітичні платформи	18
Висновки до розділу 2	19
3 ВИЗНАЧЕННЯ ВИМОГ ДО СИСТЕМИ	20
3.1 Визначення функціональних вимог	20
3.2 Визначення технічних вимог	21
3.3 Сценарії використання системи	21
3.4 Діаграма прецедентів	23
3.5 Визначення вхідних та вихідних даних	24
3.5.1 Вхідні дані	25
3.5.2 Вихідні дані	25
3.6 Опис процесу обробки даних	26
Висновки до розділу 3	26
4 ВИБІР МЕТОДІВ ПРОГНОЗУВАННЯ ТА МАТЕМАТИЧНИХ МОДЕЛЕЙ	28
4.1 Метод середнього арифметичного приросту	28
4.2 Метод ковзного середнього	29
4.3 Метод експоненційного згладжування	29
4.4 Лінійна регресійна модель	30
Висновки до розділу 4	31
5 СТРУКТУРНА СХЕМА СИСТЕМИ	32
5.1 Структурна схема системи	32
5.2 Опис взаємодії компонентів	33

5.3 Діаграма компонентів	33
Висновки до розділу 5	35
6 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ	36
6.1 Технологія клієнтської частини React	36
6.2 Технологія серверної частини Node.js	36
6.3 Система керування базами даних PostgreSQL	37
6.4 Хмарне сховище файлів Amazon S3 Bucket	38
6.5 Технологія картографічного сервісу Mapbox	38
6.6 Технологія зовнішнього джерела даних API OpenBudget	39
6.7 Контейнеризація за допомогою Docker та Docker Compose	40
6.8 Розгортання системи за допомогою AWS EC2 Instance	41
6.9 Додаткові інструменти та бібліотеки	42
6.10 Діаграма розгортання системи	43
Висновки до розділу 6	44
7 ІНФОЛОГІЧНА ТА ДАТАЛОГІЧНА МОДЕЛЬ	45
7.1 Інфологічна модель	45
7.2 Опис основних таблиць бази даних	47
7.3 Даталогічна модель системи	49
Висновки до розділу 7	52
8 ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ	53
8.1 Клієнт-серверна архітектура	53
8.2 Загальна архітектура системи	55
8.3 Реалізація клієнтської частини	57
8.4 Реалізація серверної частини	60
8.4.1 Структура серверної частини	61
8.4.2 Взаємодія модулів	65
8.5 Діаграма класів	67
8.6 Взаємодія з OpenBudget API	69
8.7 Генерація PDF-звітів	70
8.8 Діаграма послідовності для генерації звітів	72

8.9 Діаграма діяльності.....	73
Висновки до розділу 8	76
9 ОПИС ІНТЕРФЕЙСУ КОРИСТУВАЧА.....	78
Висновки до розділу 9	90
10 СТАРТАП ПРОЄКТ	91
10.1 Опис ідеї проєкту	91
10.2 Технологічний аудит ідеї проєкту	96
10.4 Аналіз ринкових можливостей запуску стартап-проєкту	97
10.4 Розроблення ринкової стратегії проєкту	113
10.5 Розроблення маркетингової програми стартап-проєкту	120
Висновки до розділу 10	127
ВИСНОВКИ	128
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	130
ДОДАТОК А	132
ДОДАТОК Б	133
ДОДАТОК В	134
ДОДАТОК Г	135
ДОДАТОК Д	136
ДОДАТОК Е	137
ДОДАТОК Ж	138
ДОДАТОК И	139
ДОДАТОК К	140

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

БД — База даних

ІС — Інформаційна система

СУБД — Система управління базами даних

API — Application Programming Interface — набір правил і методів, що визначають спосіб взаємодії програмних компонентів між собою

AWS — Amazon Web Services — хмарна платформа, що надає інфраструктурні та програмні сервіси для розгортання застосунків

CSV — Comma-Separated Values — текстовий формат даних, у якому значення розділяються комами або іншим символом

CSS — Cascading Style Sheets — мова опису стилів, що визначає зовнішній вигляд веб-сторінок

DB — Database — організоване сховище структурованих даних

DOM — Document Object Model — модель представлення HTML- та XML-документів у вигляді дерева об'єктів

EC2 — Elastic Compute Cloud — сервіс AWS для створення віртуальних серверів у хмарі

ETL — Extract, Transform, Load — процес вилучення, перетворення та завантаження даних у сховище

HTTP — HyperText Transfer Protocol — протокол передавання даних між веб-клієнтом і веб-сервером

JSON — JavaScript Object Notation — формат обміну даними, що використовується для передачі структурованої інформації

JWT — JSON Web Token — токен для безпечної автентифікації користувача, що містить зашифровані дані про сесію

PDF — Portable Document Format — формат електронних документів, придатний для збереження звітів і друкованих матеріалів

REST — Representational State Transfer — архітектурний стиль побудови вебсервісів, що визначає принципи обміну даними

S3 — Simple Storage Service — сервіс AWS для зберігання файлів у хмарних об'єктах

SQL — Structured Query Language — мова запитів для управління даними у реляційних базах

UI — User Interface — інтерфейс взаємодії користувача з програмним забезпеченням

UML — Unified Modeling Language — уніфікована мова моделювання для опису структури та поведінки систем

URL — Uniform Resource Locator — уніфікований локатор ресурсу, що визначає адресу веб-сторінки або файлу

XML — Extensible Markup Language — розширювана мова розмітки для зберігання та передачі структурованих даних

ВСТУП

Стрімкий розвиток інформаційних технологій в Україні суттєво вплинув на всі сфери суспільного життя, зокрема на державне управління та економічне планування. У сучасних умовах відкритості даних та діджиталізації процесів зростає потреба у створенні зручних і зрозумілих інструментів, які дозволяють ефективно працювати з великою кількістю публічної фінансової інформації. Особливої актуальності набуває проблема прозорості місцевих бюджетів, адже саме громади є основою децентралізованої системи управління державними коштами.

Бюджети територіальних громад формуються з різних джерел доходів і спрямовуються на реалізацію місцевих програм — освіти, медицини, інфраструктури, соціального захисту. Водночас процес контролю за ефективністю витрат часто є складним для звичайного користувача, оскільки дані подаються у неструктурованому або важкодоступному форматі. Для підвищення прозорості та зручності аналізу необхідно створювати сучасні інформаційні системи, які здатні збирати, обробляти та наочно відображати дані про використання бюджетних коштів. В даній роботі описано розробку веб-застосунку «Інформаційна система збору, аналізу та прогнозування даних про витрати місцевих бюджетів»

Об'єктом дослідження є процеси формування, використання та аналітичного оцінювання місцевих бюджетів України, які мають важливе значення для розвитку територіальних громад, прозорості управління фінансами та підвищення ефективності державного управління.

Предметом дослідження є функціональні можливості веб-застосунку, методи збору та систематизації відкритих бюджетних даних, математичні методи аналізу та прогнозування фінансових показників, а також підходи до візуалізації результатів у зрозумілому та інтерактивному форматі.

Метою дослідження є розробка веб-орієнтованої інформаційної системи, яка забезпечує автоматизований збір, обробку, аналітику та прогнозування даних про витрати місцевих бюджетів. Система має на меті підвищити прозорість

бюджетних процесів, спростити доступ громадян до фінансової інформації та сприяти підвищенню ефективності використання бюджетних коштів шляхом математичного аналізу й візуалізації ключових показників.

Задля досягнення поставленої мети необхідно вирішити такі завдання:

- актуальність проблеми, яка зумовила вибір теми роботи;
- проаналізувати предметну область фінансової аналітики місцевих бюджетів та оцінити актуальність теми;
- здійснити огляд існуючих інформаційних систем і сервісів відкритих бюджетних даних, визначити їхні переваги та недоліки;
- визначити цілі, вхідні та вихідні дані, а також структуру інформаційних масивів;
- сформулювати вимоги до функціональності, продуктивності та безпеки системи;
- дослідити та обрати методики прогнозування бюджетних показників, що будуть реалізовані в системі;
- створити сценарії використання та структурну схему системи;
- обрати технології для розробки;
- спроектувати структуру бази даних та створити ER-діаграму;
- розробити серверну та клієнтську частини системи;
- описати процес діяльності системи, механізми збору та обробки даних;
- провести тестування розробленого застосунку й оцінити точність аналітичних та прогнозних результатів.

Розроблена інформаційна система має практичне значення для органів місцевого самоврядування, аналітичних центрів, журналістів та громадських організацій, які здійснюють моніторинг витрат бюджетних коштів. Вона забезпечує можливість оперативного аналізу даних, виявлення неефективних витрат, формування аналітичних звітів і прогнозів, а також сприяє підвищенню прозорості управління фінансовими ресурсами в Україні.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

У процесі дослідження предметної області було розглянуто особливості функціонування системи місцевих бюджетів України, а також проблеми, пов'язані з відкритістю, структурованістю та аналітикою бюджетних даних. Бюджет є основним фінансовим інструментом держави, що визначає напрями використання коштів для забезпечення соціально-економічного розвитку. Місцеві бюджети, у свою чергу, є фінансовою основою діяльності територіальних громад і виконують ключову роль у реалізації регіональних програм, наданні адміністративних послуг, фінансуванні освіти, медицини, соціального захисту, інфраструктури та інших сфер життєдіяльності населення.

Процес формування місцевих бюджетів регулюється Бюджетним кодексом України та передбачає кілька етапів: планування, затвердження, виконання та звітування. У межах цих процесів органи місцевого самоврядування подають детальні фінансові документи до Міністерства фінансів, Державної казначейської служби та інших установ. Отримані дані публікуються у вигляді відкритих наборів даних на офіційних порталах, зокрема data.gov.ua, spending.gov.ua та openbudget.gov.ua

Незважаючи на доступність відкритих бюджетних даних, більшість громадян не мають зручного інструменту для їхнього перегляду та аналізу. Дані публікуються у вигляді великих неструктурованих файлів (CSV, XLSX, JSON), що ускладнює їх розуміння, порівняння та застосування для аналітичних цілей. Традиційні методи роботи з такими наборами потребують спеціальних знань у сфері фінансів або програмування, що обмежує коло користувачів і знижує ефективність публічної звітності.

Водночас, в умовах децентралізації управління фінансами, громади отримують більше повноважень у розпорядженні коштами. Це створює нові виклики у сфері контролю, ефективності використання ресурсів та прозорості звітності. Суспільство очікує на впровадження систем, які б дозволяли просто,

швидко та візуально зрозуміло оцінювати бюджетні рішення місцевих органів влади.

Було встановлено, що значна частина існуючих державних інформаційних платформ має технічні або функціональні обмеження. Наприклад, державний портал spending.gov.ua забезпечує доступ до транзакцій, але не пропонує гнучких аналітичних інструментів чи прогнозів. Система openbudget.gov.ua орієнтована переважно на візуалізацію макроекономічних показників, а не на детальну роботу з даними окремих громад. У результаті, користувачі не можуть у зручний спосіб відстежувати витрати в розрізі регіонів, категорій чи конкретних бюджетних статей.

Також було проаналізовано потреби основних цільових груп користувачів — громадських активістів, журналістів, аналітиків і звичайних громадян. Для цих категорій важливо мати інструмент, який дозволяє:

- отримувати дані з різних відкритих джерел в одному місці;
- швидко фільтрувати інформацію за роком, громадою, напрямом витрат;
- переглядати результати у вигляді інтерактивних графіків та діаграм;
- отримувати прогноз бюджетних тенденцій і показників ефективності.

Таким чином, було визначено, що предметна область потребує створення інтегрованої системи, яка об'єднує можливості збору відкритих даних, математичної обробки, побудови прогнозів та зручної візуалізації результатів. Розробка інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів дозволить розв'язати проблему доступності та зрозумілості бюджетної інформації, забезпечить прозорість витрат, сприятиме підвищенню громадського контролю та розвитку цифрової демократії в Україні.

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Під час дослідження предметної області було проаналізовано наявні державні та громадські сервіси, які надають доступ до відкритих бюджетних даних України. Метою аналізу є визначення їхніх функціональних можливостей, переваг, недоліків та потенційних напрямів удосконалення. Це дозволяє сформулювати вимоги до розроблюваної інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів.

2.1 Портал відкритих даних data.gov.ua

Портал data.gov.ua[1] є офіційною державною платформою для публікації відкритих наборів даних, створеною відповідно до Закону України «Про доступ до публічної інформації». На порталі розміщуються тисячі наборів, серед яких — фінансові звіти органів влади, державних підприємств, міністерств і місцевих рад.

Основна перевага цього ресурсу полягає у великій кількості даних, доступних у машинозчитуваному форматі (CSV, JSON, XML), що дозволяє інтегрувати їх до зовнішніх систем. Крім того, платформа підтримує REST API, що робить можливим автоматичне отримання оновлень.

Однак недоліком є відсутність аналітичного інтерфейсу: портал лише надає файли, не пропонуючи вбудованих інструментів для їхньої обробки, фільтрації чи візуалізації. Це створює бар'єр для звичайних користувачів, які не мають навичок роботи з великими обсягами даних. Також часто спостерігається невідповідність форматів даних, дублювання наборів та затримки в оновленні інформації.

Крім того, незважаючи на наявність великої кількості наборів, структура інформації часто є неоднорідною: різні розпорядники публікують дані у власному форматі, що ускладнює подальшу автоматизовану обробку. Для бюджетних показників особливо характерною є неповнота полів, відсутність детальної розбивки або наявність лише агрегованих значень, що знижує можливість

глибокого аналізу. На рисунку 2.1 наведено сторінку порталу з інформацією про доходи та видатки міського бюджету.

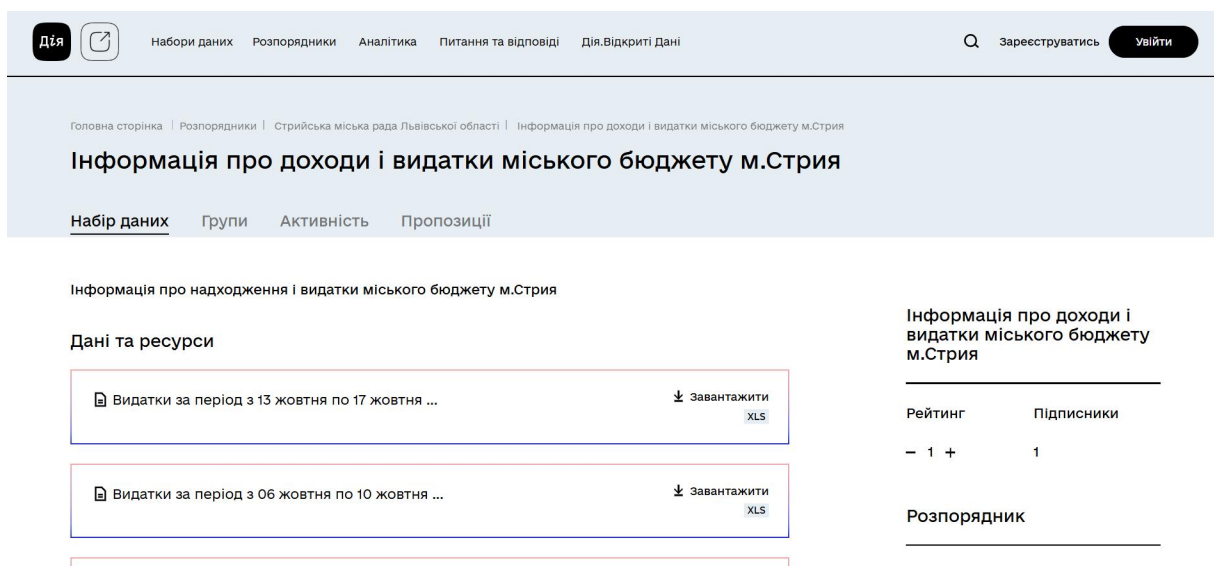


Рисунок 2.1 — Приклад відображення інформації на порталі data.gov.ua

Таким чином, data.gov.ua може виступати джерелом сирих даних для аналітичних систем, але не є готовим інструментом для оцінки чи прогнозування бюджетних показників.

2.2 Єдиний вебпортал використання публічних коштів spending.gov.ua

Портал spending.gov.ua[2] є основним державним ресурсом, що забезпечує прозорість фінансових операцій державних і комунальних установ. Він надає відкритий доступ до інформації про транзакції, пов'язані з використанням публічних коштів, включно з назвами розпорядників, сумами, контрагентами та призначенням платежів.

Основною перевагою порталу є високий рівень деталізації даних. Користувач може переглядати окремі транзакції, фільтрувати їх за розпорядниками, регіонами, датами або кодами бюджетних програм. Це забезпечує базовий рівень аналітики та контроль за витратами бюджетних організацій. На рисунку 2.2 наведено приклад пошуку за транзакціями на порталі.

spending

ГОЛОВНА ПОШУК ЗА ЗОБОВ'ЯЗАННЯМИ ПОШУК ЗА РОЗПОРЯДНИКАМИ ПОШУК ЗА ТРАНСАКЦІЯМИ

Пошук на сторінці

	ДАТА ОПЛАТИ	ПЛАТНИК	ОТРИМУВАЧ	СУМА	ПРИЗНАЧЕННЯ ПЛАТЕЖУ
ДАТА ОПЛАТИ	22.10.2025	22.10.2025	22.10.2025	22.10.2025	22.10.2025
ПЛАТНИК	* Код ЄДРПОУ	* Найменування			
ОТРИМУВАЧ	* Код ЄДРПОУ	* Найменування			
ПРИЗНАЧЕННЯ	Призначення платежу	Додаткова інформація			
СУМА					
	22.10.2025	Рахункова палата 00013540	ТОВ "Автосаміт УКР-АВТО" 14369855	2 324.45 UAH	6511010;2240;Оплата тех.обсл.автозг.дог.101 від 04.03.25;рах.2025021264;акт.2025022382 від 16.10.25;Пост.КМУ590;В т.ч. ПДВ 387,41
	22.10.2025	Четфалівська гімназія ім. Ж. Моріца 26098516	ТОВ"ЕКСПРЕС" 45498242	350.00 UAH	0611021;2240;опл. послуги з надання доступу до мережі Інтернет(послуги провайдерів);зг.дог. №078/27513 від 07.02.2025р.;та зг. акту №27513/09від 20.10.2025р.;в т.ч. ПДВ-58,33
	22.10.2025	Головський психоневрологічний інтернат 03188205	ТОВ "КІВХЛІБ" 44375369	1 388.28 UAH	0813102;2230;за хліб пшеничн.,житній ,батон ,зг.накл. №065 від 20.10.2025р договір№1 від 24.12.2024;в т.ч. ПДВ-231,38.
	22.10.2025	ХСМФХ ХОР 02011657	ТОВ "СКАЙ СОФТ" 32654545	3 178.80 UAH	0711101;2282(2273); електроенергія за вересень 2025р;зг. рах№2580 від 17,10,25 акт№2558 від 17,10,25 дог.№1 від 17,01,25; ПДВ-529,80.
	22.10.2025	ІФН НАНУ 05416952	ГУ ДПС у м.Києві 44116011	456 177.12 UAH	*101;05416952;101 2281 6541030 єдиний внесок,нарахований на суму зарплати за жовтень 2025р.(101 2281 6541030 єдиний внесок,нарахований на суму зарплати за жовтень 2025р.)
	22.10.2025	Інститут гідробіології НАН України 05417029	ПАТ "АК "Київводоканал" 03327629	3 203.50 UAH	6541030;2281; сплата за водовідвед. за 10.25; зг. дог. 26392/1-5-05-К від 29.01.25, акт КВК00029491025 від 20.10.25; в т.ч. ПДВ-533,92
	22.10.2025	ПП "ХЕРСОНТЕПЛОЕНЕРГО" 05417029	АТ "ХерсонОблЕнерго" 05417029	15 861.38 UAH	xxxx Плата за послугу з розподілу електричної енергії згідно рах.№xxxxр від 30.09.2025, акта №9пр від 30.09.2025

Рисунок 2.2 — Пошук за транзакціями на порталі Spending

Попри це, портал не дозволяє виконувати глибоку аналітику, порівняльні оцінки або побудову прогнозів. Дані представлені у вигляді таблиць без можливості інтерактивної візуалізації чи формування динамічних звітів. Для аналізу великих обсягів інформації користувачеві необхідно самостійно завантажувати та обробляти файли.

Таким чином, spending.gov.ua виконує функцію прозорого джерела даних, але не є повноцінною аналітичною системою. Вона потребує розширення у напрямі зручного подання, візуалізації та математичної обробки інформації.

2.3 Інформаційна система “Open Budget”

Інформаційна система “Open Budget”[3] створена для візуалізації державного та місцевих бюджетів. Платформа відображає основні фінансові показники у вигляді графіків і діаграм, що робить її більш зручною для пересічних користувачів.

Серед переваг системи — зрозумілий інтерфейс, можливість перегляду доходів і видатків у різних розрізах, доступ до статистичних зведень по регіонах, а також публікація офіційних бюджетних документів. Користувацький інтерфейс системи наведено на рисунку 2.3.

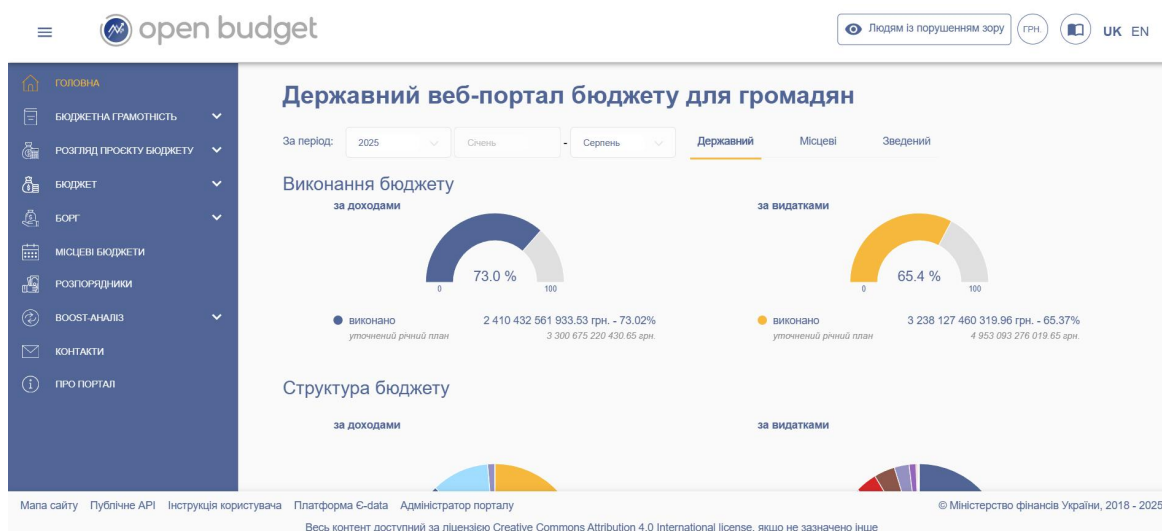


Рисунок 2.3 — Головна сторінка системи “Open Budget”

Втім, система має низку обмежень. Вона орієнтована здебільшого на огляд макрорівня, тобто відображення зведених показників державного бюджету або окремих регіонів без детального розподілу за громадами, проєктами чи статтями. Відсутні інструменти прогнозування, аналітичні розрахунки чи порівняння між громадами. Крім того, користувач не може отримати персоналізовану вибірку або налаштувати власні показники ефективності.

Таким чином, openbudget.gov.ua частково реалізує функцію візуалізації фінансових даних, проте не забезпечує глибокого аналізу, обчислень та інтелектуальної обробки інформації, що є важливим для сучасних аналітичних систем.

2.4 Громадські ініціативи та незалежні аналітичні платформи

Крім державних систем, існують громадські ініціативи, спрямовані на моніторинг бюджетної прозорості. Серед них — «Texty.org.ua», «DoZorro», «YouControl» та інші платформи, що обробляють відкриті дані, зокрема у сфері закупівель, фінансів та діяльності місцевої влади.

Такі сервіси мають перевагу у швидкості оновлення та зручності візуалізації даних, однак переважно фокусуються на окремих аспектах, наприклад, державних

закупівлях або ризиках у фінансовій звітності. Вони не забезпечують цілісної картини витрат місцевих бюджетів і не реалізують прогнозних моделей.

Крім того, більшість подібних проєктів не є відкритими з технічної точки зору — користувачі не мають доступу до сирих даних через API або експорту. Це обмежує можливості повторного використання інформації для досліджень та розробки нових аналітичних інструментів.

Висновки до розділу 2

У результаті проведеного аналізу було встановлено, що наявні рішення забезпечують лише часткову реалізацію завдань аналітики місцевих бюджетів. Вони або зосереджені на публікації даних без подальшої обробки, або надають лише базову візуалізацію без аналітичних і прогнозних функцій.

Тому розробка нової інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів є доцільною та актуальною. Вона поєднує переваги автоматизованого збору відкритих даних, математичної обробки та прогнозування фінансових показників, а також сучасного веб-інтерфейсу для зручної взаємодії користувача з аналітичною інформацією.

3 ВИЗНАЧЕННЯ ВИМОГ ДО СИСТЕМИ

Метою розробки є створення веб-орієнтованої інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів України. Система повинна забезпечувати автоматичне отримання відкритих даних із державних джерел, їх обробку, аналітичне подання у вигляді графіків і таблиць, а також формування прогнозів майбутніх бюджетних показників.

Основними завданнями системи є забезпечення зручного доступу до фінансової інформації для користувачів, створення інтерактивного інтерфейсу для перегляду, аналізу та порівняння бюджетів різних громад, а також використання математичних методів для побудови прогнозів на основі історичних даних.

3.1 Визначення функціональних вимог

Інформаційна система збору, аналізу та прогнозування даних про витрати місцевих бюджетів має забезпечувати такі функціональні можливості:

- автоматизований збір та оновлення даних із відкритих державних джерел;
- збереження даних у базі даних;
- перегляд зведеної інформації про доходи та витрати місцевих бюджетів за вибраний період;
- можливість фільтрації та сортування даних за регіоном, громадою, роком, типом витрат або бюджетною статтею;
- порівняння показників між громадами або роками для оцінки динаміки змін;
- побудова інтерактивних графіків і діаграм на основі отриманих даних;
- формування аналітичних звітів і статистичних зведень;
- реалізація алгоритмів прогнозування витрат на основі історичних трендів і математичних моделей;
- експорт результатів аналізу у форматах PDF та CSV;

- авторизація користувачів з різними ролями (адміністратор, аналітик, гість);
- збереження завантажених звітів і супровідних документів;
- надання адміністраторам можливості редагувати, оновлювати або видаляти записи та перевіряти коректність даних.

3.2 Визначення технічних вимог

Інформаційна система збору, аналізу та прогнозування даних про витрати місцевих бюджетів має відповідати таким технічним вимогам:

- масштабованість і стабільна робота з великими обсягами даних;
- інтеграція з відкритими державними API для автоматичного оновлення даних;
- збереження великих або неструктурованих файлів;
- автентифікація користувачів із шифруванням паролів та захистом персональних даних;
- механізм кешування для пришвидшення обробки запитів та зниження навантаження на сервер;
- журналювання дій користувачів і помилок системи для спрощення контролю й діагностики;
- інтуїтивно зрозумілий інтерфейс із використанням інтерактивних елементів візуалізації (графіки, діаграми, таблиці);
- можливість подальшого розширення системи без суттєвих змін у її архітектурі.

3.3 Сценарії використання системи

Інформаційна система має три типи користувачів: гість, авторизований користувач (аналітик) та адміністратор. Кожен із них взаємодіє із системою через певний набір сценаріїв використання.

Сценарій 1. Перегляд зведеної інформації (гість). Користувач переходить на головну сторінку системи. На екрані відображається карта, при натисканні на яку відкривається статистика, графіки та діаграми по обраній громаді. Гість може застосувати базові фільтри (за роком або областю), після чого результати оновлюються динамічно. Додатково користувач може ознайомитися з прогнозом витрат бюджету на наступний рік за різними моделями прогнозування.

Сценарій 2. Реєстрація та авторизація (аналітик). Для доступу до розширених функцій користувач проходить реєстрацію, вводячи електронну пошту, пароль та ім'я. Після підтвердження реєстрації він може увійти у систему, отримуючи доступ до персонального кабінету. Авторизація здійснюється через безпечне з'єднання HTTPS із перевіркою пароля шляхом порівняння його хешу у базі даних.

