

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

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Євгенія СУЛЕМА

«___» _____ 2025 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія програмного
забезпечення мультимедійних та інформаційно-пошукових систем»**

спеціальності 121 Інженерія програмного забезпечення

**на тему: «Вебзастосунок для управління діаграмами ціннісної
пропозиції»**

Виконав:

студент IV курсу, групи КП-12

Мальцев Микита Ігорович _____

Керівник:

Доцент кафедри ПЗКС, к.т.н., доцент,

Люшенко Леся Анатоліївна _____

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н., доцент,

Онай Микола Володимирович _____

Рецензент:

Доцент кафедри СПСКС, к.т.н., доцент

Тарасенко-Клятченко Оксана Володимирівна _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення
мультимедійних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Євгенія СУЛЕМА

«__» _____ 2024 р.

ЗАВДАННЯ

на дипломний проєкт студенту

Мальцеву Микиті Ігоровичу

1. Тема проєкту «Вебзастосунок для управління діаграмами ціннісної пропозиції», керівник проєкту Люшенко Леся Анатоліївна, старший викладач, к.т.н., затверджені наказом по університету № 1808-С від «29» травня 2025 р.
2. Термін подання студентом проєкту «13» червня 2025 р.
3. Вихідні дані до проєкту: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - аналіз існуючих рішень поставленої задачі;
 - обґрунтування вибору засобів реалізації;
 - розроблення вебзастосунку;
 - аналіз розробленого вебзастосунку.
5. Перелік обов'язкового графічного матеріалу:
 - діаграма діяльності основного сценарію використання (креслення);
 - схема бази даних (креслення);
 - дії користувача вебзастосунку «Scheme chef» (плакат);
 - архітектура вебзастосунку «Scheme chef» (плакат).

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., доцент		

7. Дата видачі завдання «31» жовтня 2024 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	18.11.2024	
2.	Розроблення та узгодження технічного завдання	27.11.2024	
3.	Розроблення структури вебзастосунку	18.12.2024	
4.	Підготовка матеріалів першого розділу дипломного проєкту	03.02.2025	
5.	Розроблення дизайну сторінок та графічних елементів	30.02.2025	
6.	Підготовка матеріалів другого розділу дипломного проєкту	14.03.2025	
7.	Програмна реалізація вебзастосунку	28.03.2025	
8.	Тестування вебзастосунку	17.04.2025	
9.	Підготовка матеріалів третього розділу дипломного проєкту	23.04.2025	
10.	Підготовка матеріалів четвертого розділу дипломного проєкту	01.05.2025	
11.	Підготовка графічної частини дипломного проєкту	10.05.2025	
12.	Оформлення документації дипломного проєкту	23.05.2025	

Студент

Микита МАЛЬЦЕВ

Керівник проєкту

Леся ЛЮШЕНКО

АНОТАЦІЯ

Дипломний проєкт розроблений для задоволення потреби в управлінні діаграмами ціннісних пропозицій та автоматизації розробки таких діаграм. Метою даного дипломного проєкту є створення вебзастосунку, що надаватиме функціональність для створення ціннісних пропозицій та додавання нових якостей для вже створених ціннісних пропозицій за допомогою сучасних програмних технологій.

Проєкт включав аспекти управління діаграмами ціннісних пропозицій, такі як структурування діаграм у теках, контроль доступу до діаграм та обмін створеними діаграмами із зацікавленими сторонами за допомогою URL-посилань. Автоматизація розробки діаграм ціннісних пропозицій включала надання пропозицій щодо елементів діаграми на основі доступної інформації про ціннісну пропозицію.

Вебзастосунок був розроблений на основі розподіленої клієнт-серверної архітектури. Можливість розгортання застосунку з використанням контейнеризації також була врахована на етапі планування. Технології, які використовуються для розробки застосунку, були обрані з урахуванням обраного архітектурного підходу. Зв'язок між клієнтськими та серверними компонентами застосунку здійснюється через API.

Обмін інформацією між користувачем та вебзастосунком здійснюється через протокол HTTPS, що забезпечує шифрування даних під час передачі. Для захисту даних користувача реалізовано інші заходи безпеки, такі як хешування облікових даних та same-origin policy.

В рамках дипломного проєкту було спроектовано, розроблено та протестовано застосунок відповідно до користувацьких потреб, що мали бути задоволені.

ABSTRACT

The diploma project was developed to address the need for managing Value Proposition diagrams and automating the development of such diagrams. The target of this diploma project was the creation of a web application that would provide the means for creating value proposition diagrams and adding new qualities to the existing ones with the use of modern information technology.

The project included aspects of value proposition diagram management, such as structuring the diagrams in the underlying directories, controlling access to the diagrams, and sharing the created diagrams with the stakeholders via URL links. The automation of value proposition diagram development includes suggestions for the diagram elements based on the available information about the diagram.

The web application was designed based on the distributed client-server architecture. The ability to deploy the application using containerization was also considered during the planning phase. The technologies used to develop the application were selected to account for the selected architectural approach. The communication between the client-side and server-side application components is handled via API.

The information exchange between the user and the web application is performed through the HTTPS protocol, which ensures data encryption during the transfer. Other security measures, such as credentials hashing and same-origin policy, are implemented to protect the user's data.

Within the scope of the diploma project, the application was designed, developed, and tested in accordance with the customer needs that it was meant to cover.

ДП.045440-01-90 Вебзастосунок для управління діаграмами ціннісної пропозиції.
Відомість проекту

Позначення	Найменування	Кіл-ть	Примітка
	Документація проекту		
ДП.045440-02-91	Вебзастосунок для	5	
	управління діаграмами		
	ціннісної пропозиції.		
	Технічне завдання		
ДП.045440-03-81	Вебзастосунок для	61	
	управління діаграмами		
	ціннісної пропозиції.		
	Пояснювальна записка		
ДП.045440-04-51	Вебзастосунок для	4	
	управління діаграмами		
	ціннісної пропозиції.		
	Програма та методика		
	тестування		
ДП.045440-05-34	Вебзастосунок для	11	
	управління діаграмами		
	ціннісної пропозиції.		
	Керівництво користувача		
ДП.045440-06-99	Вебзастосунок для	1	
	управління діаграмами		
	ціннісної пропозиції.		
	Діаграма діяльності		
	основного сценарію		
	використання. Схема		
	алгоритму		

[illegible]

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ” _____ 2024 р.

ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ ДІАГРАМАМИ ЦІННІСНОЇ
ПРОПОЗИЦІЇ

Технічне завдання

ДП.045440-02-91

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Леся ЛЮШЕНКО

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Микита МАЛЬЦЕВ

2024

ЗМІСТ

1. Найменування та галузь застосування.....	3
2. Підстава для розроблення	3
3. Призначення розробки.....	3
4. Вимоги до програмного продукту	3
5. Вимоги до проєктної документації	4
6. Етапи проєктування	4
7. Порядок тестування розробки.....	5

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Вебзастосунок для управління діаграмами ціннісної пропозиції.

Галузь застосування: маркетингові дослідження, консультації з питань комерційної діяльності, фінансів управління.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на дипломне проєктування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для використання маркетологами та підприємцями в якості програмного забезпечення для керування діаграмами ціннісної пропозиції, та задля їх розробки та публікування.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Вебзастосунок повинен забезпечувати такі основні функції:

- 1) створення, редагування та перегляд діаграм ціннісної пропозиції;
- 2) управління доступом до створених діаграм ціннісної пропозиції;
- 3) поширення ціннісної пропозиції за посиланням;
- 4) розміщення діаграм ціннісної пропозиції у теках;
- 5) додавання тегів до елементів ціннісної пропозиції;
- 6) додавання тегів до діаграми ціннісної пропозиції;
- 7) перегляд інформації щодо автора ціннісної пропозиції.

Додаткові вимоги:

- 1) підвищена стійкість вебзастосунку;
- 2) підтримка веббраузерів: Chrome, Microsoft Edge, Opera, Brave;
- 3) захищеність підключення користувача до вебзастосунку;
- 4) безпечне зберігання облікових даних користувача.

5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проєкту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
 - «Діаграма діяльності основного сценарію використання»;
 - «Схема бази даних вебзастосунку».

6. ЕТАПИ ПРОЄКТУВАННЯ

Вивчення літератури за тематикою проєкту	17.11.2024
Розроблення та узгодження технічного завдання.....	27.11.2024
Розроблення структури вебзастосунку	18.12.2024
Підготовка першого розділу дипломного проєкту	02.02.2025
Розроблення дизайну сторінок та графічних елементів.....	29.02.2025
Підготовка другого розділу дипломного проєкту	14.03.2025
Програмна реалізація вебзастосунку	29.03.2025
Тестування вебзастосунку	17.04.2025
Підготовка третього розділу дипломного проєкту	23.04.2025
Підготовка четвертого розділу дипломного проєкту	01.05.2025
Підготовка матеріалів графічної частини проєкту	10.05.2025
Оформлення технічної документації проєкту	24.05.2025

7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ

Тестування розробленого програмного застосунку виконується відповідно до «Програми та методики тестування».

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ” _____ 2025 р.

**ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ ДІАГРАМАМИ ЦІННІСНОЇ
ПРОПОЗИЦІЇ**

Пояснювальна записка

ДП.045440-03-81

«ПОГОДЖЕНО»

Керівник проекту:

_____ Леся ЛЮШЕНКО

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Микита МАЛЫЦЕВ

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	3
ВСТУП	6
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ	7
1.1. Загальні положення та аналіз предметної області.....	7
1.2. Аналіз існуючих рішень	8
1.3. Актуальність розробки веб-застосунку для управління діаграмами ціннісної пропозиції	13
1.4. Загальні вимоги до застосунку	14
1.5. Висновки до розділу	16
2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	18
2.1. Обґрунтування вибору бази даних	18
2.2. Обґрунтування вибору мови програмування серверної частини вебзастосунку	20
2.3. Вибір бібліотек для серверної частини.....	23
2.4. Обґрунтування вибору фреймворку для клієнтської частини вебзастосунку	25
2.5. Висновки до розділу	27
3. РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ	29
3.1. Загальний опис структури застосунку “Scheme chef”.....	29
3.2. Архітектура вебзастосунку «Scheme chef».....	35
3.3. База даних вебзастосунку «Scheme chef»	38
3.4. Висновки до розділу	44
4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБЗАСТОСУНКУ	46
4.1. Опис інтерфейсу користувача.....	46
4.2. Тестування вебзастосунку	49
4.3. Пропозиції для майбутнього покращення	54
4.4. Висновки до розділу	55
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	59
ДОДАТКИ.....	61

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ПЗ – програмний застосунок.

БД – база даних.

СКБД – Система керування базами даних.

API (Application Programming Interface) – підхід до архітектури мережових протоколів, які надають доступ до інформаційних ресурсів.

HTTP (Hyper Text Transfer Protocol Secure) – протокол, який використовується для передачі даних через комп'ютерну мережу.

HTTPS (Hyper Text Transfer Protocol) – розширення протоколу HTTP, що застосовує шифрування для комунікації у мережі.

JSON (JavaScript Object Notation) – формат обміну даними, який використовується для передачі структурованих даних між різними системами.

HTML – мова розмітки гіпертексту.

Фреймворк – це набір заздалегідь розроблених структур, бібліотек, інструментів та шаблонів, які надають базову архітектурну основу для розробки програмного забезпечення.

Логін – ідентифікатор користувача, який використовується для авторизації в системі.

Аутентифікація – це процес перевірки ідентичності користувача, щоб забезпечити, що він справді той, за кого він себе видає.

Авторизація – це процес перевірки прав доступу користувача до певних ресурсів або функцій системи.

DOM (Document object model) – структура, що використовується для представлення вебсторінки в пам'яті комп'ютера та для доступу до її елементів за допомогою скриптів.

JS (JavaScript) – мова програмування вебзастосунків. *Unit-тестування (Unit Testing)* – це метод тестування програмного забезпечення, коли окремі

компоненти або модулі програми перевіряються на коректність та працездатність.

Unit-тестування – модульне тестування.

Системне тестування (System Testing) – це метод тестування програмного забезпечення що виконується на повній, інтегрованій системі, з метою перевірки відповідності системи вимогам.

UML-діаграма – статична структурна діаграма, яка описує структуру системи, демонструє класи системи, їхні атрибути, методи й залежності між класами.

Хмарне сховище даних – це модель зберігання даних, в якій цифрові дані накопичуються в логічні розділи, а для зберігання даних застосовуються віддалені сервери.

SWOT аналіз – засіб для стратегічного планування, що полягає в розділенні чинників і явищ на чотири категорії: сильних і слабких сторін проєкту, можливостей та загроз.

Project-kickoff діаграма – засіб для організації та планування зустрічі перед початком проєкту.

Point-in-time-recovery – можливість відновлення даних до попереднього стану у визначений період часу.

REST (Representational State Transfer) – підхід до архітектури мережевих протоколів, які надають доступ до інформаційних ресурсів.

SQL ін'єкція – спосіб злому інформаційних систем, що працюють з базами даних, заснований на впровадженні в запит довільного SQL-коду.

Goroutine – це легковагий потік керований середовищем виконання Go.

Docker – інструментарій для управління ізольованими Linux-контейнерами.

Kubernetes – система автоматичного розгортання, масштабування та управління застосунками у контейнерах.

OWASP (Open Web Application Security Project) – онлайн-спільнота, яка створює вільно доступні статті, методології, документацію, інструменти та технології в галузі безпеки вебзастосунків.

Dependency injection – шаблон проєктування програмного забезпечення, що передбачає надання зовнішньої залежності програмному компоненту.

Ahead-of-time Compilation – це вид транслятора, який використовує метод компіляції перед виконанням.

SQL-транзакції – це сукупність однієї або кількох SQL-операцій, як одна логічна одиниця роботи з даними.

Same-origin policy – визначення меж взаємодії із даними, отриманими від стороннього джерела.

CORS (Cross-Origin Resource Sharing) – механізм безпеки сучасних браузерів, який дозволяє вебсторінкам використовувати дані, що не відповідають *same-origin policy*.

Хмарні послуги – це функції чи послуги, які надаються або розміщуються через інтернет на платній основі.

OIDC (OpenID Connect) – це протокол автентифікації ідентичності, який є розширенням стандарту Open Authorization (OAuth) 2.0, розроблений із метою стандартизувати процес автентифікації та авторизації користувачів під час входу для доступу до цифрових служб.

ВСТУП

Однією із методик маркетингового аналізу потреб користувачів та їх оцінки їх співвідношення із характеристиками продукту є створення діаграм ціннісних пропозицій. Процес розробки продукту із застосуванням ціннісної пропозиції вимагає проведення аналізу профілю клієнтів та властивостей продукту. Один із підходів до дослідження цільової аудиторії передбачає створення окремих ціннісних пропозицій для різних груп потенційних клієнтів, що додатково збільшує обсяг підготовчої роботи при розробленні продукту. Особливо на етапі проектування продукту, важливо цілісно проаналізувати потреби користувачів та відповідні якості продукту що розроблюється.

За статистикою, близько 39% нових продуктів зазнають невдачі на ринку [1]. До того ж близько 50% підприємств припиняють свою діяльність впродовж 5 років [2]. Розповсюдженими помилками, що призводять до створення не затребуваних на ринку продуктів є відсутність визначення ключових цілей та потреб цільової аудиторії, а також не розуміння наявних на ринку пропозицій [3]. Тому є актуальною задачею розробка програмного забезпечення, яке дозволить проводити конкурентне просування нових продуктів відповідно до ціннісної пропозиції для певних зацікавлених сторін.

Використання інформаційних технологій, що сприяють проведення аналізу продуктів та потенційних споживачів, є ключовою складовою для створення та просування сучасних та конкурентоспроможних продуктів. Завдяки цьому, актуальною є розробка веб-застосунку, що б дозволив частково автоматизувати процес створення діаграми ціннісної пропозиції.

Метою даного дипломного проєкту є створення вебзастосунку, що надаватиме функціональність для створення ціннісних пропозицій та додавання нових якостей для вже створених ціннісних пропозицій за допомогою сучасних програмних технологій.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1. Загальні положення та аналіз предметної області

Використання інформаційних технологій є важливим складником процесів, що забезпечують роботу підприємств у сучасному світі. Доступність інформації у реальному часі суттєво позначилося на спроможності людей порівнювати аналоги пропонованих товарів, та обирати такі, що найкраще задовольняють індивідуальні потреби. Сукупність таких чинників виявила потребу використовувати сучасне програмне забезпечення для рекламування продуктів на ринку, та аналізувати відповідність продуктів потребам користувачів.

Галузь маркетингу відзначається збільшенням рівня застосування інформаційних систем, спрямованих на автоматизацію низки процесів, що забезпечують рекламну діяльність. Програмне забезпечення застосовується як для безпосереднього розміщення рекламних матеріалів на платформах розміщення оголошень, так і для оптимізації пошуку рекламованих товарів, управління контентом, планування структури рекламної кампанії та маркетингової стратегії [4]. Відкритість цього виду діяльності до впровадження програмних інформаційних систем свідчить про перспективність розробки програмного забезпечення для цієї галузі.

Збільшення кількості споживачів, які можуть бути зацікавлені у придбанні продуктів компаній, забезпечується методами аналізу інформації щодо користувачів та вимог до запропонованого продукту. Одним з інструментів, що використовується для проведення такого аналізу є створення діаграм ціннісної пропозиції. Вказаний вид аналізу потреб користувача та властивостей продукту також використовується при проектуванні нових продуктів та продуктів, які із незначними змінами можуть бути виведені на ринок. Отже актуальним є застосунок, який дозволить автоматизувати процес створення та розвиток ціннісної пропозиції для різних продуктів споживання.

1.2. Аналіз існуючих рішень

Задля подальшої оцінки актуальності розробки вебзастосунку для управління ціннісними пропозиціями, було проведено аналіз існуючих програмних рішень. Під час дослідження було виявлено програмні продукти, що можуть використовуватись як товари-замінники для побудови діаграм ціннісних пропозицій, такі як draw.io та Canva. а також програмні продукти що забезпечують штучне генерування ціннісних пропозицій, такі як valueprop.dev.

Проведемо більш детальний аналіз зазначених програмних продуктів, які можуть використовуватись для створення діаграм, та для автоматичної генерації діаграми ціннісної пропозиції.

1.2.1. Draw.io

Draw.io – це кросплатформне програмне забезпечення, що надає функціональність для створення креслень, таких як UML-діаграми, каркасні моделі, алгоритмів задач та інших діаграм, що можуть бути побудовані на основі вбудованих складових елементів, в тому числі — і ціннісних пропозицій. Створені у цьому програмному продукті креслення можуть буди завантажені на пристрій користувача у якості зображення, або як файл формату .drawio, що дозволяє продовжити редагування на іншому пристрої, використовуючи даний засіб для створення діаграм. Також надається можливість зберігати діаграми у хмарних сховищах даних, таких як Google Drive та One Drive.

Засіб для створення діаграм Draw.io може використовуватись у браузері клієнтів, що використовують будь-яку операційну систему. Також, продукт може бути встановлений як окремий застосунок для операційних систем Windows, Linux та macOS [5]. На рис. 1.1 показано вебінтерфейс застосунку.

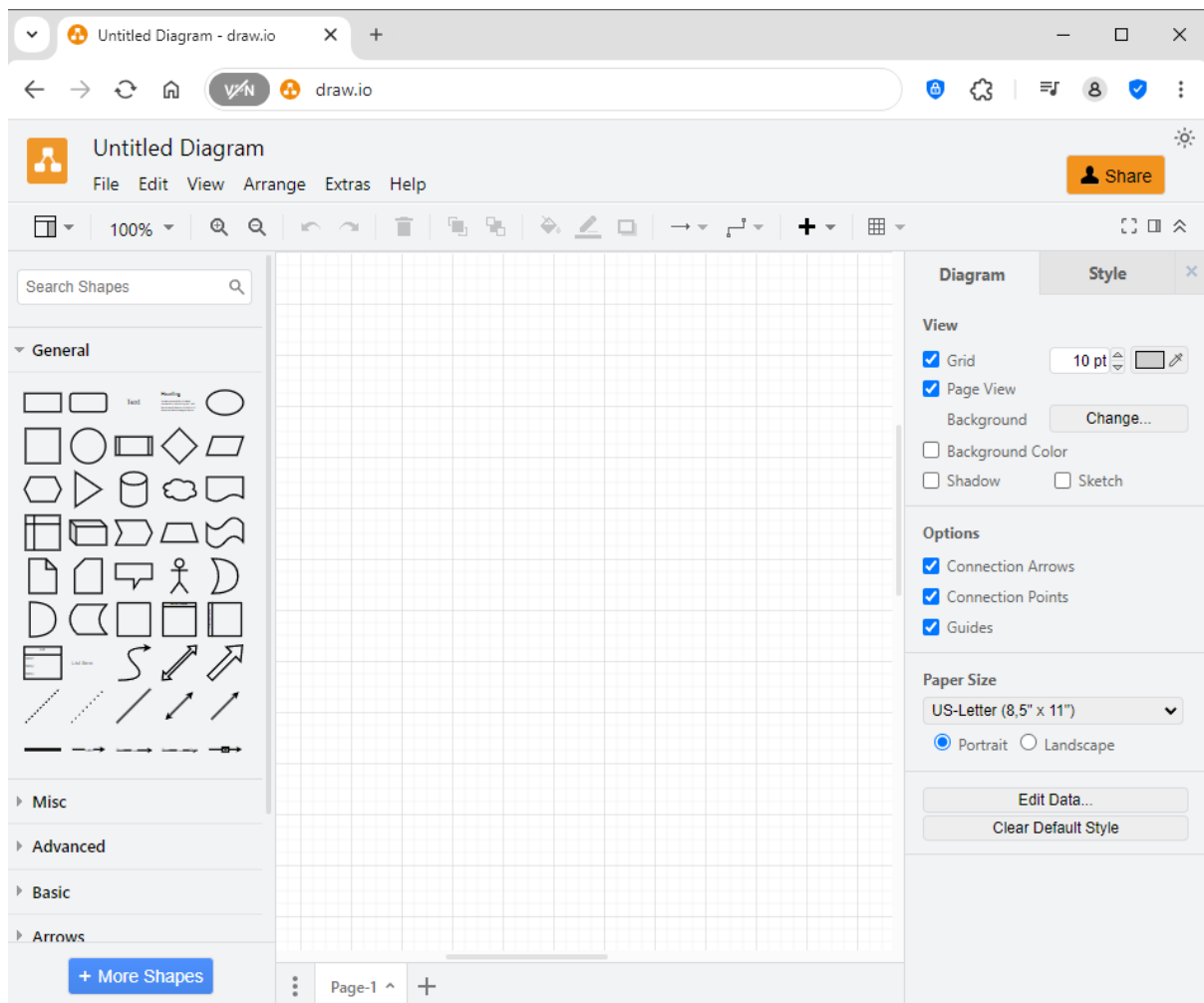


Рис. 1.1. Вебінтерфейс застосунку Draw.io

Переваги Draw.io:

- 1) Можливість побудови діаграм різних типів.
- 2) Контроль користувача над місцем зберігання даних.
- 3) Можливість використовувати застосунок без інтернет-з'єднання.

Недоліки Draw.io:

- 1) Відсутність спеціалізованих шаблонів та підказок щодо ціннісних пропозицій.
- 2) Відсутність вбудованих засобів задля організації груп діаграм.
- 3) Потреба авторизуватись у сторонніх сервісах задля поширення діаграм.
- 4) Відсутність інструментів аналізу тексту.
- 5) Відсутність генерації текстів ціннісної пропозиції.

1.2.2. Canva

Canva – це програмний продукт що надає низку засобів для командної кооперації для створення графічного дизайну, діаграм, презентацій та інших мультимедійних продуктів (рис. 1.2). Програмний засіб надає також низку професійних шаблонів, зокрема для проведення SWOT-аналізу, Project-kickoff діаграм, а також ціннісних пропозицій. Окрім цього, програмний продукт надає можливості для колаборації в реальному часі при створенні діаграм групою з кількох користувачів.

Окремо слід зазначити вбудовану підтримку плагінів від сторонніх розробників. За допомогою таких плагінів можливо розширити наявну функціональність Canva, та таким чином отримати більш спеціалізований інструмент для роботи з ціннісними пропозиціями.

