

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

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

Індивідуальний дослідницький проєкт

**на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інтегровані інформаційні системи»
спеціальності 126 «Інформаційні системи та технології»
на тему: « Система підтримки навчального процесу університету»**

Виконав:

студент IV курсу, групи ІА-82
Березовський Віталій Віталійович

Керівник:

Доцент кафедри ІСТ, к.ф.-м.н.

Жереб Костянтин Анатолійович

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

Студент _____

Київ – 2022 року

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

Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАВДАННЯ

на індивідуальний дослідницький проєкт студенту

Березовському Віталію Віталійовичу

1. Тема проєкту «Системи підтримки навчального процесу університету», керівник проєкту Жереб Костянтин Анатолійович, доцент кафедри ІСТ, к.ф.-м.н
2. Термін подання студентом проєкту: 15 червня 2022 року
3. Вихідні дані до проєкту: Веб-застосунок для підтримки та контролю навчального процесу у вищих навчальних закладах, який надає змогу слідувати за усіма деталями навчання.
4. Зміст пояснювальної записки: призначення та галузь, технічні характеристики, огляд та аналіз існуючих рішень, розробка UML діаграм програмного додатку, програмна реалізація.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) структурна схема бази даних, діаграма варіантів використання, діаграма послідовностей, структурна схема.
6. Дата видачі завдання 1 грудня 2021 року

Календарний план

№ з/п	Назва етапів виконання індивідуального дослідницького проєкту	Термін виконання етапів проєкту	Примітка
1	Призначення та галузь застосування	13.03.2022	
2	Огляд існуючих рішень	30.03.2022	
3	Вибір технологій для розробки системи	10.04.2022	
4	Розробка архітектури веб-додатку	18.04.2022	
5	Розробка UML діаграм програмного додатку	01.05.2022	
6	Розробка веб-додатку	01.06.2022	
7	Оформлення документації проєкту	01.06.2022	

Студент

Віталій БЕРЕЗОВСЬКИЙ

Керівник

Костянтин ЖЕРЕБ

АНОТАЦІЯ

Березовський В.В. Системи підтримки навчального процесу університету. КПІ ім. Ігоря Сікорського, Київ, 2022.

Проект містить 60 с. тексту, 17 рисунків, 1 таблицю, посилання на 15 літературних джерел, та 4 конструкторських документів.

Ключові слова: навчальний процес, система підтримки, веб-додаток, автоматизація, клієнт-серверна архітектура.

Об'єктом розробки є система підтримки навчального процесу університету.

Метою розробки є створення веб-застосунку , який зможе оптимізувати та полегшити навчальний процес у вищих навчальних закладах. Dodatok є чудовим рішенням для співробітників деканату вищого навчального закладу, оскільки вони отримають автоматизований інструмент для швидкого адміністрування, що дозволяє миттєво обчислювати великі обсяги інформації, вивчати статистику та оброблювати великі обсяги даних.

SUMMARY

Berezovsky V.V. University educational support systems. Igor Sikorsky KPI, Kyiv, 2022.

The project contains 60 pages. text, 17 figures, 1 table, references to 15 literature sources, and 4 design documents.

Keywords: learning process , support system, web application, automation, client-server architecture.

The object of development is a system to support the educational process of the university.

The aim of the development is to create a web application that can optimize and facilitate the initial process in higher education. The application is a great solution for the staff of the dean's office of the higher education institution, as they will receive an automated tool for fast administration, which allows you to instantly calculate large amounts of information, study statistics and process large amounts of data.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер ексем.	Примітка
1			Документація загальна			
2						
3			Знову розроблена			
4						
5	A4	IA82.020БАК.003 ПЗ	Пояснювальна записка	61		
6	A3	IA82.020БАК.003 Д1	Система підтримки	1		
7			навчального процесу			
8			університету. Структурна			
9			схема бази даних			
10						
11	A3	IA82.020БАК.003 Д2	Система підтримки	1		
12			навчального процесу			
13			університету. Діаграма			
14			варіантів використання.			
15						
16	A3	IA82.020БАК.003 Д3	Система підтримки	1		
17			навчального процесу			
18			університету. Діаграма			
19			послідовностей.			
20						
21	A3	IA82.020БАК.003 Д4	Система підтримки	1		
22			навчального процесу			
23			університету. Структурна			
24			схема.			
25						
26						
27						
28						