Сценарій 3. Аналітичний запит і завантаження звітів (аналітик). Авторизований користувач обирає громаду, період та тип витрат, після чого система виконує запит до бази даних. На основі отриманої вибірки система будує діаграми, порівняльні таблиці та індикатори ефективності. Користувач може налаштовувати відображення результатів, додавати власні коментарі та зберігати створений аналітичний звіт у своєму профілі.

Сценарій 4. Формування прогнозу. Користувач обирає метод прогнозування (ковзне середнє, експоненційне згладжування або лінійну регресію) для обраної громади. Система виконує розрахунки на основі історичних даних та генерує прогноз із зазначенням довірчого інтервалу. Результати виводяться у вигляді графіка.

Сценарій 5. Управління даними (адміністратор). Адміністратор може переглядати усі наявні записи у базі даних, оновлювати дані з державних джерел, видаляти некоректні записи та контролювати зв'язки між таблицями. Він також має можливість переглядати журнал дій користувачів, журнал історії синхронізації з державними джерелами, а також запускати примусову синхронізацію.

Сценарій 6. Управління користувачами (адміністратор). Адміністратор може оновлювати дані користувачів, змінювати ролі або блокувати доступ у випадку порушень політики безпеки.

3.4 Діаграма прецедентів

Для кращого розуміння функціоналу інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів було створено діаграму прецедентів, яка описує очікувану поведінку системи та її функціональні можливості.

Діаграма прецедентів[4] — це тип UML-діаграми, який моделює взаємодію користувачів із системою. Вона дозволяє візуалізувати, які саме дії можуть виконувати користувачі, та визначити основні функціональні вимоги. Діаграма складається з акторів (користувачів), прецедентів (сценаріїв використання) та зв'язків між ними.

На наведеній діаграмі (кресленик в Додатку Б) зображено діаграму прецедентів для інформаційної системи збору, аналізу та прогнозування витрат.

У даній системі виділено три основних актора:

- Гість (неавторизований користувач);
- Авторизований користувач (аналітик);
- Адміністратор системи.

Авторизований користувач наслідує дії гостя, а адміністратор — дії авторизованого користувача, маючи розширені повноваження для управління даними та користувачами.

Варіанти використання системи гостем:

- перегляд загальної інформації про систему та її мету;
- перегляд зведених даних про бюджети у вигляді таблиць або графіків;
- застосування базових фільтрів (за регіоном або роком) та сортування, пошук по витратам;
- прогнозування майбутніх бюджетних витрат на основі історичних даних;

- реєстрація в системі та вхід;
- ознайомлення з аналітичними звітами у відкритому доступі.

Варіанти використання системи авторизованим користувачем:

- усі дії гостя;
- авторизація в системі;
- редагування профілю користувача;
- створення власних запитів до бази даних — примусове оновлення прогнозів;
- формування аналітичних звітів;
- експорт отриманих результатів у форматах PDF або CSV.

Варіанти використання системи адміністратором:

- усі дії гостя;
- усі дії авторизованого користувача;
- управління користувачами (створення, редагування, видалення акаунтів);
- перевірка коректності та актуальності даних;
- примусова інтеграція з відкритими джерелами державних даних;
- контроль інтеграції з відкритими державними джерелами даних.

3.5 Визначення вхідних та вихідних даних

У процесі проектування інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів було визначено склад та структуру даних, що надходять до системи, зберігаються у базі даних та відображаються користувачу. Для ефективного функціонування система має підтримувати роботу з великими обсягами фінансової інформації, отриманої з відкритих державних джерел, зокрема порталів spending.gov.ua, openbudget.gov.ua та data.gov.ua.

Вхідні та вихідні дані системи класифікуються за джерелом, призначенням та способом обробки.

3.5.1 Вхідні дані

До вхідних даних належить інформація, яка надходить у систему для подальшої обробки, аналізу або збереження. Основними типами вхідних даних є:

- фінансові показники — дані про доходи та видатки місцевих бюджетів у розрізі статей, категорій, кодів економічної класифікації (КЕКВ) та програмної класифікації видатків (КПКВК)[5];

- дані про громади — офіційні назви, коди, область, кількість населення, географічна прив'язка;

- інформація про бюджетні періоди — рік, квартал або місяць, до якого належить звіт;

- історичні дані — попередні роки, що використовуються для формування динаміки та побудови прогнозних моделей;

- користувацькі дані — облікові записи користувачів, права доступу, персональні налаштування, сформовані запити, завантажені аналітичні звіти;

- адміністративні дані — інформація про дії користувачів, журнал системних подій, параметри інтеграції з API державних порталів.

Отримання вхідних даних здійснюється у двох режимах: автоматичне завантаження через API державних порталів та ручне завантаження адміністраторами або аналітиками.

Перед збереженням усі дані проходять етапи перевірки, нормалізації та уніфікації, що забезпечує узгодженість структури та унеможлиблює дублювання.

3.5.2 Вихідні дані

Вихідні дані системи — це результат обробки та аналітики, який подається користувачу у зручному форматі. До них належать:

- зведені таблиці з основними фінансовими показниками за вибраний період;

- інтерактивні графіки, гістограми, кругові діаграми розподілу витрат;

- порівняльні звіти за громадами або роками;
- результати прогнозів на наступні бюджетні періоди;
- аналітичні звіти, доступні для завантаження;
- повідомлення про результати оновлення даних або помилки при імпорті.

Система забезпечує інтерактивну взаємодію: користувач може застосовувати фільтри, змінювати параметри візуалізації, експортувати результати або зберігати власні налаштування аналітичної панелі.

3.6 Опис процесу обробки даних

Під час роботи системи відбувається послідовність етапів:

- a) збір вхідних даних — з OpenBudget API або через ручне завантаження;
- b) перевірка валідності — перевірка форматів і значень, фільтрація некоректних записів;
- c) збереження в базі — нормалізована структура в PostgreSQL;
- d) аналітична обробка — обчислення статистики;
- e) прогнозування — математичні моделі (за потреби);
- f) генерація звітів — створення PDF з таблицями та графіками;
- g) зберігання результатів — в базі та S3;
- h) надання користувачу — відображення на веб-інтерфейсі та завантаження PDF.

Висновки до розділу 3

У процесі визначення вимог до інформаційної системи було сформовано цілісне бачення функціональних можливостей та технічних характеристик, необхідних для забезпечення повного циклу роботи з бюджетними даними. Було визначено, що система має забезпечувати автоматизований збір відкритих даних, їхнє збереження, обробку та подальшу аналітичну інтерпретацію, включно з формуванням прогнозів майбутніх витрат. На основі аналізу предметної області

та вже існуючих рішень було обґрунтовано набір функціональних можливостей, які є критично важливими для ефективної взаємодії користувачів із даними, зокрема підтримку фільтрації, порівняння, побудови інтерактивних графіків та формування звітів.

У межах розділу було визначено технічні вимоги, спрямовані на забезпечення масштабованості, стабільності роботи з великими обсягами інформації та безпеки даних. Було обрано архітектурний підхід, що дозволяє системі функціонувати з урахуванням високого навантаження, інтеграції з API зовнішніх джерел та можливості подальшого розширення.

Було сформовано сценарії використання системи для різних категорій користувачів – гостей, аналітиків та адміністраторів. Завдяки цьому було визначено повний спектр взаємодії із системою, починаючи від перегляду базових показників і завершуючи виконанням спеціалізованих аналітичних операцій та керування даними. Таке структурування дозволило забезпечити чітке розмежування функцій і створити основу для подальшого проектування інтерфейсу та внутрішніх модулів.

Було також уточнено склад вхідних і вихідних даних, що дало змогу сформулювати вимоги до структури бази даних і до механізмів обробки інформації. Це дозволило закласти фундамент для моделювання архітектури системи та реалізації математичних алгоритмів прогнозування, які використовуватимуться на наступних етапах розробки.

Таким чином, у розділі було сформовано комплексну систему вимог, що визначає функціональну, технічну та інформаційну структуру майбутнього програмного продукту. Отримані результати створюють методологічну основу для подальшого проектування архітектури, обрання технологій та реалізації компонентів інформаційної системи.

4 ВИБІР МЕТОДІВ ПРОГНОЗУВАННЯ ТА МАТЕМАТИЧНИХ МОДЕЛЕЙ

Під час розробки інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів важливе значення має застосування математичних методів, які дозволяють формувати достовірні прогнози на основі історичних фінансових показників. Метою використання прогнозних моделей є отримання оцінки майбутніх витрат для кожної громади та виявлення можливих тенденцій, що допомагають у прийнятті управлінських рішень.

Було проведено аналіз основних підходів до прогнозування економічних показників: регресійних, стохастичних, методів часових рядів та згладжування. На основі дослідження обрано найбільш доцільні для реалізації в рамках веб-системи методики, що забезпечують баланс між точністю, простотою реалізації та швидкістю обчислень.

4.1 Метод середнього арифметичного приросту

Найпростішим методом прогнозування є розрахунок майбутнього показника за допомогою середнього темпу приросту[6] за попередні періоди. Він використовується для формування базового прогнозу, коли зміни витрат мають стабільний тренд.

Середній темп приросту визначається за формулою (4.1):

$$T_{cp} = \frac{1}{n-1} \sum_{i=1}^{n-1} \frac{X_{i+1} - X_i}{X_i}, \quad (4.1)$$

де X_i — значення витрат у році i ;

n — кількість років, що враховуються для прогнозу.

Тоді прогнозне значення витрат на наступний період обчислюється за формулою (4.2):

$$X_{n+1} = X_n * (1 + T_{cp}), \quad (4.2)$$

Цей метод дозволяє отримати початкову оцінку тенденції зміни витрат та використовується як базовий алгоритм для первинного прогнозу в системі.

4.2 Метод ковзного середнього

Метод ковзного середнього[6] є класичним способом згладжування часових рядів. Він дає змогу усунути випадкові коливання та виявити основну тенденцію у зміні бюджетних показників.

Значення прогнозу визначається як середнє арифметичне кількох останніх спостережень, формула (4.3):

$$\hat{X}_{t+1} = \frac{1}{k} \sum_{i=t-k+1}^t X_i, \quad (4.3)$$

де k — кількість періодів, що враховуються у середньому;

X_i — значення витрат у періоді i ;

X_{t+1} — прогнозоване значення на наступний період.

Метод ковзного середнього дозволяє згладити випадкові коливання у бюджетних показниках і є ефективним для короткострокового прогнозування при стабільній динаміці.

4.3 Метод експоненційного згладжування

Для підвищення точності прогнозу при наявності незначних коливань у часових рядах використовується метод експоненційного згладжування[6]. Його особливість полягає у тому, що останнім спостереженням надається більша вага, ніж попереднім.

Розрахунок прогнозного значення виконується за формулою (4.4):

$$F_{t+1} = \alpha * X_t + (1 - \alpha) * F_t, \quad (4.4)$$

де F_{t+1} — прогноз на наступний період;

X_t — фактичне значення у поточному періоді;

α — коефіцієнт згладжування ($0 < \alpha < 1$);

F_t — прогноз на попередній період.

Коефіцієнт α визначає, наскільки швидко прогноз реагує на зміни у даних: чим він більший, тим менше вплив мають старі спостереження. Значення α підбирається експериментально для кожного показника, зазвичай у межах 0.2—0.5.

4.4 Лінійна регресійна модель

Для середньо- та довгострокових прогнозів застосовується лінійна регресійна модель, яка враховує тренд змін у бюджетних витратах. Така модель описується рівнянням (4.5):

$$Y_t = a + b * t, \quad (4.5)$$

де Y_t — прогнозоване значення показника у момент часу t ;

a — початкове значення (вільний член);

b — коефіцієнт тренду, що показує середню зміну показника за одиницю часу.

Коефіцієнти a та b визначаються методом найменших квадратів за формулою (4.6):

$$b = \frac{n \sum t_i Y_i - \sum t_i \sum Y_i}{n \sum t_i^2 - (\sum t_i)^2}, \quad a = \bar{Y} - b * \bar{t}, \quad (4.6)$$

де n — кількість спостережень.

Лінійна регресійна модель дозволяє виявляти тренди та прогнозувати витрати з урахуванням їх динаміки, що робить її корисною для стратегічного планування місцевих бюджетів.

Висновки до розділу 4

У межах інформаційної системи передбачено реалізацію кількох методів прогнозування з можливістю вибору користувачем алгоритму для розрахунків. Для короткострокових прогнозів доцільно використовувати методи ковзного середнього та експоненційного згладжування, які забезпечують швидке реагування на зміни у витратах. Для довгострокового аналізу застосовується лінійна регресія, що дозволяє моделювати тренди на основі історичних даних.

Результати прогнозів зберігаються в базі даних у таблиці і можуть бути візуалізовані у вигляді графіків та таблиць у веб-інтерфейсі. Крім того, система формує індикатори точності прогнозу (середню абсолютну похибку та відсоток відхилення), що дає змогу користувачу оцінити надійність отриманих результатів.

5 СТРУКТУРНА СХЕМА СИСТЕМИ

Для формування чіткої моделі функціонування інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів було розроблено структурну схему, яка відображає основні компоненти системи, зв'язки між ними та взаємодію із зовнішніми джерелами даних.

5.1 Структурна схема системи

Для візуалізації загальної архітектури інформаційної системи була розроблена структурна схема, що демонструє її основні модулі, зовнішні зв'язки та потоки даних. Система побудована за принципом клієнт—серверної архітектури. На наведеній діаграмі (кресленик в Додатку В) зображено структурну схему для інформаційної системи збору, аналізу та прогнозування даних.

Основні елементи структурної схеми:

- користувацький інтерфейс (Frontend) реалізує взаємодію користувача з системою, забезпечує перегляд аналітичних даних, візуалізацію графіків, роботу з фільтрами, створення звітів та прогнозів; дані отримуються через REST API від серверної частини;

- сервер додатка (Backend) здійснює обробку запитів користувачів, виконує авторизацію, обробляє аналітичні запити, розраховує прогнозні показники та передає результати у фронтенд; забезпечує взаємодію з базою даних і зовнішніми API;

- база даних містить структуровану інформацію про громади, бюджети, витрати, користувачів, звіти та прогнози; зберігає історичні дані для подальшої аналітики;

- хмарне сховище використовується для зберігання великих файлів звітів, документів та резервних копій бази даних;

— зовнішні джерела даних (spending.gov.ua, openbudget.gov.ua, data.gov.ua) забезпечують надходження первинної інформації про витрати бюджетів; інтеграція реалізована через API-запити та періодичні автоматичні оновлення;

— модуль аналітики та прогнозування відповідає за математичну обробку даних, формування статистичних показників, побудову графіків та реалізацію алгоритмів прогнозування;

— модуль керування користувачами та безпекою реалізує автентифікацію, авторизацію, контроль ролей, а також захист переданих даних через протокол HTTPS.

5.2 Опис взаємодії компонентів

У процесі роботи системи користувач через браузер надсилає запит до серверу. Серверна частина виконує обробку запиту, звертається до бази даних або зовнішніх джерел, після чого повертає результати у форматі JSON. Дані відображаються на інтерфейсі у вигляді інтерактивних таблиць і графіків.

Якщо користувач формує прогноз, сервер передає запит у модуль прогнозування, який виконує розрахунки за обраним методом, записує результати у базу та повертає їх у вигляді графічної візуалізації. Усі звіти, створені користувачем, можуть бути експортовані у форматі PDF або збережені у хмарному сховищі.

5.3 Діаграма компонентів

У системі визначено кілька великих компонентів, кожен з яких виконує окрему функціональну роль. На наведеній діаграмі (кресленик в Додатку Г) зображено діаграму компонентів[7] для інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів.

Центральним елементом є веб-браузер, через який користувач взаємодіє з клієнтською частиною. Користувач переглядає карту, статистичні показники, формує звіти та ініціює запити до серверної частини.

Клієнтська частина працює як окремий компонент, що реалізує інтерфейс користувача. У цьому компоненті відображається карта, панель з інформацією про громаду, аналітичні графіки та панель прогнозування. Компонент надсилає REST-запити[8] до серверної частини для отримання аналітичних даних, прогнозів, статистики, метаданих про громади та бюджети.

Серверна частина представлена окремим великим компонентом, який об'єднує підсистеми авторизації та автентифікації, обробки статистики, прогнозної аналітики, керування громадами та бюджетами, а також модуль синхронізації з OpenBudget API. Саме сервер відповідає за отримання та агрегування бюджетних даних, обробку запитів клієнта, виконання математичних обчислень для прогнозів та повернення результатів у клієнтську частину. З клієнтською частиною сервер взаємодіє через протокол HTTPS.

Окремим компонентом у структурі виступає база даних. Серверна частина зчитує та записує дані у таблиці, що містять інформацію про громади, бюджети, структуру витрат, кеш прогнозів, користувачів, сесії та сформовані звіти. База даних не взаємодіє безпосередньо з клієнтом — лише через сервер.

Модуль синхронізації з OpenBudget API представлений як незалежний компонент всередині серверної частини. Він ініціює запити до зовнішніх державних API, отримує дані про бюджети, структуру видатків та місячні показники, обробляє їх і зберігає у локальну базу. Він також співпрацює з компонентом кешування, що дозволяє мінімізувати повторні запити в інтернет.

Хмарне сховище для PDF-файлів звітів є окремим зовнішнім компонентом. Сервер взаємодіє з ним при генерації аналітичних звітів — завантажує готові PDF-файли у S3-bucket та повертає клієнту посилання для завантаження.

Компонент керування користувачами та безпекою відповідає за автентифікацію, видачу токенів, ведення сесій та маршрутизацію прав доступу.

Він співпрацює з таблицями користувачів та сесій у базі даних, а також є точкою контролю для доступу до аналітичних модулів і функцій створення звітів.

Зовнішні джерела даних, такі як OpenBudget API, є окремими компонентами за межами системи. Вони отримують запити від модуля синхронізації серверної частини та повертають первинні дані, що потім обробляються системою.

Разом усі ці компоненти утворюють цілісну архітектуру, де користувач взаємодіє з інтерфейсом; інтерфейс — із сервером; сервер — із базою даних, S3-сховищем та зовнішніми API. Така структура забезпечує масштабованість, модульність, надійність та можливість централізованого оновлення та аналізу бюджетних даних.

Висновки до розділу 5

У межах створення структурної схеми системи було визначено загальну архітектуру майбутнього програмного продукту та сформовано логічну модель взаємодії всіх ключових компонентів. Було обґрунтовано застосування клієнт–серверного підходу, що забезпечує розмежування функціональних рівнів, підвищує масштабованість та спрощує подальший розвиток системи. У процесі моделювання було встановлено ролі кожного компонента – клієнтської частини, серверної логіки, бази даних, модуля синхронізації з державними API та хмарного сховища, а також описано послідовність обміну даними між ними.

Було сформовано детальне представлення взаємодії клієнта з сервером через REST API, визначено порядок виклику модулів аналітики та прогнозування, а також механізми збереження та завантаження звітів. Структурна схема дозволила узгодити логіку роботи системи з вимогами, сформованими на попередньому етапі, та забезпечити цілісне бачення функціональної архітектури.

Таким чином, у розділі було створено основу для подальшого проектування компонентів, визначення внутрішніх модулів та розробки докладної архітектурної документації, що гарантує узгодженість функціональних і технічних рішень у наступних етапах реалізації системи.

6 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ

Вибір технологій для розробки інформаційної системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів був здійснений на основі вимог до надійності, продуктивності, масштабованості та зручності розгортання системи у вебсередовищі. З огляду на архітектурні особливості проєкту, було прийнято рішення реалізувати систему за принципом клієнт—серверної архітектури, де кожен компонент відповідає за свій рівень обробки даних: інтерфейс користувача, сервер додатка, база даних і хмарне сховище.

6.1 Технологія клієнтської частини React

Для створення інтерфейсу користувача було обрано React, одну з найпопулярніших бібліотек JavaScript для побудови динамічних вебзастосунків.

React забезпечує високу швидкість відображення даних завдяки віртуальному DOM[9] і компонентному підходу до розробки. Це дозволяє створювати масштабовану структуру інтерфейсу, де кожен елемент (графік, таблиця, панель фільтрів, звіт) є окремим незалежним компонентом.

Серед основних переваг React ефективне оновлення інтерфейсу без перезавантаження сторінки, можливість повторного використання компонентів, простота інтеграції з іншими бібліотеками (наприклад, Recharts для побудови графіків або Axios для запитів до API), адаптивність під різні пристрої завдяки CSS-фреймворкам (Tailwind CSS, Bootstrap).

Завдяки React користувач отримує швидкий, інтуїтивний та візуально приємний інтерфейс для роботи з великими обсягами аналітичних даних.

6.2 Технологія серверної частини Node.js

Серверну логіку реалізовано за допомогою Node.js — середовища виконання JavaScript, яке забезпечує асинхронну, подієорієнтовану обробку

запитів. Це дає змогу досягати високої продуктивності навіть при великій кількості одночасних користувачів.

Node.js дозволяє легко реалізувати REST API, через який клієнтська частина взаємодіє із сервером. Для побудови API було використано фреймворк Express.js[10], який спрощує маршрутизацію, обробку запитів та керування проміжними обробниками (middleware).

Основні переваги Node.js у контексті проєкту: можливість використання єдиної мови програмування (JavaScript) як для клієнта, так і для сервера; висока швидкість обробки запитів; наявність великої екосистеми бібліотек NPM; зручна інтеграція з PostgreSQL, S3 та зовнішніми API.

Node.js забезпечує логіку збору даних, валідацію, збереження у базі, виконання математичних розрахунків для прогнозування та формування аналітичних звітів.

6.3 Система керування базами даних PostgreSQL

Для зберігання структурованих даних було обрано PostgreSQL — реляційну систему керування базами даних із відкритим вихідним кодом.

Основні переваги PostgreSQL у даному проєкті: підтримка складних SQL-запитів та агрегаційних функцій, необхідних для аналітики; надійна робота з великими обсягами даних; підтримка транзакційності та збереження цілісності даних; можливість розширення за рахунок користувацьких функцій і тригерів; підтримка типів JSONB, що дає змогу комбінувати структуровані та напівструктуровані дані.

У базі даних PostgreSQL зберігаються усі масиви інформації: дані про громади, бюджети, витрати, прогнози, користувачів та журнали подій. Така структура дозволяє виконувати швидкий пошук, групування та обчислення статистичних показників у реальному часі.

6.4 Хмарне сховище файлів Amazon S3 Bucket

Для збереження великих або неструктурованих файлів (аналітичних звітів, PDF-документів) доцільним буде використання Amazon S3 Bucket[11]. Це хмарне рішення забезпечує високу надійність, безпеку та масштабованість.

Інтеграція S3 із Node.js виконується через офіційну бібліотеку AWS SDK, що дозволяє завантажувати, зчитувати та видаляти файли безпосередньо з інтерфейсу адміністратора.

Переваги S3 у даному проєкті: автоматичне резервне копіювання та захист від втрати даних; простота масштабування відповідно до обсягу збережених файлів; гнучка система прав доступу (public/private); висока швидкість доступу до файлів незалежно від кількості користувачів.

Таким чином, Amazon S3 доповнює реляційну базу даних, забезпечуючи ефективне збереження великих звітів і статистичних документів.

6.5 Технологія картографічного сервісу Mapbox

Для реалізації інтерактивної карти в системі було використано сервіс Mapbox, платформу для побудови геоінформаційних інтерфейсів у вебзастосунках. Mapbox надає інструменти для високопродуктивного рендерингу карт, роботи з геоданими та інтеграції користувацьких взаємодій — таких як вибір об'єктів, навігація та масштабування.

Основна причина вибору Mapbox — підтримка векторних карт, які рендеряться безпосередньо на клієнті за допомогою WebGL. Це забезпечує плавну роботу, високу деталізацію та можливість динамічних змін стилів без перезавантаження сторінки. Завдяки цьому система може швидко реагувати на дії користувача навіть при великій кількості геометрій (області, райони, територіальні громади).

У межах проєкту Mapbox використовується для таких задач: відображення адміністративно-територіального поділу України (області, громади); вибір

громади за кліком на карті із подальшим завантаженням її бюджетних даних; адаптивне масштабування в залежності від рівня деталізації, що дозволяє переходити від всієї країни до окремої громади.

Інтеграція Mapbox у клієнтську частину здійснюється через бібліотеку `mapbox-gl`, яка надає програмний інтерфейс для додавання шарів, обробки подій (`click`, `hover`), керування камерою та стилізацією об'єктів. У поєднанні з React це дозволило створити сучасний, швидкий та зручний картографічний модуль.

6.6 Технологія зовнішнього джерела даних API OpenBudget

Для отримання офіційних даних про бюджети територіальних громад, областей та інших адміністративно-територіальних одиниць України в системі використовується API OpenBudget[12], створене Міністерством фінансів України. Це відкрите державне API, яке забезпечує доступ до детальних фінансових даних у розрізі програмної, функціональної та економічної класифікацій, а також по періодах виконання бюджету.

API OpenBudget є основним джерелом фактичних значень бюджетних показників, на основі яких будується аналітика, статистика та прогнозування в системі.

У проєкті API використовується для таких задач завантаження щомісячних даних про запланованих та фактичних видатків за всіма класифікаціями; отримання інформації для синхронізації бюджетів громад, включно з кодами бюджетів, назвами та метаданими; оновлення внутрішньої бази даних з офіційних джерел.

API повертає дані у форматі JSON або CSV, зокрема:

- `rep_period` — період (місяць.рік), наприклад "09.2024";
- `plans_amt` — планові призначення на накопичувальній основі;
- `fakt_amt` — фактичні витрати за аналогічним принципом;
- `classification_type` — PROGRAM, FUNCTIONAL або ECONOMIC;

— `cod_cons_mb_pk` — код програмної статті (КПК), економічної або функціональної класифікації.

Всі дані надходять у накопичувальному форматі, тому система виконує додаткові обрахунки для визначення правильних річних та квартальних значень (вибір останнього доступного місяця року).

Синхронізація відбувається за допомогою окремого модуля, що:

- a) формує запит до API OpenBudget за конкретним бюджетним кодом, класифікацією, роком та періодом;
- b) завантажує великі обсяги даних (деякі бюджети містять десятки тисяч рядків);
- c) зберігає дані у таблицю;
- d) фіксує статус, обсяг даних та можливі помилки у окремій таблиці;
- e) для запобігання дублюванню запитів система веде облік часу останньої синхронізації кожного бюджету.

До переваг використання API OpenBudget можна віднести офіційність, актуальність, глибину деталізації, відкритість та автоматизацію

Інформаційна система побудована таким чином, що всі аналітичні розрахунки базуються на даних із OpenBudget. Саме тому коректна інтеграція з API є ключовою частиною функціональності.

Використання API OpenBudget дає можливість створити єдину, централізовану платформу для аналізу місцевих бюджетів, яка працює з актуальними та уніфікованими державними даними.

6.7 Контейнеризація за допомогою Docker та Docker Compose

У системі планується застосовувати Docker для ізоляції всіх компонентів застосунку та забезпечення стабільного середовища незалежно від операційної системи чи локальних налаштувань розробника. Контейнеризація дозволяє запускати frontend, backend та базу даних PostgreSQL у вигляді окремих сервісів,

кожен з яких має власний набір залежностей і конфігурацій. Це усуває проблеми, пов'язані з різницею у версіях бібліотек, інструментів чи мовних пакетів.

Docker Compose буде використано для координації всіх сервісів. Один YAML-файл визначає їхню структуру, порядок запуску, мережеві налаштування та точки монтування. Завдяки цьому система запускається однією командою, а залежності між сервісами (наприклад, підключення backend до бази даних) конфігуруються автоматично. Такий підхід спрощує розгортання як у середовищі розробки, так і у продакшні, дозволяючи швидко відновити роботу застосунку на новому сервері або робочій машині.

Завдяки контейнеризації процес оновлення, тестування та масштабування системи стає передбачуваним і контрольованим — кожен компонент працює у власному визначеному середовищі, яке не змінюється з часом. Це суттєво підвищує надійність і повторюваність процесів, що є ключовим для аналітичних систем, які працюють з великими обсягами даних та зовнішніми API.

6.8 Розгортання системи за допомогою AWS EC2 Instance

AWS EC2 (Elastic Compute Cloud)[13] обрано як платформу для розгортання системи. EC2 надає віртуальні сервери в хмарі з повним контролем над операційним середовищем та конфігурацією.