Вказаний програмний продукт окрім використання вебверсії застосунку у браузері, також можна встановити на низку операційних систем, а саме: Android, ChromeOS, iOS, iPadOS, Linux, macOS, Windows [6].

Переваги Canvas:

- 1) Можливість колаборації в реальному часі при створенні діаграм.
- 2) Публікація діаграм за допомогою створення посилання.
- 3) Наявність шаблонів для створення діаграм різних типів.
- 4) Підтримка плагінів, що розширюють функціональність програмного забезпечення.
- 5) Можливість групувати діаграми у теки.

Недоліки Canvas:

- 1) Відсутність функціональності для автоматизації створення ціннісних пропозицій.
- 2) Не підтримується надання підказок щодо елементів ціннісної пропозиції.
- 3) Відсутність простору для перегляду ціннісних пропозицій, опублікованих користувачами застосунку.

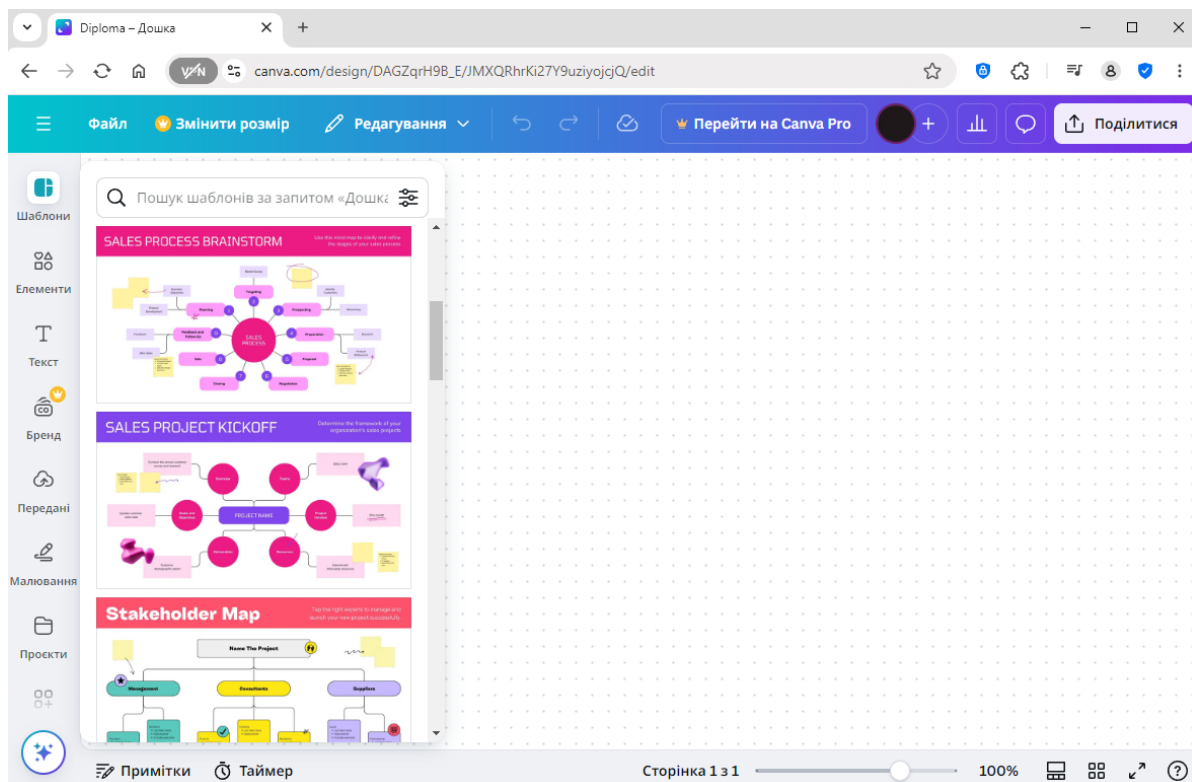


Рис. 1.2. Вебінтерфейс застосунку Canva

1.2.3. *valueprop.dev*

Valueprop.dev – це вебзастосунок, який дозволяє генерувати діаграму ціннісної пропозиції, на основі наданого опису продукту, щодо якого розробляється ціннісна пропозиція. Вебзастосунок надає обмежені можливості щодо редагування створених діаграм. Також, не передбачено можливостей для перегляду списку власних діаграм. Водночас, застосунок не містить функціональності задля керування доступом до ціннісної пропозиції. При створенні нової діаграми надається єдине посилання доступу, що надає права як перегляду так і редагування діаграми. Через це його використання у галузі маркетингу може бути недоцільно. Користувацький інтерфейс вебзастосунку для генерації ціннісної пропозиції valueprop.dev зображено на рис. 1.3.

На відміну від аналогів, даний застосунок доступний виключно в інтернет-браузерах, таких як Chrome, Opera, Microsoft Edge.

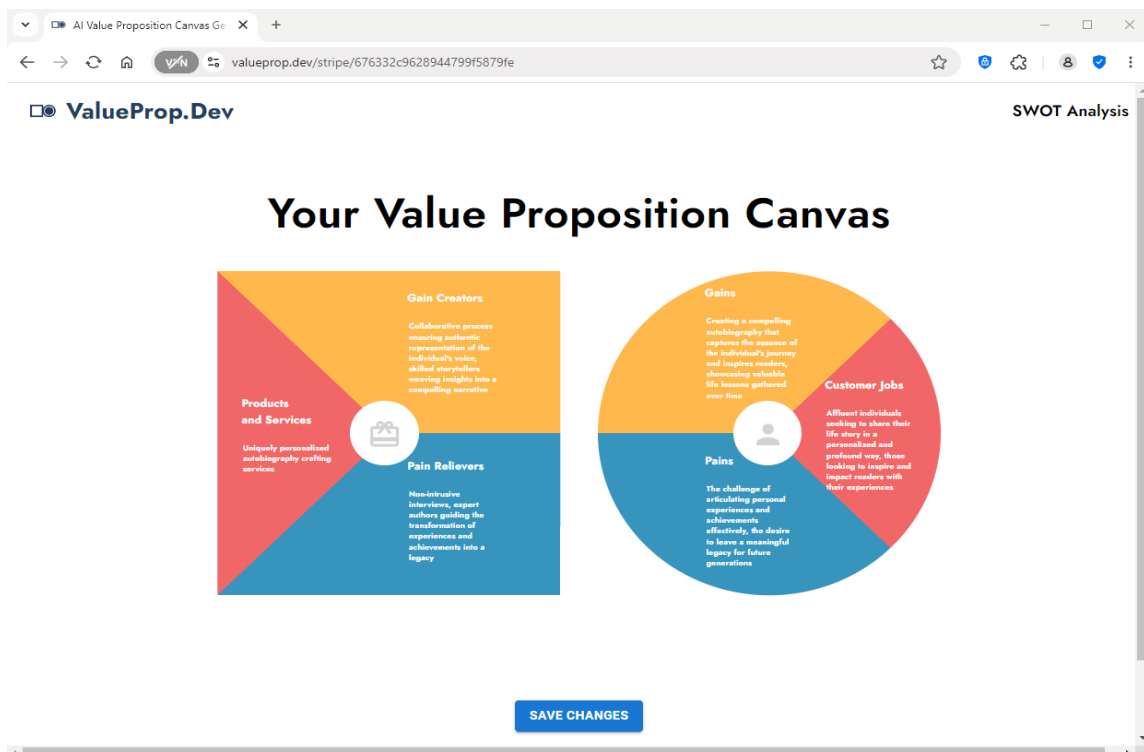


Рис. 1.3. Вебінтерфейс застосунку valueprop.dev

Переваги valueprop.dev:

- 1) Спеціалізовані підказки щодо створення ціннісної пропозиції.
- 2) Наявність можливості генерувати елементи ціннісної пропозиції за допомогою штучного інтелекту.

Недоліки valueprop.dev:

- 1) Відсутність безплатної версії.
- 2) Неможливість додати більше одного компонента ціннісної пропозиції для кожної категорії елементів.
- 3) Не передбачена можливість групування створених діаграм у спільній теці.

Отже, жоден із досліджуваних програмних продуктів не є прямим конкурентом розроблюваного програмного забезпечення, через відсутність засобів автоматизації процесу створення діаграм ціннісних пропозицій, визначених у аналізі предметної області.

Таким чином, програмні продукти, що можуть застосовуватись для створення діаграм ціннісної пропозиції, мають функціональність, що задовольняє лише частину потреб для управління ціннісною пропозицією.

1.3. Актуальність розробки веб-застосунку для управління діаграмами ціннісної пропозиції

Створення діаграми ціннісної пропозиції, яка не лише відображатиме відомі характеристики продукту та потреби поточних користувачів, а й відкриватиме нові можливості для створення нових конкурентних переваги продукту, а також вказуватиме на нові групи користувачів, які можуть бути зацікавлені у використанні продукту. Застосунок дозволить сконцентруватися на визначені визначних характеристик товарів. Функціональність для управління створеними діаграмами та можливість надавати діаграми зацікавленим сторонам також буде передбачена у застосунку. Перевіримо, чи наявні рішення надають функціональність задля задоволення вказаних потреб, виконавши їх порівняльний аналіз у табл. 1.1.

Таблиця 1.1

Порівняння функціональності застосунків

Застосунок	Draw.io	Canva	valueprop.dev
Наявність шаблонів ціннісної пропозиції	—	+	+
Отримання підказок щодо елементів ціннісної пропозиції	—	—	—
Вбудована можливість поширення діаграми за допомогою посилання	—	+	+
Керування доступом до редагування діаграми	+	+	—

Продовження табл. 1.1

Можливість перегляду опублікованих ціннісних пропозицій спільноти	—	—	—
Групування діаграм у спільні теки	—	+	—
Наявність безкоштовної версії	+	+	—
Можливість штучного створення ціннісної пропозиції на основі опису продукту	—	—	+

Таким чином можна побачити, що наявні на ринку програмні продукти не поєднують в собі усю необхідну функціональність, яка потрібна для автоматизації процесу створення ціннісних пропозицій. В той же час, це обумовлює актуальність розробки спеціалізованого програмного забезпечення. Тому застосунок “Scheme chef” буде містити необхідні властивості, щоб не лише дозволити автоматизацію створення діаграм ціннісної пропозиції, а також, щоб розроблений продукт відповідав основним вимогам для безпечного та безперешкодного його використання у процесах роботи підприємств.

1.4. Загальні вимоги до застосунку

Можна визначити перелік вимог, що мають передбачати створення програмного забезпечення, яке відповідатиме очікуванням до сучасного програмного продукту.

Основними функціональними вимогами, які повинні забезпечити відповідність розробки потребам та очікуванням користувачів є:

- можливість створення акаунту у веб-застосунку на основі електронної пошти;
- наявність шаблону для створення ціннісної пропозиції;

- можливість додавати, змінювати та видаляти елементи ціннісної пропозиції;
- отримання підказок щодо елементів ціннісної пропозиції, на основі часткового введення їх змісту;
- пошук елементів ціннісної пропозиції відповідно до присвоєних їм тегів;
- можливість доступу до перегляду змісту діаграми за посиланням;
- наявний механізм управління доступом до зміни вмісту діаграм;
- можливість групувати діаграми в окремі теки;
- користувацький інтерфейс, що сприяє безперешкодній роботі із ціннісною пропозицією;
- автоматична генерація ціннісної пропозиції на основі пошуку елементів найбільш відповідних темі діаграми.

Вебзастосунок має задовольняти такі нефункціональні вимоги:

- вебзастосунок доступний щонайменше 99% часу впродовж кожного місяця [7];
- користувацький інтерфейс доступний із таких браузерів як: Chrome, Microsoft Edge, Opera, Firefox;
- конфіденційна інформація користувачів не розголошується при роботі застосунку;
- опції конфігурації передаються застосунку безпечним чином.

Через збільшену кількість загроз щодо можливих атак на програмне забезпечення, слід окремо звернути увагу на вимоги до безпеки. Вебзастосунок має бути захищеним від загроз, що можуть впливати на конфіденційність даних користувачів та на безвідмовну роботу програмного продукту. У табл. 1.2 наведу основні види таких загроз, у відповідності із вимогами щодо безпеки, які мають гарантувати стійкість програмного забезпечення від зловмисних дій.

Задля дотримання якості розроблюваного програмного продукту, розробка супроводжуватиметься ручним тестуванням програмного

застосунку. Надійність вебзастосунку буде забезпечуватись за рахунок використання кращих практик програмування, а також за рахунок використання сучасних технологій розробки програмного забезпечення.

Таблиця 1.2

Вимоги до безпеки застосунку

Загроза	Вимога безпеки
Клієнти отримують доступ до закритих профілів інших користувачів та до закритих ціннісних пропозицій, підбираючи ідентифікатор у командному рядку	Ідентифікатори доступу до закритої інформації мають бути такими, що значною мірою ускладнюють підбір ідентифікатора користувачами.
Користувачі отримують доступ до перегляду та зміни інформації в базі даних, за допомогою виконання SQL-ін'єкцій, що призводить до порушення цілісності даних та перебоїв в роботі ПЗ	Механізм створення запитів до бази даних є стійким до SQL-ін'єкцій
Конфіденційні дані користувачів розкриваються, через їх збереження у журналі дій програмного продукту	Засіб ведення журналу дій програмного продукту видаляє конфіденційну інформацію, перед введенням записів у журнал дій

1.5. Висновки до розділу

У галузі маркетингу широко застосовуються інформаційні технології, які автоматизують процеси, пов'язані із рекламною діяльністю, що допомагає покращити результати маркетингових кампаній. Впровадження програмного забезпечення для управління діаграмами ціннісної пропозиції сприятиме подальшому покращенню процесів ведення рекламної діяльності.

Аналіз наявних рішень для створення діаграм ціннісної пропозиції, таких як Draw.io та Canva показав наявність у них функціональності для створення діаграм, проте вони не містили специфічної функціональності,

необхідної для автоматизації створення ціннісних пропозицій та управління ними. Водночас, такий продукт як Valueprop.dev не містив засобів для ефективного управління створеними ціннісними пропозиціями, а також його можливості із покращення наявних ціннісних пропозицій є значно обмежені.

Провівши порівняльний аналіз наявних рішень, було визначено потребу у розробці програмного продукту, що поєднуватиме в собі засоби, необхідні для автоматизації процесу створення діаграм ціннісної пропозиції. У разі забезпечення основних вимог до програмного продукту для управління ціннісними пропозиціями, такий інструмент покращить аналіз відповідності продукту потребам користувача.

Було сформовано вимоги до вебзастосунку для управління діаграмами ціннісної пропозиції, що мають забезпечити наявність функціональності для автоматизації створення та покращення діаграм ціннісної пропозиції, а також основні можливості для управління, контролю доступу та поширення створених діаграм. Також, було визначено нефункціональні вимоги, що мають сприяти стабільній роботі продукту на більшості цільових платформах. Задля відповідності застосунку встановленим вимогам, було зазначено необхідність проведення тестування розроблюваного програмного продукту.

2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Обґрунтування вибору бази даних

Вибір бази даних є ключовим етапом проєктування інформаційної системи. Обране рішення має відповідати типу даних що зберігатимуться та зв'язкам між такими даними. Створюване програмне забезпечення взаємодіятиме зі структурованими даними визначеного типу. Також, система значною мірою покладатиметься на зв'язки між сутностями бази даних. Через це, розглянемо найбільш часто вживані бази даних реляційного типу.

SQL Server це СКБД розроблена компанією Microsoft, що використовується для управління структурованими даними. Однією із характеристик цієї бази даних є високий рівень швидкодії, який доступний без додаткових налаштувань. Вказана СКБД відповідає вимогам ACID, що забезпечує цілісність даних при виконанні операцій із даними.

Іншою особливістю SQL Server є висока розвиненість вбудованих засобів для моніторингу та підтримки працездатності бази даних. Вбудована функціональність СКБД дозволяє створювати резервні копії для point in time recovery інформації у базі даних.

Визначною ознакою SQL Server є здатність її інтеграції з іншими продуктами Microsoft, такими як засоби для моніторингу систем та аналітики даних. Це є значною перевагою використання СКБД у системах, у яких широко залучене інше програмне забезпечення компанії Microsoft.

Слід зазначити, що SQL Server є ліцензованим продуктом. Його безплатна версія має обмеження функціональності. Використання ж повної версії СКБД може призвести до значних витрат на ліцензію продукту.

Одним із недоліків даного програмного продукту є більш жорсткі вимоги до апаратного забезпечення, ніж вимоги інших СКБД розглянутих у цьому розділі.

PostgreSQL є потужною системою керування реляційними базами даних, яка має ряд переваг при створенні вебзастосунків. PostgreSQL надає широкий спектр вбудованих типів даних, включаючи числові, текстові, дати/часу, а також складні типи, такі як масиви. Можливість використання масивів дозволяє зберігати та обробляти структуровані дані без необхідності створення додаткових таблиць, що спрощує розробку та покращує продуктивність.

Найбільші постачальники хмарних послуг, такі як AWS, Google Cloud Platform та Microsoft Azure, пропонують керовані сервіси PostgreSQL. Це дозволяє легко розгортати та масштабувати бази даних PostgreSQL у хмарному середовищі.

PostgreSQL підтримує різні методи масштабування, включаючи вертикальне та горизонтальне масштабування. Можливість налаштування основної бази даних та реплік дозволяє розподіляти навантаження та забезпечувати високу доступність та відмовостійкість.

PostgreSQL повністю відповідає вимогам ACID у всіх конфігураціях. Завдяки цьому PostgreSQL є особливо придатним для застосунків, де цілісність даних є критично важливою.

PostgreSQL надає широкий спектр типів індексів, включаючи B-tree, hash, GiST, GIN та SP-GiST. Це дозволяє оптимізувати пошук даних для різних типів запитів та даних [10].

MySQL широко використовуваною системою керування базами даних, але вона має певні особливості, які важливо враховувати при виборі СКБД. MySQL надає достатній набір основних типів даних, таких як цілі числа, числа з плаваючою комою, рядки та дати/час. Однак, MySQL має менш розвинену підтримку складних типів даних. Так MySQL не підтримує вбудовані типи даних для масивів, що може зменшити ефективність роботи застосунку у окремих випадках.

MySQL широко підтримується всіма основними постачальниками хмарних послуг, такими як AWS, Google Cloud Platform та Microsoft Azure.

Це забезпечує легкість розгортання та керування базами даних MySQL у хмарному середовищі.

MySQL підтримує реплікацію для масштабування читання та забезпечення високої доступності. Ця СКБД може масштабуватися як вертикально так і горизонтально, що надає достатню гнучкість для проєктування інформаційної системи.

MySQL забезпечує також забезпечує відповідність ACID, але лише при використанні певних рушіїв пошуку, зокрема InnoDB та NDB Cluster.

MySQL також підтримує індексацію, але вибір типу індексу обмежений B-tree та R-tree індексами.

На основі проведеного аналізу, було обрано PostgreSQL для побудови інформаційної системи. Ця СКБД надає більшу гнучкість у зберіганні різних типів даних, гарантує цілісність даних у всіх конфігураціях, та має ширші можливості для оптимізації швидкодії. До того ж, повна функціональність даної СКБД доступна на безоплатній основі.

2.2. Обґрунтування вибору мови програмування серверної частини вебзастосунку

Розроблюваний вебзастосунок має клієнт-серверну архітектуру. Для забезпечення достатньої швидкодії, та сумісності було проведено огляд мов програмування серверної частини застосунку.

Мова програмування Go є сучасною мовою програмування, яка часто використовується для розробки вебзастосунків. Ця мова програмування надає можливості створення розширюваних та високопродуктивних програмних систем. За допомогою даної мови програмування можна створювати програмні продукти для таких операційних систем як Windows, Linux, macOS та інших. Створене програмне забезпечення може використовуватись із процесорами архітектури x86, x86-64, ARM, ARM64, MIPS, MIPS64. Однією з переваг цієї мови програмування є можливість використання програмного продукту без встановлення додаткових бібліотек

або інших програмних засобів. Це дозволяє більш ефективно використовувати контейнеризацію для розгортання створеного програмного забезпечення.

Java – це об'єктно-орієнтована мова програмування, яка широко використовується для розробки серверної частини вебзастосунків. Вона забезпечує можливість створення масштабованих, безпечних та кросплатформених програмних рішень. Програмне забезпечення, написане на Java, може виконуватись на різних операційних системах, зокрема Windows, Linux, macOS та інших, за умови наявності віртуальної машини Java. Мова Java також надає численні бібліотеки та фреймворки, що полегшують розробку застосунків різної складності. Завдяки широкій підтримці спільноти та постійному оновленню, Java залишається актуальним інструментом для розробників програмного забезпечення.

PHP є популярною мовою програмування, що спеціалізується на розробці вебзастосунків. Використовуючи PHP, розробники можуть створювати динамічні та інтерактивні вебзастосунки, які можуть взаємодіяти з базами даних і обробляти користувацькі запити в реальному часі. Однією з ключових переваг PHP є вбудована підтримка роботи з різноманітними базами даних, зокрема MySQL, PostgreSQL та SQLite. Завдяки великому вибору фреймворків, таких як Laravel, Symfony та CodeIgniter, PHP дає змогу ефективно створювати вебзастосунки. Для запуску PHP-застосунків необхідно мати встановлений вебсервер, такий як Apache або Nginx, а також інтерпретатор PHP.

На основі проведеного аналізу було обрано мову програмування Go для розробки серверної частини вебзастосунку. Вибір мови програмування Go для розробки API обумовлений рядом технічних переваг, які сприяють ефективності, надійності та безпеці вебзастосунків:

У мові передбачена вбудована підтримка взаємодії із JSON, який є стандартом для обміну інформацією у вебзастосунках, що спрощує обробку

вхідних даних. Стандартна бібліотека `net/http` забезпечує зручні інструменти для обробки HTTP-запитів, що дозволяє ефективності розробки REST API.

Для обраної мови програмування доступно багато бібліотек для роботи з базами даних, які забезпечують безпечне підключення та виконання запитів. Найбільш вживаними бібліотеками для взаємодії із реляційними базами даних є: `database/sql`, `GORM`, `sqlx`.

Однією з вимог до безпеки застосунку вказаних у табл. 1.2, яку задовольняють вказані вище бібліотеки, є протидія SQL-ін'єкціям.

Для розробки серверної частини застосунку `Scheme chef` обираємо мову програмування `Go`, яка підтримує паралельне виконання коду за допомогою `Goroutines`. Це дозволяє збільшити ефективність обробки великої кількості одночасних запитів. `Goroutines` споживають менше обчислювальних ресурсів, ніж засоби впровадження багатопотоковості у мовах `Java` та `PHP`. До того ж використання `Goroutines` спростить застосування ефективних засобів багатопотоковості при розробці `Scheme chef`.

Оскільки `Go` – це мова зі статичною типізацією, що дозволить виявляти помилки на етапі компіляції. Це спростить процес тестування розроблюваного ПЗ та сприятиме підвищенню надійності та стабільності серверної частини вебзастосунку "`Scheme chef`". Компіляція коду `Go` в машинний код забезпечує вищу продуктивність ПЗ у порівнянні із інтерпритованими мовами програмування, такими як `PHP`. Дана особливість важлива для серверної частини вебзастосунку, що має обробляти запити із високою швидкістю.

`Go` є мовою високим чином пристосованою для контейнеризації, оскільки скомпільовані програми `Go` мають невеликий розмір і не потребують додаткових залежностей. Це дозволяє легко створювати та розгортати контейнеризовані вебзастосунки за допомогою `Docker` та `Kubernetes`.

Таким чином, Go є засобом для розробки, здатним передбачити безпеку та належну швидкодію вебзастосунків. При цьому, властивості даної мови програмування забезпечує сумісність із сучасними засобами контейнеризації та архітектурами обчислювальних систем. Окрім зазначених переваг використання мови, слід вказати ще й на окремі недоліки мови:

Go не підтримує класичне успадкування, яке є ключовим компонентом ООП. Проте, Go надає інші механізми, такі як інтерфейси та композиція, які дозволяють досягти подібної функціональності. У певних випадках, такий підхід навіть може забезпечити більшу гнучкість при розробці.