IA82.020БАК.003 ТП

Зм.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Березовський В.В.			Система підтримки навчального процесу університету. Відомість проекту	Літ.	Аркуш	Аркушів
Керівн.		Жереб К.А.				Т	1	1
						КПІ ім. Ігоря Сікорського		
Затв.						Група IA-82		

**Пояснювальна записка
до індивідуального дослідницького проєкту на
тему:
«Система підтримки навчального процесу
університету»**

Київ – 2022 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	5
ВСТУП.....	6
1 ПРИЗНАЧЕННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	8
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	9
2.1 Система «Alma»	9
2.3 Система «Fedena».....	13
2.4 Система « Campus Cafe».....	14
2.5 Порівняння систем	15
3 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ.....	18
3.1 Мова програмування JavaScript	18
3.2 Веб фреймворки JavaScript.....	20
3.3 Css фреймворк JavaScript	23
3.3.1 Bootstrap	24
3.3.2 Bulma	25
3.3.3 Tailwind	26
3.4 Серверна частина. Firebase.....	27
3.4.1 Безкоштовний для початку	28
3.4.2 Швидкість розробки	29
3.4.3 Усе в одній платформі	29
3.4.4 Працює від Google	30
3.4.5 Фокус на розробці фронтенду	30
3.4.6 Безсерверність	31
3.4.7 Машинне навчання	31
3.4.8 Моніторинг помилок	32

					IA82.020BAK.003 ПЗ			
Зм.	Лист	№ докум.	Підпис		Системи підтримки навчального процесу університету. Пояснювальна записка	Літ.	Арк.	Аркушів
Розробив	Березовський В.В.					Т	2	62
Перевірив	Жереб К.А.					КПШ ім. Ігоря Сікорського Група ІА-82		
Затв.								

3.4.9 Резервне копіювання	32
4 РОЗРОБКА АРХІТЕКТУРИ ВЕБ-ДОДАТКУ	33
4.1 Що взагалі таке архітектура веб-застосунків?	33
4.2 Компоненти інфраструктури архітектури веб-додатків	34
4.2.1 Інтерфейс користувача	35
4.2.2 Веб-сервер.....	35
4.2.3 Сервер бази даних	35
4.2.4 DNS.....	36
4.2.5 Проміжне програмне забезпечення для обміну повідомленнями	36
4.2.6 Балансувальник навантаження	37
4.2.7 Кеш	37
4.2.8 CDN	37
4.3 Як працюють сучасні веб-програми?.....	38
4.4 Типи архітектури веб-додатків	39
4.4.1 Архітектури фронтенду.....	39
4.4.1 Архітектури бекенду.....	40
4.5 Архітектура веб-додатків: Найкращі практики розробки.....	41
5 РОЗРОБКА ВЕБ-ДОДАТКУ	44
5.1 Створення проекту	44
5.2 Налаштування Vue Router	45
5.3 Налаштування Firebase	46
5.4 Налаштування Vuex	48
5.5 Створення компонентів	50
5.6 Реєстрація та вхід до кабінету.....	52
5.7 Користувачі.....	53

5.7.1	Адміністрація	54
5.7.2	Вчителі	55
5.7.3	Студенти	56
5.8	Адаптивність.....	57
ВИСНОВКИ.....		59
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних.