Вибір EC2 обумовлений потребами системи та вимогами до інфраструктури. Система складається з кількох компонентів, розгорнутих у Docker контейнерах: веб-клієнт на базі React, Node.js сервер та база даних PostgreSQL. EC2 дозволяє розгорнути всі компоненти на одному інстансі, забезпечуючи їхню взаємодію через внутрішню мережу та спрощуючи управління.

Важливою перевагою є гнучкість масштабування. EC2 дозволяє змінювати тип інстансу залежно від навантаження, додавати інстанси для балансування навантаження та автоматично масштабувати ресурси. Це критично для системи, яка може мати змінне навантаження залежно від кількості користувачів та обсягу оброблюваних даних.

Економічна ефективність також є важливим фактором. EC2 пропонує різні моделі оплати, включаючи оплату за фактичне використання, що дозволяє оптимізувати витрати, особливо на етапі розробки та тестування. Можливість використання резервних інстансів або Spot інстансів додатково знижує вартість експлуатації.

Безпека та надійність є ключовими вимогами для системи, що працює з фінансовими даними. EC2 інтегрується з іншими сервісами AWS, такими як AWS S3 для зберігання файлів, AWS VPC для ізоляції мережі та AWS IAM для управління доступом. Це забезпечує комплексний підхід до безпеки та дозволяє використовувати перевірені інструменти AWS.

Технічна сумісність з Docker є ще одним важливим аспектом. EC2 підтримує роботу з Docker контейнерами, що дозволяє використовувати docker-compose для оркестрації всіх компонентів системи. Контейнеризація забезпечує ізоляцію компонентів, спрощує розгортання та дозволяє легко масштабувати окремі частини системи незалежно одна від одної.

6.9 Додаткові інструменти та бібліотеки

Для підвищення ефективності розробки та якості продукту було обрано такі додаткові інструменти:

- Axios — для взаємодії з REST API;
- Chart.js або Recharts — для побудови інтерактивних графіків і діаграм;
- JWT (JSON Web Token) — для реалізації безпечної авторизації користувачів;
- Nodemon — для автоматичного перезапуску сервера під час розробки;
- dotenv — для зберігання конфіденційних параметрів підключення до бази даних і S3.

6.10 Діаграма розгортання системи

Діаграма розгортання (кресленик в Додатку Д) відображає архітектуру системи та взаємодію компонентів. Система розгорнута на AWS EC2 і складається з кількох рівнів.

Користувач взаємодіє через веб-браузер. Запити надходять до Nginx Web Server, який працює в Docker контейнері для клієнта на AWS EC2. Nginx обслуговує статичні файли зібраного React-додатку, що містить інтерфейс для перегляду бюджетів, генерації звітів та адміністрування. React-клієнт використовує MapBox API для відображення інтерактивних карт регіонів та територіальних громад.

Клієнтська частина спілкується з бекендом через REST API. Backend працює в окремому Docker контейнері як Node.js сервер на базі Express Web Framework. Express обробляє HTTP-запити, автентифікацію, авторизацію та бізнес-логіку. Для роботи з базою даних використовується PostgreSQL клієнт (pg), який виконує SQL-запити до PostgreSQL, що працює в окремому Docker контейнері на тому ж EC2 інстансі. PostgreSQL зберігає дані про бюджети, користувачів, звіти, структури бюджетів та іншу інформацію.

Система інтегрується з зовнішніми сервісами. Backend виконує ETL-синхронізацію з OpenBudget API для отримання актуальних даних про державні бюджети. Згенеровані PDF-звіти зберігаються в AWS S3 Bucket, що забезпечує надійне зберігання та швидкий доступ до документів. Backend завантажує звіти в S3 та надає користувачам посилання для завантаження.

Всі компоненти розгорнуті як Docker контейнери на одному AWS EC2 інстансі, що спрощує розгортання та масштабування. Контейнери ізольовані, але можуть спілкуватися через внутрішню мережу Docker. Така архітектура забезпечує модульність, легке оновлення компонентів та високу доступність системи.

Висновки до розділу 6

Вибраний технологічний стек — React + Node.js + PostgreSQL + S3 Bucket — забезпечує сучасний, гнучкий і масштабований підхід до створення веборієнтованої аналітичної системи. Окремо було обґрунтовано застосування Docker і Docker Compose для контейнеризації та створення ізольованого, портативного середовища розгортання.

Обраний технологічний стеку дозволяє створити зручний і швидкий інтерфейс користувача; забезпечити стабільну роботу сервера з великою кількістю запитів; гарантувати надійне збереження структурованих та неструктурованих даних; реалізувати автоматичну інтеграцію з відкритими державними джерелами; підтримувати подальший розвиток системи та розширення її функціоналу без зміни основної архітектури.

Таким чином, обрані технології оптимально відповідають функціональним і технічним вимогам до системи, забезпечуючи високу продуктивність, гнучкість та надійність у роботі.

7 ІНФОЛОГІЧНА ТА ДАТАЛОГІЧНА МОДЕЛЬ

Для ефективного зберігання та обробки даних в інформаційній системі було спроектовано реляційну базу даних. Основною метою проєктування бази є забезпечення логічної цілісності даних, мінімізація дублювання інформації та створення зручної структури для аналітичних запитів і прогнозних розрахунків.

Процес проєктування бази даних включав аналіз предметної області, визначення сутностей, їхніх атрибутів та взаємозв'язків між ними. Результатом стала ER-діаграма (Entity—Relationship Diagram), яка відображає логічну структуру даних, що використовуються системою.

7.1 Інфологічна модель

ER-діаграма[14, с. 13] відображає основні сутності, їхні атрибути та типи зв'язків між ними. У системі реалізовано кілька ключових сутностей, які забезпечують роботу аналітичної платформи для перегляду, обробки та прогнозування місцевих бюджетів.

На наведеній діаграмі (кресленик в Додатку Е) зображено інфологічну модель для інформаційної системи.

Основні сутності системи включають в себе:

Region (Область). Довідник адміністративних областей України. Містить назву, код та координати центру області. Використовується для відображення на мапі та групування громад.

Community (Громада). Містить інформацію про територіальні громади: назва, код, належність до області. Пов'язана з Region (зв'язок N:1).

Budget (Бюджет громади). Зберігає метадані про бюджет громади за певний рік: код бюджету, назва, тип бюджету, рік. Пов'язаний з Community (зв'язок N:1).

Monthly Indicators (Помісячні показники). Агреговані підсумки бюджету за кожен місяць (дохід, видатки, баланс). Пов'язані з Budget (зв'язок 1:N).

Budget Structure (Структура бюджету). Головна транзакційна таблиця. Містить деталізовані дані з OpenBudget за всіма типами бюджетів (громади, області, райони тощо): періоди, класифікації, показники плану/факту, коди програм/функцій/економічних статей. Ця сутність не пов'язана зовнішнім ключем із Budget, оскільки містить усі бюджети України, а таблиця Budget — лише бюджети громад.

Forecast (Кеш прогнозів). Сутність зберігає результати прогнозування за різними методами (середній приріст, ковзне середнє, експоненціальне згладжування, регресія). Містить: код бюджету, метод, прогнозовані значення. Пов'язується через код бюджету з Budget Structure.

User (Користувач). Зареєстровані користувачі системи з ролями: user — може формувати та зберігати звіти; admin — управляє користувачами, запуском синхронізації, переглядом логів.

Report (Аналітичний звіт). Містить згенеровані PDF-файли аналітичних звітів, які зберігаються у S3. Прив'язується до користувача.

Допоміжні сутності:

User Sessions (Сесії користувачів). Зберігає токени автентифікації, час створення та завершення сесії.

API Sync Log (api_sync). Таблиця логів синхронізації даних з OpenBudget. Фіксує: endpoint, час, статус, отримані обсяги даних, помилки.

User Log (Log). Журнал дій користувачів системи (вхід, створення звіту, запуск синхронізації тощо). Прив'язується до користувача.

Region — Community: зв'язок «один до багатьох», кожна область має багато громад.

Community — Budget: зв'язок «один до багатьох», оскільки кожна громада може мати кілька бюджетів за різні роки.

Budget — Monthly Indicators: зв'язок «один до багатьох», кожен бюджет має 12 місячних агрегованих показників.

Budget Structure — Forecast Cache: зв'язок «один до багатьох», адже кожен бюджет може мати кілька прогнозів (по різних класифікаційних типах та методах).

User — Report: зв'язок «один до багатьох», один користувач може створювати декілька аналітичних звітів.

User — User Sessions: зв'язок «один до багатьох», кожен користувач може мати активні токени.

User — Log: зв'язок «один до багатьох», дії користувача фіксуються у журналі подій.

7.2 Опис основних таблиць бази даних

В таблиці 7.1 надано інформацію про основні таблиці бази даних в системі.

Таблиця 7.1 — Опис основних таблиць бази даних інформаційної системи

Назва таблиці	Призначення	Основні поля	Тип зв'язку
region	зберігає інформацію про області України	id, name, code, center_lat, center_lng	1:N з communities
communities	довідник територіальних громад	id, name, code, region_id	N:1 з region; 1:N з budgets
budget	містить дані про бюджети громад за роками	id, community_id, year, total_income, total_expense, balance	1:N з expenses; 1:1 з forecasts
monthly_indicators	агреговані помісячні бюджетні показники	id, budget_id, month, income, expense, balance	N:1 з budgets
budget_structure	транзакційні дані: планові та фактичні витрати, класифікації, програми, економічні коди показників	id, rep_period, classification_type, fund_type, cod_budget, cod_cons_mb_pk, name, plans_amt, fakt_amt	
forecast_cache	зберігає результати прогнозування (кеш)	id, budget_code, classification_type, method, params, series, forecast	1:N з budget
users	містить інформацію про користувачів системи та їх ролі	id, email, role, password_hash	1:N з reports, 1:N з logs, 1:N з user_sessions
user_sessions	зберігає токени	id, user_id, token_hash	N:1 з users

Назва таблиці	Призначення	Основні поля	Тип зв'язку
	авторизації користувачів		
reports	зберігає аналітичні звіти, створені користувачами	id, user_id, title, file_url, created_at, is_public	N:1 з users
logs	фіксує дії користувачів	id, user_id, action, timestamp, details	N:1 з users
api_sync	логи синхронізації з OpenBudget	id, endpoint, last_synced, status, details	автономна таблиця, не має FK

Між таблицями region і communities встановлено зв'язок 1:N — кожна область може містити багато територіальних громад.

Між таблицями communities і budgets реалізовано зв'язок 1:N, оскільки одна громада може мати кілька бюджетів за різні роки.

Між budgets і monthly_indicators існує зв'язок 1:N — кожен бюджет має власну динаміку доходів, видатків і балансу по місяцях.

Таблиця budget_structure не має фізичного зв'язку з budgets, оскільки містить дані для всіх бюджетів України незалежно від того, чи є бюджет у таблиці budgets. Вона використовується як транзакційне сховище імпортованих показників з OpenBudget.

Таблиця forecast_cache має логічний зв'язок з бюджетами через budget_code та classification_type. Один бюджет може мати кілька прогнозів — за різними класифікаціями та методами прогнозування (зв'язок 1:N).

Зв'язок між users і reports — 1:N, оскільки один користувач може створювати багато аналітичних звітів.

Зв'язок між users і logs — 1:N, адже всі події, пов'язані з користувачем (авторизація, дії в системі), фіксуються у журналі.

Зв'язок між users і user_sessions — 1:N, що дає можливість одному користувачу мати кілька активних сесій (токенів).

7.3 Даталогічна модель системи

На основі попередньо сформованої інфологічної моделі була побудована даталогічна модель, яка визначає точний склад таблиць, їх поля, типи даних та зв'язки між ними.

Якщо інфологічна модель описує систему на концептуальному рівні, то даталогічна конкретизує її у вигляді фізичної структури майбутньої бази даних.

Побудова даталогічної моделі здійснювалася з урахуванням низки факторів:

- особливості даних, отриманих з API OpenBudget;
- необхідність зберігати транзакційну фінансову інформацію у великому обсязі;
- розмежування доступу та можливість створення звітів користувачами;
- наявність ролей та механізму авторизації;
- потреба зберігати кешовані результати прогнозування.

У даталогічній моделі відображено всі таблиці бази даних та їхні ключі.

Структура включає як основні об'єктні таблиці (users, budgets, communities тощо), так і транзакційні (budget_structure, monthly_indicators), а також допоміжні (user_sessions, user_reports).

На рисунку 7.1 наведена даталогічна модель інформаційної системи, що відображає остаточну структуру з урахуванням реальних вимог до зберігання бюджетних даних та роботи з користувачами.

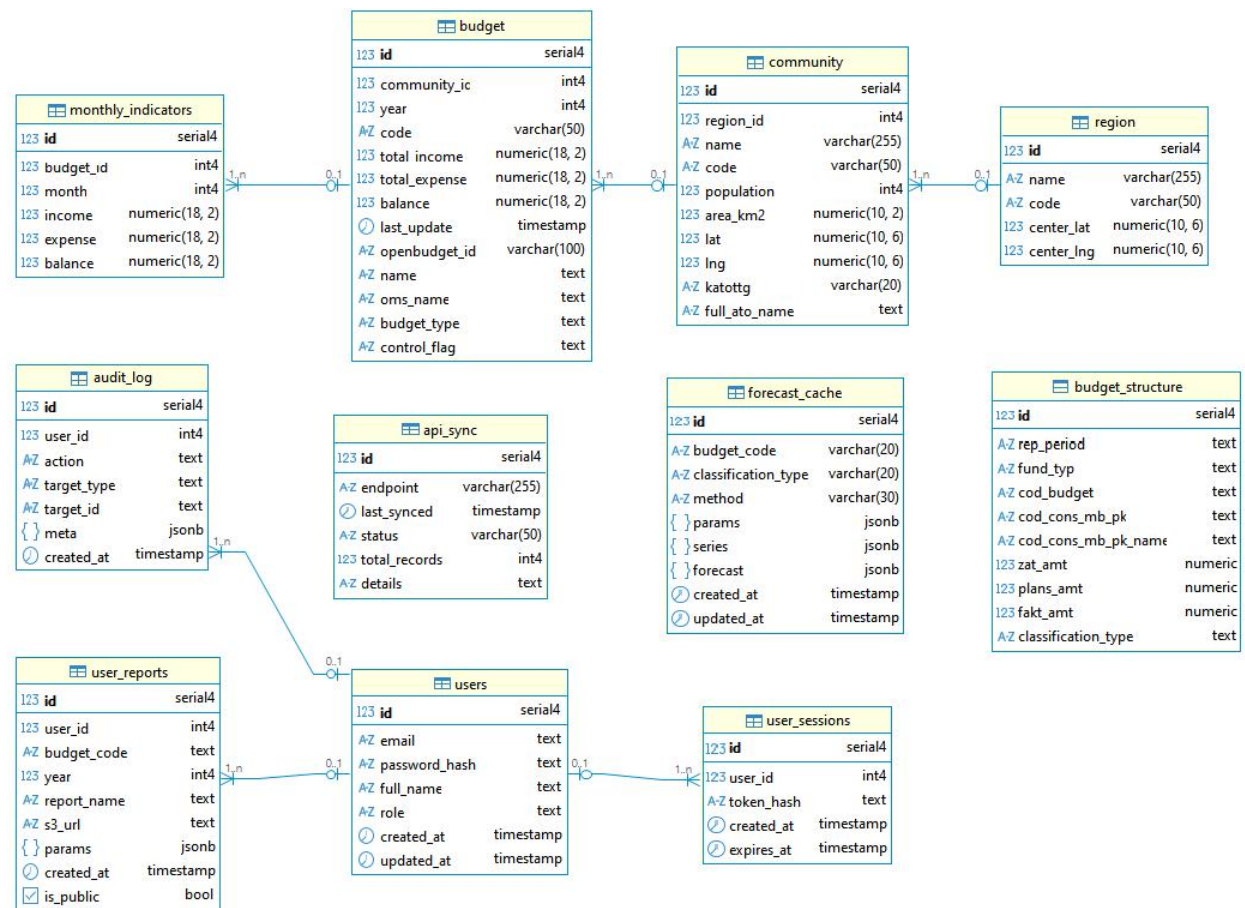


Рисунок 7.1 — Даталогічна модель інформаційної системи

На таблиці 7.2 наведено опис полів таблиці «Регіони».

Таблиця 7.2 — Опис полів таблиці «Регіони»

Назва	Тип	Опис	Назва в базі даних
ідентифікатор	integer	первинний ключ області	id
Назва області	varchar	Повна назва регіону	name
код регіону	integer	офіційний код області	code
геокоординати широти	numeric	центр області (широта)	center_lat
геокоординати довготи	numeric	центр області (довгота)	center_lng

На таблиці 7.3 наведено опис полів таблиці «Громади».

Таблиця 7.3 — Опис полів таблиці «Громади»

Назва	Тип	Опис	Назва в базі даних
ідентифікатор	integer	первинний ключ громади	id
регіон	integer	зовнішній ключ на таблицю region.id	region_id
назва громади	varchar	назва територіальної громади	name
бюджетний код громади	integer	офіційний бюджетний код, використовується в OpenBudget	code
населення	integer	кількість жителів громади	population
площа території	numeric	площа громади у км ²	area_km2
геолокація (широта)	numeric	координата широти для візуалізації на карті	lat
геолокація (довгота)	numeric	координата довготи	lng
КАТОТТГ	integer	офіційний код КАТОТТГ громади	katottg
повна назва АТО	text	повна адміністративно-територіальна назва громади	full_ato_name

В таблиці 7.4 наведено опис полів таблиці «Бюджет».

Таблиця 7.4 — Опис полів таблиці «Бюджет»

Назва	Тип	Опис	Назва в базі даних
ідентифікатор	integer	первинний ключ бюджету	id
громада	integer	зовнішній ключ на таблицю community.id	community_id
рік бюджету	integer	рік, до якого належить бюджет	year
код бюджету	integer	офіційний код бюджету	code
загальні видатки	numeric	підсумкова сума видатків за рік	total_expense
ідентифікатор OpenBudget	varchar	внутрішній ID бюджету у державному API OpenBudget	openbudget_id
назва бюджету	text	офіційна назва бюджету	name
назва органу місцевого самоврядування	text	назва установи, що адмініструє бюджетт	oms_name

Таблиця 7.5 — Опис полів таблиці «Структура бюджету»

Назва	Тип	Опис	Назва в базі даних
ідентифікатор	integer	первинний ключ	id
звітний період	text	місяць та рік записання витрати	rep_period
тип фонду	text	тип фонду: T (total), C (general), S (special)	fund_typ
код бюджету	integer	офіційний код бюджету	cod_budget
програмний код класифікації	integer	програмний код класифікації	total_expense
назва програмного коду класифікації	text	назва відповідного коду з API OpenBudget	cod_cons_mb_pk_name
затверджено (затрати)	numeric	початкові затверджені планові показники	zat_amt
уточнений план	numeric	уточнені планові показники	plans_amt
виконано (факт)	numeric	значення поля ФАКТ_АМТ — фактичні видатки за звітний період (накопичувальні)	fakt_amt
тип класифікації	text	одне з трьох: program, functional, economic. Визначається при синхронізації	classification_type

Висновки до розділу 7

Розроблена структура бази даних забезпечує логічну цілісність і нормалізацію даних, можливість масштабування при збільшенні обсягу інформації, простоту виконання складних SQL-запитів, ефективну підтримку аналітичних операцій та побудови прогнозів та гнучку інтеграцію з зовнішніми джерелами даних через API.

8 ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ

Архітектура програмної системи визначає її внутрішню структуру, компоненти та способи взаємодії між ними. Від якості архітектурних рішень залежить масштабованість, продуктивність, надійність і здатність до подальшого розвитку. Для аналітичних систем, які працюють з великими обсягами даних і регулярно взаємодіють із зовнішніми сервісами, архітектура має забезпечувати стабільність, швидку обробку запитів і можливість розширення функціоналу без суттєвих змін у фундаментальних модулях.

Обрана архітектура застосунку побудована за принципами модульності та чіткого розподілу відповідальностей. Кожний компонент системи реалізує власну функціональність: клієнтська частина відповідає за відображення та взаємодію, серверна — за бізнес-логіку та управління даними, база даних — за структуроване зберігання інформації. Завдяки такому поділу система легко підтримується, а в разі потреби окремі модулі можуть бути замінені або масштабовані незалежно від інших.

8.1 Клієнт-серверна архітектура

Клієнт-серверна архітектура є базовою моделлю побудови системи та визначає спосіб взаємодії між її основними компонентами. У межах даного проєкту використовується класичний варіант архітектури, у якому логіка обробки даних зосереджена на сервері, а клієнтська частина відповідає за відображення інформації та взаємодію з користувачем. Такий підхід дозволяє забезпечити високу продуктивність, гнучкість у розширенні функціоналу та зручність підтримки.

У системі клієнт виконується у веб-браузері і реалізований за допомогою React. Він відповідає за формування інтерфейсу, який відображає ключові дані — бюджетні показники, графіки, статистичні таблиці, прогнози видатків, а також

інтерактивну карту з вибором громади. Всі дані клієнт отримує через REST API, які надає серверна частина.

Сервер побудовано на Node.js з використанням Express. Саме сервер виконує основні операції — отримує дані з OpenBudget API, зберігає їх у PostgreSQL, виконує агрегацію бюджетних показників, формує статистику, обчислює прогнози чотирма методами, керує користувачами, обробляє авторизацію, генерує PDF-звіти та зберігає їх у S3. Завдяки асинхронній природі Node.js сервер може обробляти велику кількість одночасних запитів, що є важливим при роботі з великими бюджетними наборами даних.

База даних PostgreSQL забезпечує надійне й структуроване зберігання інформації. У ній розміщено громади, бюджети, детальну структуру доходів і видатків, кеш прогнозів, користувачів, сесії, логи взаємодій і згенеровані звіти. Окремим компонентом виступає таблиця `budget_structure`, яка містить повний набір бюджетних даних, отриманих із зовнішнього API. Таблиця не залежить від локальної таблиці `budget` і таким чином може охоплювати всі бюджети України, навіть якщо вони не занесені до довідника громад.

Взаємодія між клієнтом і сервером здійснюється через HTTP-запити. Фронтенд отримує структуровану інформацію у форматі JSON та відображає її у вигляді графіків, діаграм, таблиць і картографічних шарів. Використання цього механізму дозволяє чітко розділити логіку та інтерфейс, що спрощує масштабування: при необхідності сервер може бути замінений, розширений або розподілений на декілька незалежних сервісів без змін у клієнтській частині.

Ключовою перевагою клієнт-серверної архітектури в контексті цього проєкту є можливість ефективної роботи з великими обсягами бюджетних даних. До клієнта потрапляють лише необхідні результати розрахунків, тоді як сервер виконує всі важкі операції — обробку десятків тисяч записів, підсумовування, перерахунки, вибір останніх періодів, прогнозування та формування звітів. Це забезпечує високу швидкість роботи інтерфейсу та зручність взаємодії для користувачів.

8.2 Загальна архітектура системи

Інформаційна система побудована як багатокомпонентна клієнт-серверна платформа, у якій окремі частини виконують чітко визначені функції, але взаємодіють між собою через стандартизовані інтерфейси. Такий підхід робить систему гнучкою, масштабованою та придатною до подальшого розвитку.

Загальна структура включає чотири основні підсистеми: клієнтський застосунок, серверну логіку, базу даних та зовнішні API. Додатково використовуються сервіси для збереження звітів (Amazon S3) та контейнеризації (Docker). Кожен компонент системи взаємодіє з іншими через зрозумілі та формалізовані запити, що дозволяє оновлювати або замінювати будь-яку частину без впливу на інші модулі.

Клієнтська частина відповідає за відображення даних, взаємодію з користувачами та формування запитів до сервера. Вона працює у браузері та реалізована за допомогою React, що забезпечує високу швидкість інтерфейсу. Значна частина користувацького досвіду побудована на інтерактивній карті громади, реалізованій за допомогою Mapbox GL JS, через яку користувач обирає бюджет для аналізу.

Серверна частина реалізована на Node.js та Express. Вона опрацьовує запити, виконує бізнес-логіку, агрегує дані, викликає зовнішні API OpenBudget, записує інформацію до бази, формує статистику та прогнозні моделі, генерує звіти у форматі PDF і передає клієнту необхідні дані. Окремий модуль відповідає за авторизацію користувачів і роботу з сесіями.

База даних PostgreSQL виступає основним джерелом збереження структурованої інформації. У ній зберігаються громади, бюджети, бюджетна структура, кеш прогнозів, користувачі, сесії та згенеровані звіти. Особливе місце займає таблиця `budget_structure`, яка є ядром системи аналітики, оскільки містить повну деталізацію видатків за всіма типами бюджетної класифікації.

Для наочності загальні компоненти системи та їхні ролі наведено у таблиці.

Таблиця 8.1 — Основні компоненти системи та їх призначення

Компонент	Призначення	Основні функції
React-клієнт	клієнтський інтерфейс	відображення статистики, графіків, прогнозів; інтерактивна карта; формування запитів до API
Mapbox GL JS	картографічний модуль	відображення карти України; вибір громади; візуалізація геоданих
Node.js + Express сервер	серверна логіка	збір даних з OpenBudget; обробка запитів; генерація прогнозів; формування PDF-звітів; керування користувачами
PostgreSQL	основна база даних	зберігання громад, бюджетів, бюджетної структури, прогнозів, користувачів, логів
Amazon S3	файлове сховище	зберігання PDF-аналітичних звітів
Docker / Docker Compose	контейнеризація	автоматизований запуск клієнта, сервера та бази даних у спільному середовищі
OpenBudget API	зовнішнє джерело даних	отримання сирих бюджетних показників для подальшого аналізу

У рамках загальної структури система працює наступним чином: користувач взаємодіє з інтерфейсом у браузері, обираючи громаду та переглядаючи доступні дані. React надсилає HTTP-запит до серверної частини, де Express обробляє його, звертається до бази даних або виконує обчислення, після чого формує відповідь у форматі JSON. Дані повертаються на клієнт, де React оновлює інтерфейс, відображаючи графіки, таблиці та діаграми.

Завдяки чітко структурованій взаємодії між компонентами система легко розширюється — можна додавати нові модулі прогнозування, різні типи звітів, нові API або додаткові картографічні функції, не змінюючи основи архітектури. Це робить проєкт придатним для подальшого масштабування, інтеграції та модернізації.

8.3 Реалізація клієнтської частини

Клієнтська частина інформаційної системи реалізована за допомогою React, що забезпечує модульність, високу продуктивність та зручність подальшого розширення інтерфейсу. Компонентний підхід дозволяє розділити інтерфейс на окремі логічні блоки, кожен з яких відповідає за власну функцію, що значно спрощує розвиток та підтримку системи. Основою взаємодії з даними є інтерактивна карта України, через яку користувач у природний спосіб обирає потрібний регіон, після чого система автоматично завантажує статистику, структуру витрат та прогнозні моделі. Такий підхід робить інтерфейс інтуїтивним та доступним для користувачів без спеціальної підготовки.

Фронтенд побудований як набір взаємопов'язаних модулів, що забезпечують відображення картографічної інформації, статистики, аналітики, роботи з профілем користувача та генерації звітів. Інтеграція з Mapbox GL JS дозволяє реалізувати швидку та точну візуалізацію геоданих, плавну навігацію та інтерактивну роботу з територіями. Завдяки оптимізації рендерингу та адаптивній верстці інтерфейс коректно працює на різних пристроях та стабільно реагує на дії користувача.

Узагальнено, клієнтська частина є динамічним інструментом доступу до бюджетних даних, який поєднує геоінформаційну взаємодію, аналітику та прогнозування у зручній та зрозумілій формі.

Архітектурно фронтенд складається з кількох ключових модулів, які відображено на рисунку 8.1.