Відсутність конструкцій try/catch, що широко застосовується в інших мовах програмування, змушує використовувати інші засоби для обробки помилок. У свою чергу, це може сприяти створенню більш досконалії обробці помилок засобами мови Go.

Відносно невелика, порівняно з деякими іншими мовами, кількість сторонніх бібліотек може обмежувати вибір засобів для розробки мовою Go. Однак, стандартна бібліотека Go є дуже потужною і надає багато необхідних інструментів, що зменшує залежність від сторонніх бібліотек.

Незважаючи на ці особливості, Go залишається надзвичайно ефективною та надійною мовою для розробки серверної частини вебзастосунків. Його простота, продуктивність та підтримка багатопотоковості роблять його придатним для створення високопродуктивних та масштабованих програмних продуктів.

2.3. Вибір бібліотек для серверної частини

Проведемо огляд доступних бібліотек для побудови серверної частини вебзастосунку. Найбільший вплив на створення серверної частини застосунку матиме вибір бібліотеки для маршрутизації запитів. Важливо обрати бібліотеку що надаватиме широкі можливості для обробки запитів,

та буде сумісною із іншими бібліотеками, що можуть використовуватись при розробці.

Бібліотека `gorilla/mux` є маршрутизатором HTTP-запитів для Go, який часто використовується для створення REST API. Вона дозволяє визначати маршрути на основі різних критеріїв, таких як метод HTTP, шлях URL, параметри запиту та заголовки. Також ця бібліотека передбачає застосування регулярних виразів для маршрутизації запитів. Бібліотека `gorilla/mux` сумісна з іншою бібліотекою – `gorilla/sessions`, що надає функціональність для керування користувацькими сесіями. Варто також зазначити наявність великої спільноти розробників, що підтримують працездатність бібліотеки та створюють оновлюють бібліотеки.

У мові програмування Go також є вбудована бібліотека для обробки HTTP запитів. Бібліотека містить вбудований засіб маршрутизації запитів до різних обробників, що пов'язує URL-шлях з функцією-обробником. Також у бібліотеці є структура для обробки для більш складних сценаріїв маршрутизації. Ця бібліотека може інтегруватися зі сторонніми бібліотеками, проте для стабільної взаємодії із сторонніми засобами може знадобитись додаткова конфігурація.

Завдяки інтеграції з іншими бібліотеками, що будуть використовуватись при розробці програмного продукту, вибір бібліотеки `gorilla/mux` сприятиме ефективності розробки вебзастосунку “Scheme chef”.

Окремо слід дослідити доступні засоби задля забезпечення безпеки зберігання такої інформації як пароль користувача. Обрана технологія має бути основою механізму авторизації користувачів. На основі паролів користувачів мають бути створені хеші, які не можна декодувати до початкового паролю. Бібліотека, що буде використовуватись, має створювати хеші, та здійснювати перевірку відповідності введених при вході в акаунт паролів до збережених хешів.

Найбільш поширені засоби для хешування та перевірки паролів – бібліотеки `crypto/bcrypt` та `alexedwards/argon2id`. Визначу, яка з вказаних бібліотек найкраще задовольнятиме вимоги до безпеки зберігання паролів:

Найбільш широко використовуваним засобом для хешування паролів та перевірки їх відповідності при авторизації є `crypto/bcrypt`. Алгоритм `Bcrypt` має низку переваг над алгоритмами хешування, таких як `SHA256`, які передбачають що для одного й того самого набору вхідних даних буде створено однаковий хеш. Натомість, алгоритм `crypto/bcrypt` може створювати різні хеші для однакового набору вхідних даних, при цьому не втрачаючи можливість перевірити збір вхідних даних та хешу. Цей алгоритм шифрування потребує помірного об'єму оперативної пам'яті для створення хешу, що може ускладнити спроби підібрати початковий пароль.

Бібліотека для створення та перевірки хешів `alexedwards/argon2id` також надає засоби для створення хешів на основі однакових вхідних даних. Однак, алгоритм створення та перевірки хешу більш ресурсномісткий, завдяки чому досягається збільшений рівень захищеності хешу від спроб підбору паролю. Використання даного засобу шифрування рекомендовано організацією OWASP як більш надійної альтернативи `bcrypt` [11].

Виконавши аналіз наявних засобів хешування паролів, оберу бібліотеку `argon2id`. Хоча її використання вимагатиме більше ресурсів серверної частини застосунку, підвищений рівень безпеки паролів користувачів сприятиме підвищенню рівня безпеки інформаційної системи.

2.4. Обґрунтування вибору фреймворку для клієнтської частини вебзастосунку

При обранні програмного засобу для розробки клієнтської частини вебзастосунку, слід зважати на стабільну роботу фреймворку та на доступність стратегій розгортання створеного програмного продукту. Розгляну такі сучасні засоби для розробки вебзастосунків як `React`, `Angular` та `Flutter`:

React – це бібліотека для розроблення вебзастосунків мовою програмування JavaScript. Однією з вагомих переваг React є застосування Virtual DOM [12], яка сприяє зменшенню кількості операцій, необхідних для реагування на зміну стану застосунку. Відповідно до стратегії Virtual DOM зміна стану застосунку впливатиме лише на компоненти, вхідні дані для яких зазнали змін. Такий підхід дозволяє пришвидшити швидкодію застосунку. React дозволяє створювати компоненти користувацького інтерфейсу, що можуть повторно використовуватися. Такі компоненти містять власну логіку, що сприяє підвищенню ефективності тестування та налагодження вебзастосунку. Для розгортання програмного продукту, можна використати системи розповсюдження статичних файлів, що сприятиме підвищенню доступності застосунку при завантаженні.

Фреймворк для розробки вебзастосунків Angular містить низку засобів для створення сучасного програмного забезпечення на мові програмування TypeScript. Перевагою написання вебзастосунку за допомогою TypeScript є використання статичної типізації, що спрощує процес налагодження застосунку на етапі розробки. Також, у Angular передбачена система *dependency injection*, застосування якої дозволяє організовувати програмний код у вигляді модулів. Даний фреймворк також містить стратегії оптимізації роботи вебзастосунку, такі як *Ahead-of-time Compilation*. Впровадження такої стратегії дозволяє зменшити розміри застосунку після компіляції. Angular регулярно оновлюється, задля забезпечення стабільності роботи та сумісності із сучасними технологіями.

Іншим фреймворком для створення вебзастосунків є Flutter. Однією з його переваг є кросплатформність. Застосунок, створений за допомогою Flutter можна скомпілювати для використання як вебзастосунок, або встановлюваний застосунок для таких платформ як Android, iOS, Windows та Linux. Також, суттєвою перевагою використання Flutter є значна кількість вбудованих елементів інтерфейсу користувача, а також наявність вбудованих анімацій для компонентів інтерфейсу.

Виконавши аналіз доступних засобів для розробки, для створення користувацької частини вебзастосунку можна окреслити основні переваги застосування React. Його використання дозволяє створювати програмний продукт що задовольнятиме сучасні потреби у швидкодії застосунку, завдяки використанню модульної архітектури та механізму Virtual DOM. Завдяки створенню незалежних компонентів з ізольованою логікою, можна буде пришвидшити та спростити процес розробки програмного забезпечення. Вебзастосунок, створений із використанням React може бути розміщений на спеціалізованих системах розповсюдження статичних файлів, що сприятиме підвищенню часу завантаження вебзастосунку з боку кінцевих користувачів.

Зазначені вище переваги підкреслюють високу придатність React для створення сучасних вебзастосунків. Завдяки цьому, задля створення користувацького інтерфейсу програмного продукту “Scheme chef” буде обрано бібліотеку React.

2.5. Висновки до розділу

Проведення аналізу засобів створення програмного забезпечення є важливою складовою розробки.

Було обрано СКБД Postgres, як таку, що найкраще задовольняє вимоги щодо гнучкості типів даних. Також, проведений аналіз доступних засобів управління даними виявив що Postgres надає більш широкі можливості для оптимізації операцій отримання даних. Окремим вагомим чинником що сприяв вибору цієї СКБД була наявність засобів забезпечення цілісності даних, які доступні для будь-яких рушіїв Postgres.

Розроблюване ПЗ використовуватиме клієнт-серверну інфраструктуру. При цьому, завдяки використанню даної архітектури, користувацький інтерфейс зможе бути частково працездатним у разі недоступності серверної частини. Серверна частина вебзастосунку

здійснюватиме обробку користувацьких запитів, включно з авторизацією користувачів, та взаємодіятиме із базою даних.

Обрана мова програмування Go може застосовуватись для створення серверної частини вебзастосунків та забезпечувати високий рівень швидкодії завдяки багатопотоковій обробці запитів. Програмне забезпечення, створене за допомогою мови програмування Go, пристосоване для застосування у сучасній архітектурі ПЗ, яка передбачає контейнеризацію.

Для створення сучасної користувацької частини вебзастосунку було передбачено використання бібліотеки React, яка завдяки своїй архітектурі, здатна забезпечити високу швидкодію програмного продукту. Також, використання React дозволяє використовувати сучасні стратегії розміщення вебзастосунків, що позитивною мірою вплине на швидкість завантаження інтерфейсу користувача.

3. РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ

3.1. Загальний опис структури застосунку “Scheme chef”

Задля задоволення основних потреб користувачів, розроблюваний вебзастосунок має задовольняти наступні вимоги:

- 1) Наявність шаблону для створення ціннісної пропозиції.
- 2) Можливість автоматично генерувати елементи ціннісної пропозиції, відповідно до теми діаграми.
- 3) Можливість управління доступом до діаграми ціннісної пропозиції.
- 4) Можливість групувати діаграми ціннісної пропозиції у теки.
- 5) Можливість поширювати діаграму за допомогою посилання.

Для задоволення вказаних вимог, та задля забезпечення працездатності застосунку, необхідно створити перелік вебсторінок, що містять вказані у табл. 3.1 елементи:

Таблиця 3.1

Структура сторінок клієнтської частини вебзастосунку

Номер сторінки	Назва сторінки	Розділ сторінки	Призначення структурного елементу
1	Головна інформаційна сторінка	Верхня панель сторінки	Відображення емблеми застосунку, навігація між інформаційними сторінками застосунку, перенаправлення на сторінку авторизації
		Центральний простір інформаційної сторінки	Надання інформації щодо призначення застосунку, відображення основних переваг застосунку, опис властивостей застосунку
		Нижня інформаційна панель	Навігація між інформаційними сторінками застосунку, стисла інформація щодо розробника застосунку

Продовження табл. 3.1

2	Інформаційна сторінка щодо розробника застосунку	Верхня панель сторінки	Відображення емблеми застосунку, навігація між інформаційними сторінками застосунку, перенаправлення на сторінку авторизації
		Центральний простір інформаційної сторінки	Інформація щодо розробника, посилання на сторонні інформаційні ресурси щодо розробника, контакти для запитів, відгуків та скарг щодо застосунку
		Нижня інформаційна панель	Навігація між інформаційними сторінками застосунку, стисла інформація щодо розробника застосунку
3	Сторінка реєстрації	Верхня панель сторінки	Повернення до головної сторінки, відображення назви застосунку
		Панель вводу інформації для реєстрації	Отримання інформації користувача для реєстрації, надання інструкцій щодо реєстрації, повернення до головної сторінки у разі успішної реєстрації
		Нижня панель сторінки	Стисла інформація щодо вебзастосунку
4	Сторінка авторизації	Верхня панель сторінки	Повернення до головної сторінки, відображення назви та емблеми застосунку, перехід до інших сторінок вебзастосунку
		Панель вводу інформації для авторизації	Отримання інформації користувача для авторизації, надання інструкцій щодо авторизації, повернення до головної сторінки у разі успішної авторизації

Продовження табл. 3.1

4	Сторінка авторизації	Нижня панель сторінки	Стисла інформація щодо вебзастосунку
5	Сторінка перегляду колекцій діаграм	Верхня панель сторінки	Відображення емблеми застосунку, навігація між сторінками застосунку, вихід із сесії застосунку
		Центральний простір сторінки	Перегляд переліку колекцій діаграм ціннісної пропозиції, створення нових колекцій діаграм
		Нижня панель сторінки	Стисла інформація щодо вебзастосунку
6	Сторінка перегляду діаграм у колекції	Верхня панель сторінки	Відображення емблеми застосунку, навігація між сторінками застосунку, перехід до вибору колекції діаграм, вихід із сесії застосунку
		Центральний простір сторінки	Перегляд переліку діаграм ціннісної пропозиції у обраній колекції, Створення нових діаграм, створення доступу за посиланням
		Нижня панель сторінки	Стисла інформація щодо вебзастосунку, навігація між сторінками вебзастосунку
7	Сторінка редагування діаграми ціннісної пропозиції	Верхня панель сторінки	Відображення емблеми застосунку, навігація між сторінками застосунку, у відповідності до наявності активної користувацької сесії, вихід із сесії застосунку

7	Сторінка редагування діаграми ціннісної пропозиції	Верхня панель простору редагування	Керування доступом до діаграми, створення доступу за посиланням, керування розміщенням діаграми у колекціях, обрання типу елемента для розміщення на діаграмі, налаштування рушія автоматичної генерації елементів ціннісної пропозиції
		Права інформаційна панель сторінки	Відображення інформації про автора діаграми, відображення та редагування опису діаграми, відображення та редагування тегів діаграми
		Центральна панель	Перегляд елементів ціннісної пропозиції, переміщення елементів ціннісної пропозиції, редагування елементів ціннісної пропозиції, додання тегів для елементів ціннісної пропозиції, отримання згенерованих елементів ціннісної пропозиції
		Нижня панель сторінки	Стисла інформація щодо вебзастосунку
8	Сторінка керування інформацією користувача	Верхня панель сторінки	Відображення емблеми застосунку, навігація між сторінками застосунку, вихід із сесії застосунку
		Центральна панель	Відображення та редагування інформації користувача, відображення та редагування доступу до персональних даних
		Нижня панель сторінки	Стисла інформація щодо вебзастосунку

На рис. 3.1 зображено основний сценарій використання – створення елементів ціннісної пропозиції. Наведений порядок дій передбачає можливість автоматичної генерації елементів ціннісної пропозиції. Також передбачена можливість створювати елементи діаграми вручну.

На початку сценарію використання, користувач обирає режим взаємодії із діаграмою ціннісної пропозиції. Вибір режиму взаємодії здійснюється за допомогою випадного списку.

У разі обрання режиму додання елементів діаграми, користувач натискає на сектор діаграми, до якого бажає додати елемент. Кожна з двох секцій діаграми містить три сектори. Подальші дії визначаються станом перемикача автоматичної генерації ціннісної пропозиції. За замовчуванням автоматична генерація елементів ціннісної пропозиції не застосовується. Режим автоматичної генерації елементів діаграми можна застосувати, натиснувши на перемикач.

При ввімкненій автоматичній генерації – елемент буде створено без запитів на ввід даних зі сторони користувача. Якщо ж було обрано режим введення даних вручну – буде відображене спливаюче вікно із формою для вводу інформації.

У режимі додавання елементів ціннісної пропозиції можна додавати теги до створених компонентів діаграми. Для цього необхідно натиснути на елемент діаграми правою кнопкою миші. Користувачу буде відображено діалогове вікно із формою редагування тегів елементу діаграми. Додані теги діаграми будуть збережені після закриття діалогового вікна.

Для зміни позиції створених елементів діаграми можна обрати режим переміщення елементів діаграми ціннісної пропозиції. У даному режимі спливаюче вікно вводу інформації не буде відображатися при натисканні на простір сектору діаграми. Компонент діаграми може бути переміщений лише у межах сектору діаграми, у якому даний елемент був створений.

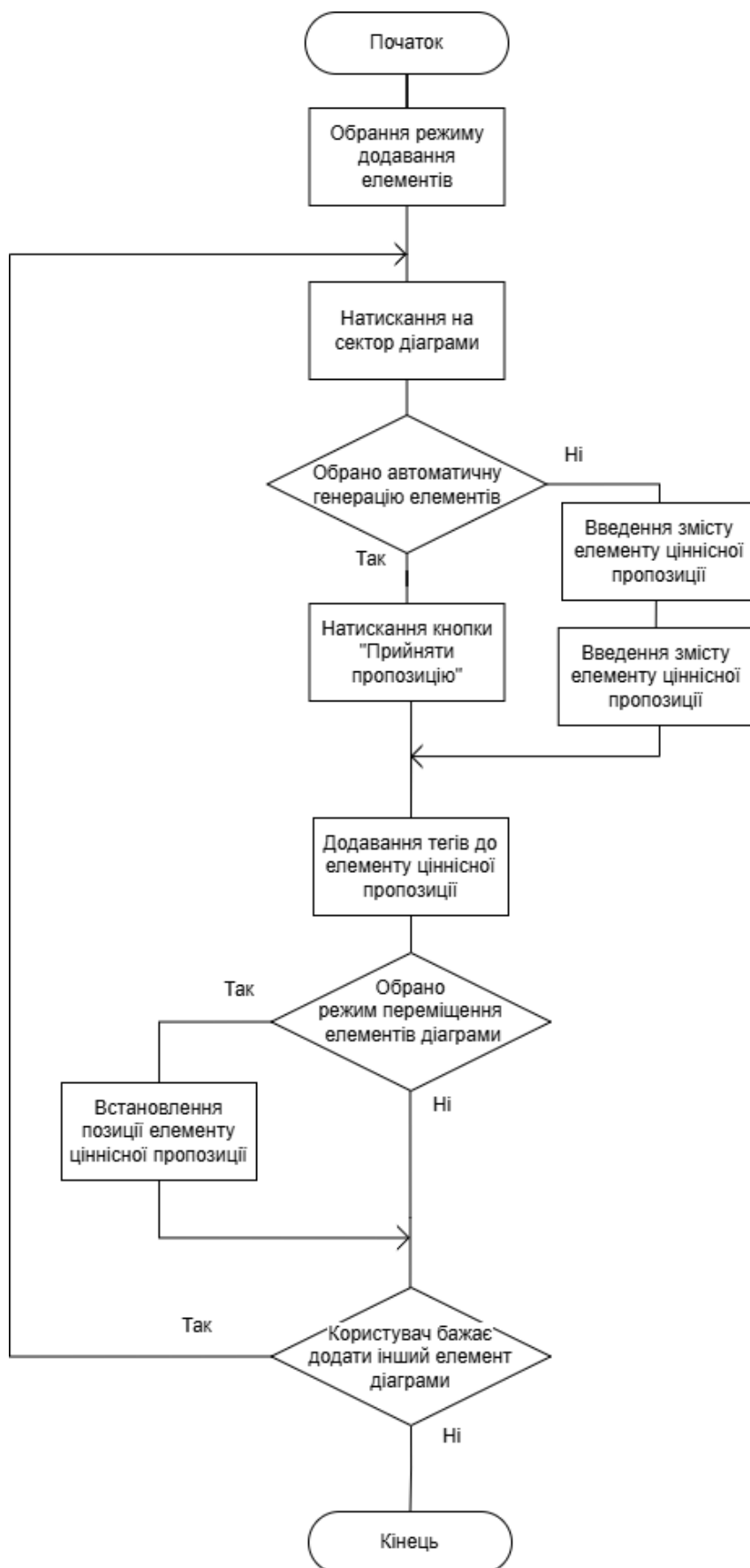


Рис. 3.1. Діаграма діяльності основного сценарію використання

Зареєстровані користувачі можуть виконувати такі дії (рис. 3.2):

- авторизація у вебзастосунок та поновлення сесії користувача;
- створення діаграм ціннісної пропозиції;
- перегляд, редагування та видалення власних діаграм ціннісної пропозиції;
- перегляд публічних діаграм ціннісної пропозиції за посиланням;
- перегляд та редагування діаграм ціннісної пропозиції за посиланням;
- автоматична генерація ціннісної пропозиції;
- додавання тегів до діаграми та до елементів ціннісної пропозиції;
- створення посилань для поширення ціннісної пропозиції;
- отримання підказок щодо елементів ціннісної пропозиції;
- перегляд інформації щодо автора ціннісної пропозиції.



Рис. 3.2. Дії користувача вебзастосунку «Scheme chef»

3.2. Архітектура вебзастосунку «Scheme chef»

Вебзастосунок спроектований на основі розподіленої клієнт-серверної архітектури, що розділяє серверну частину та користувацьку

частину вебзастосунку. Завдяки цьому, клієнтська частина є частково працездатною у разі недоступності серверної частини. Взаємодія клієнтської та серверної частин вебзастосунку відбувається за допомогою API (рис. 3.3).

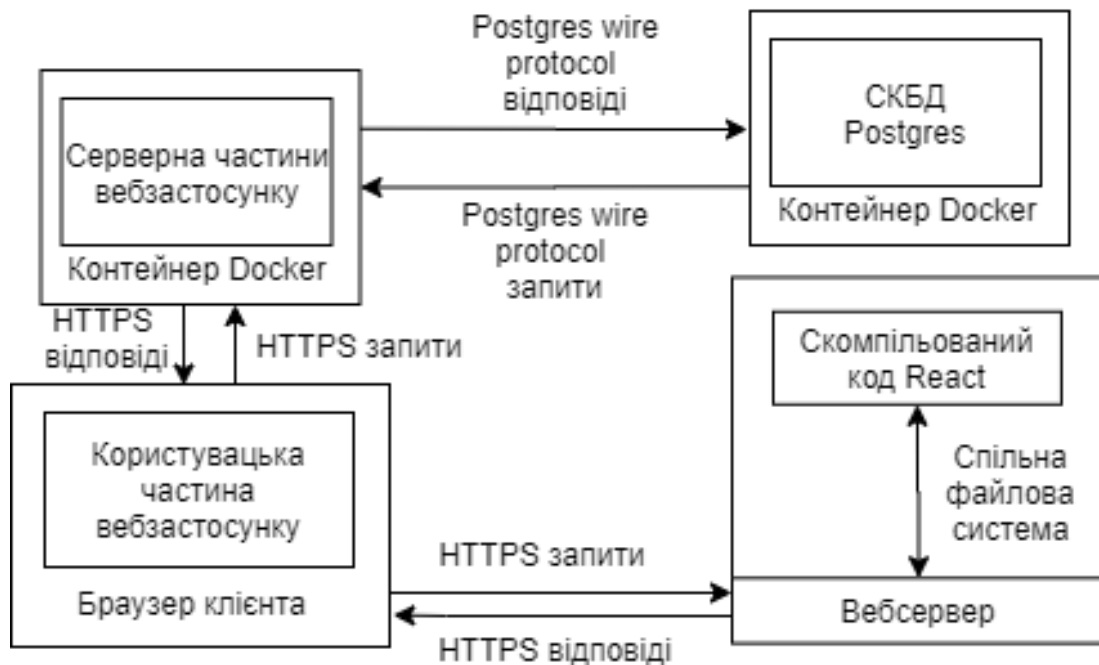


Рис. 3.3. Архітектура вебзастосунку «Scheme chef»

Користувацька частина вебзастосунку – це набір сторінок користувацького інтерфейсу. Його використання має забезпечувати можливість надсилання запитів до серверної частини застосунку. Клієнтська частина вебзастосунку надає засоби для відображення та редагування ціннісних пропозицій та управління ними.

Структура клієнтської частини вебзастосунку передбачає розподілення на рівні файлової системи сторінок, модулів та інших ресурсів:

- components – директорія що містить складові частини користувацького інтерфейсу. Компоненти у даній директорії можуть бути розміщені у окремих вкладених директоріях, відповідно до їх призначення;

- `structure` – директорія, що містить дані, які визначають перелік компонентів вебзастосунку для відображення. За допомогою компонентів у даній директорії авторизованим та не авторизованим може буде відображено різний вміст сторінок;
- `pages` – директорія, що містить сторінки вебзастосунку;
- `public` – директорія, яка містить допоміжні ресурси. Такими можуть бути зображення, що відображаються на сторінках вебзастосунку;
- `app.js` – забезпечує маршрутизацію запитів користувача до сторінок застосунку;
- `index.js` – відповідає за ініціалізацію користувацької частини вебзастосунку.

Серверна частина вебзастосунку працює у контейнері Docker. Такий підхід дозволяє використовувати різні методи розміщення серверної частини застосунку. Серверна частина забезпечує:

- авторизацію користувачів та створення облікових записів;
- перевірку дійсності користувацьких сесій;
- зв'язок із базою даних задля отримання запитаної інформації;
- перевірку рівня доступу користувача до ресурсів;
- надання запитаної інформації користувачам.