JS – JavaScript

ES – ECMAScript

AJAX – Asynchronous JavaScript and XML

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

JSON - JavaScript Object Notation

PDF - Portable Document Format

UI - User interface

UX - User experience

HTTP - Hypertext Transfer Protocol

HTTPS - Hypertext Transfer Protocol Secure

IP - Internet Protocol,

TCP - Transmission Control Protocol

AMQP - Advanced Message Queuing Protocol

API - Application Programming Interface

REST - Representational state transfer

URL - Uniform Resource Locator

ОС – операційна система

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		5

ВСТУП

Протягом усієї історії людства люди ставали розумнішими, сильнішими і освіченішими. Знання передавалися через книги, перекази історій та іншими шляхами. Але були й випадки, коли важливі джерела для історій могли загубитися і набути нечитаного вигляду. Завдяки вченим ми змогли вигадати поняття інтернет, яке перевернуло історію. Так люди стали швидше і зручніше здобувати знання, знаходити необхідну інформацію та багато іншого. Здавалося б, що все йде своєю чергою та завдяки інтернет джерелам люди зможуть самі здобувати знання. Але це виявилось правдою не на сто відсотків. Завжди цінувалися суворі вчителі, лектори, чиї історії та знання могли змінювати свідомості людей докорінно. Однак у сучасному світі така система освіти зазнала невеликого краху, у світі з'явилися важковиліковні хвороби в деяких країнах почалися війни і вести звичну систему освіти стало складніше. На допомогу прийшли онлайн заняття. Однак і тут не все так добре. Стало складно контролювати навчальний процес, стежити за всіма його учасниками, оформляти документи та тримати все в одній єдиній системі.

Саме тому ми вирішили створити систему контролю освітнього процесу університету, яка враховуватиме недостатній функціонал уже наявних аналогів на просторах інтернету.

Об'єктом дослідження є система підтримки навчального процесу.

Предметом є веб-додаток для підтримки навчального процесу з реалізацією на основі фреймворку Vue.js та платформи Firebase

Метою даного індивідуального дослідницького проєкту є розробка веб застосунку з назвою «Campus+», який вирішує проблематику існуючих аналогів. Ідея полягає у зручній комунікації між представниками навчального процесу, детальній інформації про групи, студентів вчителів, їх предмет, посиланні на онлайн заняття та решти інформації, необхідної для усього процесу.

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		6

В ході роботи над індивідуальним дослідницьким проєктом було виконано наступні задачі:

- огляд і порівняння існуючих систем для системи контроль навчального процесу;
- вибір технологій для розробки системи;
- розробка архітектури веб-додатку;
- розробка структурної схеми процесів у додатку;
- розробка UML- діаграм системи;
- дизайн та розробка застосунку.

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		7

1 ПРИЗНАЧЕННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Програмне забезпечення, розроблене під час виконання індивідуального дослідницького проєкту, дозволить автоматизувати роботу адміністрації університету та викладачів, тобто – зручно контролювати навчальний процес закладу. Додаток дозволяє легко контролювати успішність, активність та результативність студентів. Це можливо завдяки функціональності цього веб-додатку та його зручному програмному інтерфейсу. А також повністю підконтрольний вищезгаданим органам управління та представникам вузу.

Розроблений проєкт може бути використаний для контролю та забезпечення якісної освіти студентів та представників вищих навчальних закладів. Система дозволяє легко налаштувати та надати повну інформацію всім учасникам навчального процесу. Викладачі та лектори мають можливість легко керувати навчальним процесом, створювати лекції, виставляти оцінки та додавати інформацію про свої предмети. Додаток є чудовим рішенням для працівників деканату вищого навчального закладу, оскільки вони отримають автоматизований інструмент для швидкого адміністрування, який дозволяє миттєво обчислювати великі обсяги інформації, вивчати статистику та обробляти великі обсяги даних. Весь процес простий, оскільки інтерфейс зручний і працює на будь-якому сучасному пристрої зі звичайним браузером та доступом до Інтернету.

Висновки до розділу 1

У даному розділі розглянуто та поставлено задачу на розробку веб-застосунку, який має свою актуальність та відмінність від вже уснуючих аналогів у мережі інтернету.