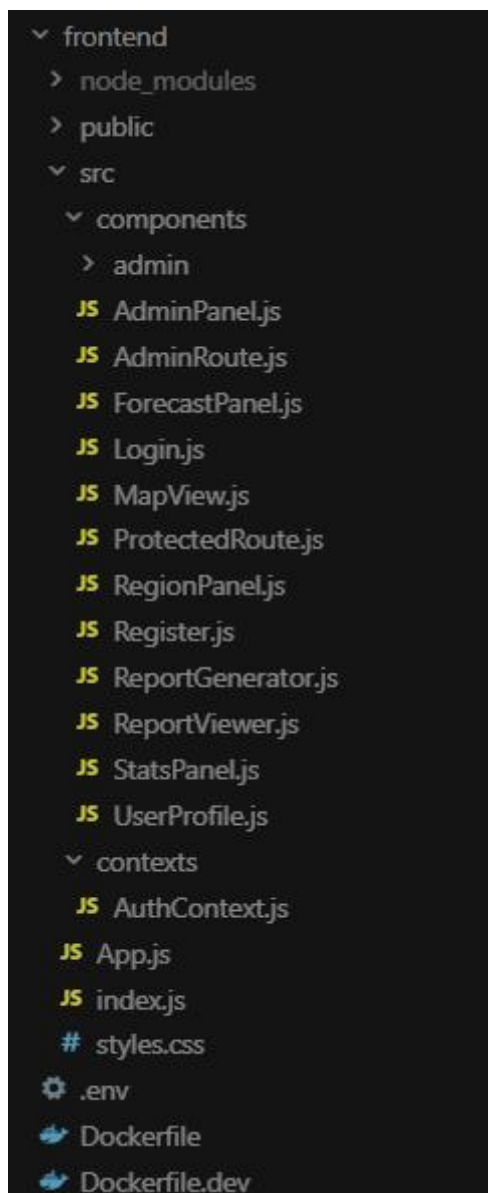


Рисунок 8.1 — Структура клієнтської частини

Компонент MapView реалізує карту та логіку вибору громади через Mapbox GL JS[15]. Компоненти RegionPanel та StatsPanel відповідають за відображення детальної інформації — структури видатків, графіків, статистики за роками, даних виконання бюджету. Окремий компонент ForecastPanel реалізує роботу з алгоритмами прогнозування, включаючи відображення графіка та вибір методу розрахунку.

Компоненти Login та Register забезпечують автентифікацію та реєстрацію користувачів. Login містить форму входу з валідацією облікових даних та

обробкою помилок, а Register дозволяє новим користувачам створити обліковий запис з введенням email, пароля та повного імені.

Компонент UserProfile надає функціонал управління профілем авторизованого користувача, дозволяючи оновлювати особисті дані (email, повне ім'я) та змінювати пароль з перевіркою поточного пароля.

Компонент ReportGenerator реалізує інтерфейс для генерації PDF звітів про бюджети. Він містить форму для вибору параметрів звіту (код бюджету, тип, рік), налаштування опцій включення різних розділів статистики, параметри прогнозування та можливість збереження звіту з назвою та налаштуванням публічності. Компонент також відображає список власних згенерованих звітів користувача.

Компонент ReportViewer забезпечує перегляд публічних та приватних звітів. Він містить таби для перемикання між публічними звітами та власними звітами користувача, дозволяє завантажувати PDF файли та відображає метадані звітів.

Компонент AdminPanel є головним інтерфейсом для адміністраторів системи та містить три основні розділи. Підкомпонент UserManagement забезпечує управління користувачами: перегляд списку, редагування даних, зміну паролів, перегляд та видалення звітів користувачів, а також видалення облікових записів. Підкомпонент SyncManagement дозволяє адміністраторам керувати синхронізацією даних з OpenBudget API: перегляд результатів попередніх синхронізацій, запуск нових синхронізацій для обраних років та моніторинг статусу виконання. Підкомпонент BudgetStructureModeration надає інструменти для модерації структури бюджетів: створення, редагування та видалення структурних елементів бюджетів у системі.

Компоненти ProtectedRoute та AdminRoute реалізують захист маршрутів на рівні React Router. ProtectedRoute перевіряє автентифікацію користувача та перенаправляє на сторінку входу, якщо користувач не авторизований. AdminRoute додатково перевіряє наявність адміністраторських прав, забезпечуючи доступ до адмін панелі тільки для користувачів з відповідною роллю.

React взаємодіє з сервером через REST API. При зміні обраної громади або року надсилаються запити для отримання бюджетної структури, прогнозів чи згенерованих користувачем звітів. Для роботи з сервером використовуються `fetch`-запити або `Axios`, залежно від контексту. Дані приходять у форматі JSON і передаються у компоненти через стан додатка (`state`).

Важливу роль у візуалізації відіграє бібліотека `Recharts`, яка використовується для побудови інтерактивних графіків. Це дозволяє в реальному часі оновлювати статистику видатків, динаміку за роками, прогнозні значення та інші показники. Завдяки адаптивній верстці інтерфейс коректно працює на різних розмірах екранів, включно з планшетами.

На клієнті реалізовано також зручну систему навігації та індикаторів завантаження. Під час виконання тривалих операцій, наприклад розрахунку прогнозу або формування PDF-звіту, користувач бачить відповідні статуси. Додатково передбачено авторизацію користувачів: на фронтенді зберігається токен сесії, який передається у кожному запиті до серверної частини.

Узагальнено клієнтська частина слугує інтерактивним інтерфейсом для перегляду територіальних даних, аналітики та прогнозування бюджетної інформації. React забезпечує швидкість роботи, модульність і гнучкість, а `Mapbox` додає географічну інтуїтивність, роблячи систему зручною для кінцевого користувача.

8.4 Реалізація серверної частини

Серверна частина інформаційної системи побудована на платформі `Node.js` із використанням фреймворка `Express.js`, що забезпечує чітке розмежування логіки, масштабованість та зручність подальшої підтримки. Архітектура орієнтована на модульність і застосовує принципи MVC, де контролери відповідають за обробку HTTP-запитів, сервіси — за бізнес-логіку, а маршрутизатори — за організацію ендпоінтів. Такий підхід дозволяє просто

розширювати систему без порушення існуючої структури та забезпечує стабільну роботу навіть за умов зростання обсягів даних.

8.4.1 Структура серверної частини

У серверній частині можна виділити чотири ключові шари: маршрутизацію, контролери, сервісні модулі та допоміжні скрипти. Вони взаємодіють у межах узгодженої схеми, де кожен компонент виконує свою вузькоспеціалізовану роль. Структуру серверної частини продемонстровано на рисунку 8.2.

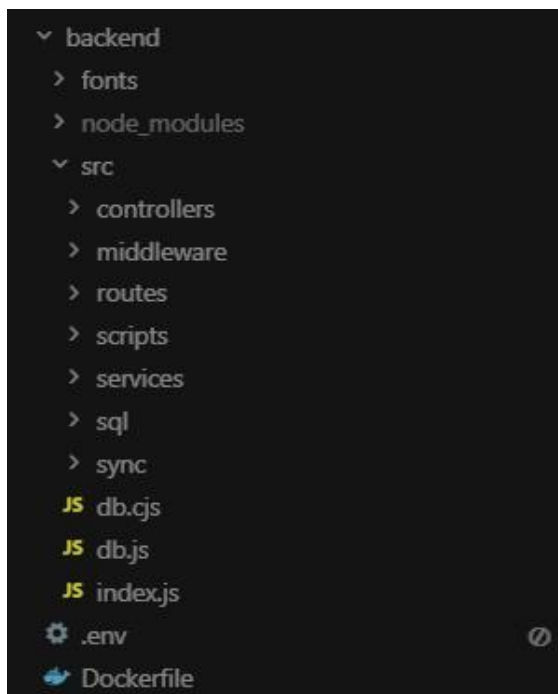


Рисунок 8.2 — Структура серверної частини

Маршрутизатори (routes) відповідають за те, щоб кожен HTTP-запит потрапляв у відповідний контролер. Логіка маршрутів розділена на бюджети, громади, регіони, структура видатків, статистичні дані, прогнозування та процеси синхронізації. Це робить API логічно організованим і дозволяє легко орієнтуватися у структурі ендпоінтів.

Контролери (controllers) безпосередньо обробляють запити користувача або клієнтської частини, проводять базову валідацію параметрів і звертаються до

сервісів. Вони формують відповіді у форматі JSON і не містять бізнес-логіки чи доступу до бази даних. Наприклад, BudgetController повертає метадані бюджету, StructureController працює з деталізацією видатків, ForecastController управляє процесом обчислення та повернення прогнозів, а SyncController контролює оновлення даних з OpenBudget API. AuthController обробляє автентифікацію та реєстрацію користувачів, ReportsController керує генерацією та завантаженням PDF звітів, AdminController надає функціонал адміністративного управління системою, а CommunityController та RegionController працюють з даними про територіальні громади та регіони.

Основне навантаження виконується у сервісах (services). Усі SQL-запити, складні математичні обчислення, формування статистики, обробка прогнозів, кешування результатів та інтеграція з MapBox API винесені саме сюди. Завдяки цьому контролери залишаються «чистими», а логіка добре структурованою. Сервіс StructureStatsService прораховує місячну, квартальну та річну статистику; ForecastService реалізує методи прогнозування; ForecastCacheService зберігає і повертає кешовані результати; SyncService відповідає за взаємодію з OpenBudget API. AuthService керує генерацією та валідацією JWT токенів, управлінням сесіями користувачів; UserService забезпечує роботу з обліковими записами користувачів, включаючи хешування паролів та валідацію; ReportsService відповідає за генерацію PDF звітів з використанням PDFKit, формування графіків та таблиць; S3Service забезпечує інтеграцію з AWS S3 для зберігання та завантаження PDF файлів; AdminService містить логіку адміністративного управління користувачами та даними системи; BudgetService, CommunityService та RegionService працюють з відповідними типами даних у базі.

Middleware (middleware) містить проміжне програмне забезпечення для обробки запитів. Модуль auth.middleware.js реалізує три основні middleware-функції: authenticate для обов'язкової перевірки JWT токена та ідентифікації користувача, optionalAuthenticate для опціональної автентифікації, яка дозволяє деяким ендпоінтам працювати як з авторизованими, так і з неавторизованими

користувачами, та `authorize` для перевірки прав доступу на основі ролі користувача, що забезпечує контроль доступу до адміністративних функцій.

Модуль підключення до бази даних (`db.js`) ініціалізує пул з'єднань з PostgreSQL за допомогою бібліотеки `pg`, налаштовує параметри підключення на основі змінних середовища та експортує пул для використання в сервісах. Це забезпечує ефективне управління з'єднаннями з базою даних та дозволяє обробляти одночасно кілька запитів.

Головний файл сервера (`index.js`) ініціалізує Express додаток, налаштовує `middleware` для обробки JSON, CORS та підключає всі маршрути API. Він також містить ендпоінт `health check` для моніторингу стану сервера та запускає HTTP сервер на вказаному порту.

Директорія `sql` містить SQL скрипти для міграцій та оновлення структури бази даних. Ці скрипти використовуються для додавання нових таблиць, колонок або зміни існуючих структур даних без необхідності ручного виконання SQL команд.

Директорія `sync` містить модулі синхронізації даних з зовнішніми джерелами. Основний модуль `syncOpenBudget.cjs` реалізує ETL процес для завантаження даних з OpenBudget API, включаючи трансформацію даних та їх збереження в локальну базу даних.

Допоміжні функції винесені в директорію `scripts`. Тут розміщено інструменти для імпорту довідників, обробки помилкових синхронізацій, отримання геолокації з MapBox API, синхронізації даних по роках та масового завантаження структур бюджетів. Ці скрипти виконуються вручну або можуть бути інтегровані в `cron`-завдання. Деталізація та призначення ключових модулів та файлів серверної частини відображена в Таблиці 8.2.

Таблиця 8.2 — Головні модулі та файли серверної частини

Модуль/Файл	Призначення
<code>index.js</code>	центральна точка входу. Ініціалізує Express, підключає маршрути, сервісні модулі, логування, робить з'єднання з PostgreSQL та запускає веб-сервер

Модуль/Файл	Призначення
controllers/	містить контролери, що реалізують бізнес-логіку обробки HTTP-запитів. Викликають відповідні сервіси й формують HTTP-відповідь
controllers/ForecastController.js	керує процесом прогнозування: отримує параметри, перевіряє кеш, запускає сервіс прогнозування, повертає клієнту фінальний результат
controllers/StructureController.js	забезпечує доступ до структури бюджету (budget_structure) та різних типів класифікацій
controllers/SyncController.js	керує запуском та контролем процесів синхронізації з OpenBudget API
services/	містить модулі, що взаємодіють з базою даних, виконують агрегації, обчислення та отримання інформації
services/BudgetService.js	робота з метаданими бюджетів: пошук, фільтрація, оновлення, зв'язки з громадами
services/StructureStatsService.js	виконує складні SQL-агрегації для статистики: помісячні, поквартальні, річні показники, топ-10, структура видатків, динаміка
services/ForecastService.js	реалізація математичних методів прогнозування (арифметичний приріст, ковзне середнє, експоненційне згладжування, лінійна регресія)
services/ForecastCacheService.js	відповідає за кешування прогнозів у таблиці forecast_cache для швидкої відповіді на повторні запити
services/SyncService.js	головний модуль синхронізації: завантажує дані з OpenBudget API, обробляє, чистить і записує до budget_structure
routes/	директорія з усіма маршрутами (API endpoints), що пов'язують контролери з HTTP-запитами клієнта
scripts/syncOpenBudget.cjs	головний автоматизований скрипт збору даних з OpenBudget. Використовується за розкладом або вручну
scripts/fetchCoordinatesMapbox.cjs	скрипт отримання географічних координат громад через MapBox API

8.4.2 Взаємодія модулів

Взаємодія модулів серверної частини реалізована відповідно до шаблону MVC, де кожний компонент виконує чітко визначену роль. Під час обробки HTTP-запиту від клієнта процес проходить декілька логічних етапів, забезпечуючи розділення відповідальності, ефективну обробку даних та масштабованість системи.

Загальна послідовність взаємодії модулів виглядає так:

- a) клієнт надсилає HTTP-запит до API (наприклад, запит на отримання структури бюджету або прогнозу);
- b) маршрутизатор (routes) приймає запит і визначає, який контролер відповідає за його обробку;
- c) контролер (controllers) виконує перевірку параметрів, ініціює звернення до відповідного сервісного модуля і визначає формат даних у відповіді;
- d) сервісний модуль (services) звертається до бази даних, виконує обчислення, агрегації або інші бізнес-операції;
- e) за потреби використовується кеш прогнозів (ForecastCacheService), який дозволяє повертати результат без повторних обчислень;
- f) сервіс повертає результат у контролер, який формує фінальну структуру відповіді;
- g) контролер повертає відповідь клієнту у форматі JSON.

На рисунку 8.3 наведено діаграму взаємодії модулів.

Діаграма взаємодії модулів системи

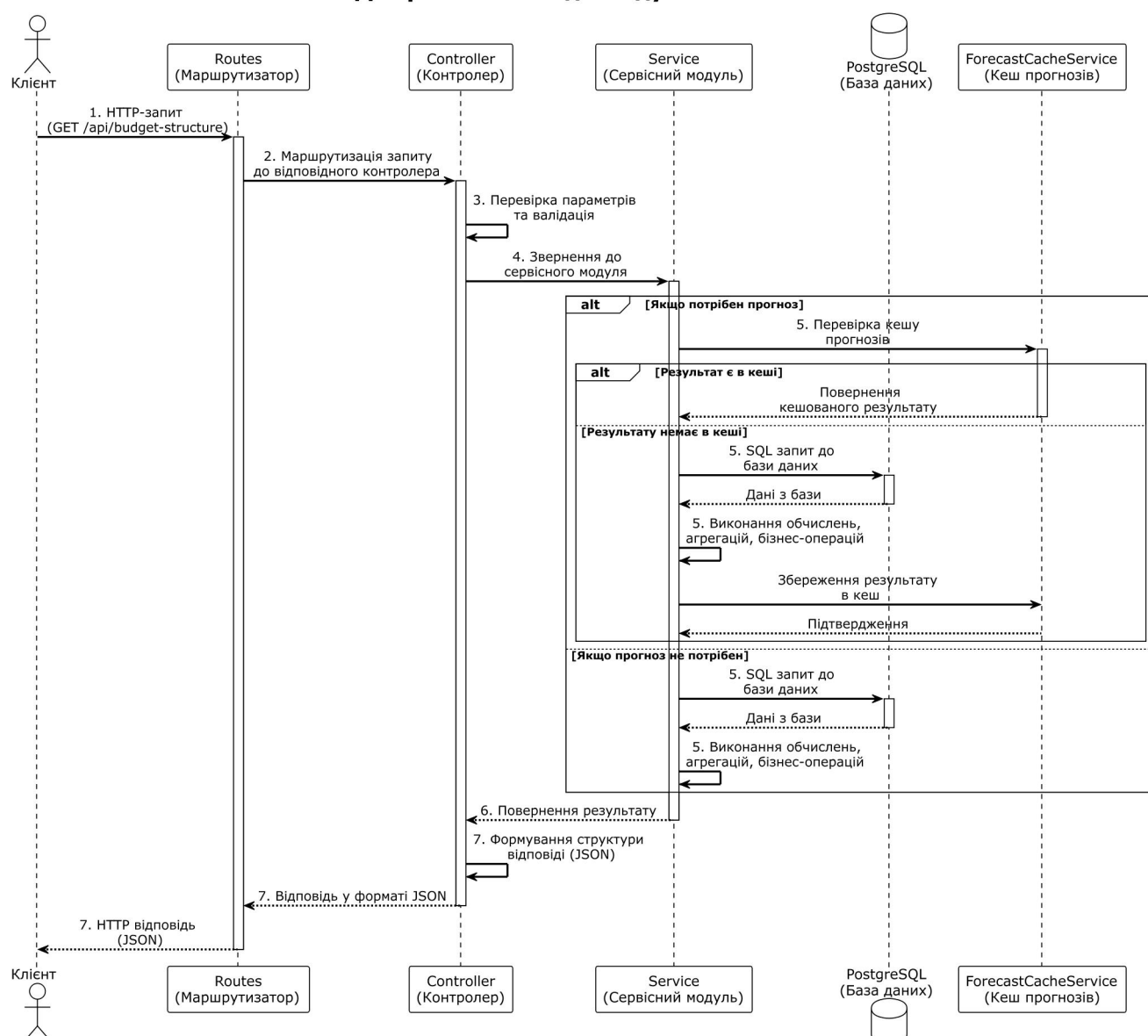


Рисунок 8.3 — Діаграма взаємодії модулів системи

Усі операції з PostgreSQL виконуються виключно у сервісних модулях. Це забезпечує централізацію доступу до даних і спрощує оптимізацію SQL-запитів. Оскільки таблиця `budget_structure` містить сотні тисяч записів, сервер активно використовує агрегацію в SQL, CTE (WITH), фільтрацію за останнім доступним місяцем та віконні функції[16]. Обчислення максимально переносяться на рівень бази даних, що суттєво знижує навантаження на сервер і дозволяє швидко отримувати підсумкові статистичні дані.

Система прогнозування реалізована в модулі ForecastService, де застосовано кілька математичних методів: середній приріст, ковзне середнє, експоненціальне згладжування та лінійну регресію. Результати прогнозів зберігаються у таблиці forecast_cache. Це дає змогу уникати повторних обчислень і прискорює відповіді API. При повторному виклику прогноз повертається з кешу, якщо не було запиту на примусове оновлення.

Синхронізація даних виконується через модуль SyncService, який звертається до відкритого державного API, отримує дані бюджетної структури по визначених бюджетах, зберігає дані в таблицю budget_structure і фіксує статуси операцій в таблиці api_sync. Разом з відповідними скриптами цей модуль забезпечує регулярне оновлення бюджетної структури та можливість повторного виконання невдалих синхронізацій. Такий підхід дозволяє підтримувати актуальний стан даних у системі та забезпечує надійність процесу оновлення.

8.5 Діаграма класів

Діаграма класів, що зображена на кресленику в Додатку Ж, відображає структуру системи та взаємозв'язки між компонентами.

Архітектура розділена на кілька пакетів. Пакет Backend - Controllers містить контролери, які обробляють HTTP-запити: AuthController для автентифікації та управління профілем, ReportsController для генерації та завантаження PDF звітів, ForecastController для прогнозування, SyncController для синхронізації даних, AdminController для адміністративного управління, а також контролери для роботи з бюджетами, структурами, регіонами та громадами.

Пакет Backend - Services містить сервіси з бізнес-логікою. UserService забезпечує управління користувачами, включаючи створення, пошук, оновлення та видалення облікових записів. AuthService відповідає за генерацію та валідацію JWT токенів, управління сесіями користувачів. ReportsService реалізує генерацію PDF звітів з використанням PDFKit, інтегрується зі StructureStatsService та ForecastService для отримання статистичних даних та прогнозів. ForecastService

містить статичні методи для різних алгоритмів прогнозування: арифметичне зростання, ковзне середнє, експоненційне згладжування та лінійну регресію. ForecastCacheService забезпечує кешування результатів прогнозування для оптимізації продуктивності. StructureStatsService надає статичні методи для обчислення місячної, квартальної та річної статистики, а також топ-10 кодів та динаміки змін. S3Service реалізує інтеграцію з AWS S3 для завантаження, завантаження та видалення PDF файлів, а також генерацію presigned URL для приватних файлів. SyncService керує процесом синхронізації даних з OpenBudget API через запуск окремих скриптів.

Пакет Backend - Middleware містить AuthMiddleware, який реалізує три функції: authenticate для обов'язкової перевірки JWT токена, optionalAuthenticate для опціональної автентифікації та authorize для перевірки прав доступу на основі ролі користувача.

Пакет Backend - Database містить DatabasePool, який забезпечує пул з'єднань з PostgreSQL та виконує SQL запити. Всі сервіси використовують DatabasePool для доступу до бази даних.

Пакет Frontend - Context містить AuthContext, який керує станом автентифікації в React додатку, зберігає інформацію про користувача та токен, надає методи для входу, реєстрації та виходу, а також формує заголовки авторизації для HTTP запитів.

Взаємозв'язки між компонентами показують, що контролери використовують сервіси для виконання бізнес-логіки, сервіси взаємодіють з базою даних через DatabasePool, а ReportsService інтегрується з іншими сервісами для отримання необхідних даних. Middleware використовує AuthService та UserService для перевірки автентифікації, а Frontend через AuthContext взаємодіє з Backend контролерами через HTTP -запити.

Така архітектура забезпечує розділення відповідальностей, де контролери обробляють HTTP запити, сервіси містять бізнес-логіку, а база даних зберігає дані. Це спрощує підтримку та розширення системи.

8.6 Взаємодія з OpenBudget API

Інформаційна система інтегрується з відкритим державним сервісом OpenBudget API, який надає доступ до детальних даних про місцеві бюджети України. Ця інтеграція є ключовим елементом роботи системи, оскільки саме з цього джерела надходять первинні транзакційні дані про видатки, плани та виконання бюджетів за всіма класифікаціями. Система не генерує бюджетну інформацію самостійно, а працює як аналітичний шар над офіційними даними державного порталу.

Взаємодія з OpenBudget API реалізована через спеціалізований сервісний модуль SyncService, який формує HTTP-запити до ендпоінтів[17] державного API, отримує CSV-файли, які перетворює у оброблені масиви JSON-даних і зберігає їх у локальну базу даних. Запити виконуються для кожного бюджету окремо, із врахуванням року, типу класифікації та періодичності (у системі використовується помісячна деталізація).

Перед збереженням дані проходять нормалізацію: формується таблиця `budget_structure`, у якій кожен рядок відповідає певному коду класифікації, бюджетному періоду та значенням план/факт. Оскільки OpenBudget подає дані у накопичувальному форматі, логіка синхронізації враховує лише останній доступний місяць для кожного року. Це дозволяє уникати дублювання і забезпечує коректне обчислення річних підсумків та динаміки.

Механізм синхронізації реалізований як послідовність операцій. Спочатку система визначає, для яких бюджетів необхідно отримати актуальні дані. Далі формується URL запити до API, який включає код бюджету, тип видатків, періодичність та рік. Після отримання відповіді здійснюється перевірка коректності структури даних. Далі інформація очищується від дублювань і зберігається в таблицю `budget_structure`.

Кожен виклик фіксується в таблиці `api_sync`, де зберігається час виконання операції, статус, кількість отриманих рядків і, в разі помилки, її опис. Це забезпечує прозорість і можливість подальшого аналізу стабільності синхронізації.

У випадку некоректної роботи державного API, синхронізація не переривається глобально — система дозволяє повторно виконувати запити через окремі службові скрипти. Оновлення бюджетних структур може бути викликане як адміністратором вручну, так і виконуватися автоматично через планувальник завдань.

Отримані з OpenBudget API дані слугують основою для всіх аналітичних модулів. На їх основі формується статистика за роками, кварталами та місяцями; обчислюється структура видатків та ТОП-10 бюджетних статей; створюються вхідні ряди для прогнозування бюджетних показників; формуються PDF-звіти, які користувач може завантажити.

Головною перевагою інтеграції є актуальність і достовірність інформації. Оскільки OpenBudget є офіційним державним ресурсом, усі дані, що використовує система, відповідають фактичним показникам виконання бюджетів.

8.7 Генерація PDF-звітів

Функціональність формування PDF-звітів є одним із ключових інструментів системи, оскільки дозволяє користувачам отримувати структуровані аналітичні документи щодо бюджетів громад. Звіт формується автоматично на основі актуальних даних, збережених у локальній базі, включаючи річні підсумки, структуру видатків, статистику за класифікаціями та дані прогнозування. Завдяки цьому користувач отримує повноцінний аналітичний документ, який може бути використаний для внутрішньої роботи, презентацій, моніторингу або контролю виконання бюджетних програм.

Генерація PDF реалізована на серверній частині як окремий модуль ReportService. Він відповідає за підготовку даних, формування макета сторінок і експорт у PDF-формат. Щоб забезпечити високу якість відтворення графічних елементів (діаграм, таблиць, інфографіки), система використовує HTML-шаблони, які рендеряться у PDF за допомогою pdf-lib.

Під час формування звіту сервіс отримує дані з кількох модулів: статистичні показники з `StructureStatsService`, прогнозні ряди з `ForecastService`, метадані бюджету з таблиці `budget`, а також структуру видатків, сформовану у вигляді діаграм. Усі дані агрегуються в єдину модель, яка підставляється в HTML-шаблон.

PDF формується на сервері як тимчасовий файл, після чого завантажується до S3-сховища. Посилання на сформований документ зберігається в таблиці `user_reports`, а сам файл може бути переглянутий або завантажений користувачем у будь-який момент.

Для зберігання PDF-файлів використовується об'єктне сховище S3-сумісного типу. Такий підхід дозволяє не перевантажувати сервер та уникнути збереження великих бінарних об'єктів у базі даних. Після завершення рендерингу PDF завантажується у вказаний bucket через S3-SDK, а система отримує постійне унікальне посилання.

У таблиці `user_reports` зберігається ідентифікатор користувача, бюджет, рік, назва звіту, параметри генерації та URL-адреса файлу. Оскільки звіти можуть містити великі обсяги аналітичної інформації, вони є незмінними після генерації, а повторне формування здійснюється лише за потреби користувача.

Після генерації PDF звіт стає доступним у кабінеті користувача. Інтерфейс React отримує список звітів через REST-ендпоінт, а також дозволяє відкрити PDF у новій вкладці або завантажити його локально. Усі звіти прив'язані до конкретного користувача, що забезпечує приватність документів та логічну роздільність даних.

Користувач отримує історію згенерованих документів, що дозволяє порівнювати зміни між роками, аналізувати ефективність бюджетних програм і фіксувати динаміку видатків. Завдяки цьому PDF-звіт стає повноцінним зовнішнім аналітичним інструментом, який доповнює інтерактивний інтерфейс системи.

8.8 Діаграма послідовності для генерації звітів

Діаграма послідовності, яку наведено на кресленику в Додатку И, описує процес генерації PDF звіту про бюджет регіону або територіальної громади.

Процес починається з дій користувача у веб-інтерфейсі: вибір параметрів (код бюджету, тип, рік, опції) та натискання кнопки "Генерувати". Frontend формує HTTP POST запит до `/api/reports/generate` з параметрами та JWT токеном в заголовку `Authorization`.

Backend отримує запит і виконує `middleware` автентифікації, який перевіряє валідність JWT токена та ідентифікує користувача. Після успішної автентифікації запит передається до `ReportsController`, який валідує обов'язкові параметри.

`ReportsController` викликає `ReportsService` для генерації PDF. `ReportsService` виконує послідовні запити до `StructureStatsService` для отримання статистичних даних. Спочатку запитується місячна статистика через SQL запит до PostgreSQL, який повертає дані про фактичні та планові видатки по місяцях. Потім аналогічно запитується квартальна статистика, яка містить агреговані дані по кварталах. Далі запитується річна статистика для загального огляду за рік.