Серверна частина складається з окремих модулів. Частина з них забезпечує виконання функціональних вимог застосунку. Інша частина відповідає за зв'язок із зовнішніми сутностями. Також, були створені модулі, що збирають моніторингові дані стосовно роботи серверної частини вебзастосунку.

Взаємодія серверної частини вебзастосунку відбувається на основі `Postgres wire protocol`. Він забезпечує обробку складних запитів, ефективність передачі даних та авторизацію для здійснення запитів [13]. Даний протокол також надає можливість використання SQL транзакцій. Серверна частина вебзастосунку використовує пакет `pq` разом із бібліотекою `database/sql` для застосування зазначеного вище протоколу.

Клієнт взаємодіє із користувацькою та серверною частинами за допомогою протоколу HTTPS. Використання даного протоколу забезпечує захищеність даних під час їх транспортування мережею. Також, щодо взаємодії клієнта із вебзастосунком застосовано same-origin policy та CORS policy. Вони дозволяють обмежувати доступ до ресурсів вебзастосунку на основі характеристик запиту. До них відноситься доменне ім'я, HTTP заголовки запиту та метод HTTP.

Вебзастосунок “Scheme chef” використовує архітектуру, засновану на таких технологіях, як React, JavaScript, Go та PostgreSQL. Клієнтська частина вебзастосунку розроблена на мові програмування JavaScript, із використанням бібліотек React. Мова програмування Go використовується для створення API, що використовується користувацькою частиною вебзастосунку. Використання бази даних PostgreSQL дозволяє зберігати дані вебзастосунку та керувати ними. Обробка та управління користувацькими даними, збереженими у СКБД PostgreSQL, здійснюється серверною частиною вебзастосунку.

3.3. База даних вебзастосунку «Scheme chef»

Для зберігання даних, що мають чітку структуру та велику кількість зв'язків між сутностями, використовується реляційна база даних Postgres. Інформація розміщена у таблицях: users (табл. 3.2), boards (табл. 3.3), access (табл. 3.4), cards (табл. 3.5), card_to_board (табл. 3.6), tags (табл. 3.7), board_to_tag (табл. 3.8), card_to_tag (табл. 3.9), folders (табл. 3.10), folder_to_user (табл. 3.11), folder_to_board (табл. 3.12).

Інформація користувача зберігається у таблиці users. Вона містить дані для ідентифікації користувача у інформаційній системі, а також облікові дані користувача, такі як електронна пошта, ім'я та фамілія. Таблиця також зберігає ідентифікатор публічності профілю. Дані щодо діаграм ціннісної пропозиції користувача знаходяться у окремій таблиці boards.

Таблиця 3.2

Опис таблиці “users”

Назва поля даних	Тип даних	Призначення
id	Serial (INT)	Унікальний ідентифікатор елемента у базі даних
uid	UUID	Унікальний ідентифікатор для доступу за посиланням
is_public	Boolean	Визначення доступності профіля до перегляду
firstname	Varchar(32)	Ім'я користувача
secondname	Varchar(32)	Фамілія користувача
username	Varchar(64)	Логін користувача
email	Varchar(100)	Електронна пошта користувача
password_hash	Varchar(98)	Хеш паролю користувача

Таблиця 3.3

Опис таблиці “boards”

Назва поля даних	Тип даних	Призначення
id	Serial (INT)	Унікальний ідентифікатор елемента у базі даних
name	Varchar(64)	Назва діаграми
description	Varchar(512)	Опис діаграми

Слід звернути увагу на обраний підхід збереження інформації щодо керування доступом до діаграм. Таблиця access визначає рівень доступу до діаграми, а також – перелік ідентифікаторів користувачів, що можуть застосовувати його. Кілька елементів таблиці access можуть відповідати елементу таблиці boards. При спробі користувача отримати доступ до

діаграми ціннісної пропозиції, за uid з таблиці access буде перевірено, чи наявний ідентифікатор користувача у полі target_ids.

Таблиця 3.4

Опис таблиці “access”

Назва поля даних	Тип даних	Призначення
access_uid	UUID	Унікальний ідентифікатор для доступу до діаграми за посиланням
access_level	INT	Рівень доступу до діаграми на перегляд та модифікацію
board_id	INT	Ідентифікатор діаграми
target_ids	Integer[]	Масив ідентифікаторів користувачів, що можуть використовувати посилання для доступу

Елементи таблиці cards додаються до діаграм ціннісної пропозиції за допомогою допоміжної таблиці асоціації card_to_board. При цьому, одна картка може бути додана до багатьох діаграм. Через це, дані щодо позиції картки на діаграмі зберігаються у таблиці асоціації.

Таблиця 3.5

Опис таблиці “cards”

Назва поля даних	Тип даних	Призначення
id	Serial (INT)	Унікальний ідентифікатор елементу у базі даних
uid	UUID	Ідентифікатор картки для доступу за кодом UUID
card_type	INT	Тип картки ціннісної пропозиції

Продовження табл. 3.5

card_text	Varchar(128)	Текст картки ціннісної пропозиції
-----------	--------------	-----------------------------------

Таблиця 3.6

Опис таблиці “card_to_board”

Назва поля даних	Тип даних	Призначення
card_id	INT	Ідентифікатор картки, доданої до діаграми
board_id	INT	Ідентифікатор діаграми
x	INT	Позиція X картки на діаграмі
y	INT	Позиція Y картки на діаграмі
z	INT	Позиція Z картки на діаграмі
endorsements	Integer[]	Масив ідентифікаторів користувачів, що відмітили відповідність картки до діаграми

Інформація щодо тегів розподілена схожим чином – назви тегів зберігаються у таблиці tags, теги асоціюються із картками за допомогою допоміжної таблиці card_to_board. Теги також можуть бути додані безпосередньо до діаграми, за допомогою допоміжної таблиці board_to_tag, яка поєднує тег із відповідною діаграмою.

Таблиці асоціації тегів містять масив ідентифікаторів користувачів, які відмітили відповідність тегу до діаграми. Ці дані використовуються задля того, щоб користувач міг відмітити відповідність тегу до діаграми не більше одного разу.

Таблиця 3.7

Опис таблиці “tags”

Назва поля даних	Тип даних	Призначення
id	Serial (INT)	Унікальний ідентифікатор елемента у базі даних
uid	UUID	Ідентифікатор для доступу за UUID
tag_text	Varchar(64)	Текст тегу

Таблиця 3.8

Опис таблиці “board_to_tag”

Назва поля даних	Тип даних	Призначення
board_id	INT	Ідентифікатор діаграми
tag_id	INT	Ідентифікатор тегу
endorsements	Integer[]	Масив ідентифікаторів користувачів, що відмітили відповідність тегу до діаграми

Таблиця 3.9

Опис таблиці “card_to_tag”

Назва поля даних	Тип даних	Призначення
card_id	INT	Ідентифікатор картки
tag_id	INT	Ідентифікатор тегу
endorsements	Integer[]	Масив ідентифікаторів користувачів, що відмітили відповідність тегу до картки

Розміщення діаграми ціннісної пропозиції у течії відбувається за допомогою таблиці `folders`, що містить дані щодо теки, та таблиць асоціації `folder_to_user` та `folder_to_board`.

Таблиця 3.10

Опис таблиці “folders”

Назва поля даних	Тип даних	Призначення
id	Serial (INT)	Унікальний ідентифікатор елемента у базі даних
uid	UUID	Ідентифікатор папки для доступу за посиланням
name	Varchar(64)	Назва теки із діаграмами
description	Varchar(512)	Опис теки із діаграмами

Таблиця 3.11

Опис таблиці “folder_to_user”

Назва поля даних	Тип даних	Призначення
folder_id	INT	Ідентифікатор теки
user_id	Int	Ідентифікатор користувача
access_uid	UUID	Ідентифікатор зв'язку

Таблиця 3.12

Опис таблиці “folder_to_board”

Назва поля даних	Тип даних	Призначення
folder_id	INT	Ідентифікатор теки
board_id	INT	Ідентифікатор діаграми

Структура бази даних та зв'язки між її сутностями зображені на схемі бази даних вебзастосунку (рис. 3.4). Відображено типи зв'язків між сутностями бази даних. Серед таких зв'язків – один до одного, один до багатьох або без зв'язків, один до багатьох та щонайменше один зв'язок. Вказана структура бази даних забезпечує відповідність роботи із даними інформаційної системи.

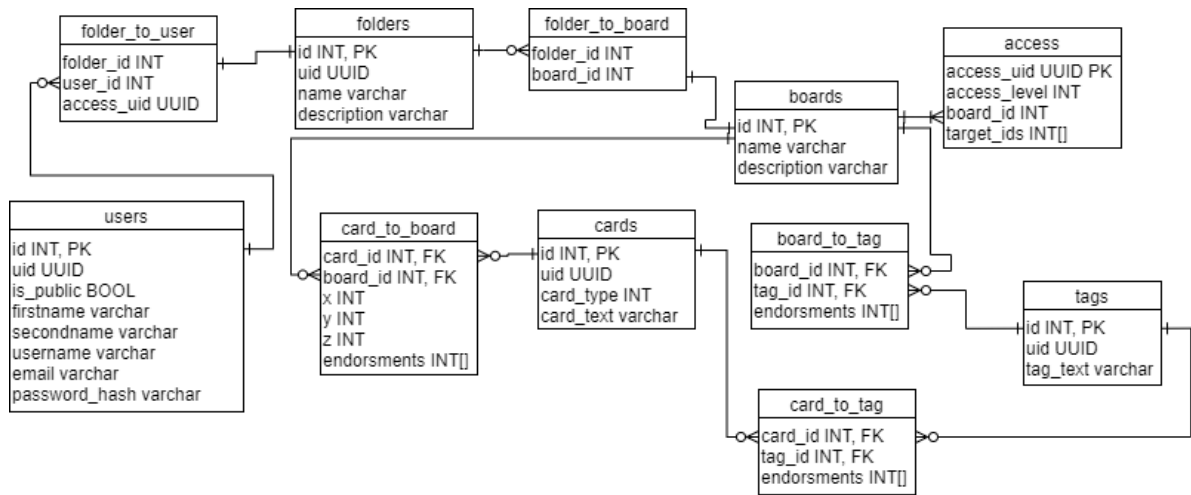


Рис. 3.4. Схема бази даних вебзастосунку

3.4. Висновки до розділу

Вебзастосунок «Scheme chef» спроектовано на основі розподіленої архітектури, що розділяє серверну частину та клієнтську частину на окремі компоненти, та збільшує стійкість вебзастосунку до недоступності окремих модулів інформаційної системи.

Користувачка частина забезпечує візуальний інтерфейс, що надає засоби для взаємодії із серверною частиною вебзастосунку. Основними сторінками вебзастосунку є сторінки, що забезпечують управління діаграмами ціннісної пропозиції. Сторінка редагування ціннісної пропозиції надає можливості для редагування вмісту діаграми, встановлення розташування елементів діаграми. Також, ця сторінка дозволяє застосовувати автоматичну генерацію елементів ціннісної пропозиції. Сторінки керування теками діаграм та керування діаграмами всередині теки забезпечують навігацію між діаграмами. На сторінці управління діаграмами теки також можна налаштувати доступ до діаграми за посиланням.

Серверна частина вебзастосунку забезпечує виконання операцій над даними користувача та над діаграмами ціннісної пропозиції. Авторизація та реєстрація користувачів, разом із управлінням користувацькими сесіями теж забезпечується серверною частиною вебзастосунку “Scheme chef”. Серверна частина також забезпечує додавання заголовків HTTP запитів, що

необхідні для застосування налаштувань безпеки. Такими налаштуваннями є same-origin policy та CORS policy. Серверна частина взаємодіє із базою даних для управління інформацією користувачів вебзастосунку. Для забезпечення ефективного передавання даних, використовується Postgres wire protocol. Він також надає можливість застосовувати транзакції SQL для забезпечення цілісності даних, при виконанні послідовності запитів до бази даних.

Вебзастосунок використовує базу даних PostgreSQL для збереження даних інформаційної системи та управління ними. Зазначена СКБД підтримує збереження типів даних, таких як масиви, необхідні для роботи окремих частин функціональності вебзастосунку. Ця база даних також підтримує широкий набір методів для покращення ефективності роботи із даними. Вказана СКБД підтримується найбільшими надавачами хмарних послуг, та може бути розгорнута у контейнері Docker. Описані властивості надають широкий набір опцій для розгортання інформаційної системи, та спрощують процес підтримки її працездатності

4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБЗАСТОСУНКУ

4.1. Опис інтерфейсу користувача

На сторінці створення та редагування діаграми (рис. 4.1) верхня панель містить елементи навігації. Використовуючи їх, користувач може потрапити на сторінки управління теками, на головну інформаційну сторінку та на сторінку управління персональними даними. Нижче, у вікні реалізовано простір управління діаграмою ціннісної пропозиції. У панелі керування засобами взаємодії можна обрати режим додавання, переміщення, редагування або видалення елементів діаграми. Верхня панель керування режимами роботи застосунку також містить перемикач режиму автоматичної генерації елементів ціннісної пропозиції.

Центральний простір взаємодії із діаграмою забезпечує обробку користувацьких дій при натисненні на діаграму, або при переміщенні її елементів. Центральний простір взаємодії із діаграмою містить два компоненти ціннісної пропозиції. Кожен із таких компонентів розподілений на 3 секції. Елементи діаграми не можуть бути переміщені за межі секції, у яких їх створили.

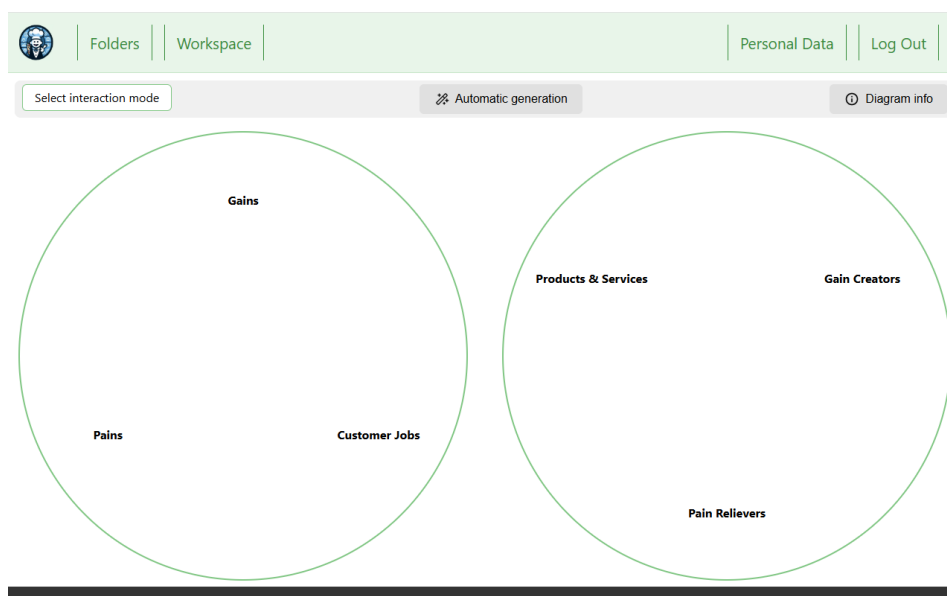


Рис. 4.1. Сторінка створення та редагування діаграми

Лише авторизовані користувачі можуть переглядати сторінки, що потребують взаємодії із серверною частиною вебзастосунку. Не авторизовані користувачі отримають повідомлення про необхідність увійти до облікового запису.

Для отримання користувацьких даних у режимі додавання або редагування елементів ціннісної пропозиції використовується спливаюче вікно (рис. 4.2). Форма користувацького вводу передається до спливаючого вікна як вкладений елемент. Вказана форма містить елементи для введення тексту компонента, а також для додавання та видалення тегів елементу. При використанні даної форми у режимі редагування, Внесені зміни автоматично надсилаються до серверної частини вебзастосунку.

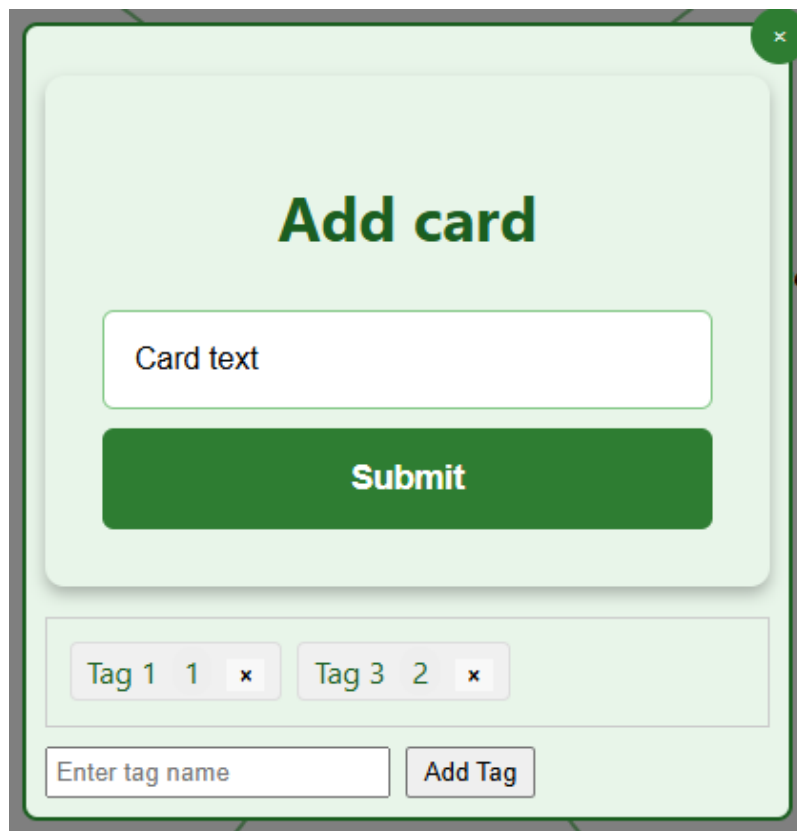


Рис. 4.2. Спливне вікно додавання елементу ціннісної пропозиції

Сторінка управління теками діаграм відображує перелік доступних директорій із діаграмами. Для кожної теки відображається її назва, опис та дата створення. При натисканні на інформаційну картку діаграми

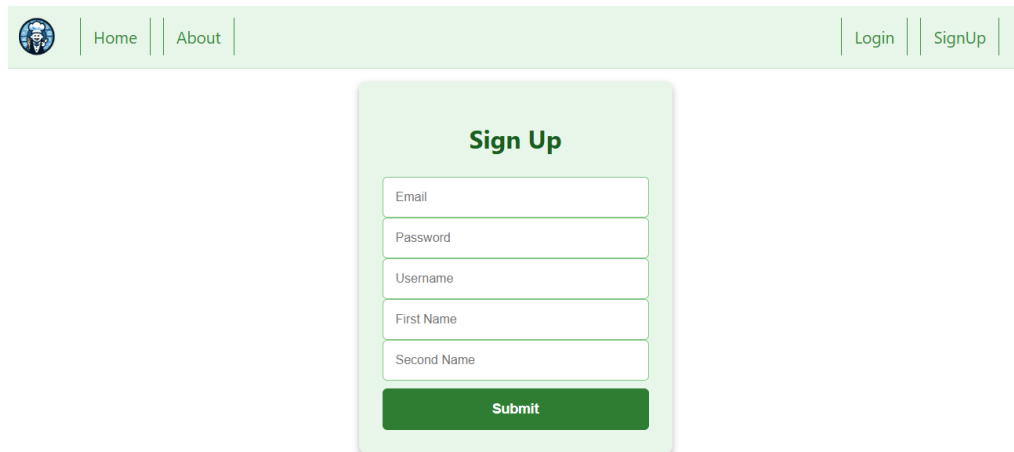
користувач буде направлений на сторінку управління діаграми у теці. Інформаційні картки також містять виклик елементу управління текою. За допомогою нього можна редагувати назву та опис діаграми. На сторінці також розміщено засіб для створення нових тек для діаграм ціннісної пропозиції.

Сторінка управління діаграмами ціннісної пропозиції певної теки надає можливості для перегляду та редагування інформації щодо діаграм. Користувач може переглядати та редагувати назву, опис та теги діаграми. Також, на сторінці передбачено засіб для створення нових діаграм ціннісної пропозиції. Натиснувши на кнопку отримання довідки, користувач може переглянути інформацію щодо автора діаграми. Інформація щодо автора буде відображена у відповідності до налаштувань показу особистої інформації користувача. Користувач також може створювати доступ за посиланням, використовуючи дану сторінку.

При додаванні тегу до діаграми ціннісної пропозиції, перевіряється наявність такого тегу на діаграмі. Якщо тег не було додано раніше – він буде закріплений за діаграмою. У випадку коли тег був доданий іншим користувачем – буде збільшено показник відповідності тегу до діаграми. Ступінь відповідності відображається на елементі тегу. Він являє показує кількість користувачів що відмітили даний тег на діаграмі.

Сторінка реєстрації (рис. 4.3) містить поля для вводу даних користувача. При натисканні на кнопку створення облікового засобу буде перевірено вміст запитаних елементів вводу інформації. Якщо користувач ввів не всю запитану інформацію, або у випадку коли формат дані введено у невірному форматі, користувачу буде відображено помилку. У разі коректного введення даних, інформація буде надіслана до серверної частини вебзастосунку. Якщо користувача із введеною електронною поштою та логіном не містяться у системі, буде створено обліковий запис. У цьому випадку користувача буде направлено на сторінку перегляду тек із діаграмами. Верхня панель із посиланнями для навігації у застосунку

відображає різний набір посилань для авторизованих та не авторизованих користувачів.



The image shows a web application interface. At the top, there is a light green navigation bar. On the left, there is a circular logo with a person's face. To the right of the logo are links for 'Home' and 'About'. On the far right of the navigation bar are links for 'Login' and 'SignUp'. Below the navigation bar, centered on the page, is a 'Sign Up' form. The form has a light green background and a dark green border. It contains five input fields: 'Email', 'Password', 'Username', 'First Name', and 'Second Name'. Below these fields is a green button labeled 'Submit'.

Рис. 4.3. Форма створення облікового запису

4.2. Тестування вебзастосунку

Задля забезпечення якості розробленого програмного продукту, на всіх етапах розробки застосовувалось тестування розроблюваного програмного забезпечення. Тестування вебзастосунку перевірялося у два етапи. Першим етапом перевірялося відповідність роботи окремих компонентів програмного продукту. Коли для окремих компонентів не було виявлено несправностей – використовувалось ручне системне тестування.

Для перевірки роботи окремих компонентів інформаційної системи застосовувалось модульне тестування. На цьому етапі тестування окремі компоненти вебзастосунку тестувалися без урахування їх зв'язку із іншими модулями. Даний вид тестування дозволив контролювати якість розроблюваного ПЗ на етапі модифікації та покращення існуючих модулів. Виявлення невідповідної роботи інформаційної системи на етапі розробки її модулів сприяло своєчасному виправленню несправностей. Виконання юніт-тестування значно швидше ручної перевірки взаємодії компонентів

системи. Воно також забезпечує застосування сталого набору визначених сценаріїв тестування.

Ручне системне тестування проводилося із метою виявити потенційні проблеми при взаємодії між складниками системи. Воно застосовувалось у разі успішного проведення юніт-тестування. Таким чином, звужувалась область пошуку проблем та зменшувався час, необхідний для усунення несправностей.

Системне тестування включало перевірку взаємодій користувацької частини із серверною частиною вебзастосунку. Це дозволило виявити та усунути проблеми взаємодії вказаних систем, а також допоміжних модулів, що задіяні у системі. Прикладом виявлених проблем у взаємодії компонентів інформаційної системи була відсутність налаштованого HTTPS, невірна конфігурація CORS policy. Завдяки застосуванню системного тестування дані недоліки були вчасно усунені. Також, системне тестування включало перевірку взаємодії модулів серверної частини вебзастосунку із базою даних. Завдяки цьому, було виявлено та полагоджено помилки невідповідності типів даних, що надсилалися до таблиць.

Перелік сценаріїв тестування, що досліджувалось під час роботи над створенням вебзастосунку “Scheme chef” наведено у табл. 4.1.