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		8

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Сьогодні на ІТ-ринку вже є широкий асортимент програмних продуктів, що дозволяють частково автоматизувати процес навчання у вищих навчальних закладах. Проте важливо зазначити, що кожен програмний продукт має свій підхід до проблеми управління та є рішенням для конкретного навчального закладу. Незалежно від платформи, кожна система управління пропонує унікальний спосіб реалізації функціональних можливостей, які можна адаптувати до будь-якої платформи. Тому дуже важливо проаналізувати ринок існуючого програмного забезпечення, а не обмежувати лише його готовими рішеннями.

У рамках огляду розглядаються різноманітні програмні продукти для вирішення питань автоматизації навчальних процесів студентів та систем управління в усьому закладі. Крім того, для детального аналізу та вивчення використовуються Alma, Classter, Fedena, Campus Cafe та інші популярні програмні системи[1]. Слід зазначити, що кожен з цих інструментів використовує унікальні методи та прийоми для досягнення базової функціональності.

2.1 Система «Alma»

Alma — це хмарне рішення для управління університетом або іншим начальним процесом, розроблене для закладів будь-якого розміру в освітньому секторі. Основні функції допомагають головним користувачам слідкувати за процесом, створювати спеціальні звіти, керувати шляхами реєстрації та вступу до вибраного учбового закладу, а також відстежувати комунікацію між її представниками. За допомогою цієї системи користувачі можуть відстежувати усі головні новини учбового процесу, отримувати повідомлення будь-якого типу у зрічній системі. Це зображено на рисунку 2.1.