Якщо користувач включив опцію прогнозування, `ReportsService` викликає `ForecastService` для генерації прогнозних даних. `ForecastService` виконує SQL запит до бази даних для отримання історичних даних за попередні періоди, після чого виконує обчислення прогнозу за обраними методами (арифметичне зростання, ковзне середнє, експоненційне згладжування або регресійний аналіз). Результати прогнозування повертаються до `ReportsService`.

Після отримання всіх необхідних даних `ReportsService` створює PDF документ за допомогою бібліотеки `PDFKit`. Процес включає ініціалізацію документа, додавання заголовків, таблиць зі статистикою, графіків та діаграм, які візуалізують дані. Якщо включено прогнозування, до документа додаються розділи з прогнозованими даними та графіками трендів. Після завершення формування PDF документ конвертується в буфер байтів, який повертається до `ReportsController`.

Якщо користувач вказав назву звіту (`report_name`), `ReportsController` викликає метод `saveReportWithPDF` для збереження звіту. Цей метод спочатку завантажує PDF файл в AWS S3, використовуючи унікальний ключ, сформований на основі ідентифікатора користувача, дати та назви звіту. Після успішного завантаження S3 повертає URL файлу. Далі в базу даних PostgreSQL вставляється запис в таблицю `user_reports`, який містить ідентифікатор користувача, код бюджету, рік, назву звіту, URL файлу в S3, параметри генерації та прапорець публічності звіту. Після збереження повертаються метадані створеного звіту.

`ReportsController` повертає PDF буфер та метадані до Backend, який формує HTTP відповідь з кодом 200 OK, встановлює заголовок `Content-Type` як `application/pdf` та відправляє PDF файл до Frontend.

Frontend отримує PDF файл у відповіді, створює `blob` об'єкт з отриманих даних та ініціює автоматичне завантаження файлу на пристрій користувача. Після успішного завантаження користувач отримує повідомлення про успішну генерацію звіту, а якщо звіт було збережено, він також оновлює список власних звітів для відображення нового запису.

Діаграма послідовності демонструє взаємодію між компонентами системи, показуючи послідовність викликів методів, активацію об'єктів під час виконання операцій та альтернативні шляхи виконання залежно від наявності опцій прогнозування та збереження звіту. Це забезпечує повне розуміння архітектури процесу генерації звітів та дозволяє ідентифікувати потенційні місця оптимізації або розширення функціональності.

8.9 Діаграма діяльності

Діаграма діяльності, що наведена на кресленику в Додатку К, описує послідовність дій для авторизованого користувача та адміністратора.

Робота починається з відкриття веб-додатку. Якщо користувач не авторизований, система запитує email та пароль. Після введення даних запит надсилається на сервер, де виконується перевірка облікових даних. При успішній

автентифікації сервер генерує JWT токен, який зберігається в сесії користувача, після чого відбувається перенаправлення на головну сторінку. Якщо дані некоректні, користувач отримує повідомлення про помилку та може спробувати знову.

Після автентифікації користувач має доступ до інтерактивної карти регіонів України, побудованої на базі MapBox API. При кліку на регіон або територіальну громаду система відображає панель з інформацією про вибраний бюджет, включаючи дані про бюджет, видатки та статистику. В окремій вкладці користувач може переглянути прогноз по витратах на наступний рік з вибраним методом прогнозування.

Одна з основних функцій — генерація звітів. Користувач переходить до генератора звітів, де обирає параметри: регіон, рік, тип бюджету, додаткові фільтри та опції візуалізації. Після налаштування параметрів запит надсилається на сервер, який створює PDF звіт на основі вибраних даних. Згенерований звіт зберігається в AWS S3 для надійного зберігання, а метадані про звіт записуються в базу даних PostgreSQL. Користувач отримує посилання на згенерований звіт та може його завантажити або поділитися ним.

Система також надає можливість перегляду звітів. Користувач може перейти до розділу перегляду, де система завантажує список доступних звітів. Публічні звіти відображаються всім користувачам, тоді як приватні звіти доступні тільки їх авторам. При виборі звіту система завантажує PDF з S3 або регенерує його, якщо файл недоступний, після чого відображає звіт користувачу. Приватний звіт користувач може видалити.

Управління профілем дозволяє користувачу оновлювати свої дані, зокрема email та повне ім'я. При зміні даних система відправляє запит на сервер, який зберігає оновлену інформацію в базі даних. Користувач також може змінити пароль, для чого необхідно ввести поточний пароль, новий пароль та підтвердити його. Система перевіряє валідність нового пароля, після чого хешує його та оновлює в базі даних. Після завершення роботи користувач може вийти з системи, що призводить до видалення токена з сесії та завершення авторизованого сеансу.

Адміністратор має доступ до всього функціоналу звичайного користувача, а також до додаткових адміністративних функцій через спеціальну адмін панель. Після входу в систему адміністратор переходить до адмін панелі, де має доступ до трьох основних розділів: управління користувачами, управління синхронізацією та модерація структури бюджетів.

У розділі управління користувачами адміністратор може переглядати список всіх зареєстрованих користувачів системи та їх інформацію. Адміністратор має можливість редагувати дані будь-якого користувача, змінюючи email, повне ім'я або роль користувача, причому зміни зберігаються в базі даних. Також адміністратор може змінювати паролі користувачів без необхідності знання поточного пароля, що корисно у випадках втрати доступу. Адміністратор може переглядати звіти будь-якого користувача та видаляти їх при необхідності. Видалення відбувається як з бази даних, так і з S3 сховища. У випадках порушення правил або інших причин адміністратор може видалити користувача з системи, що призводить до видалення всіх пов'язаних даних користувача.

Управління синхронізацією дозволяє адміністратору контролювати процес завантаження даних з зовнішнього OpenBudget API. Адміністратор може переглядати результати попередніх синхронізацій, аналізувати їх статуси та виявляти помилки. При необхідності адміністратор може запустити нову синхронізацію, обравши рік, для якого потрібно завантажити дані. Після запуску система підключається до OpenBudget API, завантажує актуальні дані про бюджети регіонів та територіальних громад, обробляє та трансформує дані відповідно до внутрішньої структури системи, зберігає їх в PostgreSQL та оновлює статус синхронізації. Результати синхронізації відображаються адміністратору для аналізу та моніторингу.

Модерація структури бюджетів надає адміністратору можливість керувати структурними елементами бюджетів у системі. Адміністратор може переглядати список всіх структур бюджетів, створювати нові структури, вводячи необхідні дані та зберігаючи їх в базі даних. Також можливе редагування існуючих структур, що дозволяє оновлювати інформацію про бюджетні категорії, статті витрат та

інші структурні елементи. При необхідності адміністратор може видаляти структури бюджетів, після підтвердження дії система видаляє відповідні записи з бази даних.

Після виконання адміністративних дій адміністратор може повернутися до основного функціоналу системи та використовувати її як звичайний користувач, маючи доступ до всіх функцій перегляду карт, генерації звітів та управління профілем.

Діаграма діяльності демонструє послідовність операцій, умовні переходи між станами системи та логіку прийняття рішень на різних етапах роботи користувачів. Вона відображає як базовий функціонал для авторизованих користувачів, так і розширені можливості адміністративного управління системою.

Висновки до розділу 8

У межах розділу було здійснено повний цикл реалізації інформаційної системи, який охопив побудову клієнтської та серверної частин, створення механізмів збору й обробки даних, інтеграцію з зовнішніми API та організацію сервісних функцій, необхідних для надійної роботи платформи.

Використання клієнт-серверної архітектури забезпечило логічне розділення функціональних рівнів і дозволило створити гнучкий та продуктивний механізм взаємодії між інтерфейсом користувача та серверною логікою. Клієнтська частина, побудована на React, реалізує інтерактивний інтерфейс, підтримує роботу з картою, аналітичними графіками та формами для створення звітів. Серверна частина, реалізована на Node.js із застосуванням Express, забезпечує надійну бізнес-логіку, обробку даних, взаємодію з PostgreSQL, формування прогнозів та генерацію PDF-документів.

Особливу увагу було приділено модульності: маршрути, контролери, сервіси та службові скрипти чітко відокремлені, що підвищує масштабованість та зручність супроводу. Інтеграція з OpenBudget API дала змогу автоматизувати

наповнення даними, а синхронізаційний модуль гарантує актуальність інформації в системі.

Важливою складовою стали інструменти глибокої аналітики: модуль прогнозування, підготовлений за допомогою математичних методів; формування статистики; механізми побудови структури видатків; динаміка виконання бюджетів за роками. Завдяки їм система надає не лише агреговані дані, а й створює повноцінний аналітичний контекст.

Генерація PDF-звітів із подальшим збереженням у S3-сховищі перетворює систему на інструмент, придатний для практичної роботи фінансових відділів, органів місцевого самоврядування та аналітичних служб. Додатково була реалізована система логування та моніторингу, що забезпечує прозорість дій користувачів і стабільність роботи платформи.

У результаті побудована інформаційна система є комплексним рішенням, яке поєднує сучасні веб-технології, механізми інтерактивної аналітики та автоматизоване отримання державних фінансових даних. Система відповідає вимогам масштабованості, надійності та розширюваності, що створює основу для подальшого розвитку та впровадження додаткових функціональних можливостей.

9 ОПИС ІНТЕРФЕЙСУ КОРИСТУВАЧА

Для перевірки працездатності та правильності функціонування розробленої інформаційної системи було проведено комплексне тестування її основних інтерфейсів і функцій. Демонстрація здійснювалася у веб-браузері, оскільки система повністю працює як веб-застосунок і не потребує встановлення додаткового програмного забезпечення. Доступ до системи надається через сервер, що працює в контейнерах Docker та виконує всі операції з базою даних, зовнішніми API та генерацією звітів.

Після переходу за посиланням користувач потрапляє на головну сторінку, яка містить карту України з можливістю вибору територіальної громади. Інтерфейс побудовано таким чином, щоб основний функціонал був доступний без реєстрації, а додаткові можливості — створення звітів та їх збереження — відкривалися після авторизації. На рисунку 9.1 наведено головний екран системи, який бачить користувач після завантаження сторінки.

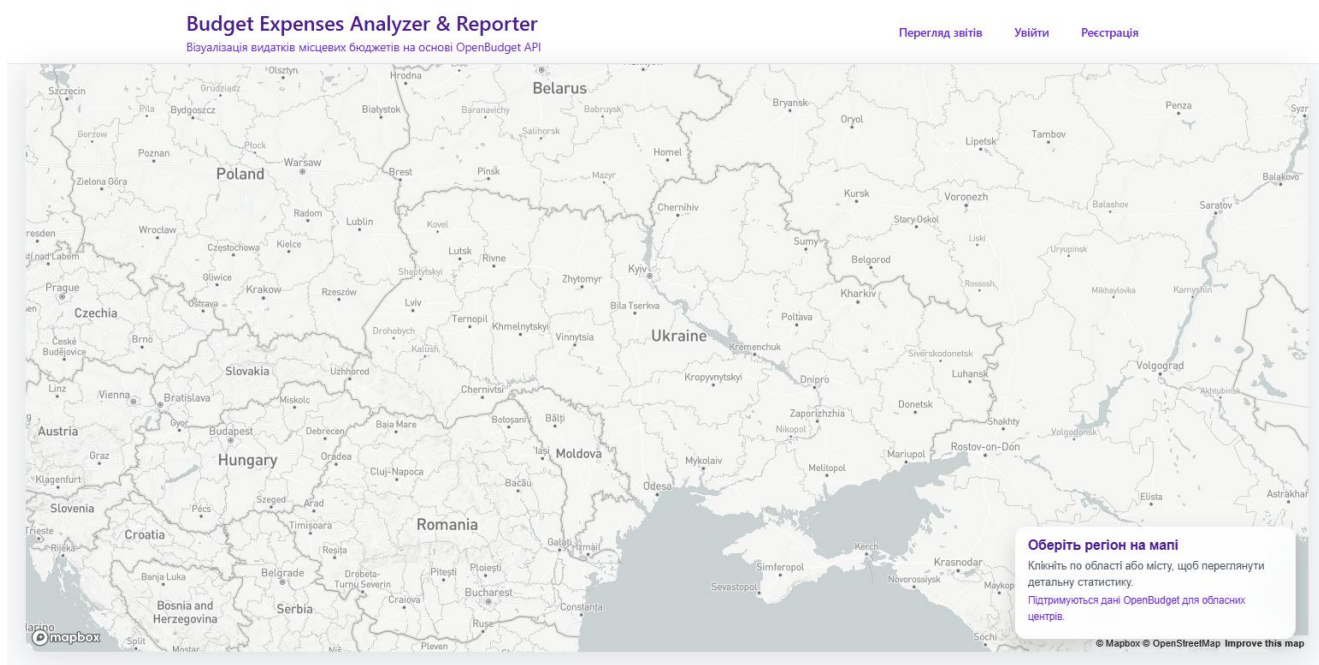


Рисунок 9.1 — Головний екран системи з інтерактивною картою

Карта є основним інструментом навігації. Користувач може збільшувати масштаб, пересуватися територією та натискати громади. Після вибору громади праворуч відкривається інформаційна панель, у якій відображається загальна довідка про вибрану адміністративну одиницю — назва, область, основні параметри. Користувач може обрати іншу громаду цієї ж області в цій панелі, тип класифікації та рік витрат. Приклад панелі огляду громади наведено на рисунку 9.2.

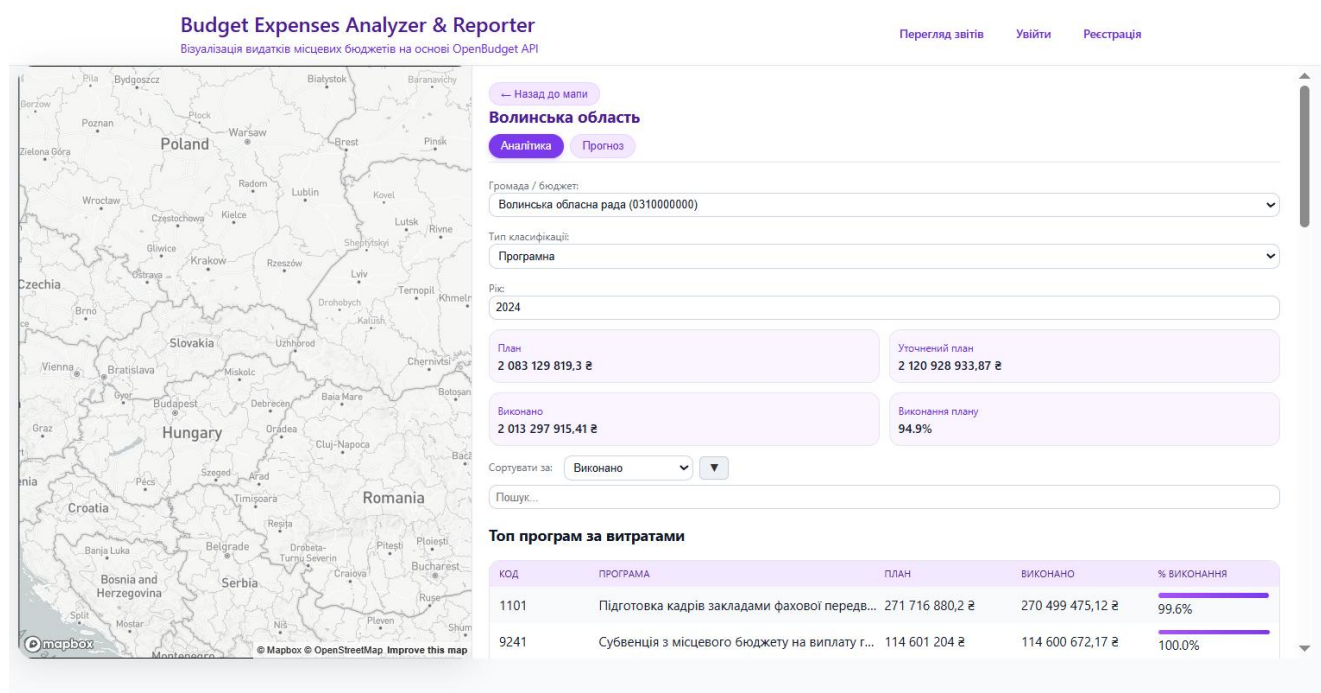


Рисунок 9.2 — Інформаційна панель вибраної громади

Далі в цій панелі користувач може переглядати статистику бюджету. Відображаються загальні витрати за рік: планові, планові з уточненням, фактичні витрати та відсоток виконання плану. Інтерфейс подає найбільші програми витратків у вигляді списку та графічного розподілу. Програми за видатками можна сортувати за різними показниками, а також шукати програму по назві. Це дозволяє швидко проаналізувати, на що громада витрачає найбільшу частину бюджету. На рисунку 9.3 наведено приклад відображення Топ програм за видатками у вигляді списку.

Топ програм за витратами

КОД	ПРОГРАМА	ПЛАН	ВИКОНАНО	% ВИКОНАННЯ
1101	Підготовка кадрів закладами фахової передв...	271 716 880,2 ₴	270 499 475,12 ₴	99,6%
9241	Субвенція з місцевого бюджету на виплату г...	114 601 204 ₴	114 600 672,17 ₴	100,0%
1033	Надання загальної середньої освіти спеціаліз...	92 192 200 ₴	91 519 248,77 ₴	99,3%
3242	Інші заходи у сфері соціального захисту і соц...	76 770 000 ₴	76 645 960,75 ₴	99,8%
9320	Субвенція з місцевого бюджету за рахунок з...	94 682 613 ₴	73 562 443,6 ₴	77,7%
1032	Надання загальної середньої освіти спеціаль...	68 708 000 ₴	68 672 687,05 ₴	99,9%
1022	Надання загальної середньої освіти спеціаль...	61 894 987,43 ₴	61 685 907,87 ₴	99,7%
9350	Субвенція з місцевого бюджету на забезпеч...	61 609 356 ₴	56 288 620,65 ₴	91,4%
7462	Утримання та розвиток автомобільних доріг ...	57 119 351,12 ₴	55 978 512,84 ₴	98,0%
9310	Субвенція з місцевого бюджету на здійсненн...	51 990 700 ₴	50 616 875,16 ₴	97,4%

Рисунок 9.3 — Відображення Топу програм за витратами

Наступними відображається Статистика бюджету, в якій відображено динаміку витрат за місяцями у протиставленні планові витрати та фактичні витрати, а також поквартальна статистика у аналогічному протиставленні (рисунок 9.4).



Рисунок 9.4 — Відображення Динаміки бюджету за місяцями та кварталами

Далі користувачу доступні графіки по структурі видатків у відсотках та топ-10 програмами за видатками, що продемонстровано на рисунку 9.5.

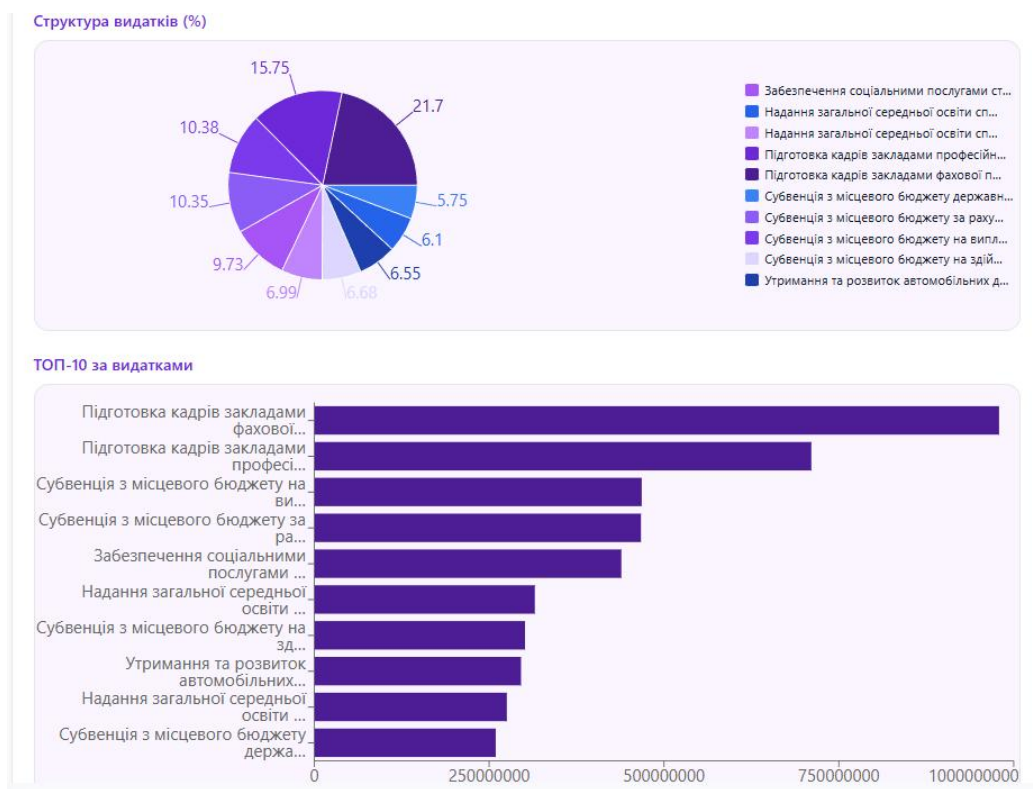


Рисунок 9.5 — Відображення структури видатків у відсотках та топ-10 видатків у вигляді графіків

Останнім графіком на сторінці статистики є динаміка бюджету громади за роками, що відображено на рисунку 9.6

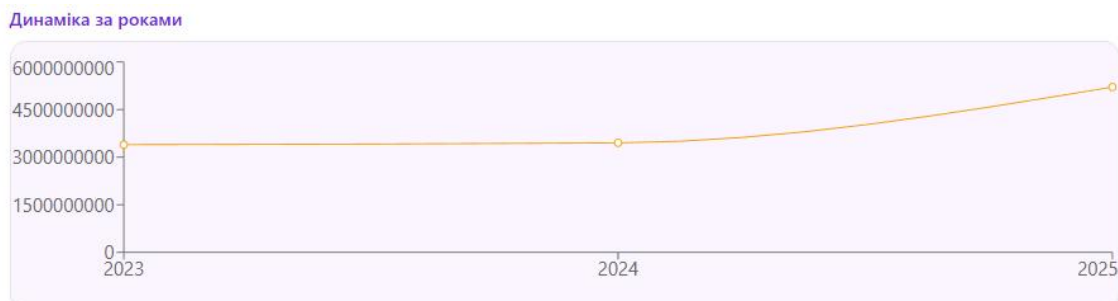


Рисунок 9.6 — Графік динаміки бюджету обраної громади за роками

При необхідності аналітичного прогнозу, користувач може перемкнутись на вкладку Аналітика, в якій відображено прогноз бюджету громади на наступний рік, за умови наявності інформації хоча б за два роки витрат громади (рисунок 9.7). На цій вкладці користувач так само може обрати іншу громаду з області зі списку, тип класифікації та останній рік, по якому буде зроблено прогноз.

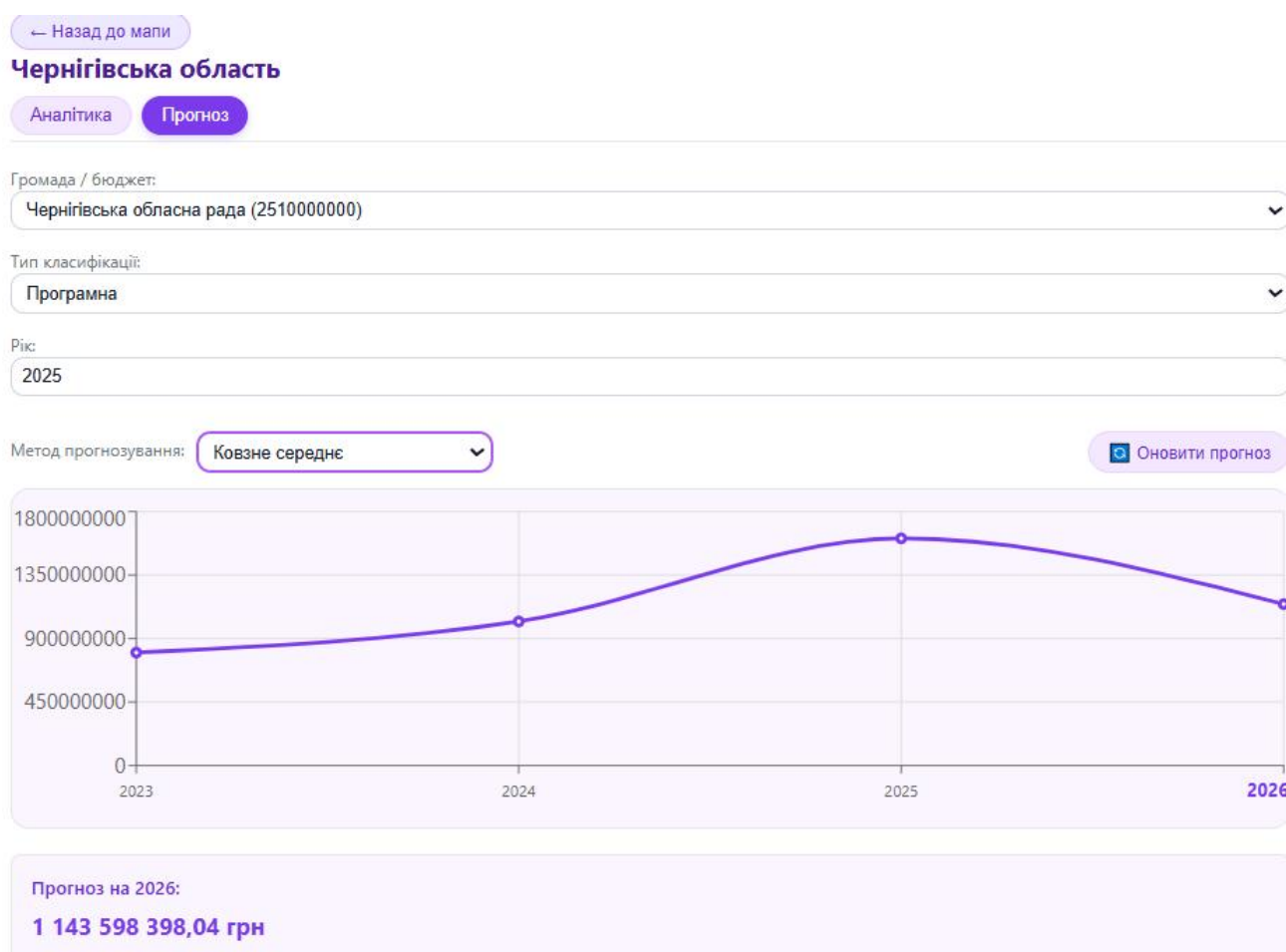


Рисунок 9.7 — Графік прогнозування бюджету громади на наступний рік

Для користувачів доступно чотири методи прогнозування та можливість примусово оновити прогноз (рисунок 9.8).



Рисунок 9.8 — Методи прогнозування бюджетів та примусове оновлення прогнозу

Неавторизований користувач також має змогу продивитись публічні доступні звіти по бюджетах (рисунок 9.9) та завантажити їх.

Budget Expenses Analyzer & Reporter

Візуалізація видатків місцевих бюджетів на основі OpenBudget API

Перегляд звітів

Увійти

Реєстрація

Перегляд звітів

Переглядайте публічні звіти та свої збережені звіти

Публічні звіти

Звіт2024

Публічний

Код бюджету: 1010000000

Рік: 2023

Створено: 5 грудня 2025 р.

Тип: program

Завантажити PDF

Звіт 2024 КиївОбл

Публічний

Код бюджету: 1010000000

Рік: 2024

Створено: 3 грудня 2025 р.

Тип: program

Завантажити PDF

ЗвітІваноРівнеОбл...

Публічний

Код бюджету: 1710000000

Рік: 2024

Створено: 3 грудня 2025 р.

Тип: program

Завантажити PDF

Звіт2023ВолиньОбл...

Публічний

Код бюджету: 0310000000

Рік: 2024

Створено: 3 грудня 2025 р.

Тип: program

Завантажити PDF

Рисунок 9.9 — Сторінка перегляду та завантаження звітів

Після натискання на кнопку завантаження звіт завантажиться користувачу у вигляді PDF-файлу (рис. 9.10).