Таблиця 4.1

Перелік сценаріїв тестування вебзастосунку “Scheme chef”

№	Опис тесту	Дія користувача	Очікуваний результат
1	Перевірка доступності застосунку за допомогою захищеного з'єднання	Користувач відкриває сторінку вебзастосунку в браузері	Вміст інформаційної сторінки відображається, підключення відбувається за допомогою HTTPS

Продовження табл. 4.1

2	Перевірка відображення помилки у разі введення не повних даних для реєстрації	Користувач не заповнює щонайменше одне поле для вводу даних	Відображається помилка щодо недостатності введених даних
3	Перевірка отримання ідентифікатора користувачької сесії після успішної реєстрації або авторизації	Користувач надсилає дані коректні для реєстрації або авторизації	Браузер користувача отримує та зберігає cookie файл із ідентифікатором сесії користувача
4	Перевірка відображення помилки при реєстрації якщо користувач із введеною електронною поштою або логіном вже існує	При реєстрації користувач вводить логін або електронну пошту що вже існує у базі даних	Користувач отримує помилку щодо того, що користувач із таким логіном чи електронною поштою вже існує
5	Перевірка відображення помилки при введення недійсних даних при авторизації	При авторизації користувач вводить дані яких немає у базі даних	Відображення помилки щодо даних авторизації
6	Перевірка отримання доступу до сторінок, що потребують авторизації, без активної сесії користувача	Користувач відкриває посилання на сторінку управління діаграм без попередньої авторизації	Користувач автоматично направлений на сторінку авторизації

7	Перевірка відображення переліку посилань у верхній панелі навігації при наявності активної сесії користувача	Користувач із активною сесією користувача відкриває будь-яку сторінку веб застосунку	Верхня панель навігації містить перелік посилань для користувачів із активною сесією
8	Перевірка відображення переліку посилань у верхній панелі навігації для не авторизованих користувачів	Не авторизований користувач відкриває будь-яку сторінку веб застосунку	Верхня панель навігації містить перелік посилань для не авторизованих користувачів
9	Перевірка автоматичного перенаправлення на сторінку управління теками діаграм після успішної авторизації	Користувач успішно авторизується у вебзастосунку	Користувач направлений на сторінку управління теками діаграм ціннісної пропозиції
10	Перевірка додавання нової теки діаграми ціннісної пропозиції	Користувач натискає на кнопку додавання нової теки, вводить назву та опис теки, натискає кнопку надсилання	Користувачу відображено діалогове вікно. Після успішного вводу даних – тека збережена у базі даних, користувач направлений на сторінку управління діаграмами у створеній теці
11	Перевірка редагування інформації теки ціннісної пропозиції	Користувач натискає на кнопку редагування на картці відповідної теки, редагує отриману інформацію та натискає кнопку надсилання	Відкривається діалогове вікно, із полями, заповненими на основі поточної інформації. Оновлена інформація зберігається у БД

Продовження табл. 4.1

12	Перевірка перенаправлення користувача до сторінки управління діаграмами ціннісної пропозиції у теці	Користувач натискає на інформаційну картку теки із діаграмами	Користувач направлений на сторінку управління діаграмами у обраній теці
13	Перевірка додавання нової ціннісної пропозиції	Користувач натискає на кнопку додавання ціннісної пропозиції, вводить її назву та опис та натискає кнопку надсилання	Відкривається діалогове вікно додавання ціннісної пропозиції. Після надсилання форми – створена діаграма додається до БД, користувач направляється
14	Перевірка редагування інформації ціннісної пропозиції із вікна управління діаграмами всередині теки	Користувач натискає на кнопку редагування інформації діаграми, змінює дані у діалоговому вікні та натискає кнопку надіслати	З'являється заповнене діалогове вікно, відповідно до поточної інформації у БД. Після успішного надсилання даних – інформація на сторінці оновлена.
15	Перевірка зміни тегів діаграми ціннісної пропозиції	У діалоговому вікні створення або редагування діаграми користувач вводить назву тегу та натискає кнопку додати	Якщо введений тег вже доданий до діаграми – збільшується показник відповідності тегу до діаграми
16	Перевірка створення доступу за посиланням для ціннісної пропозиції	Користувач натискає на кнопку керування доступом до діаграми, у діалоговому вікні обирає рівень доступу за посиланням	Користувач отримує спливаюче повідомлення із посиланням для доступу до діаграми

Продовження табл. 4.1

17	Перевірка перегляду ціннісної пропозиції за посиланням	Користувач переглядає діаграму ціннісної пропозиції за посиланням	Відкривається ціннісна пропозиція, відповідно до ідентифікатору доступу у посиланні
18	Перевірка додавання елементів у вікні редагування ціннісної пропозиції	У режимі додавання елементів діаграми користувач натискає на секцію діаграми ціннісної пропозиції, вводить текст елементу, додає теги у діалоговому вікні та натискає на кнопку надсилання	На місце натискання додається елемент ціннісної пропозиції із текстом, введеним у формі.
19	Перевірка переміщення елементів ціннісної пропозиції у вікні редагування	У режимі переміщення елементів діаграми користувач затискає ліву кнопку миші на елементі діаграми та переміщає елемент ціннісної пропозиції	Елемент ціннісної пропозиції можуть бути переміщені лише в межах секції діаграми, де він був створений
20	Перевірка застосування автоматичної генерації елементів ціннісної пропозиції	У режимі додавання нових елементів із ввімкненим режимом автоматичної генерації користувач натискає на сектор діаграми	До діаграми додано елемент, згенерований відповідно до наявної інформації про діаграму

4.3. Пропозиції для майбутнього покращення

Для розробленого програмного продукту існує декілька напрямів розширення його можливостей. Метою розвитку інформаційної системи має бути розширення можливостей для її застосування, поглиблення наявних визначних особливостей. Виявлені можливості включають:

1. Додати можливість інтеграції із системами командної взаємодії:
Надати можливість управляти діаграмами ціннісної пропозиції у вкладеному вікні, що може бути інтегровано із системами командної взаємодії.
2. Розширити доступні засоби генерації ціннісної пропозиції:
Розробити опції для автоматичної генерації ціннісної пропозиції що враховуватимуть контекст не лише створюваної діаграми, а й теки, у якій знаходиться діаграма. Поглибити можливості для дослідження області застосування діаграми ціннісної пропозиції при автоматичній генерації елементів.
3. Покращити засоби для управління проектами: Створити засоби для об'єднання користувачів у групи. Забезпечити управління доступом до діаграм ціннісної пропозиції у відповідності до приналежності користувача до певної групи.
4. Надати можливість спільної роботи над діаграмою у реальному часі: Створити функціональність що дозволить одночасну роботу кількох користувачів над однією діаграмою одночасно. Це сприятиме використанню вебзастосунку для командної роботи.
5. Реалізувати можливість авторизації за допомогою сторонніх сервісів: Додати опцію авторизації за допомогою найбільш відомих надавачів OIDC авторизації. Серед них – Google, Okta, Meta та інші.

Описані напрями для покращення інформаційної системи здатні забезпечити розвиток створеного програмного продукту. Реалізація вказаних пропозицій здатна забезпечити більш якісне задоволення потреб користувачів вебзастосунку.

4.4. Висновки до розділу

Було проаналізовано компоненти розробленого застосунку, що надають функціональність для управління діаграмами ціннісної пропозиції. Серед розроблених складових вебзастосунку можна виокремити групи

модулів, відповідно до їх призначення. Частина функціональності забезпечує базові потреби при роботі із вебзастосунком, такі як управління доступом до внутрішніх ресурсів. Інша частина розроблених компонентів інформаційної системи надають можливості для управління діаграмами ціннісної пропозиції. До них також належать засоби автоматизації створення діаграм ціннісної пропозиції.

На усіх етапах створення програмного продукту, відповідно до потреб кожної стадії розробки, застосовувались методи відповідні методи тестування. Для перевірки окремих модулів вебзастосунку використовувалось unit-тестування. Це гарантувало проходження модулем сталого набору тестів при кожній перевірці. Після успішного тестування компонентів, застосовувалось ручне системне тестування, задля перевірки правильності взаємодії між модулями системи, та її архітектурними компонентами. Застосовані заходи сприяли покращенню якості кінцевого продукту, завдяки своєчасній ідентифікації збоїв.

Аналіз можливостей для покращення інформаційної системи виявив достатній потенціал для розвитку проєкту. Наведені пропозиції здатні поліпшити процес взаємодії користувача із застосунком. Також, було досліджено можливості покращення вебзастосунку для розширення області його застосування.

ВИСНОВКИ

Метою даного дипломного проєкту є створення вебзастосунку, що надаватиме функціональність для створення ціннісних пропозицій та додавання нових якостей для вже створених ціннісних пропозицій за допомогою сучасних програмних технологій

При проектуванні програмного продукту було проведено аналіз наявних рішень, що можуть застосовуватись для управління ціннісними пропозиціями. На його основі було виявлено потреби користувачів що не задовольнялися наявними програмними засобами. Для ідентифікованих потреб користувача, були створені вимоги до програмного забезпечення.

Задля задоволення ключових вимог до вебзастосунку, було проведено огляд доступних засобів для створення інформаційних систем. Для зберігання даних та управління ними було обрано СКБД PostgreSQL. Мову програмування Go було обрано як засіб для створення серверної частини програмного продукту. А для створення користувацької частини вебзастосунку було обрано мову програмування JavaScript із застосуванням бібліотеки React.

За допомогою обраних технологій було розроблено вебзастосунок, що спроектований на основі розподіленої клієнт-серверної архітектури, що забезпечує збільшений рівень відмовостійкості системи. Сторінки вебзастосунку було створено на основі сформованої діаграми дій користувача, задля забезпечення відповідності створеного програмного продукту сценаріям його використання.

У результаті, було розроблено вебзастосунок “Scheme chef”, що надає можливості для: управління діаграмами ціннісної пропозиції та структурою їх розташування у теках, автоматизації створення діаграм ціннісної пропозиції та їх редагування. Вебзастосунок також дозволяє поширювати створені діаграми для перегляду за посиланням.

Розроблене програмне забезпечення також має потенціал для розширення можливостей його застосування. У ході виконання роботи були виокремлені пропозиції для вдосконалення вебзастосунку, що сприятимуть подальшому розвитку створеного продукту.

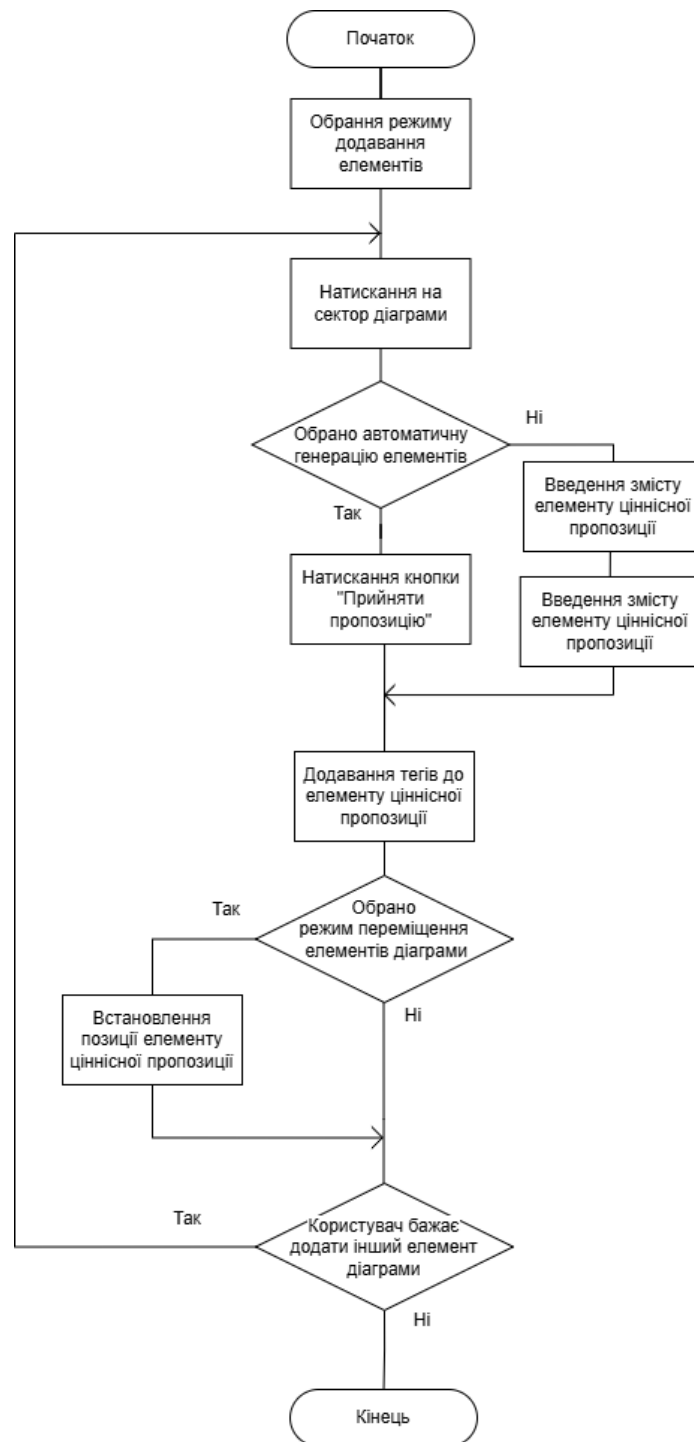
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. The best vs. the rest – knowledge hub 2.0. [Електронний ресурс]. – Режим доступу: <https://community.pdma.org/knowledgehub/bok/product-design-and-development-tools/best-vs-the-rest>. – Дата доступу: 25.12.2024.
2. The percentage of businesses that fail. [Електронний ресурс]. – Режим доступу: <https://luisazhou.com/blog/businesses-that-fail>. – Дата доступу: 28.12.2024.
3. Why over 80% of product launches fail. [Електронний ресурс]. – Режим доступу: <https://forabilis.com/why-over-80-of-product-launches-fail>. – Дата доступу: 29.12.2024.
4. The role of IT in marketing, and why it's important. [Електронний ресурс]. – Режим доступу: <https://deliveredsocial.com/the-role-of-it-in-marketing-and-why-its-important>. – Дата доступу: 29.12.2024.
5. Releases jgraph/drawio-desktop. [Електронний ресурс]. – Режим доступу: <https://github.com/jgraph/drawio-desktop/releases>. – Дата доступу: 03.01.2025.
6. Technical requirements. [Електронний ресурс]. – Режим доступу: <https://www.canva.com/help/technical-requirements>. – Дата доступу: 10.03.2025.
7. System reliability: when 99% is better than 100%. [Електронний ресурс]. – Режим доступу: <https://www.nearform.com/insights/system-reliability-when-99-is-better-than-100/>. – Дата доступу: 15.01.2025.
8. Go routines vs. Java threads: unleashing efficiency and simplicity. [Електронний ресурс]. – Режим доступу: <https://www.linkedin.com/pulse/go-routines-vs-java-threads-unleashing-efficiency-simplicity-khan/>. – Дата доступу: 23.01.2025.
9. Agarwal S. Why Golang's concurrency conquers over PHP. [Електронний ресурс]. – Режим доступу: <https://dev.to/thesaltree/why-golangs-concurrency-conquers-over-php-16mh>. – Дата доступу: 26.01.2025.

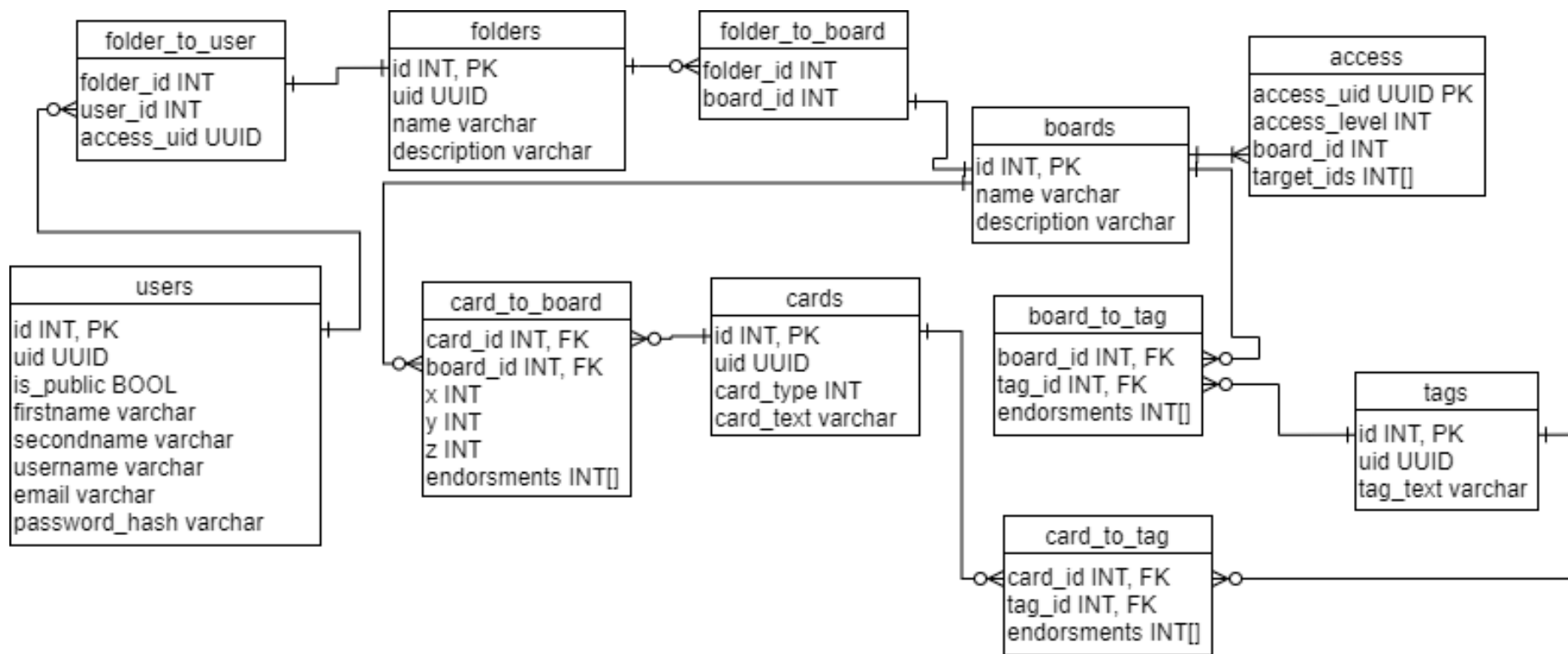
10. PostgreSQL vs MySQL – difference between relational database management systems. [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql>. – Дата доступа: 05.02.2025.
11. Password storage. [Электронный ресурс]. – Режим доступа: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html. – Дата доступа: 13.04.2025.
12. Virtual DOM and internals. [Электронный ресурс]. – Режим доступа: <https://legacy.reactjs.org/docs/faq-internals.html>. – Дата доступа: 26.05.2025.
13. Understanding postgres wire protocol. [Электронный ресурс]. – Режим доступа: <https://keploy.io/blog/technology/what-is-postgres-wire-protocol>. – Дата доступа: 29.04.2025.

ДОДАТКИ

Додаток 1
Копії графічних матеріалів



ДП.045440-06-99. Вебзастосунок
для управління діаграмами
ціннісної пропозиції. Діаграма
діяльності основного сценарію
використання. Схема алгоритму

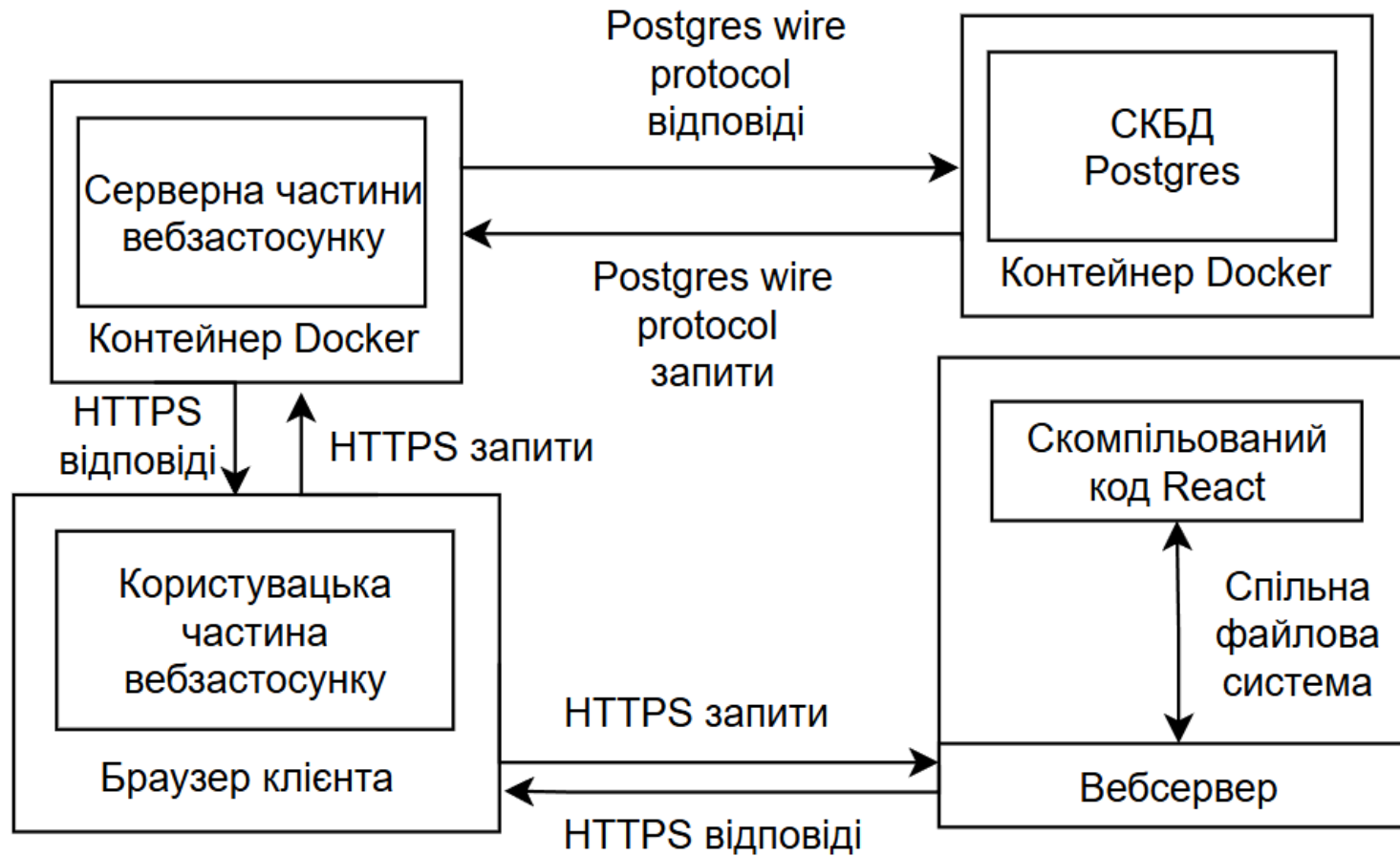


ДП.045440-07-99. Вебзастосунок для управління діаграмами ціннісної пропозиції. Схема бази даних вебзастосунку. ER-діаграма

Дії користувача вебзастосунку «Scheme chef»



Архітектура вебзастосунку



Додаток 2
Лістинг програми

Фрагменти коду ініціалізації користувацької частини App.js

```
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Home from './pages/home';
import LoginPage from './pages/login';
import UserRegistration from './pages/registration';
import Workspace from './pages/workspace';
import User from './pages/user';
import Diagrams from './pages/diagrams';
import About from './pages/about';
import FoldersList from './pages/folders';

function App() {
  return (
    <div style={{ display: 'flex', flexDirection: 'column', height: '100%' }}>
    <Router>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="*" element={<NotFound />} />
        <Route path="/login" element={<LoginPage />} />
        <Route path="/signup" element={<UserRegistration />} />
        <Route path="/about" element={<About />} />
        <Route path="/workspace" element={<Workspace />} />
        <Route path="/folders" element={<FoldersList />} />
        <Route path="/diagrams" element={<Diagrams />} />
        <Route path="/user" element={<User />} />
      </Routes>
    </Router>
    </div>
  );
}

export default App;
```