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		9

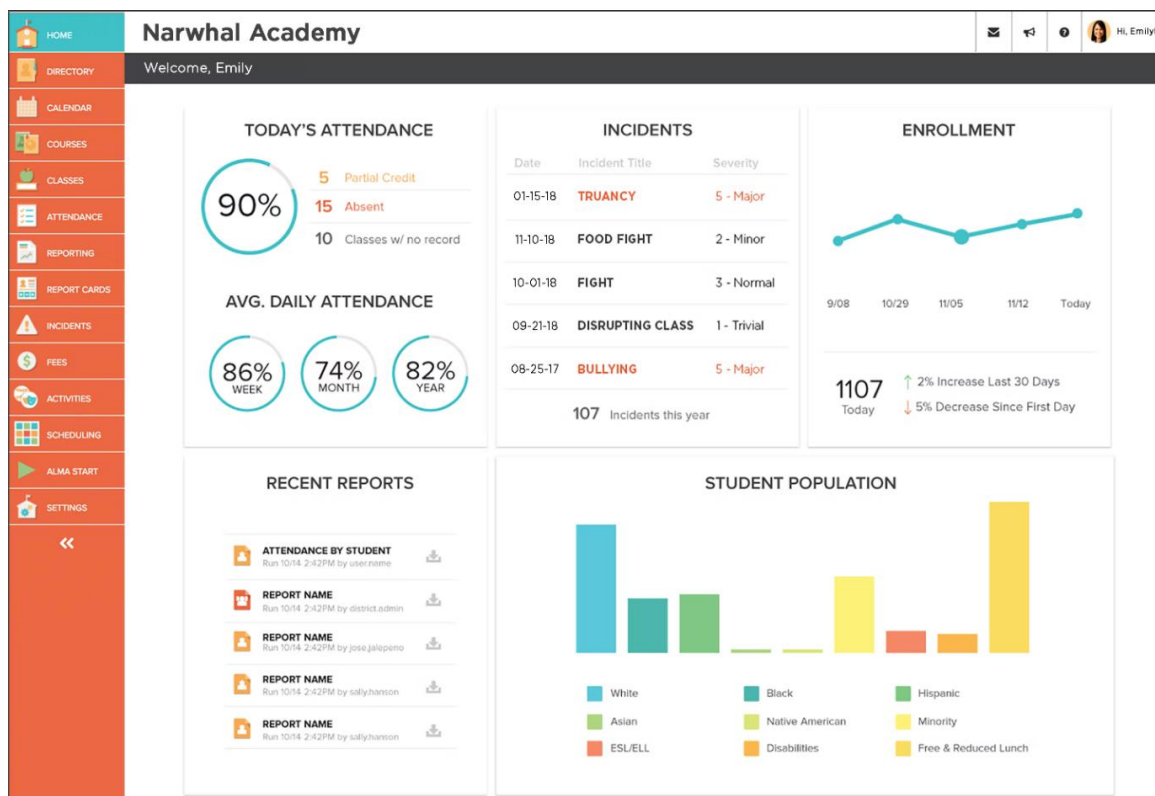


Рисунок 2.1 – Асистент для початку роботи та ознайомлення з додатком

Рішення дозволяє реєструвати відвідування та налаштовувати журнали оцінок. Користувачі можуть створювати профілі студентів, а батьки та вчителі можуть переглядати оцінки своїх студентів, відвідування та примітки. Сервіс також дозволяє освітнім закладам керувати запланованими заняттями та відвідуваннями.

Додаток дозволяє користувачам розробляти персоналізовані курси для різних здібностей і автоматично узгоджувати курси з календарем. Рішення дозволяє адміністрації спілкуватися з персоналом, студентами через дошки оголошень, електронну пошту, групові повідомлення та портали для лекторів і студентів. Адміністратори університету можуть налаштувати доступ на основі ролей до створених звітів. Alma пропонує послуги на основі підписки та підтримку клієнтів за допомогою телефону, електронної пошти та онлайн-запитів.

2.2 Система «Classter»

Classter — універсальна студентська інформаційна система для вищої освіти. Додаток який надає змогу звільнитися від розрізненої ІТ-інфраструктури. Він пропонує модульну хмарну платформу, яка є економічно ефективним рішенням для управління освітніми закладами будь-якого розміру. Використання Classter є зручним для створення комплексних, повністю контрольованих та безпечних порталів та програм для всіх користувачів.

Програма надає викладачам, учням та іншим особам доступ до всієї особистої інформації та виконує всі онлайн-процеси за допомогою порталів Classter, що легко налаштовуються.

Повністю конфігурована архітектура: ландшафт, що постійно змінюється, навчання у вищій освіті вимагає гнучкої та надійної ІТ-інфраструктури. Classter пропонує ІТ-менеджерам хмарне середовище, що повністю конфігурується, для управління університетами та коледжами. Об'єднана функціональність знижує складність обслуговування та синхронізації розрізнених систем, забезпечуючи єдиний досвід роботи у різних відділах та місцях.

Що стосується цифровізації процесів прийому та зарахування: Classter пропонує ефективний безпаперовий процес прийому, який допомагає організаціям вищої освіти скоротити кількість адміністративних завдань, що повторюються, і покращити якість прийому абітурієнтів. Прийомні комісії можуть легко створювати робочі процеси зарахування, керувати приймальними документами та згодами, а також отримувати доступ до даних у режимі реального часу та комплексних звітів за заявами. Додаток надає змогу використовувати ефективні цифрові процеси для вчителів, а саме: спеціальний портал для вчителів, що пропонує просте у використанні та інтуїтивно зрозуміле середовище для управління уроками, оцінками та відвідуваністю.

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		11

Викладачі можуть створити захоплююче навчальне середовище, використовуючи функціональність академічних модулів Classter та готову інтеграцію з популярними сервісами.