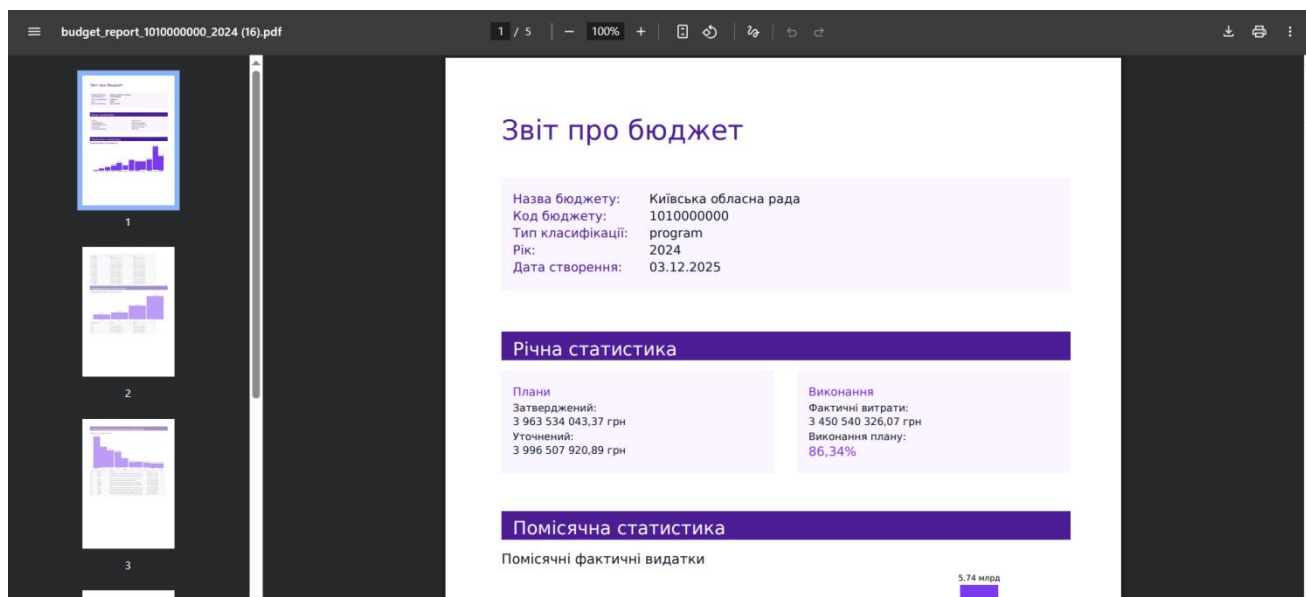


Рисунок 9.10 — Перегляд завантаженого звіту

Для того, щоб скористуватись іншими можливостями системи, користувач має зареєструватись (рис. 9.11).

Budget Expenses Analyzer & Reporter
Візуалізація видатків місцевих бюджетів на основі OpenBudget API

[Перегляд звітів](#) [Увійти](#) [Реєстрація](#)

Реєстрація

Створіть новий обліковий запис

Повне ім'я

Email

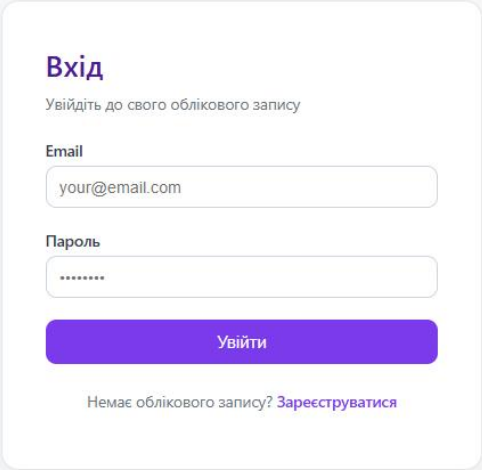
Пароль

[Зареєструватися](#)

Вже маєте обліковий запис? [Увійти](#)

Рисунок 9.11 — Сторінка реєстрації

Якщо у користувача вже існує профіль, він може увійти у нього на окремій сторінці (рисунок 9.12).



The image shows a login form titled "Вхід" (Login) with the subtitle "Увійдіть до свого облікового запису" (Log in to your account). It contains two input fields: "Email" with the placeholder "your@email.com" and "Пароль" (Password) with masked characters "*****". Below the fields is a blue button labeled "Увійти" (Log in). At the bottom, there is a link: "Немає облікового запису? [Зареєструватися](#)" (Don't have an account? [Register](#)).

Рисунок 9.12 — Сторінка входу до облікового запису

Після авторизації користувачу стають доступні нові функції. На сторінці Генератор звітів користувач може створити власний звіт для завантаження та збереження (рисунок 9.13). Користувач може обрати бюджет для звіту (за назвою, серед запропонованих по регіону, або ж за кодом бюджету), тип класифікації, рік та ввести назву звіту, якщо потрібно зберігати звіт в базі для подальшої роботи з ним.

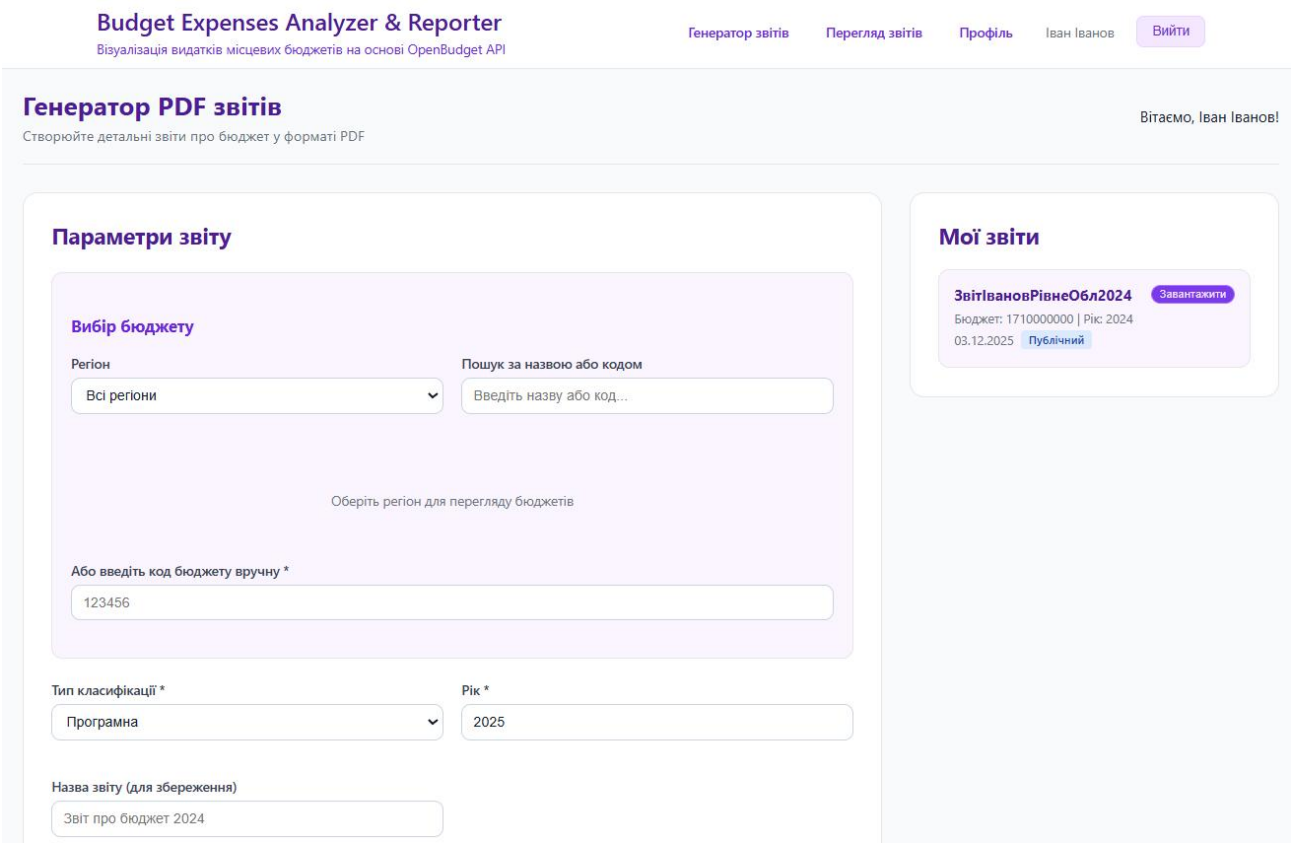


Рисунок 9.13 — Сторінка генерації PDF звітів

Окрім цих параметрів, користувач може обрати вміст звіту та параметри для прогнозування витрат (рисунок 9.14).

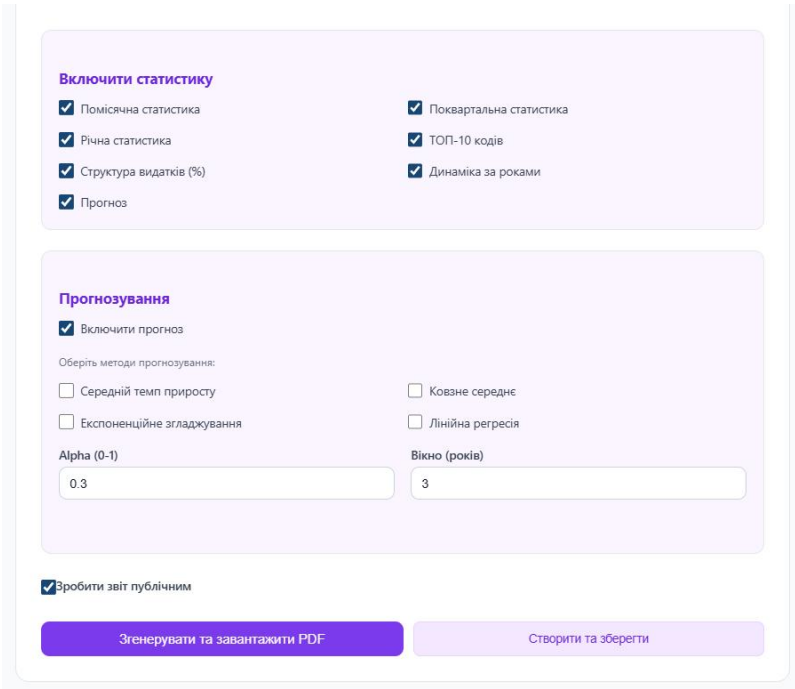


Рисунок 9.14 — Додаткові параметри для генерації звіту

Після створення користувач отримує сповіщення про успішне створення звіту та звіт стає доступним для завантаження користувачем в будь-який момент. Звіт можна завантажити на цій же вкладці (рис. 9.15).

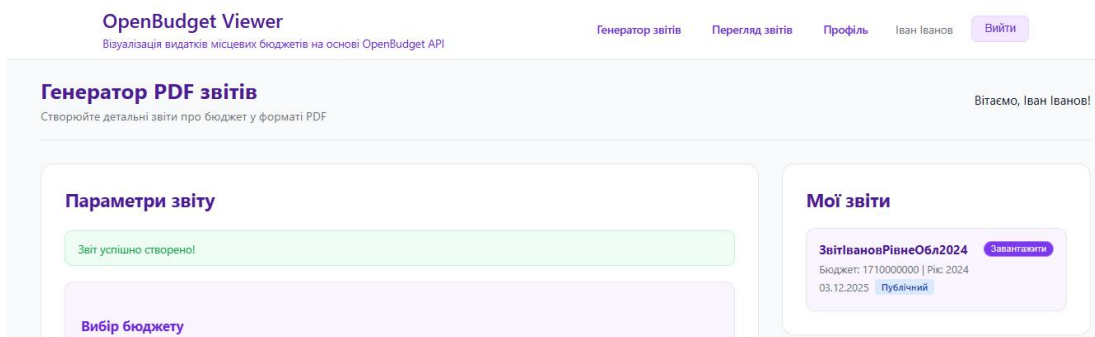


Рисунок 9.15 — Результат створення звіту

Окрім цього, користувач може переглянути та завантажити публічні та створені ним звіти у вкладці Перегляд звітів. В цій вкладці користувач може видалити власні звіти (рисунок 9.16).

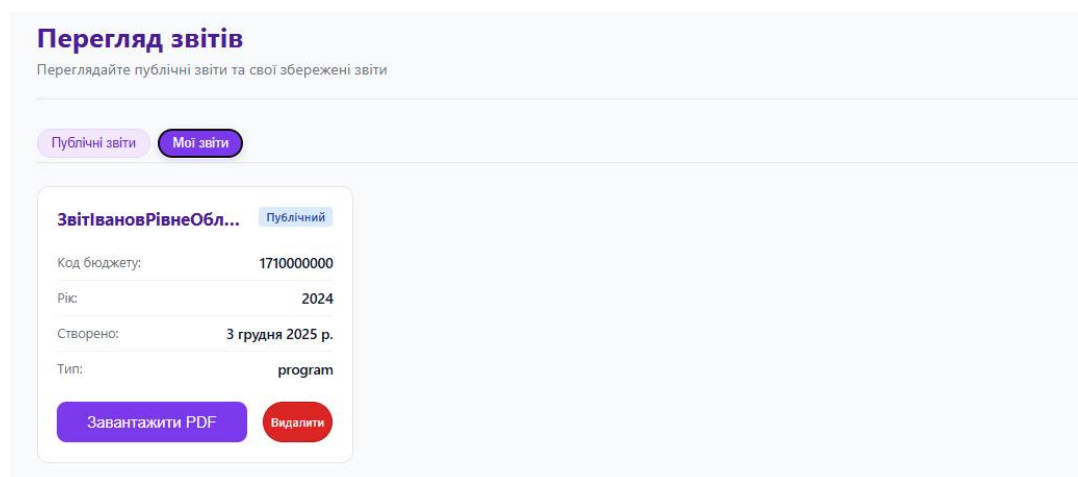


Рисунок 9.16 — Сторінка перегляду звітів користувача

За необхідністю, користувач може змінити власні дані на вкладці Профіль (рисунок 9.17).

Budget Expenses Analyzer & Reporter

Візуалізація видатків місцевих бюджетів на основі OpenBudget API

Генератор звітів

Перегляд звітів

Профіль

Іван Іванов

Вийти

Мій профіль

Керуйте своїм обліковим записом та налаштуваннями

Інформація профілю

Повне ім'я

Іван Іванов

Email

ivan.i@gmail.com

Роль

user

Роль може змінити тільки адміністратор

Зберегти зміни

Зміна паролю

Поточний пароль

Новий пароль

Мінімум 6 символів

Рисунок 9.17 — Сторінка Профілю користувача

Якщо увійти під роллю адміністратора, зверху з'являється Адмін панель, на якій доступна модерація системи. Адміністратор може здійснювати управління користувачами: змінювати їхні дані, видаляти звіти, або ж видаляти користувачів (рисунок 9.18).

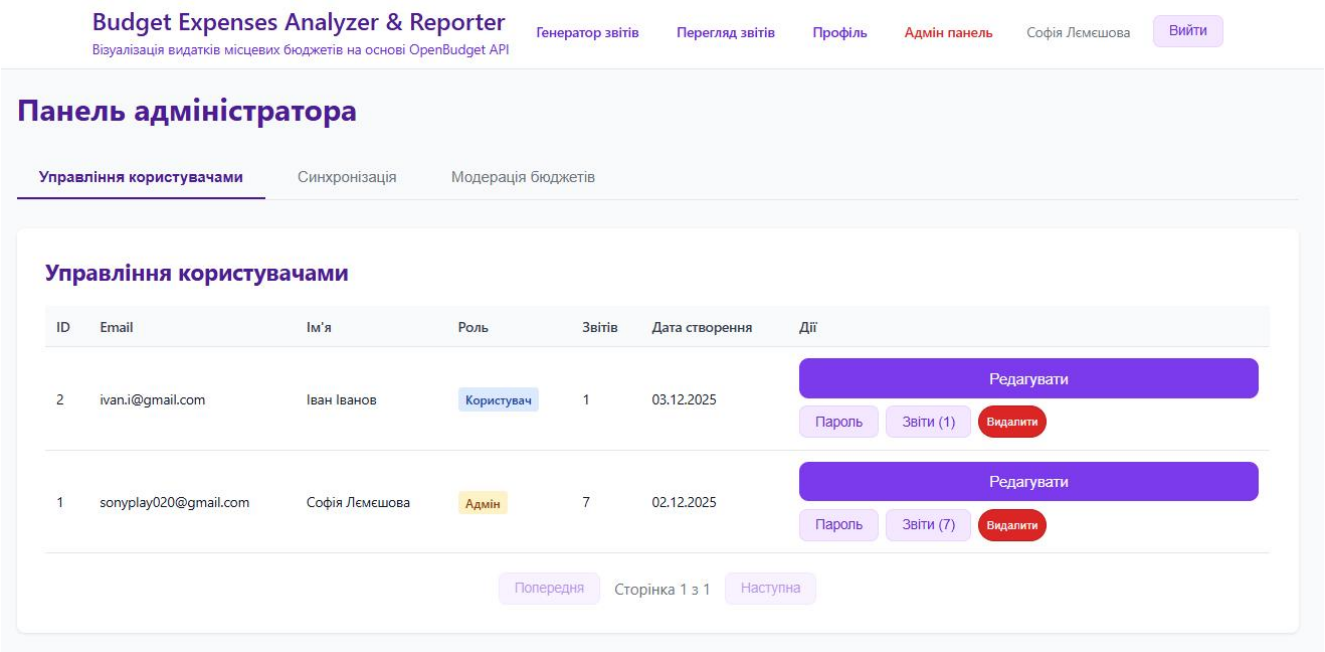


Рисунок 9.18 — Управління користувачами в Панелі адміністратора

Адміністратор може запустити примусову синхронізацію бюджетів з API, обравши рік для синхронізації, типи класифікації, періодичність, ліміт та на вибір код бюджету або ж код регіону (рисунок 9.19).

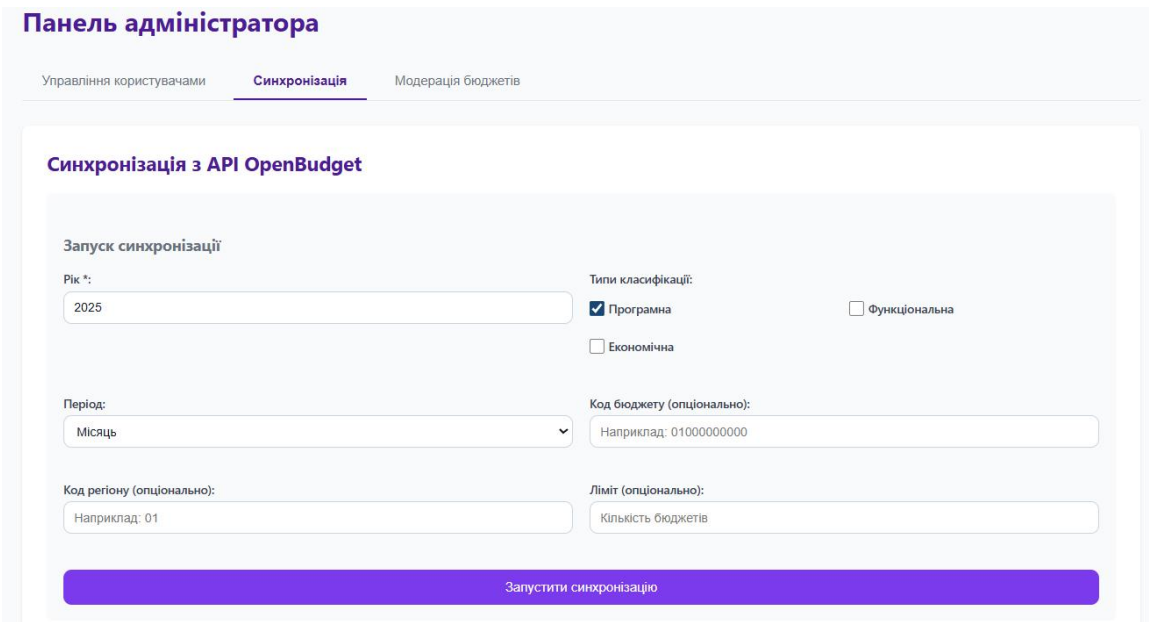


Рисунок 9.19 — Сторінка примусової синхронізації бюджетів

На цій же сторінці доступна табличка з інформацією про статус попередніх синхронізацій для спрощення роботи з системою при помилках (рисунок 9.20).

Результати синхронізації

Фільтр по endpoint...

Всі статуси

ID	Endpoint	Статус	Записів	Дата синхронізації	Деталі
46786	localBudgetData_program_2452300000_MONTH_2025	success	3342	03.12.2025, 14:21:55	Деталі
46785	localBudgetData_program_2451900000_MONTH_2025	success	2854	03.12.2025, 14:21:50	Деталі
46784	localBudgetData_program_2452200000_MONTH_2025	success	3429	03.12.2025, 14:21:48	Деталі
46783	localBudgetData_program_2451700000_MONTH_2025	success	2240	03.12.2025, 14:21:46	Деталі
46782	localBudgetData_program_2452600000_MONTH_2025	success	4865	03.12.2025, 14:21:45	Деталі
46781	localBudgetData_program_2452400000_MONTH_2025	success	2218	03.12.2025, 14:21:44	Деталі
46780	localBudgetData_program_2451800000_MONTH_2025	success	1976	03.12.2025, 14:21:43	Деталі
46779	localBudgetData_program_2452000000_MONTH_2025	success	3755	03.12.2025, 14:21:42	Деталі

Рисунок 9.20 — Перегляд результатів синхронізації

Остання вкладка, доступна для адміністратора — Модерація бюджетів, де за потреби адміністратор може змінювати або видаляти дані про вже наявні бюджети та додавати дані про нові бюджети (рисунок 9.21).

Панель адміністратора

Управління користувачами

Синхронізація

Модерація бюджетів

Модерація даних про бюджети

Увага! Ви збираєтесь змінити дані з API. Будьте обережні при редагуванні, видаленні або додаванні записів.

+ Додати запис

Фільтр по cod_budget...

Фільтр по rep_period...

Фільтр по cod_cons_mb_pk

Всі типи

ID	rep_period	cod_budget	cod_cons_mb_pk	Назва	zat_amt	plans_amt	fakt_amt	Тип	Дії
59665	12.2024	0210000000	0150	Організаційне, інформаційно-ан...	9 357 573	9 357 573	8 776 761.53	program	<div>Редагувати</div> <div>Видалити</div>
59657	12.2024	0210000000	0180	Інша діяльність у сфері держав...	130 435	130 435	128 288.57	program	<div>Редагувати</div> <div>Видалити</div>

Рисунок 9.21 — Сторінка Модерації даних про бюджети

Висновки до розділу 9

Тестування інтерфейсу користувача підтвердило коректну роботу всіх основних функцій системи та зручність взаємодії з бюджетними даними. Інтерактивна карта, панелі статистики, модулі прогнозування та генерації звітів працюють узгоджено й забезпечують повний цикл аналітичних операцій.

10 СТАРТАП ПРОЄКТ

Розділ зосереджується на маркетинговому аналізі та визначенні ринкових перспектив інформаційної системи, орієнтованої на збір, опрацювання та прогнозування витрат місцевих бюджетів. У межах дослідження розглядаються ключові маркетингові аспекти створення стартапу[18]: від етапу генерування та відбору ідей до формування концепції продукту, оцінювання потенціалу його виходу на ринок і розроблення стратегії просування. Запропонована система забезпечує можливості для комплексної візуалізації бюджетних даних, створення аналітичних звітів та побудови прогнозів, що робить її привабливою для органів місцевого самоврядування, громадських об'єднань, дослідників та усіх користувачів, зацікавлених у прозорості фінансових процесів на місцевому рівні.

Метою розділу є обґрунтування ринкової привабливості та визначення потенціалу комерціалізації розробки в умовах конкурентного середовища та глобалізаційних викликів. Основна увага приділяється аналізу можливостей упровадження запропонованого рішення, виявленню перспективних цільових сегментів, дослідженню конкурентної ситуації та формуванню ефективної маркетингової стратегії. Підсумком роботи має стати всебічна оцінка комерційного потенціалу системи та розроблення послідовного плану її успішного виходу на ринок.

10.1 Опис ідеї проєкту

Інформаційна система збору, аналізу та прогнозування витрат місцевих бюджетів — веб-додаток для візуалізації, аналізу та прогнозування бюджетних даних регіонів та територіальних громад України. Система інтегрується з офіційним OpenBudget API, надає інтерактивну карту, автоматичну генерацію звітів, прогнозування та аналітичні інструменти для різних категорій користувачів.

Основні компоненти системи та їх застосування представлені в таблиці 10.1.

Таблиця 10.1 — Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Інформаційна система збору, аналізу та прогнозування даних про витрати місцевих бюджетів	1. Візуалізація бюджетних даних на інтерактивній карті	Швидкий доступ до бюджетної інформації по регіонах, можливість перегляду та аналізу показників різних регіонів та громад
	2. Генерація детальних PDF звітів з місячною, квартальною та річною статистикою, графіками та таблицями	Економія часу на підготовку звітності, стандартизований формат представлення даних, можливість збереження та поділу звітами
	3. Прогнозування бюджетних витрат за допомогою математичних методів (арифметичне зростання, ковзне середнє, експоненційне згладжування, регресійний аналіз)	Підтримка прийняття рішень на основі даних, планування бюджетів на майбутні періоди, оцінка різних сценаріїв розвитку, зменшення ризиків фінансових помилок
	4. Аналіз структури та динаміки видатків	Глибоке розуміння структури витрат, виявлення тенденцій та аномалій, підтримка аналітичних досліджень, забезпечення прозорості даних про витрати
	5. Публічний доступ до звітів	Легкий доступ до публічної інформації про бюджети, можливість обміну аналітичними матеріалами, підвищення обізнаності про фінансову діяльність, підтримка громадського контролю
	6. Синхронізація з OpenBudget API	Економія часу на пошук та завантаження даних, автоматизація процесу збору інформації

Оцінювання конкурентоспроможності розроблюваної інформаційної системи збору, аналізу та прогнозування витрат місцевих бюджетів здійснено шляхом дослідження наявних українських платформ, що працюють з бюджетними даними. Основна мета такого аналізу — визначити сильні та слабкі

сторони існуючих рішень, оцінити їхню роль на ринку та окреслити унікальні можливості запропонованої системи.

Сегмент програмних продуктів для роботи з бюджетною інформацією в Україні сформований переважно державними офіційними порталами. Вони забезпечують доступ до первинних даних, проте зазвичай не пропонують розширених засобів глибокого аналізу, гнучкої візуалізації чи прогнозування. Така ситуація створює простір для появи інноваційних рішень, які поєднують надійні офіційні дані з сучасними інструментами аналітики, моделювання та прогнозування фінансових показників.

Для порівняльного аналізу обрано три основні платформи, що працюють з бюджетними даними в Україні.

Портал відкритих даних data.gov.ua — офіційний портал відкритих даних України, який надає доступ до різноманітних наборів даних, включаючи бюджетну інформацію. Портал забезпечує завантаження даних у різних форматах (CSV, JSON, XML), але не надає інструментів для візуалізації, аналізу чи прогнозування. Користувачам потрібно самостійно завантажувати та обробляти дані за допомогою зовнішніх інструментів. Портал працює на безкоштовній основі та не вимагає реєстрації для доступу до більшості даних.

Єдиний веб-портал використання публічних коштів spending.gov.ua — спеціалізований портал для моніторингу використання публічних коштів, який надає інформацію про видатки державних та місцевих бюджетів. Портал має базові інструменти пошуку та фільтрації даних, але обмежені можливості візуалізації та аналізу. Портал доступний безкоштовно.

Інформаційна система "Open Budget" — система, що працює з даними про державні бюджети та надає API для доступу до інформації. Система орієнтована на технічних користувачів та розробників, які можуть інтегрувати дані у власні додатки.

Запропонована інформаційна система відрізняється від конкурентів комплексним підходом до роботи з бюджетними даними, поєднуючи доступ до офіційних джерел через API з потужними аналітичними інструментами,

інтерактивною візуалізацією на карті, автоматичним прогнозуванням та генерацією PDF звітів. На відміну від конкурентів, система орієнтована на кінцевих користувачів різних категорій, надаючи інтуїтивний інтерфейс без необхідності технічних знань для роботи з даними.