Фрагменти коду секції DiagramArea.js

```
import React, { useState, useRef, useEffect } from 'react';

function getClipPathForSector(startAngle, endAngle) {
  const centerX = 50;
  const centerY = 50;

  let angleDiff = endAngle - startAngle;
  if (angleDiff <= 0) {
    angleDiff += 360;
  }
  const points = [`${centerX}% ${centerY}%`];

  const rad1 = (startAngle * Math.PI) / 180;
  const x1 = centerX + centerX * Math.cos(rad1);
  const y1 = centerY + centerY * Math.sin(rad1);
  points.push(`${x1}% ${y1}%`);

  const steps = Math.max(1, Math.ceil(angleDiff / 10));
  for (let i = 1; i <= steps; i++) {
    const currentAngle = startAngle + (angleDiff * i) / steps;
    const currentRad = (currentAngle * Math.PI) / 180;
    const xCurrent = centerX + centerX * Math.cos(currentRad);
    const yCurrent = centerY + centerY * Math.sin(currentRad);
    points.push(`${xCurrent}% ${yCurrent}%`);
  }
}
```

```

    return `polygon(${points.join(', ')});`;
  }

function getMiddleAngle([startAngle, endAngle]) {
  if (startAngle <= endAngle) {
    return (startAngle + endAngle) / 2;
  } else {
    return ((startAngle + (endAngle + 360)) / 2) % 360;
  }
}

function DraggableCard({
  id,
  content,
  initialX,
  initialY,
  containerRef,
  onDragEnd,
  allowedAngles,
}) {
  const [position, setPosition] = useState({ x: initialX, y: initialY });
  const cardRef = useRef(null);
  const draggingData = useRef({
    isDragging: false,
    startX: 0,
    startY: 0,
    initX: 0,
    initY: 0,
  });

  const clampToSector = (newX, newY) => {
    const containerRect = containerRef.current.getBoundingClientRect();
    const containerWidth = containerRect.width;
    const containerHeight = containerRect.height;
    const centerX = containerWidth / 2;
    const centerY = containerHeight / 2;

    const cardRect = cardRef.current.getBoundingClientRect();
    const cardWidth = cardRect.width;
    const cardHeight = cardRect.height;

    const cardCenterX = newX + cardWidth / 2;
    const cardCenterY = newY + cardHeight / 2;
    let dx = cardCenterX - centerX;
    let dy = cardCenterY - centerY;
    let distance = Math.sqrt(dx * dx + dy * dy);
    let angle = Math.atan2(dy, dx) * (180 / Math.PI);
    if (angle < 0) angle += 360;

    let [startAngle, endAngle] = allowedAngles;
    let clampedAngle = angle;
    if (startAngle <= endAngle) {
      if (angle < startAngle) clampedAngle = startAngle;
      else if (angle > endAngle) clampedAngle = endAngle;
    } else {
      if (angle < startAngle && angle > endAngle) {
        const distToStart = Math.abs(angle - startAngle);
        const distToEnd = Math.abs(angle - endAngle);
        clampedAngle = distToStart < distToEnd ? startAngle : endAngle;
      }
    }
  }

  const containerRadius = containerWidth / 2;

```

```

    const safeMargin = Math.max(cardWidth, cardHeight) / 2;
    const maxDistance = containerRadius - safeMargin;
    const clampedDistance = Math.min(distance, maxDistance);

    const newCenterX = centerX + clampedDistance * Math.cos((clampedAngle *
Math.PI) / 180);
    const newCenterY = centerY + clampedDistance * Math.sin((clampedAngle *
Math.PI) / 180);
    return { x: newCenterX - cardWidth / 2, y: newCenterY - cardHeight / 2
};
};

const handleMouseDown = (e) => {
  e.stopPropagation();
  draggingData.current.isDragging = true;
  draggingData.current.startX = e.clientX;
  draggingData.current.startY = e.clientY;
  draggingData.current.initX = position.x;
  draggingData.current.initY = position.y;
  document.addEventListener('mousemove', handleMouseMove);
  document.addEventListener('mouseup', handleMouseUp);
};

const handleMouseMove = (e) => {
  if (!draggingData.current.isDragging) return;
  if (!containerRef.current || !cardRef.current) return;
  const deltaX = e.clientX - draggingData.current.startX;
  const deltaY = e.clientY - draggingData.current.startY;
  let newX = draggingData.current.initX + deltaX;
  let newY = draggingData.current.initY + deltaY;

  if (allowedAngles) {
    const clampedPos = clampToSector(newX, newY);
    newX = clampedPos.x;
    newY = clampedPos.y;
  } else {

    const containerRect = containerRef.current.getBoundingClientRect();
    const cardRect = cardRef.current.getBoundingClientRect();
    const maxX = containerRect.width - cardRect.width;
    const maxY = containerRect.height - cardRect.height;
    newX = Math.max(0, Math.min(newX, maxX));
    newY = Math.max(0, Math.min(newY, maxY));
  }
  setPosition({ x: newX, y: newY });
};

const handleMouseUp = () => {
  draggingData.current.isDragging = false;
  document.removeEventListener('mousemove', handleMouseMove);
  document.removeEventListener('mouseup', handleMouseUp);
  if (onDragEnd) {
    onDragEnd(id, position);
  }
};

return (
  <div
    ref={cardRef}
    onMouseDown={handleMouseDown}
    style={{
      position: 'absolute',
      left: position.x,
      top: position.y,

```

```

        padding: '8px 12px',
        backgroundColor: '#f0f0f0',
        border: '1px solid #ccc',
        borderRadius: '4px',
        cursor: 'grab',
        userSelect: 'none',
    }}
    >
    {content}
</div>
);
}

function Sector({
  category,
  cards,
  onAddCard,
  onUpdateCardPosition,
  newCardContent,
  allowedAngles,
}) {
  const sectorRef = useRef(null);

  const handleClick = (e) => {
    if (!newCardContent) return;
    if (e.target !== sectorRef.current) return;
    const rect = sectorRef.current.getBoundingClientRect();
    const x = e.clientX - rect.left;
    const y = e.clientY - rect.top;
    const centerX = rect.width / 2;
    const centerY = rect.height / 2;
    let dx = x - centerX;
    let dy = y - centerY;
    let clickAngle = Math.atan2(dy, dx) * (180 / Math.PI);
    if (clickAngle < 0) clickAngle += 360;
    const [sectorStart, sectorEnd] = allowedAngles;
    let isInSector = false;
    if (sectorStart <= sectorEnd) {
      isInSector = (clickAngle >= sectorStart && clickAngle <=
sectorEnd);
    } else {
      isInSector = (clickAngle >= sectorStart || clickAngle <=
sectorEnd);
    }
    if (!isInSector) return;

    onAddCard(category, { content: newCardContent, x, y });
  };

  const clipPath = getClipPathForSector(allowedAngles[0],
allowedAngles[1]);
  const middleAngle = getMiddleAngle(allowedAngles);
  const labelRadiusPercent = 35;
  const labelX = 50 + labelRadiusPercent * Math.cos((middleAngle * Math.PI)
/ 180);
  const labelY = 50 + labelRadiusPercent * Math.sin((middleAngle * Math.PI)
/ 180);

  return (
    <div
      ref={sectorRef}
      onClick={handleClick}
      style={{
        position: 'absolute',

```

```

    top: 0,
    left: 0,
    width: '100%',
    height: '100%',
    clipPath: clipPath,
    overflow: 'hidden',
  }}
>
<div
  style={{
    position: 'absolute',
    left: `${labelX}%`,
    top: `${labelY}%`,
    transform: 'translate(-50%, -50%)',
    fontWeight: 'bold',
    pointerEvents: 'none',
  }}
>
  {category}
</div>
{cards.map((card) => (
  <DraggableCard
    key={card.id}
    id={card.id}
    content={card.content}
    initialX={card.x}
    initialY={card.y}
    containerRef={sectorRef}
    onDragEnd={({id, pos}) => onUpdateCardPosition(category, id, pos)}
    allowedAngles={allowedAngles}
  />
))}
</div>
);
}

```

```

const DiagramArea = ({ newCardContent, handleNewCard, sectorsData,
setSectorsData }) => {
  const handleCircleClick = (e, sectorsConfig) => {
    const rect = e.currentTarget.getBoundingClientRect();
    const x = e.clientX - rect.left;
    const y = e.clientY - rect.top;
    const centerX = rect.width / 2;
    const centerY = rect.height / 2;
    const dx = x - centerX;
    const dy = y - centerY;
    let clickAngle = Math.atan2(dy, dx) * (180 / Math.PI);
    if (clickAngle < 0) clickAngle += 360;
    sectorsConfig.forEach((sector) => {
      const [sectorStart, sectorEnd] = sector.allowedAngles;
      let isInSector = false;
      if (sectorStart <= sectorEnd) {
        isInSector = (clickAngle >= sectorStart && clickAngle <=
sectorEnd);
      } else {
        isInSector = (clickAngle >= sectorStart || clickAngle <=
sectorEnd);
      }
      if (isInSector && newCardContent) {
        const name = handleNewCard(sector.category, x, y)
        handleAddCard(sector.category, { content: name, x, y });
      }
    });
  };
};

```

```

const [nextId, setNextId] = useState(1);

const handleUpdateCardPosition = (category, cardId, newPosition) => {
  setSectorsData((prev) => ({
    ...prev,
    [category]: prev[category].map((card) =>
      card.id === cardId ? { ...card, x: newPosition.x, y: newPosition.y
    } : card
  )),
  ));
};

const circleSize = 600;
const leftCircleSectors = [
  { category: 'Customer Jobs', allowedAngles: [330, 90] },
  { category: 'Pains', allowedAngles: [90, 210] },
  { category: 'Gains', allowedAngles: [210, 330] },
];
const rightCircleSectors = [
  { category: 'Products & Services', allowedAngles: [150, 270] },
  { category: 'Gain Creators', allowedAngles: [270, 30] },
  { category: 'Pain Relievers', allowedAngles: [30, 150] },
];

return (
  <div style={{ display: 'flex', justifyContent: 'space-around', gap:
'16px' }}>

<div
  style={{
    position: 'relative',
    width: circleSize,
    height: circleSize,
    borderRadius: '50%',
    border: '2px solid #81C784',
    overflow: 'hidden',
    margin: '8px',
  }}
  onClick={(e) => handleCircleClick(e, leftCircleSectors)}
>
  {leftCircleSectors.map((sector) => (
    <Sector
      key={sector.category}
      category={sector.category}
      cards={sectorsData[sector.category]}
      onUpdateCardPosition={handleUpdateCardPosition}
      allowedAngles={sector.allowedAngles}
    />
  ))}
</div>

    <div
      style={{
        position: 'relative',
        width: circleSize,
        height: circleSize,
        borderRadius: '50%',
        border: '2px solid #81C784',
        overflow: 'hidden',
        margin: '8px',
      }}
      onClick={(e) => handleCircleClick(e, rightCircleSectors)}
    >

```

```

    {rightCircleSectors.map((sector) => (
      <Sector
        key={sector.category}
        category={sector.category}
        cards={sectorsData[sector.category]}
        onUpdateCardPosition={handleUpdateCardPosition}
        allowedAngles={sector.allowedAngles}
      />
    ))}
  </div>
</div>
);
};

export default DiagramArea;

```

Компонент модуля ввода данных ConfigurableForm.js

```

import React, { useState } from 'react';
import axios from 'axios';
import PropTypes from 'prop-types';

const ConfigurableForm = ({ fields, caption, apiEndpoint, onSuccess,
customHandleSubmit }) => {
  const [formData, setFormData] = useState(() => {
    fields.reduce((acc, field) => {
      acc[field.name] = field.value || '';
      return acc;
    }, {})
  });

  const handleInputChange = (e) => {
    const { name, value } = e.target;
    setFormData({ ...formData, [name]: value });
  };

  const defaultHandleSubmit = async (e) => {
    e.preventDefault();
    try {
      const response = await axios.post(apiEndpoint, formData, {
withCredentials: true });
      console.log('Submission successful:', response.data);
      if (onSuccess) onSuccess(response.data);
    } catch (error) {
      console.error('Error during form submission:', error);
    }
    return formData;
  };

  const handleSubmit = customHandleSubmit && ((e) => { e.preventDefault();
customHandleSubmit(formData); return formData; }) || defaultHandleSubmit;

  const formContainerStyle = {
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
    gap: '18px',
    padding: '30px 30px 30px 30px',
    backgroundColor: '#E8F5E9',
    borderRadius: '10px',
    boxShadow: '0 4px 8px rgba(0, 0, 0, 0.27)',
  };

```

```

    maxWidth: '400px',
    margin: '1rem auto',
    boxSizing: 'border-box',
  };

  const headingStyle = {
    color: '#1B5E20',
    marginBottom: '10px',
    fontSize: '2rem',
  };

  const inputStyle = {
    width: '100%',
    padding: '16px',
    borderRadius: '6px',
    border: '1px solid #81C784',
    fontSize: '1rem',
    boxSizing: 'border-box',
    backgroundColor: '#ffffff',
  };

  const buttonBaseStyle = {
    width: '100%',
    padding: '16px',
    borderRadius: '6px',
    border: 'none',
    backgroundColor: '#2E7D32',
    color: 'white',
    fontSize: '1.1rem',
    fontWeight: 'bold',
    cursor: 'pointer',
    transition: 'background-color 0.2s ease-in-out',
    marginTop: '10px',
    boxSizing: 'border-box',
  };

  const buttonHoverStyle = {
    ...buttonBaseStyle,
    backgroundColor: '#1B5E20',
  };

  return (
    <div style={formContainerStyle}>
      <h1 style={headingStyle}>{caption}</h1>
      <form onSubmit={handleSubmit}>
        {fields.map((field) =>
          field.type === "textarea" ? (
            <textarea
              key={field.name}
              name={field.name}
              placeholder={field.placeholder}
              value={formData[field.name]}
              onChange={handleInputChange}
              required={field.required || false}
              style={inputStyle}
            />
          ) : (
            <input
              key={field.name}
              type={field.type || "text"}
              name={field.name}
              placeholder={field.placeholder}
              value={formData[field.name]}
              onChange={handleInputChange}

```

```

                required={field.required || false}
                style={inputStyle}
            />
        )
    )}
    <button
        type="submit"
        style={buttonBaseStyle}
        onMouseOver={ (e) =>
            (e.target.style.backgroundColor =
buttonHoverStyle.backgroundColor)
        }
        onMouseOut={ (e) =>
            (e.target.style.backgroundColor =
buttonBaseStyle.backgroundColor)
        }
    >
        Submit
    </button>
</form>
</div>
);
};

ConfigurableForm.propTypes = {
    fields: PropTypes.arrayOf(
        PropTypes.shape({
            name: PropTypes.string.isRequired,
            type: PropTypes.string,
            placeholder: PropTypes.string,
            required: PropTypes.bool,
        })
    ).isRequired,
    caption: PropTypes.string.isRequired,
    apiEndpoint: PropTypes.string.isRequired,
    onSuccess: PropTypes.func,
};

export default ConfigurableForm;

```

Модуль ініціалізації северної частини Api.go

```

package main

import (
    "fmt"
    "net/http"
    "os"

    "github.com/gorilla/mux"
    "github.com/gorilla/sessions"
    "github.com/rs/cors"
)

type API struct {
    storage      DB
    listenTo     string
    environment  string
    sessionStore *sessions.CookieStore
}

```

```

func (s *API) Start() {
    router := mux.NewRouter()

    router.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        fmt.Fprint(w, "Hello, World!")
    })

    router.HandleFunc("/api/cards", searchCard).Methods(http.MethodGet)
    router.HandleFunc("/api/cards", addCard).Methods(http.MethodPost)
    router.HandleFunc("/api/cards", generateCard).Methods(http.MethodPost)
    router.HandleFunc("/api/cards", editCard).Methods(http.MethodPut)
    router.HandleFunc("/api/cards", deleteCard).Methods(http.MethodDelete)
    router.HandleFunc("/api/user/register",
makeHandler(s.registerUser)).Methods(http.MethodPost)
    router.HandleFunc("/api/user/auth",
makeHandler(s.authUser)).Methods(http.MethodPost)
    router.HandleFunc("/api/board",
makeHandler(s.addBoard)).Methods(http.MethodPost)
    router.HandleFunc("/api/board",
makeHandler(s.editBoard)).Methods(http.MethodPut)
    router.HandleFunc("/api/board",
makeHandler(s.deleteBoard)).Methods(http.MethodDelete)
    router.HandleFunc("/api/folders",
makeHandler(s.addFolder)).Methods(http.MethodPost)
    router.HandleFunc("/api/folders",
makeHandler(s.editFolder)).Methods(http.MethodPut)
    router.HandleFunc("/api/folders",
makeHandler(s.deleteFolder)).Methods(http.MethodDelete)
    router.HandleFunc("/api/folders",
makeHandler(s.getFolders)).Methods(http.MethodGet)
    router.HandleFunc("/api/access",
makeHandler(s.addAccess)).Methods(http.MethodPost)
    router.HandleFunc("/api/board/{access-uuid}/cards",
makeHandler(s.getCards)).Methods(http.MethodGet)
    router.HandleFunc("/api/board/{access-uuid}/cards",
makeHandler(s.addCard)).Methods(http.MethodPost)
    router.HandleFunc("/api/user/search",
makeHandler(s.searchUser)).Methods(http.MethodGet)
    router.HandleFunc("/api/user",
makeHandler(s.editUser)).Methods(http.MethodPost)

    c := cors.New(cors.Options{
        AllowedOrigins:  os.Getenv("CORS_ORIGINS"),
        AllowedMethods:  []string{http.MethodGet, http.MethodPost,
http.MethodPut, http.MethodDelete, http.MethodOptions},
        AllowedHeaders:  []string{"*"},
        AllowCredentials: true,
    })
    handler := c.Handler(router)

    http.ListenAndServe(s.listenTo, handler)
}

```

Модуль отримання діаграми за UUID uuid-board.go

```

package main

import (
    "database/sql"

```

```

    _ "github.com/lib/pq"
)

func (db *StoragePSQL) getBoardByAccessUUID(accessUUID string, user
*UserStored) (*BoardStored, int32, error) {
    connection, err := sql.Open("postgres", db.connectionString())
    if err != nil {
        return nil, 0, err
    }
    defer connection.Close()

    var board BoardStored
    var accessLevel int32
    err = connection.QueryRow(`
        SELECT b.id, b.uid, b.name, b.description, b.is_public,
a.access_level
        FROM boards b
        JOIN access a ON b.id = a.board_id
        WHERE a.access_uid = $1 AND a.target_ids @> ARRAY[$2]
    `, accessUUID, user.id).Scan(&board.id, &board.UUID, &board.Name,
&board.Description, &board.Is_public, &accessLevel)

    if err != nil {
        return nil, 0, err
    }

    return &board, accessLevel, nil
}

```

Додаток 3
Копія презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ ДІАГРАМАМИ ЦІННІСНОЇ ПРОПОЗИЦІЇ

Виконав: студент групи КП-12 Мальцев Микита Ігорович

Керівник: доцент кафедри ПЗКС, к.т.н., доцент Люшенко Леся Анатоліївна

Київ – 2025



ПОСТАНОВКА ЗАДАЧІ

Мета проєкту: Метою даного дипломного проєкту є створення вебзастосунку, що надаватиме функціональність для створення ціннісних пропозицій та додавання нових якостей для вже створених ціннісних пропозицій за допомогою сучасних програмних технологій.

Завдання:

1. Проаналізувати існуючі програмні рішення
2. Визначити вимоги до розробки
3. Обрати засоби реалізації
4. Розробити вебзастосунок відповідно до вимог
5. Протестувати розроблений вебзастосунок
6. Визначити шляхи подальшого розвитку



АКТУАЛЬНІСТЬ

Діаграма ціннісної пропозиції є потужним інструментом для проектування продуктів та послуг. За допомогою діаграм ціннісної пропозиції можна знайти властивості продуктів, які будуть важливими для користувачів. Існуючі рішення для створення таких діаграм не надають цілісного набору можливостей для створення таких діаграм та управління ними.



ІСНУЮЧІ АНАЛОГИ

Canva



VALUE

PROP



ЗБІР ВИМОГ

Після проведення аналізу існуючих аналогів було зібрано вимоги до розроблюваного вебзастосунку. Метою даного етапу проектування було знаходження потреб користувачів, що не задовольняються наявними рішеннями.

Зібрані вимоги до вебзастосунку включають наявність функціональності для створення та редагування діаграм, додавання тегів, групування діаграм у спільні теки. Також були додані вимоги щодо автоматизації створення діаграм ціннісної пропозиції.

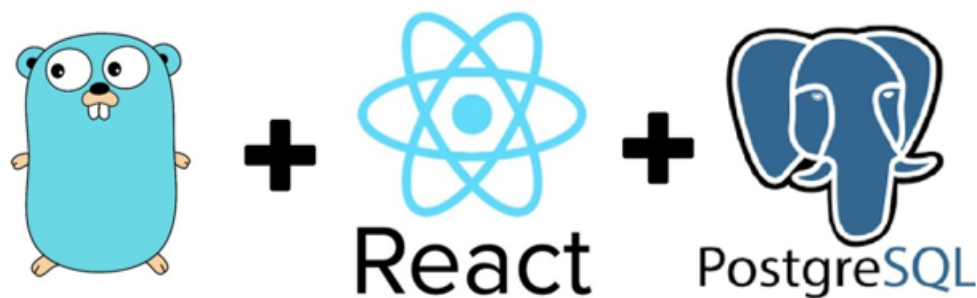
ЗАСОБИ РОЗРОБЛЕННЯ

Тип ПЗ — **вебзастосунок**.

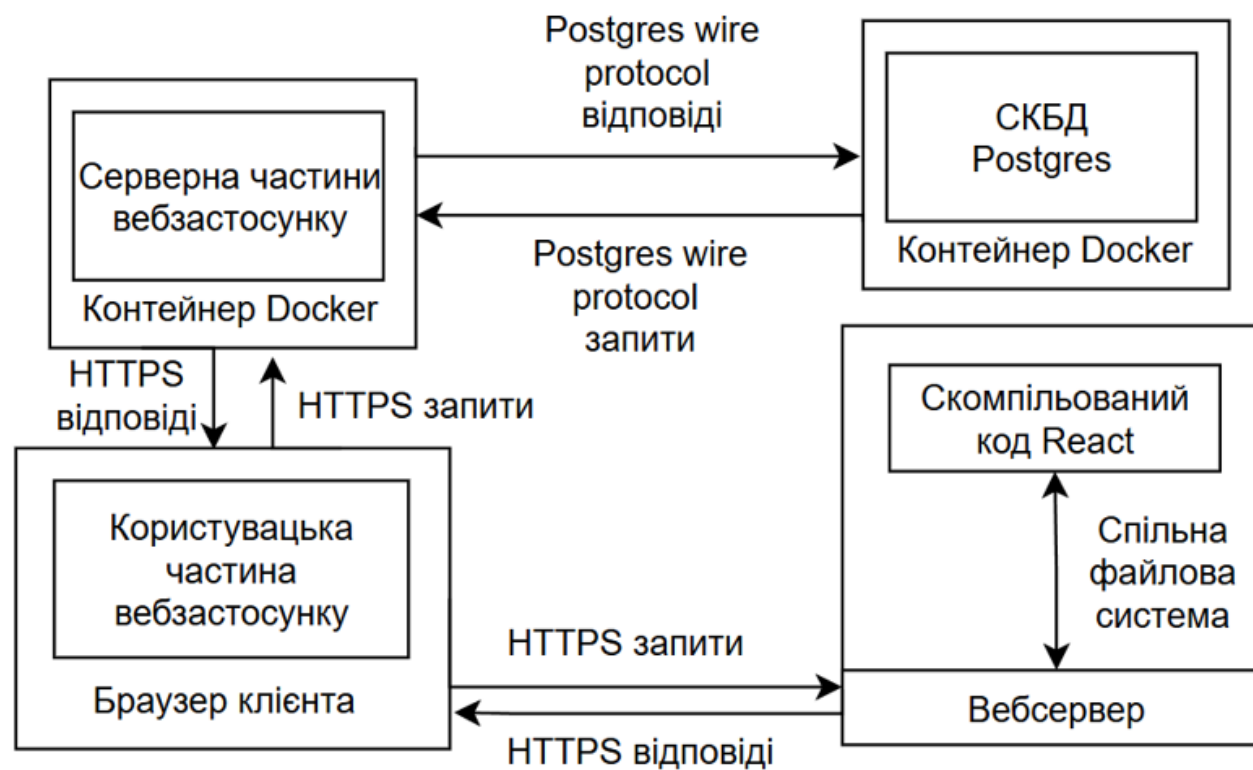
Мова програмування (сервер) — **Go**, бібліотека gorilla/mux.