Також додаток має зручний центр керування, який ми можемо побачити на рисунку 2.2

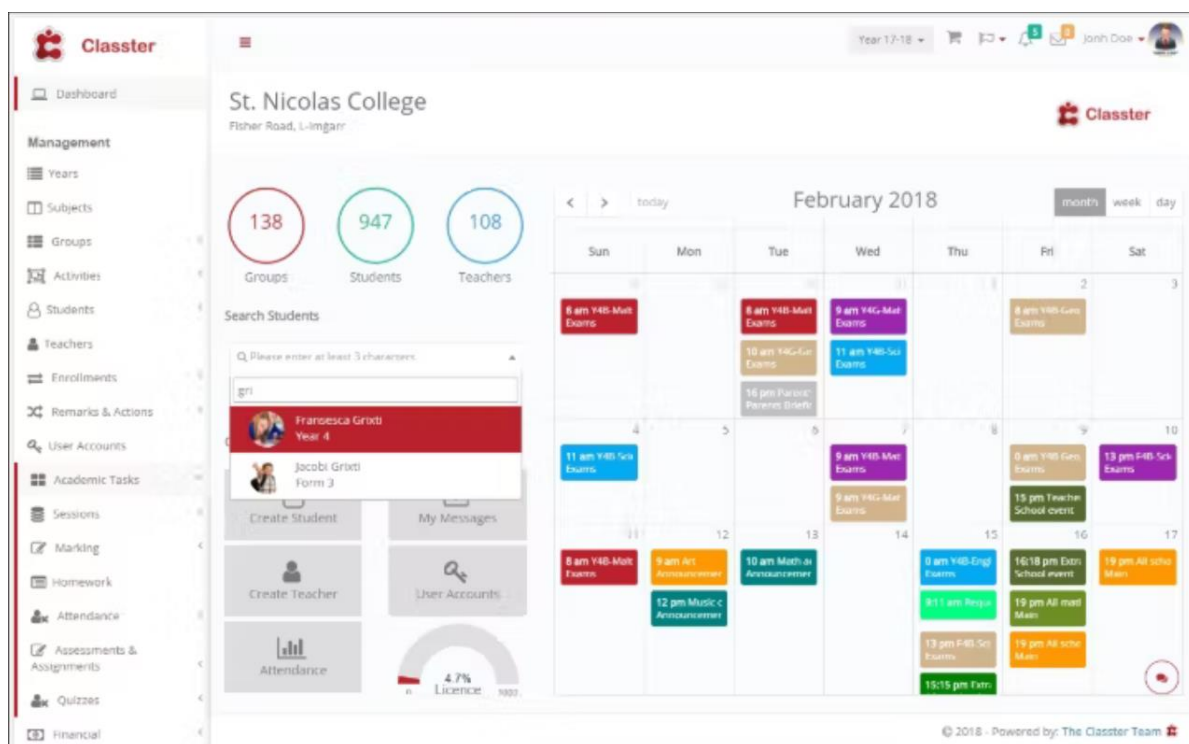


Рисунок 2.2 – Центр керування Classter

Не слід забувати про розширення можливостей учнів та підвищення залученості: створення захоплюючого середовища для учнів та забезпечення персоналізованих зв'язків у процесі навчання. Classter дозволяє організаціям вищої освіти вести централізовану БД студентів, що входять до певної системи та легко керувати низкою академічних функцій, необхідних для навчального процесу використаного закладу, таких як реєстрація, створення навчальних програм, звітність за оцінками та платежі.

2.3 Система «Fedena»

Fedena - це хмарне рішення для управління учбовим закладом від Foradian Technologies, яке включає кілька модулів для управління даними учнів, розкладом, оцінками, новинами, кадрами, фінансами і багатьом іншим.

Приладова панель рішення може бути налаштовується за допомогою "приладових панелей", щоб дати користувачам можливість відразу побачити конкретну інформацію, наприклад дані про відвідуваність, терміни складання іспитів, розклад іспитів, повідомлення, щоденні графіки, входи та виходи.

Fedena - це хмарне програмне забезпечення для планування ресурсів підприємства, яке дозволяє школам та університетам керувати електронним навчанням та повсякденними адміністративними завданнями за допомогою єдиної платформи. Система також має унікальну автоматизовану систему відвідування (рисунок 2.3).