В таблиці 10.2 показано оцінку конкурентоспроможності системи збору, аналізу та прогнозування даних про витрати місцевих бюджетів.

Таблиця 10.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

№ п/п	Техніко- економічні характеристи ки ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	spending	openbudget			
1	Глибока аналітика та розширена статистика	+	-	+/-			+
2	Прогнозуван ня бюджетних показників	+	-	-			+
3	Інтерактивна карта	+	-	+/-			+
4	Зручний сучасний інтерфейс із візуалізаціям и	+	-	+/-			+
5	Деталізація витрат	+	-	+		+	
6	Генерація PDF-звітів	+	-	-			+
7	Персональні акаунти та	+	-	-			+

№ п/п	Техніко- економічні характеристи ки ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	spending	openbudget			
	ролі користувачів						
8	Повний набір відкритих даних державного рівня	-	+	+	+		
9	Масштабован ість	+	+/-	+/-		+	
10	Оперативніст ь оновлення даних	+	+/-	+		+	

Проведений аналіз показав, що запропонована інформаційна система має суттєві конкурентні переваги порівняно з існуючими державними порталами, які здебільшого виконують функцію відкритого доступу до бюджетних даних, але не забезпечують глибокої аналітики чи інструментів прогнозування. Система відрізняється наявністю інтерактивної аналітики, блоків детальної статистики, автоматизованого прогнозування витрат та можливості формувати PDF-звіти, що робить її унікальною серед аналогів.

Основні конкуренти мають перевагу лише у масштабності наборів даних та ролі офіційних джерел, проте не пропонують розширеного функціоналу для аналізу чи прийняття управлінських рішень. Таким чином, проєкт демонструє сильні позиції за більшістю техніко-економічних характеристик та має значний потенціал для подальшого розвитку і впровадження в органах місцевого самоврядування.

10.2 Технологічний аудит ідеї проєкту

Реалізація системи збору, аналізу, візуалізації та прогнозування бюджетних даних вимагає використання сучасних вебтехнологій, інструментів обробки даних, засобів інтерактивної аналітики та зовнішніх API. Технологічний аудит дозволяє оцінити, наскільки доступні та зрілі такі технології, чи потребують вони спеціальної розробки, а також чи можуть бути використані авторами проєкту. Це дає змогу визначити технологічну здійсненність проєкту та оптимальний шлях його реалізації.

Таблиця 10.3 — Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Збір та синхронізація бюджетних даних	REST API OpenBudget, Node.js скрипти для ETL, PostgreSQL	Наявні, частково потребують власної розробки	Доступні
2	Зберігання та обробка великого обсягу даних	PostgreSQL, індексація, оптимізовані запити, кешування	Наявні	Доступні
3	Аналітика та статистична обробка	Власні модулі Node.js, SQL-агрегації, математичні методи прогнозування	Наявні, частково потребують власної розробки	Доступні
4	Побудова інтерактивної вебкарти України	Mapbox GL JS, React	Наявні	Доступні
5	Інтерактивна аналітика та графіки	React + Recharts	Наявні	Доступні
6	Прогнозування бюджетних показників	Експоненційне згладжування, ковзне середнє, регресія, середній темп приросту, кастомна реалізація на JS	Потребує власної реалізації, але методи відомі	Доступні
7	Генерація PDF-звітів	Node.js + PDFKit або Puppeteer, AWS S3 для	Наявні	Доступні

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
		зберігання		
8	Авторизація та управління користувачами	JWT, bcrypt, Node.js Sessions	Наявні	Доступні
9	Розгортання та масштабування системи	Docker, Docker Compose, AWS	Наявні	Доступні

Для реалізації проєкту обрано стек технологій Node.js + React + PostgreSQL з інтеграцією OpenBudget API, використанням Mapbox GL JS для геовізуалізації та AWS S3 для зберігання звітів. Архітектура будується на REST API, контейнеризації Docker та модульному поділі серверної частини на контролери, сервіси, маршрути та скрипти.

Проведений технологічний аудит показав, що всі необхідні технології для реалізації системи є зрілими, широко доступними й добре документованими. Вони не потребують додаткової розробки на фундаментальному рівні — лише адаптації та кастомізації під потреби системи. Усі технології повністю доступні авторам проєкту та можуть бути інтегровані в єдине рішення без технічних обмежень. Технологічна реалізація проєкту є повністю можливою, а обраний технологічний стек забезпечує надійність, масштабованість та розширюваність системи.

10.4 Аналіз ринкових можливостей запуску стартап-проєкту

У випадку аналітичної системи, яка працює з бюджетними показниками, ринковий контекст включає як державні інституції та громади, так і експертну спільноту, журналістів, громадські організації й бізнес-аналітику. Система призначена для користувачів, які працюють з відкритими бюджетними даними, виконують аналіз витрат, потребують автоматизованого прогнозування і

зацікавлені у підвищенні прозорості управління фінансами. Попит на подібні інструменти формується в умовах цифровізації публічного сектору, розвитку відкритих даних та потреби громад в інструментах планування бюджету. На таблиці 10.4 наведено попередню характеристику потенційного ринку

Таблиця 10.4 — Попередня характеристика потенційного ринку стартап-прокту

№ п/п	Показники стану ринку	Характеристика
1	Кількість основних гравців	3—5 основних платформ (openbudget.gov.ua, spending.gov.ua, data.gov.ua), незначна кількість вузькоспеціалізованих приватних сервісів
2	Загальний обсяг продаж, грн/ум. од.	Ринок некомерційний, але оцінюється високий потенціал монетизації через експертні сервіси, аналітику, консалтинг та B2G-послуги
3	Динаміка ринку	Зростає — завдяки політиці відкритих даних, реформі децентралізації та підвищенню прозорості бюджетів громад
4	Наявність обмежень для входу	Середні: потреба в експертизі роботи з бюджетними класифікаціями, складність обробки великих масивів даних, залежність від якості державних відкритих API
5	Специфічні вимоги до стандартизації	Необхідність дотримуватись бюджетної класифікації та формату відкритих даних
6	Середня норма рентабельності в галузі	Середня норма рентабельності в галузі

Ринок є привабливим для входження. Попит на інструменти бюджетної аналітики в Україні зростає завдяки активному впровадженню цифрових сервісів у публічному секторі та підвищенню потреби громад у сучасних інструментах управління фінансами. Кількість конкурентів є обмеженою, а існуючі платформи мають значні функціональні прогалини (відсутність прогнозування, недостатня аналітика, слабка інтеграція з потребами громад). Це створює можливості для впровадження інноваційного рішення.

Базова потреба на ринку формується навколо якісного доступу до бюджетних даних, їх автоматизованого аналізу, візуалізації, а також прогнозування майбутніх витрат.

Цільові аудиторії істотно відрізняються за рівнем технічної компетентності, інтенсивністю використання та очікуваною точністю результатів. Це впливає на вимоги до продукту, зокрема на дизайн інтерфейсу, функціональність та доступність інструментів. На таблиці 10.5 наведено потенційних клієнтів стартап-проєкту та їхню характеристику.

Таблиця 10.5 — Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Потреба у прозорому та якісному аналізі місцевих бюджетів	Органи місцевого самоврядування (ОМС), фінансові управління територіальних громад	Розрахунки виконуються регулярно; потреба в точності та зручності; залежність від строків формування бюджету; високий рівень відповідальності за дані	Висока достовірність, автоматизовані розрахунки, можливість прогнозування, інтерактивна аналітика, експорт звітів, надійність постачальника
2	Потреба у швидкому доступі до відкритих бюджетних даних у зручному форматі	Державні органи, аудитори, контрольні служби	Оперативність роботи; необхідність отримувати дані у стандартизованому форматі; часте використання у внутрішніх процедурах	Стандартизовані звіти, офіційні формати, логування дій, стабільність API, відповідність законодавчим вимогам
3	Потреба у візуалізації складних бюджетних	Громадські організації, аналітичні центри,	Акцент на зрозумілості та прозорості; потреба у порівняннях між громадами/роками;	Графіки, діаграми, інтерактивні панелі, можливість експорту у PDF, доступність

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	процесів	журналісти	регулярне використання матеріалів для публікацій	інструментів без спеціальної підготовки
4	Потреба у прогнозуванні бюджетних показників та моделюванні сценаріїв розвитку	Аналітики, економісти, дослідники, університети	Потреба у точності моделей та можливості налаштовувати параметри прогнозування; системне використання для наукових досліджень	Доступ до математичних методів прогнозування, можливість змінювати параметри, історія даних, прозорі алгоритми
5	Потреба у швидкій оцінці стану бюджету громади без складної підготовки	Жителі громад, активісти, місцеві ініціативи	Низький рівень технічної експертизи; асинхронне та нерегулярне використання; потреба в простому поясненні складних показників	Простий інтерфейс, інтуїтивний доступ до даних, зрозумілі пояснення, мінімум технічних бар'єрів
6	Потреба у використанні відкритих даних для створення власних сервісів або інтеграцій	ІТ-компанії, розробники аналітичних платформ	Висока технічна грамотність; акцент на стабільності API, документації, швидкості та форматах даних	API-доступ, документація, стабільні ендпоїнти, можливість інтеграції зі сторонніми системами

Для оцінки перспектив ринкового впровадження проєкту здійснено аналіз ринкового середовища, який включає визначення як факторів, що можуть стримувати розвиток системи, так і факторів, що створюють додаткові можливості для її росту. Такий підхід дозволяє оцінити потенційні ризики ще на етапі планування, сформувати готовність до зовнішніх змін та продумати реакції на ключові виклики ринку. Одночасно аналізуються сприятливі умови, які можуть

підсилити конкурентні позиції продукту та забезпечити його стійкий розвиток у довгостроковій перспективі.

У таблицях 10.6 та 10.7 подані найвагоміші загрози та можливості, впорядковані за рівнем їх значущості для інформаційної системи збору, аналізу та прогнозування витрат місцевих бюджетів.

Таблиця 10.6 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Залежність від державних відкритих API	Зміни у форматах даних або тимчасове припинення доступу до API можуть призвести до блокування роботи системи	Механізм резервного кешування даних; регулярне оновлення адаптерів API; альтернативні джерела даних
2	Висока конкуренція з боку державних порталів	Офіційні платформи мають ресурсну і правову перевагу, що може зменшувати попит на приватний продукт	Фокус на глибокій аналітиці, прогнозуванні та персоналізації, чого конкуренти не надають
3	Ризики кібератак і витоку даних	Фінансові дані є чутливими, і атаки на систему можуть вплинути на репутацію	Посилена авторизація, аудит безпеки, шифрування даних, регулярні реп-тести
4	Недостатня цифрова грамотність частини цільової аудиторії	Органи влади та громади можуть не мати фахівців для роботи з аналітичними інструментами	Спрощений інтерфейс, навчальні матеріали, відео-інструкції, онлайн-підтримка
5	Нестабільність законодавства у сфері публічних фінансів	Реформи можуть змінити структуру бюджетних класифікацій або правила подання звітності	Гнучка архітектура системи, можливість швидкої адаптації структури даних
6	Низька платоспроможність окремих громад	Частина потенційних клієнтів не може оплачувати повноцінні рішення	Введення free-моделі, грантові програми, спонсорство

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
7	Можлива недовіра до сторонніх аналітичних систем	Користувачі можуть віддавати перевагу офіційним даним та державним ресурсам	Прозорість алгоритмів, публічні методології розрахунків, співпраця з ГО та університетами

Таблиця 10.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростаючий попит на прозорість державних фінансів	Суспільство та громади активно потребують інструментів моніторингу витрат	Розвиток відкритої аналітики, інтеграція з платформами громадського контролю
2	Відсутність інструментів прогнозування на ринку	Конкуренти надають лише базову аналітику, не маючи прогнозних моделей	Позиціонування продукту як аналітичного та прогнозного рішення нового покоління
3	Широке використання відкритих даних (Open Data)	Законодавство України стимулює відкритість бюджетних даних	Активна інтеграція з іншими відкритими реєстрами, розширення масиву даних
4	Підтримка цифрових проєктів міжнародними організаціями	Є грантові програми для прозорості та цифровізації громад	Підготовка грантових заявок, партнерські проєкти з громадами
5	Необхідність громад у простих та швидких інструментах аналізу	Більшість громад не має власних ІТ-рішень та інструментів	Створення компактної панелі аналітики, можливість інтегрувати у сайт громади
6	Зростання попиту на порівняльну аналітику між громадами	Аналітики, журналісти й активісти потребують швидких порівнянь	Реалізація модулів для бенчмаркінгу та рейтингування громад
7	Розвиток е-	Держава стимулює цифровізацію і	Інтеграція з іншими

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
	урядування в Україні	використання сучасних сервісів	державними платформами, участь у цифрових реформах

Після аналізу попиту, характеристик цільових груп клієнтів та зовнішніх факторів ринкового середовища доцільно оцінити тип та інтенсивність конкуренції на ринку. Такий аналіз дозволяє зрозуміти структуру ринку, характер конкурентної боротьби та ті ринкові сили, які впливатимуть на впровадження та масштабування проєкту. Це важливо для формування ефективної конкурентної стратегії, визначення ключових переваг системи і вибору тактики подальшого розвитку.

На основі дослідження ринку державних платформ відкритих даних та аналітичних сервісів сформовано ступеневий аналіз конкуренції, наведений у таблиці 10.8.

Таблиця 10.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Тип конкуренції	Ринок складається переважно з державних платформ з відкритим доступом до даних. Фактична структура близька до монополістичної конкуренції — дані однакові, а сервіси різняться інструментами	Акцент на диференціації продукту через аналітику, прогнозування, зручність інтерфейсу та швидкість роботи
2. Рівень конкурентної боротьби	Національний рівень — конкуренти орієнтовані на український ринок та використовують однакові джерела даних	Стратегія передбачає фокусування на потребах місцевих громад, експертів з фінансів та аналітиків

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
3. Галузева ознака	Конкуренція внутрішньогалузева, всі сервіси працюють у сфері відкритих бюджетних даних	Потрібно забезпечувати більш високий рівень повноти та актуальності даних, а також додавати унікальні інструменти роботи з бюджетами
4. Конкуренція за видами товарів	Товарно-видова — усі продукти надають доступ до бюджетних даних, але не всі пропонують глибоку аналітику	Необхідно посилювати власну цінність за рахунок розширених функцій та автоматизації процесів
5. Характер конкурентних переваг	На ринку домінує нецінова конкуренція, оскільки доступ до більшості сервісів безкоштовний, а різниця формується через функціональність та зручність використання	Пріоритет — UX/UI, швидкість обробки, інтелектуальні інструменти, можливість персоналізації аналізу
6. Інтенсивність конкуренції	Не марочна конкуренція — більшість сервісів є державними або небрендованими платформами	Створення власного бренду та впізнаваної айдентики підвищить довіру користувачів та відрізнити продукт від стандартних державних порталів

Аналіз конкурентного середовища показує, що ринок бюджетних даних в Україні має помірну конкуренцію та структурований переважно державними продуктами, які часто не включають просунутих інструментів аналізу, прогнозування та візуалізації. Це створює сприятливі умови для впровадження інноваційного рішення, яке поєднує доступ до офіційних даних з аналітичними інструментами експертного рівня.

Конкуренція носить переважно неціновий характер, тому проєкт має можливість зміцнювати свої позиції завдяки технологічним перевагам, зручності використання, автоматизації та широкому набору функцій. Створення власного

бренду та розвиток додаткових сервісів може забезпечити довгострокову конкурентоспроможність та залучення нових груп користувачів.

Модель п'яти конкурентних сил М. Портера[19] дозволяє глибше оцінити умови функціонування ринку, силу конкурентного тиску та потенційні ризики для виходу проєкту на ринок. Такий аналіз охоплює вплив існуючих конкурентів, загрозу появи нових гравців, силу постачальників, силу споживачів та ризики з боку товарів-замінників. Проведення цього аналізу є важливим для визначення стратегії позиціонування проєкту, оцінки його стійкості та формування конкурентних переваг. У таблиці 10.9 наведено аналіз ринку інформаційних систем бюджетних даних в Україні за моделлю Портера.

Таблиця 10.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Характеристика для проєкту інформаційної системи бюджетного аналізу
Прямі конкуренти в галузі	Основними конкурентами є державні платформи: data.gov.ua, spending.gov.ua, openbudget.gov.ua. Їхня функціональність зосереджена на наданні доступу до даних, у той час як можливості аналітики, прогнозування і гнучких візуалізацій обмежені
Висновок щодо сили конкурентів	Інтенсивність конкуренції серед існуючих гравців є помірною, оскільки державні портали не мають комерційної мотивації активно розширювати функціональність. Це зменшує тиск на нові проєкти, що пропонують розширені можливості
Потенційні конкуренти	Можлива поява приватних аналітичних платформ, консалтингових компаній або стартапів, що працюють з відкритими даними. Бар'єри входження низькі через відкритість API та легкість доступу до даних, але високі у сфері розробки повноцінної аналітичної системи зі складними алгоритмами прогнозування
Висновок щодо загрози нових гравців	Загроза появи нових конкурентів є середньою. Найімовірніші конкуренти — ІТ-компанії, які можуть інтегрувати бюджетні дані у власні аналітичні продукти
Постачальники	Основними постачальниками є офіційні джерела даних (OpenBudget API, державні реєстри). Вони диктують формат даних, частоту оновлення та структуру API. Залежність від цих джерел є суттєвою, оскільки зміни у форматах можуть впливати на роботу системи

Складові аналізу	Характеристика для проєкту інформаційної системи бюджетного аналізу
Висновок щодо сили постачальників	Сила постачальників висока, оскільки єдиним джерелом офіційної інформації є державні сервіси. Проєкт має адаптуватися під їхні технічні обмеження
Клієнти	Потенційні клієнти — органи місцевого самоврядування, фінансові аналітики, журналісти, активісти, дослідники, громадські організації. Вони потребують точності, зручності та можливості прогнозування. Диктувати умови вони можуть через вимоги до якості та прозорості даних, а також зручності використання сервісу
Висновок щодо сили клієнтів	Сила клієнтів середня: існують альтернативні джерела даних, але немає альтернатив з таким самим рівнем аналітики. Ключова вимога — якість та прозорість
Товари-замінники	Найближчі замінники — ручний аналіз даних у Excel, BI-системи (Power BI, Tableau) та аналітичні звіти експертних організацій. Хоча вони дозволяють виконувати аналіз, але вимагають значно більше часу і не забезпечують автоматичного прогнозування
Висновок щодо загрози замінників	Загроза середня, оскільки замінники існують, але не надають такого рівня автоматизації та точності без значних зусиль користувача

Аналіз конкурентного середовища за моделлю М. Портера свідчить, що ринок бюджетної аналітики в Україні є відкритим для входження нового інноваційного продукту. Сила постачальників є найвищою, адже залежність від офіційного API критична. Інтенсивність конкуренції з існуючими платформами помірна через обмеженість їх функціональності. Загроза нових гравців та замінників середня, що робить важливим швидке впровадження унікальних можливостей — автоматичного прогнозування, генерації звітів та зручних інтерфейсів візуалізації.

Щоб бути конкурентоспроможним на цьому ринку, проєкт повинен мати такі сильні характеристики: високу якість обробки даних, глибоку аналітичну складову, стабільну інтеграцію з державними API, автоматизоване прогнозування, інтуїтивний інтерфейс, а також можливість створення професійних звітів для органів влади та громадських організацій.

Після проведених етапів аналізу — оцінки конкурентів, визначення характеристик ідеї проєкту, аналізу потреб клієнтів та зовнішнього маркетингового середовища — стає можливим сформулювати перелік ключових факторів, що визначають конкурентоспроможність запропонованої інформаційної системи. Ці фактори відображають ті властивості продукту, які забезпечують йому перевагу над іншими рішеннями на ринку та відповідають запитам користувачів і вимогам галузі. У таблиці 10.10 наведено обґрунтування основних факторів конкурентоспроможності для проєкту.

Таблиця 10.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Інтеграція з офіційними джерелами бюджетних даних (OpenBudget API)	Єдиний офіційний канал доступу до перевірених фінансових даних. Забезпечує достовірність, актуальність і повноту інформації. Державні портали не завжди пропонують зручні інструменти аналізу, тому інтеграція з API у поєднанні з розширеною аналітикою створює суттєву перевагу
2	Автоматизоване прогнозування витрат місцевих бюджетів	Жоден із конкурентів не пропонує алгоритмів прогнозування на основі статистичних моделей (ковзні середні, експоненціальне згладжування, регресія). Це задовольняє потреби аналітиків та органів місцевого самоврядування, підвищує аналітичну цінність системи і створює унікальність продукту
3	Зручна та інтуїтивна візуалізація даних	Користувачам важливо швидко отримувати інформацію у наочній формі. Інтеграція карти, графіків та динамічних панелей робить роботу з великими масивами даних значно ефективнішою, що не пропонують державні сервіси у комплексному вигляді
4	Генерація професійних PDF-звітів	Потреба у формальних, структурованих звітах характерна для бюджетних установ, громадських

№ п/п	Фактор конкурентоспроможності	Обґрунтування
		організацій та аналітичних центрів. Автоматична генерація звітів дозволяє значно скоротити час на підготовку документів, роблячи систему більш корисною у практичній роботі
5	Персоналізовані функції для різних груп користувачів (ролі user/admin)	Різні групи користувачів мають різні задачі: громадяни — перегляд, органи влади — управління, адміністратори — контроль і синхронізація. Наявність ролей робить систему гнучкою та адаптованою до потреб кожної групи клієнтів
6	Автоматизована синхронізація даних з логуванням та контролем помилок	Система здатна самостійно оновлювати дані з API, вести журнал синхронізацій та фіксувати помилки. Це знижує навантаження на користувачів і адміністратора, мінімізує ризик некоректних даних і підвищує стабільність роботи продукту
7	Можливість масштабування та контейнеризація через Docker	Дозволяє розгортати систему на різних серверах, швидко масштабувати під навантаження, спрощує технічну підтримку та оновлення. Підвищує надійність і готовність продукту до реального використання у громадах та організаціях
8	Доступність та відкритість інтерфейсу для широкої аудиторії	Метою системи є підвищення прозорості бюджетів. Надання даних у зручному вигляді для журналістів, активістів, громадян та дослідників відповідає запитам ринку й соціальній потребі в контролі фінансів
9	Підтримка глибокого аналізу (динаміка, структура, порівняння періодів)	Дає змогу проводити аналіз витрат не лише за фактами, але і в динаміці, що важливо для прийняття управлінських рішень. Державні сервіси надають переважно «сирі» дані без інструментів аналітики
10	Швидкодія системи завдяки	Оперативність обробки великих обсягів даних є

№ п/п	Фактор конкурентоспроможності	Обґрунтування
	оптимізованому бекенду та кешуванню прогнозів	критично важливою для прийняття рішень. Механізми кешування значно пришвидшують розрахунки й роблять систему конкурентоспроможною з технологічної точки зору

На основі проведених досліджень визначено набір ключових факторів конкурентоспроможності, які формують переваги проєкту на ринку інформаційних систем аналізу бюджетних даних. Головними з них є аналітичні та прогнозні можливості системи, автоматизоване формування звітів, інтеграція з офіційними джерелами даних та зручний інтерфейс для різних груп користувачів. Разом ці фактори створюють комплексне рішення, яке не має прямих аналогів серед існуючих конкурентів і здатне ефективно задовольнити потреби як органів влади, так і громадського сектору.

Після визначення ключових факторів конкурентоспроможності проєкту доцільно провести кількісне порівняння сильних та слабких сторін у контексті ринку. Такий аналіз дозволяє оцінити, наскільки ефективно система вирізняється серед існуючих платформ — data.gov.ua, spending.gov.ua та openbudget.gov.ua — та визначити напрями подальшого вдосконалення.

Таблиця 10.11 — Порівняльний аналіз сильних та слабких сторін проєкту «Інформаційна система збору, аналізу та прогнозування даних про витрати місцевих бюджетів»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з ... (назва підприємства)						
			—3	—2	—1	0	+1	+2	+3
1	Інтеграція з відкритими API та автоматичне оновлення	17				+			
2	Автоматизоване прогнозування витрат	20							+
3	Розширена візуалізація: карта, графіки,	17					+		

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з ... (назва підприємства)						
			—3	—2	—1	0	+1	+2	+3
	аналітичні панелі								
4	Генерація PDF-звітів на основі аналітики	16							+
5	Підтримка ролей користувачів та власного кабінету	14					+		
6	Автоматизована синхронізація та логування подій	15					+		
7	Масштабованість та контейнеризація через Docker	13				+			
8	Доступність інтерфейсу для широкої аудиторії	12				+			
9	Глибокий аналіз динаміки, структури та періодичних зрізів	18						+	
10	Висока швидкодія завдяки кешуванню прогнозів	14					+		

Порівняльний аналіз показує, що запропонована система має низку значних переваг над державними порталами, головним чином завдяки розширеним аналітичним можливостям, прогнозуванню та автоматичному формуванню звітів. Найсильніші позиції спостерігаються у галузях, де конкуренти взагалі не пропонують функціональності — аналітика, прогнозування, PDF-звіти, інтерактивні панелі. Водночас є й нейтральні зони, наприклад, доступність для широкого кола користувачів — оскільки державні портали також відкриті. Загалом структура балів свідчить, що система має високий потенціал для ринкового впровадження та здатна забезпечити конкурентну перевагу завдяки унікальним інструментам аналізу та прогнозування бюджетних даних.

Після проведення детального аналізу власних характеристик проєкту доцільно об'єднати отримані результати у SWOT-матрицю. Ця матриця виступає завершальним етапом оцінювання перспектив проєкту перед його впровадженням

та допомагає визначити стратегію розвитку, пріоритети вдосконалення та точки ризику. Матрицю наведено на таблиці 10.12.

Таблиця 10.12 — SWOT- аналіз стартап-проекту

<p>Сильні сторони:</p> <ol style="list-style-type: none"> 1. Наявність прогнозних моделей (експоненційне згладжування, ковзне середнє, середній приріст, регресія), яких немає у конкурентів. 2. Інтерактивна аналітична візуалізація: карта, графіки, структура витрат, динаміка. 3. Автоматизована генерація PDF-звітів із збереженням на S3. 4. Гнучка клієнт-серверна інфраструктура з кешуванням, Docker-контейнеризацією та модульною архітектурою. 5. Автоматична синхронізація та логування інтеграцій з OpenBudget API. 6. Можливість мультикористувацького доступу з ролями (user / admin). 	<p>Слабкі сторони:</p> <ol style="list-style-type: none"> 1. Обмежена публічна впізнаваність у порівнянні з державними порталами. 2. Залежність від стабільності та доступності OpenBudget API. 3. Потреба у підтримці та масштабуванні серверної інфраструктури власними силами. 4. Відсутність мобільного застосунку на поточному етапі.
<p>Можливості:</p> <ol style="list-style-type: none"> 1. Зростання попиту на прозорість місцевих бюджетів та фінансову аналітику. 2. Потенціал співпраці з громадами, журналістами, громадськими організаціями та аналітичними центрами. 3. Інтеграція з додатковими відкритими даними (держзакупівлі, реєстри послуг, демографічні дані). 4. Можливість розширення функціоналу на рівень областей, міністерств, міжрегіональної аналітики. 5. Підключення модулів машинного навчання для покращення точності прогнозів. 	<p>Загрози:</p> <ol style="list-style-type: none"> 1. Поява нових державних чи приватних платформ з подібним функціоналом. 2. Ризик зміни формату або доступності державних відкритих даних. 3. Конкуренція з боку великих аналітичних компаній, які можуть розширити фокус на бюджетну сферу. 4. Зміни у законодавстві, що можуть обмежити або ускладнити доступ до даних. 5. Можливе бажання державних порталів інтегрувати власні прогнозні модулі, що зменшить унікальність проекту.

На основі результатів SWOT-аналізу та оцінки конкурентного середовища сформовано кілька можливих стратегічних альтернатив ринкової поведінки для впровадження стартап-проєкту. Кожна альтернатива передбачає певний комплекс заходів, що мають підвищити конкурентоспроможність системи, забезпечити її впізнаваність, прискорити вихід на ринок та мінімізувати вплив ризиків. Для кожної альтернативи оцінено ймовірність отримання необхідних ресурсів і орієнтовні строки реалізації.