Мова програмування (клієнт) — **JavaScript**, бібліотека React.

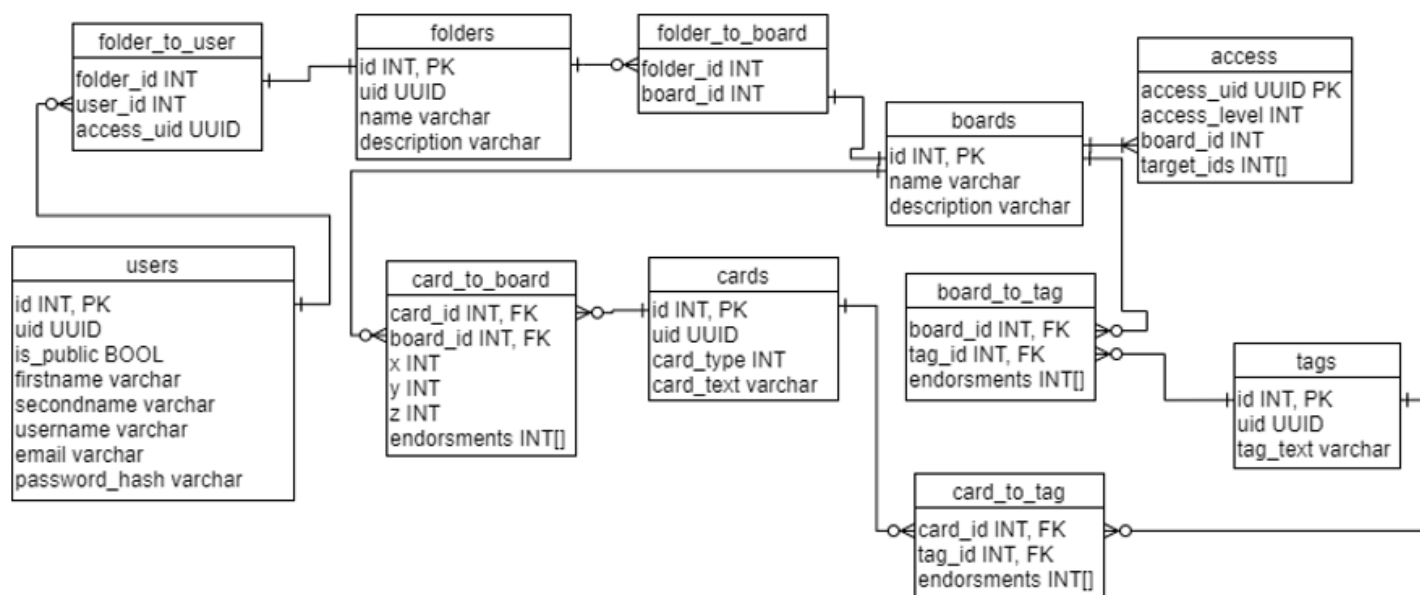
База даних — **PostgreSQL**.



АРХІТЕКТУРА СИСТЕМИ



СТРУКТУРА БАЗИ ДАНИХ



ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



ПРИКЛАД РОБОТИ ЗАСТОСУНКУ

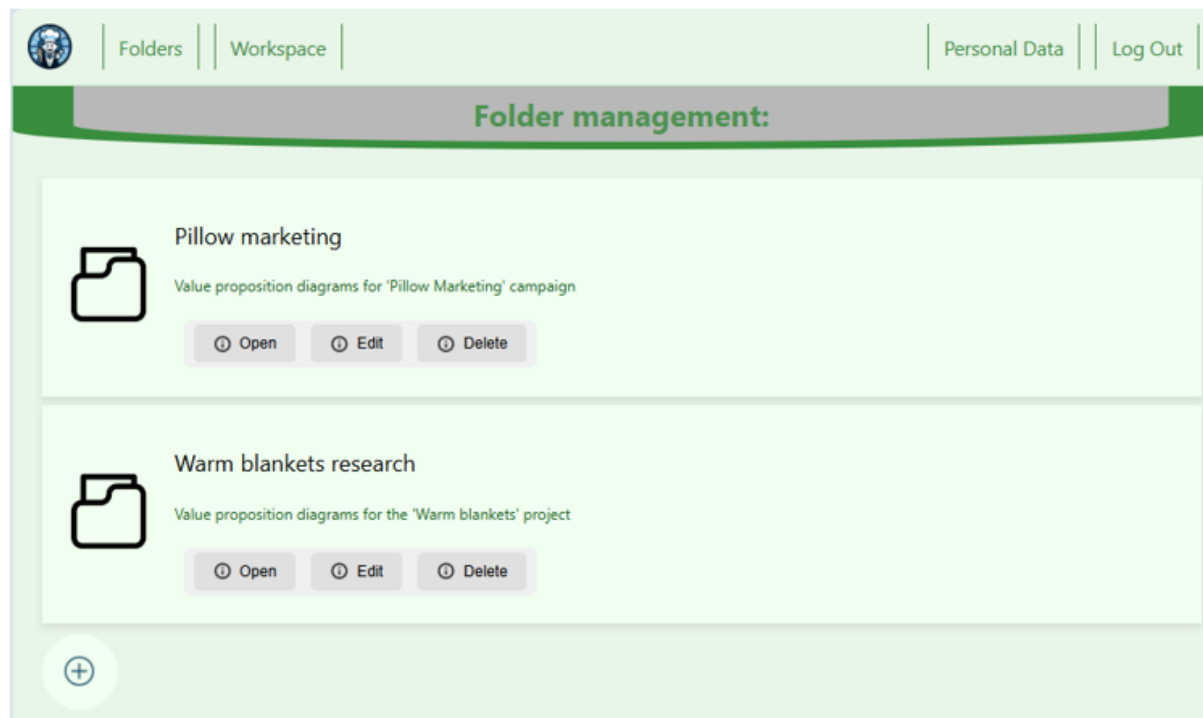


[Home](#) | [About](#) | [Login](#) | [SignUp](#)

Login

Submit

ПРИКЛАД РОБОТИ ЗАСТОСУНКУ



ПРИКЛАД РОБОТИ ЗАСТОСУНКУ

 | [Folders](#) | [Workspace](#) | [Personal Data](#) | [Log Out](#)

Diagram management:



Pillow for elderly

Pillows for elderly people

[Open](#) [Edit](#) [Delete](#)

Tag1 2 ✕ Tag2 1 ✕

[Add Tag](#)



Ergonomic pillow

Pillow that mimics the shape of the customer's head

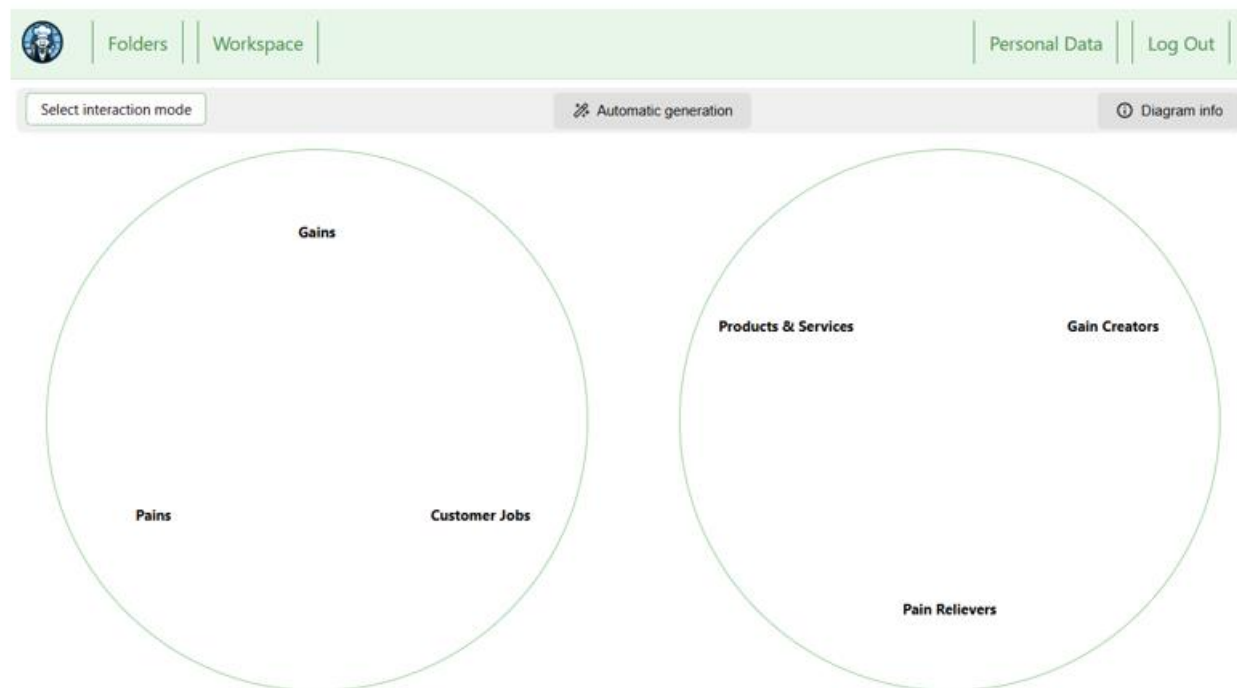
[Open](#) [Edit](#) [Delete](#)

Tag3 1 ✕

[Add Tag](#)

[+](#)

ПРИКЛАД РОБОТИ ЗАСТОСУНКУ



ПРИКЛАД РОБОТИ ЗАСТОСУНКУ



×

Add card

Card text

Submit

Enter tag name

Add Tag



ТЕСТУВАННЯ ЗАСТОСУНКУ

Модульне тестування застосовувалось для застосування сталого набору перевірок для компонентів вебзастосунку. Це допомогло виявити помилки при модифікації компонентів перевірки вхідних даних

Системне тестування використовувалось задля перевірки відповідності роботи інформаційної системи. Прикладом помилок що були виявлені на цьому етапі було використання невірного типу даних для додавання масивів до БД.

Всі виявлені помилки було виправлено.

ПОРІВНЯННЯ З АНАЛОГАМИ

Застосунок	Draw.io	Canva	valueprop.dev	Scheme Chef
Наявність шаблонів ціннісної пропозиції	-	+	+	+
Отримання підказок щодо елементів ціннісної пропозиції	-	-	-	+
Вбудована можливість поширення діаграми за допомогою посилання	-	+	+	+
Керування доступом до редагування діаграми	+	+	-	+
Групування діаграм у спільні текти	-	+	-	+
Наявність безкоштовної версії	+	+	-	+
Можливість штучного створення ціннісної пропозиції на основі опису продукту	-	-	+	+

ШЛЯХИ ПОДАЛЬШОГО РОЗВИТКУ



1. Надати можливість роботи із застосунком у вкладеному вікні
2. Покращити засіб підказок для створення елементів ціннісної пропозиції
3. Покращити засоби для управління діаграмами
4. Реалізувати можливість одночасної роботи над діаграмою кількох користувачів
5. Надати можливість авторизації за допомогою OIDC



ВИСНОВКИ

1. Проаналізовано існуючі програмні рішення.
2. Спроектовано архітектуру вебзастосунку та структуру БД
3. Розроблено серверну та клієнтську частини вебзастосунку.
4. Протестовано розробку шляхом ручного і автоматизованого тестування.
5. Визначено шляхи подальшого розвитку.

Розробка виконана у повному обсязі відповідно до Технічного завдання, тестування проведено згідно з затвердженою Програмою та методикою тестування



Дякую за увагу!

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ____ ” _____ 2024 р.

**ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ ДІАГРАМАМИ ЦІННІСНОЇ
ПРОПОЗИЦІЇ**

Програма та методика тестування

ДП.045440-04-51

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Леся ЛЮШЕНКО

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Микита МАЛЬЦЕВ

ЗМІСТ

1. Об'єкт випробувань	3
2. Мета тестування	3
3. Методи тестування.....	3
4. Засоби та порядок тестування.....	4

1. ОБ'ЄКТ ВИПРОБУВАНЬ

Вебзастосунок для управління діаграмами ціннісної пропозиції, що представляє собою клієнтську частину, створену за допомогою мови програмування JavaScript із використанням бібліотеки React, та серверної частини, створеної за допомогою мови програмування Go.

2. МЕТА ТЕСТУВАННЯ

Під час тестування необхідно перевірити наступне:

- 1) коректну роботу користувацьких сторінок вебзастосунку;
- 2) взаємодію клієнтської сторони вебзастосунку із серверною частиною;
- 3) відповідність обміну даних між серверною частиною вебзастосунку та базою даних;
- 4) безпеку даних при передачі даних та при їх зберіганні;
- 5) коректна робота вебзастосунку у різних браузерах;
- 6) відповідність створеної інформаційної системи технічним вимогам.

3. МЕТОДИ ТЕСТУВАННЯ

Задля тестування програмного застосовувався набір методів, що відноситься до стратегії тестування Gray Box Testing. Робота окремих модулів програмного продукту перевірялася за допомогою модульного тестування. Взаємодія ж усіх компонентів системи перевірялася за допомогою системного тестування. Використовуються наступні методи:

- 1) функціональне тестування;
- 2) юніт-тестування;
- 3) тестування сумісності з різними браузерами та платформами;
- 4) тестування інтерфейсу;
- 5) тестування взаємодії компонентів інформаційної системи.

4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконувалося за допомогою бібліотеки go/testing. Ручне тестування проводилося із використанням різних браузерів на різних платформах.

Працездатність веб-ресурсу перевіряється шляхом:

- 1) динамічного ручного тестування – введенням граничних та недопустимих значень в поля, які можна редагувати;
- 2) динамічного ручного тестування на відповідність функціональним вимогам;
- 3) статичного тестування коду;
- 4) тестування вебзастосунку у різних браузерах;
- 5) автоматизованого тестування модулів коду;
- 6) тестування стабільності роботи при різних умовах;
- 7) тестування інтерфейсу.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ____ ” _____ 2025 р.

**ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ ДІАГРАМАМИ ЦІННІСНОЇ
ПРОПОЗИЦІЇ**

Керівництво користувача

ДП.045440-05-34

“ПОГОДЖЕНО”

Керівник проекту:

_____ Леся ЛЮШЕНКО

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Микита МАЛЬЦЕВ

ЗМІСТ

1. Опис структури вебзастосунку	3
2. Процедура реєстрації та авторизації користувача	3
3. Управління теками діаграм ціннісної пропозиції	5
4. Управління діаграмами ціннісної пропозиції у течі	7
5. Процедура створення та редагування діаграми ціннісної пропозиції	9
6. Процедура управління інформацією користувача	10

1. Опис структури вебзастосунку

Вебзастосунок для управління ціннісними пропозиціями «Scheme chef» складається з динамічних вебсторінок. Для авторизованих користувачів доступні такі сторінки: сторінка перегляду колекцій діаграм, сторінка перегляду діаграм у колекції, сторінка редагування діаграми ціннісної пропозиції та сторінка керування інформацією користувача..

Для не авторизованих користувачів доступні статичні сторінки для отримання загальної інформації про застосунок. Не авторизованому користувачу також доступні сторінки створення облікового запису та входу у існуючий обліковий запис.

2. Процедура реєстрації та авторизації користувача

Користувач, що знаходиться на головній інформаційній сторінці (рис. 1) може перейти на сторінку створення облікового запису (рис. 2) або на сторінку авторзації (рис. 3). При вході в обліковий запис користувач вводить свою електронну пошту та пароль, на натискає на кнопку «Submit». Після успішної авторизації користувач перенаправляється на сторінку перегляду наявних тек із діаграмами ціннісної пропозиції (рис. 4).

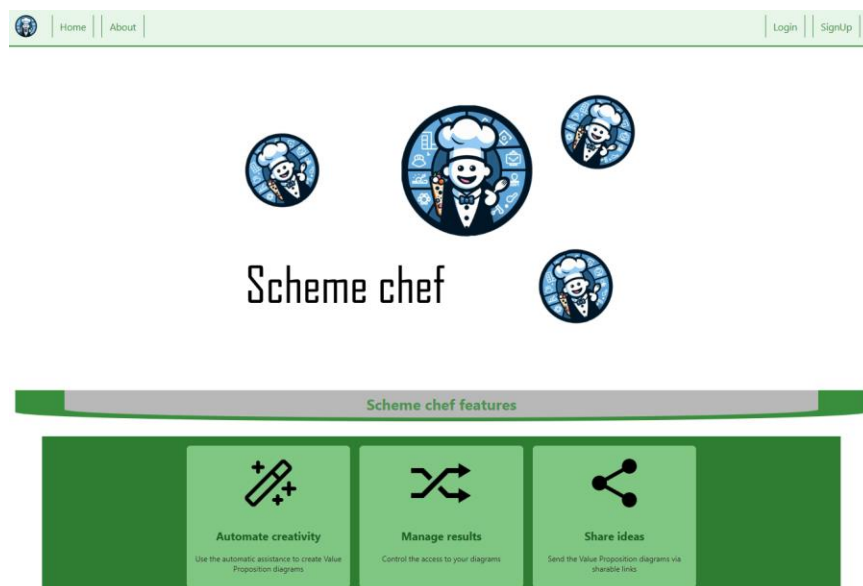
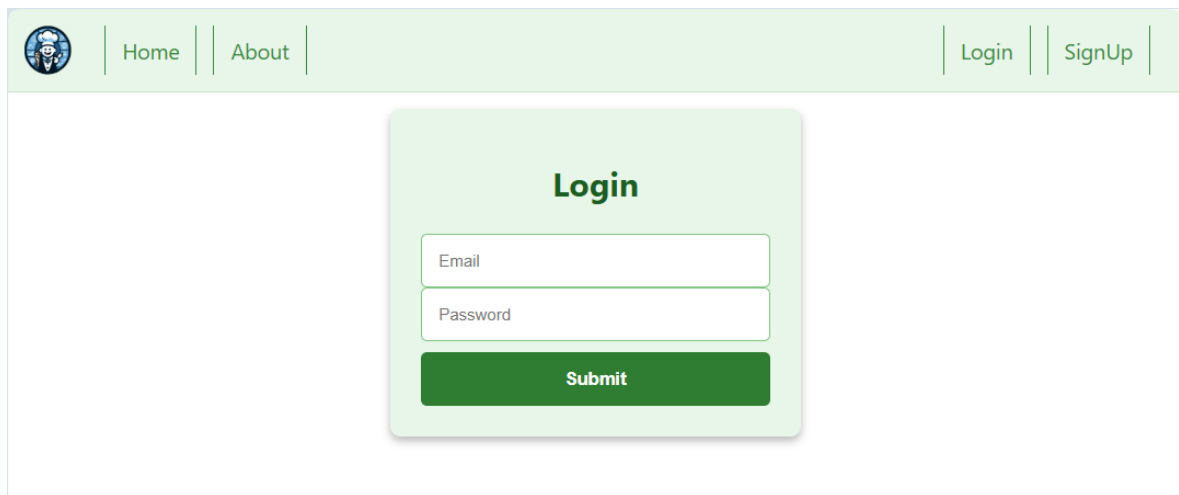


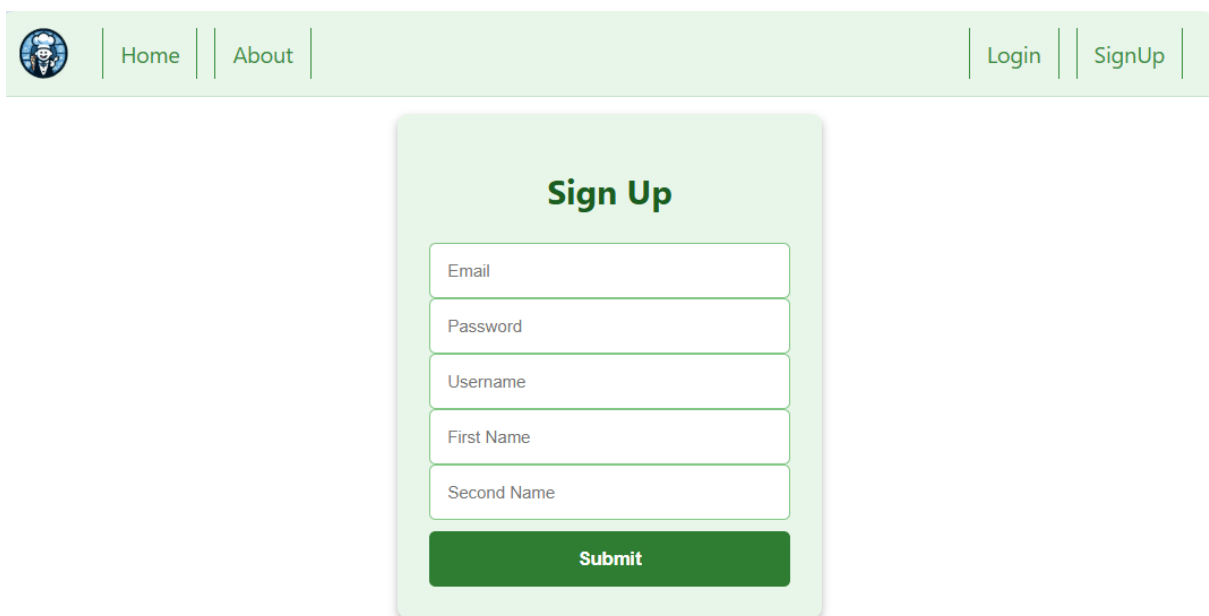
Рис. 1. Головна інформаційна сторінка



The image shows a web interface for a login page. At the top, there is a light green header bar. On the left side of the header, there is a circular logo with a person's face. To the right of the logo are two links: "Home" and "About". On the right side of the header, there are two links: "Login" and "SignUp". Below the header, the main content area is white. In the center, there is a light green rounded rectangle containing the login form. The form has a title "Login" in bold green text. Below the title, there are two input fields: "Email" and "Password". Below these fields is a green button with the text "Submit" in white.

Рис. 2. Сторінка авторизації

Для реєстрації користувач вводить свою електронну пошту, пароль, логін, ім'я та фамілію та натискає кнопку «Submit». Користувач також може перейти на сторінку авторизації, у разі якщо акаунт вже існує. При успішному створенні облікового запису користувач перенаправляється на сторінку перегляду наявних тек із діаграмами ціннісної пропозиції (рис. 4).



The image shows a web interface for a sign-up page. At the top, there is a light green header bar. On the left side of the header, there is a circular logo with a person's face. To the right of the logo are two links: "Home" and "About". On the right side of the header, there are two links: "Login" and "SignUp". Below the header, the main content area is white. In the center, there is a light green rounded rectangle containing the sign-up form. The form has a title "Sign Up" in bold green text. Below the title, there are five input fields: "Email", "Password", "Username", "First Name", and "Second Name". Below these fields is a green button with the text "Submit" in white.