Таблиця 10.13 — Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Швидкий запуск системи з базовим функціоналом: інтерактивною картою, аналітикою та прогнозуванням; публічне розгортання демоверсії; співпраця з громадами та ГО як першими користувачами	Висока (вимагає мінімальних додаткових ресурсів; технології вже реалізовані)	1—2 місяці
2	Розширення функціоналу аналітики: порівняння бюджетів між громадами, нові візуалізації, поглиблені економічні прогнози; запуск платної підписки для аналітичних центрів і медіа	Середня (потрібні додаткові розробники)	3—6 місяців
3	Інтеграція з іншими відкритими даними (Prozorro, demographic API, education/health finance data) для створення комплексної фінансово-аналітичної платформи	Низько-середня (потрібні юридичні узгодження, значний обсяг розробки)	6—12 місяців
4	Активна маркетингова стратегія: участь у хакатонах, подача на гранти, партнерства з журналістськими організаціями та аналітичними проєктами; промокампанія в освітньому сектор	Активна маркетингова стратегія: участь у хакатонах, подача на гранти, партнерства з журналістськими організаціями та аналітичними проєктами; промокампанія в освітньому сектор	2—4 місяці

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
5	Виведення B2G-версії платформи для обласних адміністрацій і громад з додатковими модулями звітності та інтеграцією з внутрішніми фінансовими системами	Низька (потрібні офіційні угоди та спеціалізований функціонал)	9—18 місяців

Серед запропонованих альтернатив найбільш доцільною для стартового етапу є альтернатива 1 — Швидкий запуск системи з базовим, але вже конкурентним функціоналом.

Причини вибору:

- забезпечує найкоротші строки виходу на ринок;
- не потребує значних додаткових ресурсів;
- дозволяє швидко отримати перших користувачів і зворотний зв'язок;
- відповідає ринковим можливостям (зростання попиту на аналітику бюджетів);
- підсилює сильні сторони системи — прогнозування, візуалізація, інтерактивна карта.

Після запуску оптимальною стратегією розвитку буде комбінування альтернатив 2 та 4: розширення аналітики та одночасний маркетинговий вихід на професійні аудиторії.

10.4 Розроблення ринкової стратегії проєкту

Першим етапом формування ринкової стратегії є визначення цільових груп потенційних користувачів системи. Для кожного сегмента оцінюється його готовність до використання продукту, можливий обсяг попиту, рівень конкуренції та простота виходу на цей сегмент. Це дозволяє встановити пріоритети ринкового охоплення та визначити найбільш перспективні напрями розвитку.

Таблиця 10.14 — Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність сприйняти продукт	Орієнтовний попит в сегменті	Інтенсивність конкуренції	Простота входу у сегмент
1	Органи місцевого самоврядування (міські та селищні ради, фінансові відділи громад)	Висока — щоденна робота з бюджетними даними та потреба у звітах	Високий, оскільки кожна громада працює з бюджетами	Низька— середня: державні портали не орієнтовані на аналітику	Середня: потрібні партнерства, але бар'єри низькі
2	Аналітичні центри, ГО, журналісти даних	Висока — зацікавленість у глибокій аналітиці та прогнозах	Середній: активні, але відносно невеликі групи	Середня: існують альтернативні інструменти, але обмежені	Висока: доступ через відкритий вебінтерфейс
3	Студенти, дослідники, економічні факультети	Середня — залежить від навчальних програм	Середній: потреба в інструментах для навчання й аналізу	Низька: майже відсутні спеціалізовані сервіси	Висока: бар'єри входу мінімальні
4	Бізнес-компанії, що працюють з територіальним плануванням та інвестиційним аналізом	Низько-середня — продукт не є основним інструментом	Низький— середній: сегмент вузький	Висока: доступні власні внутрішні аналітичні ресурси	Низька: потрібні додаткові модулі та інтеграції
5	Громадяни та активісти, які цікавляться бюджетами своїх громад	Середня — залежить від рівня цифрової грамотності	Середній	Низька: мало альтернатив з прогнозами та візуалізацією	Висока: простий доступ без спеціальних знань

Аналіз груп споживачів показує, що продукт може бути корисним для кількох сегментів, але потреби між ними різняться: органи місцевого самоврядування потребують зручної звітності; аналітичні центри — глибоких даних та прогнозів; освітній сектор — доступності та простоти.

З огляду на це обирається стратегія диференційованого маркетингу.

Система буде орієнтована на кілька сегментів, для кожного з яких формуються різні комунікаційні та функціональні акценти:

- професійна аналітика для ОМС;
- дослідницькі інструменти для аналітиків;
- навчальний доступ для студентів.

Після визначення цільових сегментів та альтернатив розвитку проєкту необхідно сформуванати базову стратегію ринкової поведінки. Її вибір залежить від конкурентних переваг стартапу, наявних ринкових можливостей та потенційних загроз. У таблиці 15 представлено узагальнення стратегічних рішень: обрану ринкову альтернативу, тип охоплення ринку, ключові конкурентні позиції продукту та базову стратегію розвитку відповідно до класифікації М. Портера.

Таблиця 10.15 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції	Базова стратегія розвитку
1	Виведення на ринок продукту зі зосередженням на професійних користувачах (ОМС, аналітичні центри) та поступовим розширенням на освітній сектор	Диференційований маркетинг — різні ціннісні пропозиції для різних сегментів	Унікальна система прогнозування бюджету; автоматизована інтеграція з OpenBudget API; глибока візуалізація бюджетних даних; простота використання; можливість формування звітів у форматі PDF; масштабованість	Стратегія диференціації, що ґрунтується на унікальних функціях аналізу, прогнозування та візуалізації
2	Розвиток сервісу як універсального	Частково масовий маркетинг (для	Відкритий доступ до частини функцій;	Стратегія диференціації зі

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції	Базова стратегія розвитку
	інструменту для бюджетної аналітики (доступний будь- якому користувачу)	громадян) у поєднанні зі спеціалізованими функціями для професійних користувачів	інтуїтивна карта вибору громад; можливість розширення модулів; створення автоматичних звітів	спеціалізацією — фокус на ключових групах, але з наявністю універсального доступу
3	Позиціонування системи як інструменту для підвищення прозорості та цифрової грамотності громад	Маркетинг для широкої аудиторії із наголосом на простоті та доступності	Відсутність аналогів із прогнозними моделями; легкий доступ до офіційних даних; сучасний інтерфейс; валідність даних	Стратегія диференціації, орієнтована на відмінність від державних порталів

На основі проведеного аналізу та оцінки ринкових можливостей обрано стратегію диференціації як основний напрям розвитку. Інформаційна система має чіткі унікальні конкурентні переваги — модулі прогнозування, глибокі аналітичні інструменти, інтеграцію з офіційними державними даними та сучасний картографічний інтерфейс.

Система пропонує різні ціннісні пропозиції для різних сегментів, що відповідає підходу диференційованого маркетингу. Такий підхід дозволяє одночасно охопити професійних аналітиків, органи місцевого самоврядування, дослідників та громадян, зберігаючи фокус на ключових функціях, які вирізняють продукт на ринку.

Оскільки ринок бюджетної аналітики в Україні частково сформований державними платформами, важливо визначити, як саме стартап позиціонуватиме себе серед них та яку тактику використовуватиме для здобуття власної частки ринку.

Таблиця 10.16 узагальнює ключові питання, що визначають стратегічну поведінку проєкту, та дозволяє обґрунтувати оптимальний вибір конкурентної стратегії.

Таблиця 10.16 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Частково. На ринку немає аналогів з модулем прогнозування і повноцінними аналітичними панелями, але є державні портали з базовими функціями перегляду даних	Основний фокус — залучення нових користувачів, які потребують аналітики та прогнозування (ОМС, експерти, дослідники). Другорядно — перехоплення частини аудиторії офіційних порталів завдяки зручності та розширеному функціоналу	Копіювання базових можливостей перегляду даних не є ключовою метою. Передбачено лише використання стандартних механізмів фільтрації та відображення, але основна ставка робиться на унікальні функції — динамічну карту, прогнозування, PDF-звіти	Стратегія заняття конкурентної ніші з елементами виклику лідеру через унікальні аналітичні функції

Проведений аналіз показує, що оптимальною є стратегія заняття конкурентної ніші. Державні портали виконують роль базових сховищ даних, але не пропонують інструментів прогнозування, деталізованої аналітики та сучасної візуалізації. Це створює можливість сфокусуватися на вузькому, але зростаючому

сегменті користувачів, які потребують інструментів глибокого аналізу — органи місцевого самоврядування, дослідники, фінансові аналітики.

Проект також частково реалізує риси стратегії виклику лідеру, оскільки пропонує функціонал, який відсутній у ключових державних гравців, та може привернути частину їх аудиторії.

Завдяки унікальності інструментів прогнозування та автоматичної генерації звітів, система має підстави сформувати власну стійку нішу на перетині відкритих даних, фінансової аналітики та цифрових сервісів для громад.

Позиціонування стартап-проекту визначає, яким чином продукт сприйматимуть потенційні споживачі серед існуючих аналогів і які асоціації формуватимуться у їхній свідомості. У таблиці 10.17 наведено систематизацію ключових параметрів, що впливають на розвиток позиції бренду стартап-проекту.

Таблиця 10.17 — Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції	Вибір асоціацій
1	Доступ до структурованих бюджетних даних; зручність перегляду; інтерактивний інтерфейс на мапі; висока швидкість роботи; сучасна візуалізація	Стратегія диференціації	Інтерактивна карта; глибока аналітика; інтуїтивний UI; швидка реакція сервера	Сучасність, зручність, доступність даних у 1 клік
2	Потреба у прогнозних бюджетних значеннях; підтримка ухвалення рішень; аналітичні моделі	Стратегія спеціалізації	Унікальний модуль прогнозування; декілька методів розрахунку; кешування результатів	Аналітичність, точність, експертність
3	Можливість формувати звіти;	Стратегія спеціалізації +	Автоматичні PDF-звіти; сховище звітів; персональний	Професійність, повна

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції	Вибір асоціацій
	автоматизація рутинної роботи; збереження документів у S3	диференціації	кабінет користувача	готовність результатів, надійність

На основі аналізу вимог цільових сегментів, конкурентного середовища та можливостей продукту визначено, що ринкова стратегія позиціонування має ґрунтуватися на асоціаціях “сучасний аналітичний інструмент для роботи з місцевими бюджетами”, який поєднує:

а) аналітичність та професійність — продукт виступає не просто переглядачем даних, а інструментом для аналізу та прогнозування, що дозволяє здійснювати прийняття рішень на основі даних;

б) зручність та інноваційність — інтерактивна карта, сучасний дизайн і швидкий пошук роблять систему відмінною від державних порталів, що позиціонує її як більш технологічне рішення;

с) надійність та повнота результатів — користувачі отримують повні звіти, прогнозні моделі та доступ до історії взаємодії з даними, що формує довіру до платформи.

Сформована стратегія позиціонування визначає стартап як інноваційну аналітичну платформу, що спеціалізується на обробці та прогнозуванні бюджетних даних територіальних громад. Ринкова позиція базується на поєднанні точності аналітики, сучасного функціоналу та високої зручності у використанні, що дозволяє вирізнити продукт серед державних сервісів та зайняти стійку нішу на ринку цифрових фінансових рішень.

10.5 Розроблення маркетингової програми стартап-проєкту

Маркетингова програма стартап-проєкту починається з формування чіткої концепції товару, що визначає, яку цінність отримають користувачі та які конкурентні переваги має продукт порівняно з наявними аналогами. Концепція товару узагальнює ключові потреби цільових аудиторій, вигоди, які продукт пропонує у відповідь на ці потреби, а також ті переваги, що забезпечують його конкурентоспроможність.

У таблиці 10.18 наведено структурований опис цих елементів, сформований на основі попереднього аналізу ринку, конкурентів, вимог споживачів та сильних сторін проєкту.

Таблиця 10.18 — Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Доступ до структурованих бюджетних даних у зручному форматі	Швидкий пошук і перегляд даних за допомогою інтерактивної карти та адаптивної аналітичної панелі	Наявні державні сервіси подають дані у громіздкому табличному форматі; відсутня інтерактивність та зручна візуалізація
2	Аналітика та оцінка бюджетних показників	Глибока статистика: помісячні, поквартальні та річні показники; автоматичні розрахунки; структура витрат	Відсутність подібних інструментів на порталах OpenBudget або Spending; автоматизована аналітика значно економить час користувача
3	Прогнозування витрат	Мульти-модельні прогнози (середній приріст, ковзне середнє, експоненційне згладжування, регресія) з кешуванням результатів	Прогнозні модулі відсутні у всіх трьох конкурентах; унікальна функціональність на ринку
4	Можливість створювати професійні звіти	Генерація PDF-документів з аналітикою, графіками та прогнозами; збереження у S3,	Державні платформи не дозволяють формувати зручні для презентацій або службового використання звіти

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
		повторне завантаження	
5	Економія часу аналітиків, журналістів, громадських організацій	Автоматизація рутинних операцій: агрегація даних, розрахунок показників, формування звітів	У конкурентів більшість операцій виконується вручну, без можливості швидкої обробки
6	Простота використання	Інтуїтивний інтерфейс, інтерактивна карта, панель вибору параметрів, мінімум зайвих кроків	Державні сервіси перевантажені інформацією, мають складну навігацію та низьку UX-оптимізацію
7	Доступність та сучасність	Веб-доступ без встановлення програмного забезпечення; адаптація під різні пристрої	Конкуренти мають застарілий інтерфейс і повільно оновлюються (особливо data.gov.ua)

Таблиця 10.19 структурує продукт на трьох рівнях: товар за задумом, товар у реальному виконанні, товар із підкріпленням, що дозволяє комплексно представити його як для споживачів, так і для партнерів, інвесторів чи потенційних користувачів системи.

Таблиця 10.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Базова потреба — доступний, швидкий та прозорий аналіз фінансової інформації місцевих бюджетів. Основна вигода — можливість оперативно отримувати аналітику, структуру витрат та прогнозування без необхідності ручної обробки даних. Товар створює основу для підвищення прозорості діяльності громад, ефективності прийняття управлінських рішень та полегшує роботу аналітиків, журналістів й активістів		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Інтерактивна карта України з можливістю вибору громади	Нм	Вр
	2. Аналітичні панелі з деталізованою статистикою	Нм	Вр/Тл

Рівні товару	Сутність та складові		
	(місяць, квартал, рік)		
	3. Модулі прогнозування бюджетних видатків на основі математичних моделей	Нм	Тл
	4. Автоматична генерація PDF-звітів із графіками, таблицями та прогнозами	Нм	Вр/Ор
	5. Сховище звітів у S3 та можливість повторного завантаження	Нм	Тх
	6. Простий, сучасний інтерфейс у світлій темі		
	7. Логування, безпека, система авторизації	Нм	Е
	8. API для взаємодії з іншими системами громад	Нм	Тх/Ор
		Нм	Тх
	Якість: Система побудована на сучасному стеку (Node.js, PostgreSQL, React, Mapbox). Тестування включає модульні та інтеграційні перевірки, контроль коректності даних, кешування запитів до API, стабільність роботи при великому обсязі даних.		
	Пакування: Пакуванням виступає інтерфейс користувача — веб-портал із інтуїтивною структурою та сучасним дизайном. Документація та короткі інструкції забезпечують легкість освоєння		
III. Товар із підкріпленням	До продажу: Безкоштовний онлайн-доступ; відсутність складної інсталяції; демонстраційні приклади громад; можливість тестування функцій без реєстрації		
	Після продажу: Регулярні оновлення моделей прогнозування; підтримка користувачів; розширення функціоналу; оновлення даних у фоновому режимі; автоматичні повідомлення про помилки; можливість завантажувати нові звіти		
Захист від копіювання: Унікальне поєднання інтерактивної карти, прогнозних моделей, автоматичної генерації звітів та кешованої взаємодії з OpenBudget API. Можливість зареєструвати авторські права на програмний код, візуальні рішення та алгоритмічні моделі як ноу-хау. Складність відтворення комплексної інфраструктури (ETL-процеси, S3, бекенд-сервіси, інтелектуальні моделі) створює природний ринковий бар'єр			

Визначення цінових меж є важливим етапом формування маркетингової програми, оскільки саме ціна визначає можливість ринкового впровадження проєкту та його фінансову життєздатність.

Для інформаційної системи аналізу та прогнозування бюджетних даних ключовими орієнтирами є: наявні комерційні сервіси аналітики (BI-платформи), державні портали з базовою функціональністю (які є безкоштовними), а також бюджет громад та організацій, що можуть виступати цільовими користувачами.

Таблиця 10.20 — Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Державні портали з бюджетними даними (spending.gov.ua, openbudget.gov.ua) є безкоштовними, але не містять розширеної аналітики та прогнозування	Комерційні BI-платформи (Power BI, Tableau) на рівні від 300 до 1500 грн/міс для базових ліцензій	Органи місцевого самоврядування та громадські організації мають обмежений бюджет на IT-рішення; журналісти та аналітики часто використовують безкоштовні інструменти	Нижня межа: 0 грн (free-доступ для базових функцій). Верхня межа: 200—400 грн/міс за розширену аналітику та прогнозування
2	Інші відкриті API та набори даних — безкоштовні	SaaS-рішення для аналітики даних у держсекторі стартують від 5000—15000 грн/рік	Для великих громад можливе виділення бюджету на цифрові сервіси; для малих — потреба у доступній підписці	Нижня межа: безкоштовний модуль. Верхня межа: 5000 грн/рік для великих громад
3	Excel-аналіз (ручний збір та обробка) має	Аналітичні дашборди	Доходи журналістів, студентів та	Нижня межа: доступна

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	високі трудовитрати, але нульову вартість	комерційного типу вартують 200—1000 грн/міс	аналітичних центрів невисокі, тому вони схильні використовувати доступні рішення	підписка 50—100 грн/міс. Верхня межа: 200—300 грн/міс для розширеного пакету

Оскільки продукт є цифровою інформаційною системою, канали збуту та способи донесення продукту до користувача суттєво відрізняються від матеріальних товарів. У цьому контексті ключовими є такі питання: як користувачі здійснюють вибір продуктів подібного типу, які функції має виконувати постачальник, якої глибини має бути канал збуту та чи варто залучати посередників.

Таблиця 10.21 демонструє оптимальний підхід до побудови збутової системи для запропонованого стартап-проекту.

Таблиця 10.21 — Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Представники громад шукають сервіси онлайн, орієнтуються на офіційність, зручність та доступність	Надання демонстрацій, презентацій, технічних характеристик; консультації щодо інтеграції; техпідтримка	Прямий канал	Власна система збуту через сайт, онлайн-презентації та офіційні пропозиції для громад через Prozorro
2	Громадські організації та	Маркетинг,	Прямий	Прямі продажі через

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	журналісти шукають інструменти самостійно, орієнтуючись на відгуки, прозорість та наявність безкоштовного доступу до базових функцій	інформаційна підтримка, робота з партнерами, надання безкоштовних пакетів та навчальних матеріалів	канал	сайт, SEO/контентна стратегія, партнерські програми з аналітичними центрами
3	Аналітичні центри та студенти обирають продукти самостійно, порівнюючи інструменти за функціональністю та зручністю	Надання пробних версій, документації, відкритого API, швидкого старту для користувачів	Прямий канал	Онлайн-реєстрація без посередників, можливість самостійного підключення та активації

Ефективне просування цифрового продукту залежить від глибокого розуміння поведінки цільових аудиторій, їхніх інформаційних звичок та очікувань від сервісів аналітики бюджетних даних. Запропонована концепція комунікацій базується на результатах аналізу попиту, позиціонування, характеристик цільових груп та стратегій конкурентної поведінки. Оскільки продукт належить до сфери gov-tech та civic-tech, особливе значення мають прозорість, офіційність, достовірність джерел та орієнтація на суспільну користь.

На таблиці 10.22 наведено структуру концепції маркетингових комунікацій.

Таблиця 10.22 — Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій клієнтів	Ключові позиції для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Представники місцевого самоврядування	Офіційний сайт, email-розсилки,	Прозорість, надійність, автоматизація,	Показати практичну користь для	Єдина платформа для прозорого

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій клієнтів	Ключові позиції для позиціонуван ня	Завдання рекламного повідомлення	Концепція рекламного звернення
	орієнтуються на офіційність, корисність інструменту, можливість автоматизувати звітність та роботу з бюджетами	вебінари, презентації для ОМС, Prozorro Market, Facebook-групи ОМС	офіційні дані	роботи громади: як система економить час та забезпечує контроль за бюджетом	управління бюджетом громади
2	Журналісти та ГО шукають інструменти для розслідувань, аналітики, порівнянь, працюють з великими масивами даних	Соціальні мережі, Telegram, LinkedIn, тематичні конференції, співпраця з аналітичними центрами	Глибока аналітика, швидкість роботи, доступ до історичних даних, можливість виявляти аномалії	Підкреслити аналітичну цінність та можливість знаходити важливі суспільні проблеми через дані	Аналітичний інструмент для контролю за публічними фінансами
3	Студенти, дослідники та аналітики шукають доступні та зрозумілі інструменти для навчання та досліджень	Соціальні мережі, YouTube-демо, освітні платформи, партнерства з університетами	Доступність, простота використання, навчальні матеріали	Пояснити, що інструмент можна легко освоїти та використати для навчання і досліджень	Бюджетні дані в зручному форматі для досліджень і навчання

Сформована концепція маркетингових комунікацій дозволяє вибудувати цілісну систему просування, яка враховує специфіку різних сегментів ринку та підкреслює конкурентні переваги стартап-проекту. Зосередженість на офіційності,

прозорості, точності даних і високій аналітичній цінності забезпечує продукту довіру серед професійних користувачів та сприяє формуванню стійкого ринкового позиціонування.

Запропонований підхід забезпечує узгодженість між концепцією товару, його позиціонуванням, системою збуту, специфікою споживачів та вимогами ринку, формуючи завершену маркетингову програму стартап-проєкту.

Висновки до розділу 10

Проведений аналіз підтверджує наявність реальних ринкових можливостей для комерціалізації інформаційної системи збору, аналізу та прогнозування витрат місцевих бюджетів. Попит на інструменти роботи з бюджетними даними стабільно зростає, а сегменти цільових користувачів — органи місцевого самоврядування, громадські організації, журналісти та аналітики — демонструють високу готовність використовувати подібні рішення. Рентабельність роботи в цьому ринковому середовищі є сприятливою, а бар'єри входження залишаються відносно низькими завдяки доступності відкритих державних даних. Конкурентний аналіз показав, що наявні платформи зосереджені переважно на наданні доступу до даних, тоді як запропонований проєкт вирізняється наявністю аналітичних та прогнозних інструментів, що формує його ключову перевагу. SWOT-аналіз засвідчив, що сильні сторони — автоматизація, прогнозування та інтерактивні візуалізації — дозволяють ефективно протистояти ринковим загрозам.

Оцінка ринкових альтернатив показала, що найбільш доцільною стратегією впровадження є поступовий вихід на ринок із фокусом на ключових сегментах користувачів, насамперед органах місцевого самоврядування.

У цілому результати аналізу підтверджують, що проєкт є ринково перспективним та технологічно готовим до впровадження. Подальша імплементація системи є доцільною та здатна забезпечити стійку конкурентну позицію на ринку gov-tech рішень України.

ВИСНОВКИ

У процесі виконання роботи було всебічно досліджено предметну область відкритих бюджетних даних та проаналізовано сучасний стан інструментів, що забезпечують прозорість і доступність фінансової інформації органів місцевого самоврядування. Було встановлено, що наявні сервіси є фрагментованими, не забезпечують глибокої аналітики та не містять механізмів прогнозування, що підкреслює актуальність створення інтегрованої інформаційної системи нового типу. На основі цього було сформульовано комплекс вимог до майбутнього програмного продукту, що охоплюють функціональні, технічні, інформаційні та архітектурні аспекти.

У ході роботи було розроблено архітектуру інформаційної системи, побудовану за принципами клієнт–серверної взаємодії, визначено структуру бази даних, механізми інтеграції з державними API та логіку обробки великих обсягів бюджетних даних. Було створено інфологічну та даталогічну моделі, структурну схему та діаграми, які забезпечують цілісне уявлення про роботу всіх компонентів системи. Вибір технологічного стеку було обґрунтовано з урахуванням вимог до масштабованості, продуктивності, безпеки та зручності подальшого розвитку. Для реалізації клієнтської частини було використано React, для серверної – Node.js та Express, а для зберігання даних – PostgreSQL і хмарне сховище Amazon S3.

Особлива увага була приділена математичним методам аналізу та прогнозування. Було обрано, досліджено та реалізовано кілька моделей прогнозування бюджетних витрат, серед яких метод середнього приросту, ковзне середнє, експоненційне згладжування та лінійна регресія. Це забезпечило можливість формування прогнозів розвитку бюджетних показників, що значно розширює практичну цінність створеної системи.

Розроблена інформаційна система продемонструвала здатність автоматично отримувати, зберігати та аналізувати дані про витрати місцевих бюджетів, а також формувати графічні візуалізації та генерацію звітів. Реалізований функціонал підтвердив відповідність системи поставленим вимогам, а проведені тестування

засвідчило коректність роботи основних модулів і достовірність прогнозних результатів.

Підсумком виконання роботи стало створення комплексного веб-орієнтованого інструменту, який може бути корисним для органів місцевого самоврядування, аналітичних центрів, журналістів, дослідників та громадськості. Система сприяє підвищенню прозорості бюджетних процесів, формує підґрунтя для прийняття обґрунтованих управлінських рішень, забезпечує можливість порівняльного аналізу та прогнозування фінансових тенденцій, а також дозволяє оперативно оцінювати динаміку витрат місцевих бюджетів у розрізі періодів та категорій. Завдяки інтеграції з державними відкритими даними та реалізації сучасних математичних методів обробки інформації система демонструє потенціал стати інструментом довгострокового моніторингу фінансової сталості громад і підвищення ефективності управління ресурсами.

Результати роботи свідчать про можливість подальшого удосконалення програмного продукту, розширення його функціональності, масштабування для аналізу нових типів даних та інтеграції у державні цифрові платформи, що відкриває перспективи для розвитку більш широкої екосистеми аналітичних сервісів у сфері публічних фінансів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Портал відкритих даних data.gov.ua. URL: <https://data.gov.ua/>
2. Єдиний вебпортал використання публічних коштів Spending. URL: <https://spending.gov.ua/new/>
3. Інформаційна система “Open Budget”. URL: <https://openbudget.gov.ua/>
4. Діаграма прецедентів. URL: http://ni.biz.ua/16/16_10/16_105305_opisanie-pretседentov.html
5. Про затвердження складових Програмної класифікації видатків та кредитування місцевого бюджету. URL: https://zakon.rada.gov.ua/rada/show/v07932_01-17#Text
6. Nielsen A. Practical Time Series Analysis: Prediction with Statistics and Machine Learning. O'Reilly Media, 2019. 504 p.
7. Larman. Applying Uml and Patterns : Complete Vid Crse: Uml Slipcase. Pearson Education, Limited, 2001. 291 p.
8. Allamaraju S. RESTful Web Services Cookbook. O'Reilly Media, Incorporated, 2010. 311 p.
9. React DOM. URL: <https://react.dev/reference/react#react-dom>
10. Max B. Express. js: Web Development with Node and Express. Independently Published, 2019. 109 p.
11. Gulabani S. Practical Amazon EC2, SQS, Kinesis, and S3. Berkeley, CA : Apress, 2017. 223 p.
12. API OpenBudget. URL: <https://confluence-ext.spending.gov.ua/pages/viewpage.action?pageId=364490>
13. Mishra P. Cloud Computing with AWS. Berkeley, CA : Apress, 2023. 145 p.
14. Teorey Toby J., Fry James P. Design of database structures : First Edition. The University of Michigan : Prentice Hall, 1982. 287 p.
15. Mapbox GL JS API Reference. URL: <https://docs.mapbox.com/mapbox-gl-js/api/>

16. Dombrovskaya H., Novikov B., Bailliekova A. PostgreSQL Query Optimization. Berkeley, CA : Apress, 2024. 105 p.

17. Ендпоінти API OpenBudget. URL: <https://api.openbudget.gov.ua/swagger-ui.html#/BudgetIndicators>

18. Harnish V. Mastering the Rockefeller Habits: What You Must Do to Increase the Value of Your Growing Firm. Gazelles, Inc., 2002. 51 p.

19. Timlon J., Zakaria R. Porter's Five Forces. URL: <https://sk.sagepub.com/foundations/porter-five-forces>