Рис. 3. Сторінка реєстрації

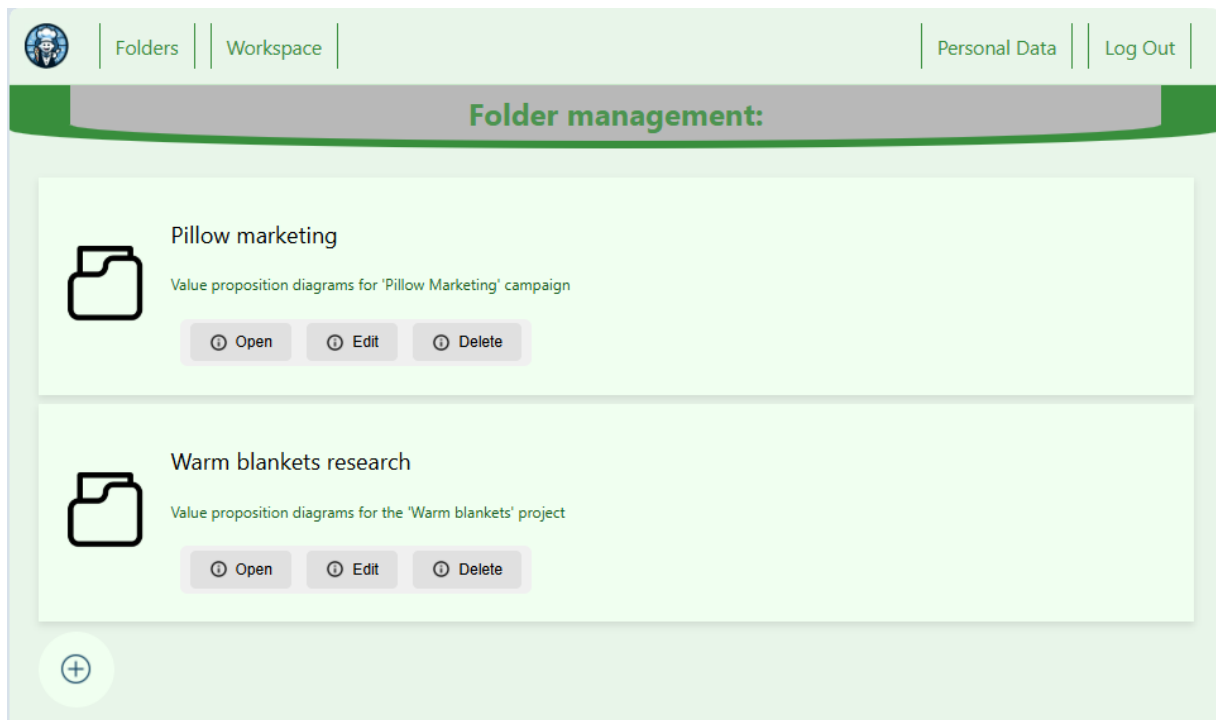


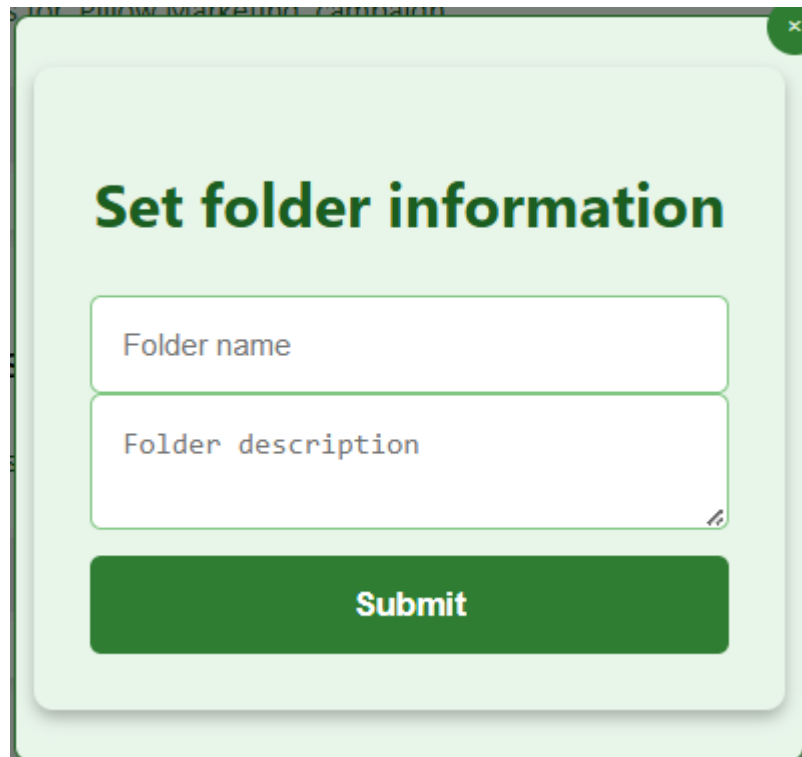
Рис. 4. Сторінка управління теками діаграм ціннісної пропозиції

3. Управління теками діаграм ціннісної пропозиції

На сторінці управління теками діаграм ціннісної пропозиції (рис. 4) користувач може додавати нові теки, натиснувши на кнопку із позначкою додавання. Користувачу буде відкрито модальне вікно встановлення інформації щодо теки (рис. 5). Після заповнення даних щодо теки та натискання кнопки «Submit» теку буде створено. Користувача буде перенаправлено на сторінку управління діаграмами у теці (рис. 6). Користувач також може перейти на сторінку управління діаграмами у теці (рис. 6) для існуючої теки, натиснувши на кнопку «Open».

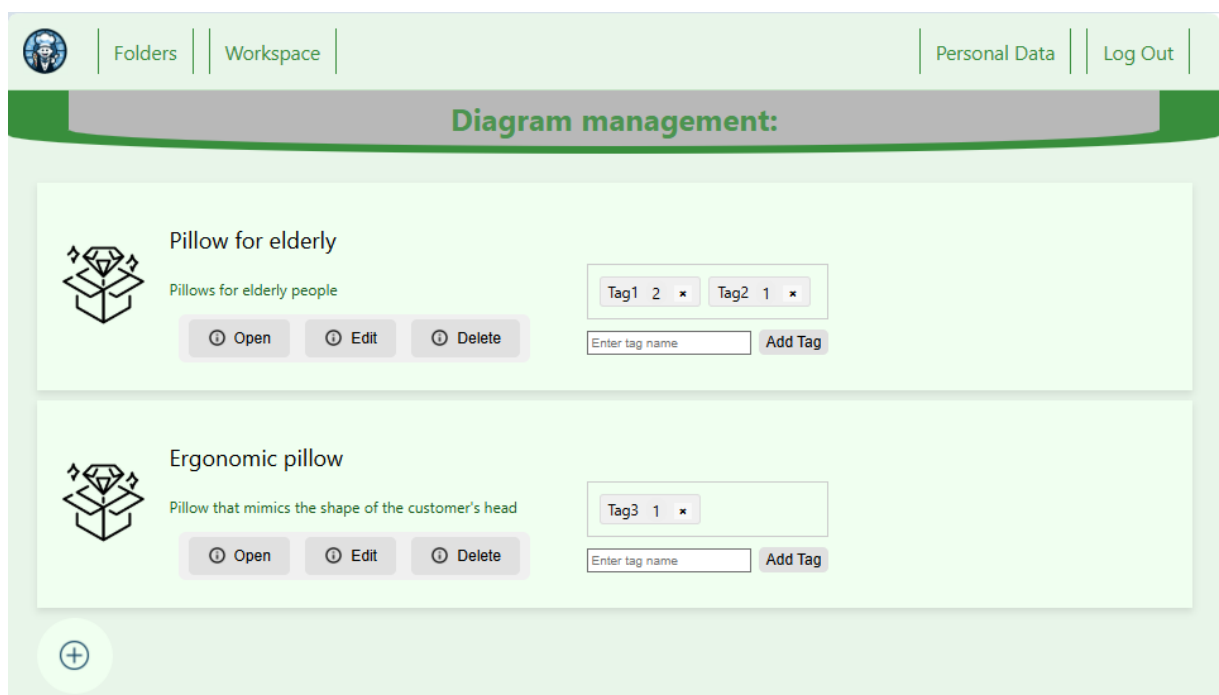
Користувач також може редагувати інформацію щодо наявних діграм ціннісної пропозиції, натиснувши на кнопку «Edit». Користувачу буде відкрито модальне вікно встановлення інформації щодо теки (рис. 5). Поточна інформація щодо теки буде заповнена у поля редагування. Після оновлення інформації щодо теки та натискання кнопки «Submit» інформація буде оновлена. Користувач не буде перенаправлений на іншу сторінку.

Користувач також може видалити теку із діаграмами ціннісної пропозиції, натиснувши на кнопку «Delete».



A modal window titled "Set folder information" with a close button in the top right corner. It contains two text input fields: "Folder name" and "Folder description". Below the fields is a green "Submit" button.

Рис. 5. Модальне вікно редагування



The "Diagram management" interface features a top navigation bar with links for "Folders", "Workspace", "Personal Data", and "Log Out". The main content area lists two diagrams, each with a box icon and a title:

- Pillow for elderly**: Subtitle "Pillows for elderly people". It has "Open", "Edit", and "Delete" buttons. To the right, it shows "Tag1 2" and "Tag2 1" with an "Add Tag" button and an "Enter tag name" input field.
- Ergonomic pillow**: Subtitle "Pillow that mimics the shape of the customer's head". It also has "Open", "Edit", and "Delete" buttons. To the right, it shows "Tag3 1" with an "Add Tag" button and an "Enter tag name" input field.

A plus icon in a circle is located at the bottom left of the interface.

Рис. 6. Сторінка управління діаграмами ціннісної пропозиціями у теці

4. Управління діаграмами ціннісної пропозиції у теці

Для створення діаграми ціннісної пропозиції, користувач може натиснути на кнопку із позначкою додавання на сторінці управління діаграмами ціннісної пропозиції у теці (рис. 6). Користувачу буде відкрите модальне вікно встановлення інформації щодо теки (рис. 7).

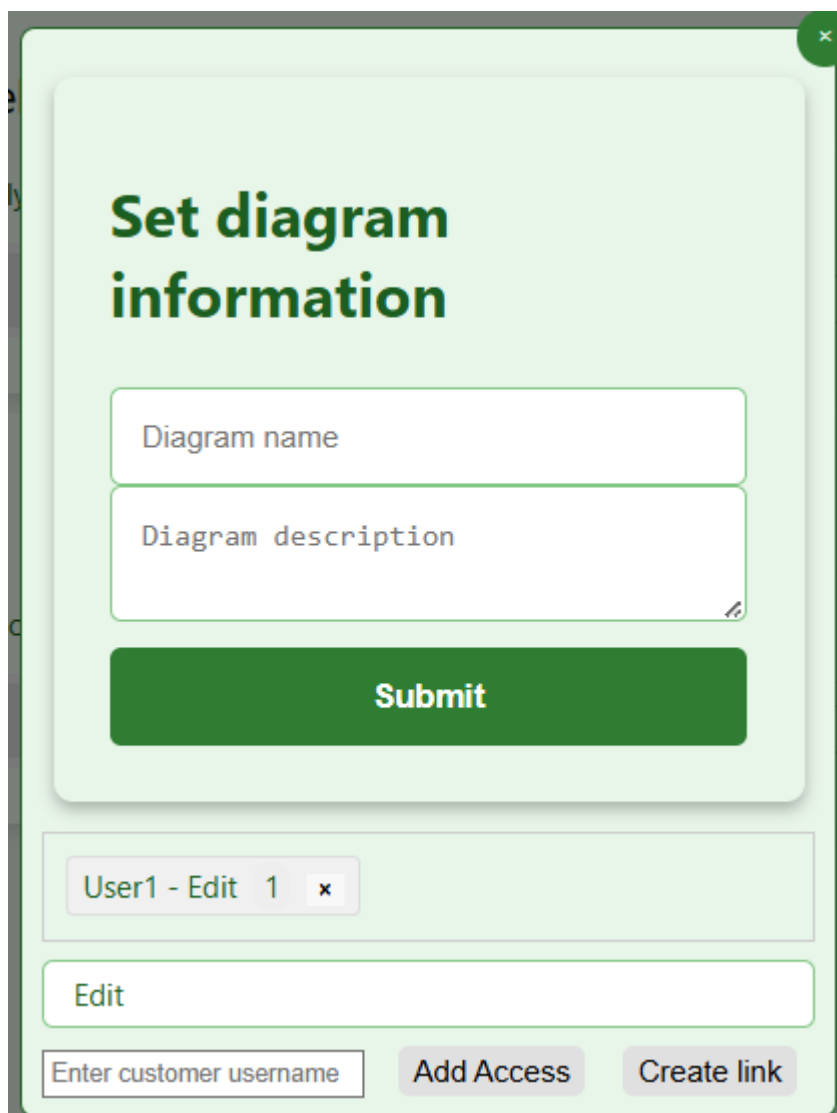
A modal window titled "Set diagram information" with a green header and a close button in the top right corner. It contains two text input fields: "Diagram name" and "Diagram description". Below these fields is a large green "Submit" button. At the bottom of the modal, there is a section with a tab labeled "User1 - Edit 1" and a close button. Below the tab is an "Edit" button. At the very bottom of the modal are three buttons: "Enter customer username", "Add Access", and "Create link".

Рис. 7. Модальне вікно встановлення інформації щодо теки

Після введення даних щодо діаграми та натискання на кнопку «Submit» користувача буде перенаправлено на сторінку редагування ціннісної пропозиції (рис. 9). Користувач також може перейти на сторінку

редагування існуючої діаграми ціннісної пропозиції натиснувши кнопку «Open».

Користувач може редагувати інформацію щодо наявних діаграм ціннісної пропозиції, а також керувати доступом до діаграми натиснувши кнопку «Edit». Користувачу буде відкрите модальне вікно встановлення інформації щодо теки (рис. 7). Наявна інформація щодо діаграми ціннісної пропозиції буде заповнена автоматично. Після зміни інформації щодо діаграми та натискання кнопки «Submit» інформацію буде змінено. Для створення посилання доступу до діаграми слід обрати бажаний рівень доступу у списку режимів досупу (рис. 8).

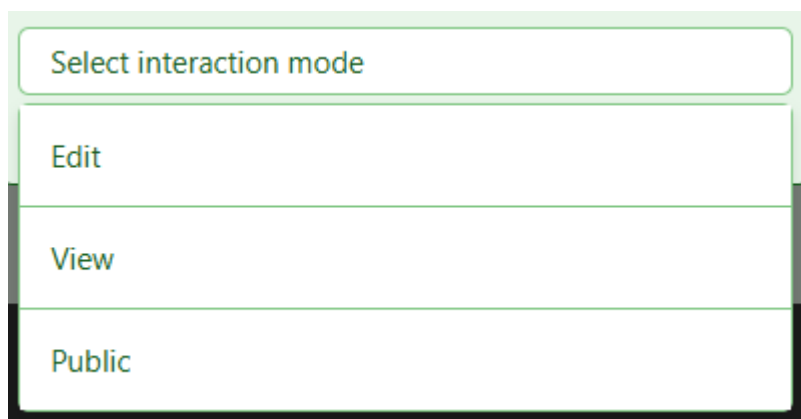


Рис. 8. Список режимів доступу до діаграми

Якщо було обрано рівень доступу «Public», вказувати логіни користувачів не потрібно. У випадку коли було обрано рівень доступу «Edit» або «View», у модальному вікні встановлення інформації щодо теки слід вказати логіни користувачів, які матимуть вказаний рівень доступу до діаграми. Після введення логіну користувача, слід натиснути на кнопку «Add access». Після додавання переліку користувачів, користувач може натиснути на кнопку «Create link» для створення посилання для доступу до діаграми. Для видалення діаграми ціннісної пропозиції на сторінці управління діаграмами ціннісної пропозиції у теці слід натиснути кнопку «Delete» біля відповідної діаграми.

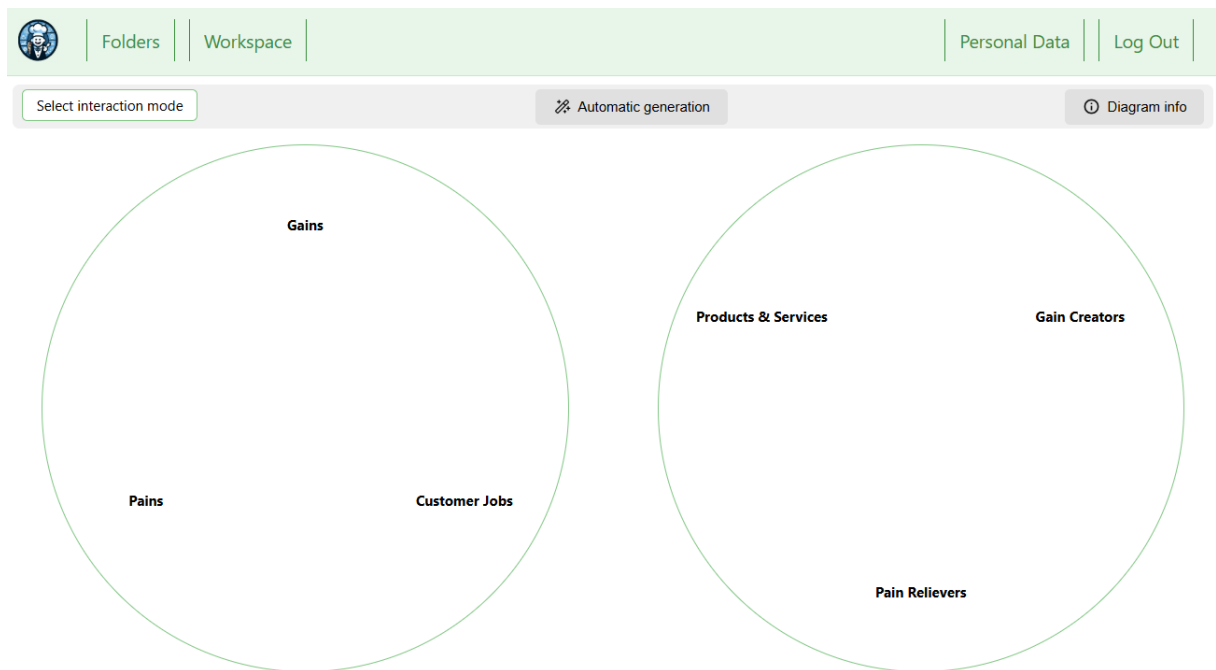


Рис. 9. Сторінка створення та редагування діаграми

5. Процедура створення та редагування діаграми ціннісної пропозиції

На сторінці створення та редагування ціннісної пропозиції (рис. 9) користувач може обрати режим взаємодії із діаграмою у списку вибору режиму взаємодії. Користувач може обрати режим додавання елементів діаграми, виделення або редагування.

При обраному режимі додавання елементів, користувач може натиснути на сектор діаграми, до якого користувач бажає додати елемент діаграми. Буде відкрите модальне вікно встановлення інформації щодо діаграми (рис. 10). Після введення вмісту елементу діаграми та натискання кнопки «Submit» елемент буде додано. Для зміни елементу ціннісної пропозиції можна натиснути на нього у режимі редагування елементів. Модальне вікно буде відкрите із заповненою інформацією щодо елементу. Для збереження змін слід натиснути на кнопку «Submit». Модальне вікно також містить простір для редагування тегів елементу. Щоб додати тег слід ввести його назву у відповідне вікно та натиснути кнопку «Add tag». Якщо вказаний тег вже був доданий до діаграми, буде збільшено показник

відповідності тегу до діаграми. Для видалення тегу треба натиснути на позначку видалення на позначці тегу.

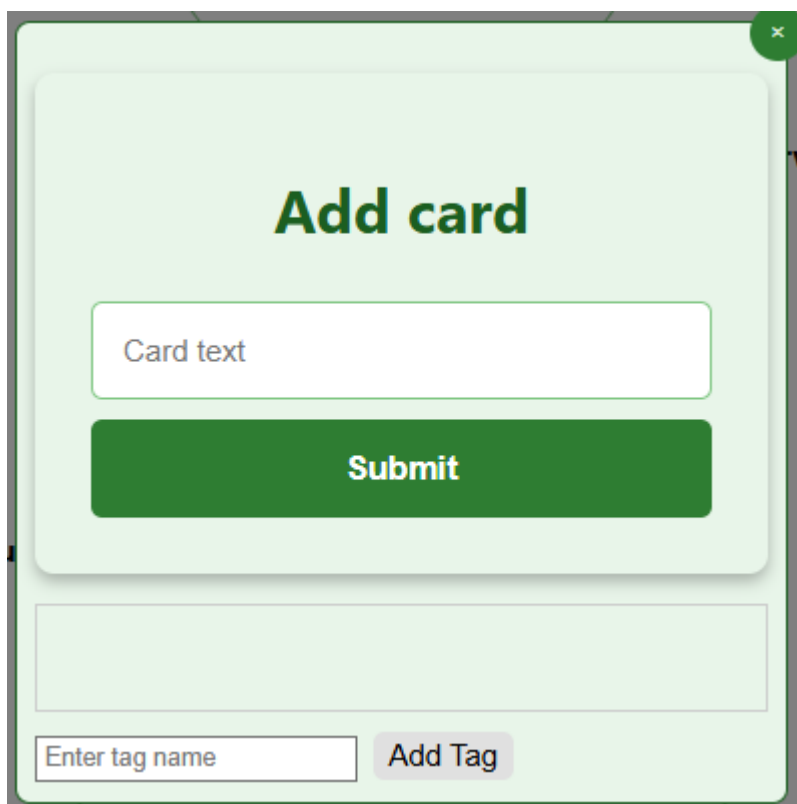
A modal window titled "Add card" with a green background. It contains a text input field labeled "Card text", a green "Submit" button, a larger empty text area, and at the bottom, a text input field labeled "Enter tag name" next to a grey "Add Tag" button. A close button (X) is in the top right corner.

Рис. 10. Модальне вікно встановлення інформації щодо елемента діаграми

Елементи діаграми можна переміщувати, встановивши режим переміщення елементів у списку режимів взаємодії. Елементи можуть бути переміщені лише в межах сектору, де вони були створені. У режимі видалення елементів діаграми, при натисканні на елемент діаграми його буде видалено.

6. Процедура управління інформацією користувача

Щоб переглянути інформацію щодо поточного користувача, на верхній панелі навігації для авторизованих користувачів можна обрати сторінку «Personal data». Користувач буде направлений на сторінку управління власною інформацією (рис. 10).

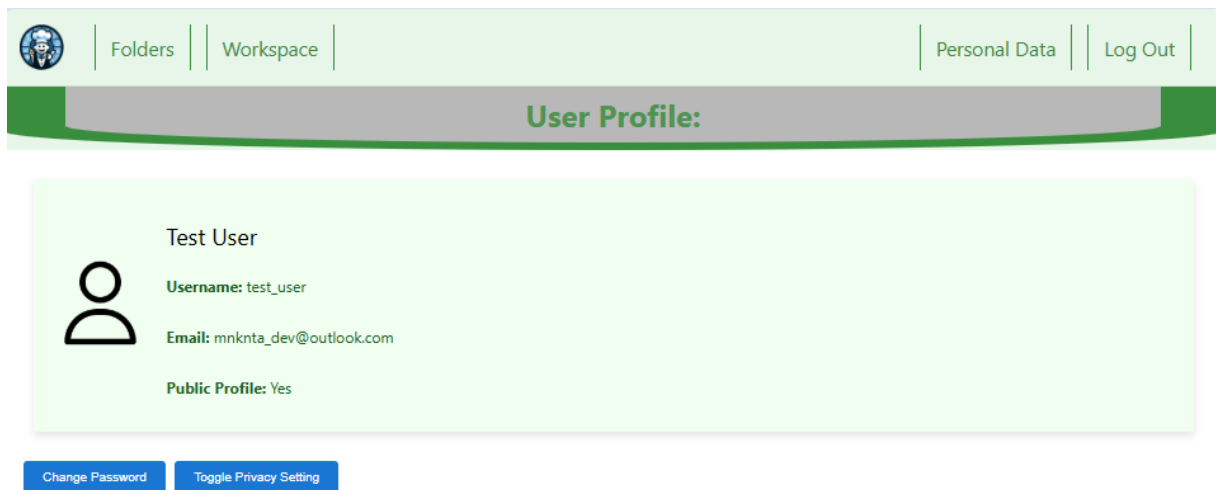


Рис. 11. Сторінка управління інформацією користувача

Для зміни паролю користувача, можна натиснути на кнопку зміни паролю. Буде відкрите модальне вікно для зміни паролю (рис. 12). Після введення поточного паролю, старого паролю та підтвердження правильності введення паролю, користувач може натиснути на кнопку «Submit» для внесення нового паролю.

The image shows a modal window titled 'Change Password' with a green border and a close button in the top right corner. Inside the modal, there are three input fields: 'Current password', 'New Password', and 'Repeat new password'. Below these fields is a large green button labeled 'Submit'.

Рис. 12. Модальне вікно зміни паролю