Student	Sun 01	Mon 01	Tue 01	Wed 01	Thu 01	Fri 01	Sat 01	Sun 01	Mon 01	Tue 01	Wed 01	Thu 01	Fri 01	Sat 01	Sun 01	Mon 01	Tue 01
Adele		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Annie		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Alice		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Ann		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Aash		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Betty		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Davi		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Bab		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Bobby		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Camely		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Dabi		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Dina		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X
Ewator		X	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		X	X

Рисунок 2.3 – Відвідуваність Fedena

Стандартні функціональні можливості, плагіни та інтеграції Fedena, включаючи систему керування навчанням Moodle, мають відкритий вихідний код, тому їх можна налаштовувати при необхідності відповідно до робочих процесів та вимог.

Інші модулі рішення включають реєстрацію заявок, експорт даних, інтеграцію з Tally, інтеграцію з електронною поштою, генератор розкладу, створення форм та тестування. Також цей додаток пропонує послуги в рамках річної підписки, яка включає підтримку телефоном, електронною поштою та онлайн-ову базу знань.

2.4 Система « Campus Cafe»

Campus Cafe - це хмарна інформаційна система для студентів, яка допомагає керувати функціями прийому, студентів, бізнес-офісу та розвитку випускників. Основні функції включають відстеження заяв, кандидатів та статусів, створення профілю кандидата, управління контентом та створення звітів та іспитів. Він розроблений для задоволення потреб малих та середніх коледжів та університетів. Інтерфейс програми зображено на рисунку 2.4 .

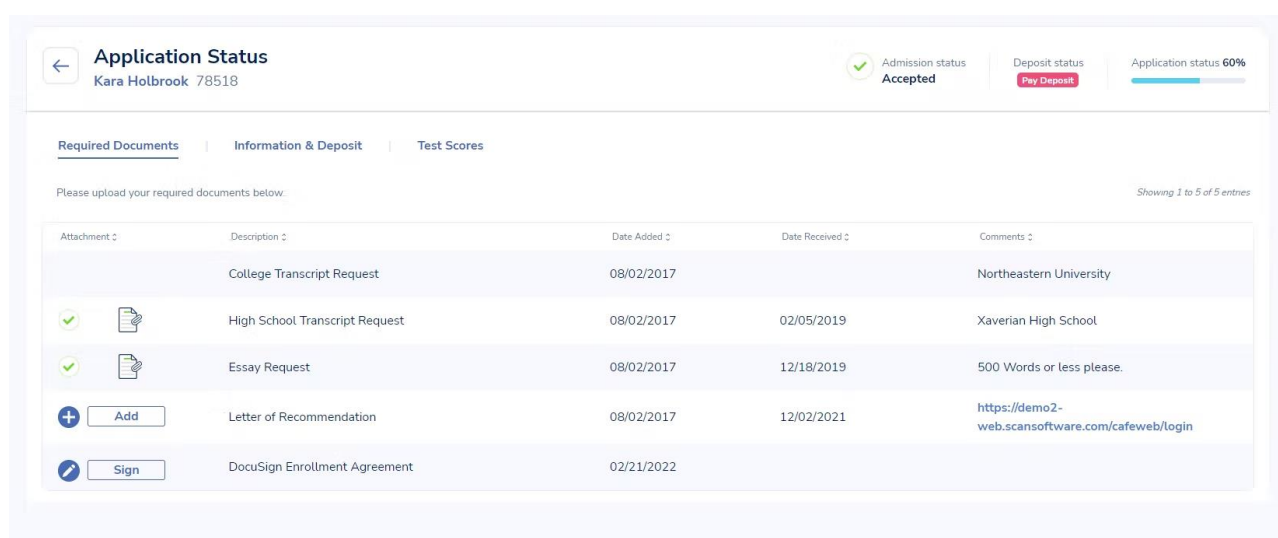


Рисунок 2.4 – Інтерфейс програми Campus Cafe

Campus Cafe може використовуватися для створення єдиного запису для всього життєвого циклу студента, від абітурієнта до випускника, та керує робочими процесами у різних галузях університету. Він також опрацьовує такі операційні аспекти, як управління контактами, складання розкладу, академічні завдання та управління відвідуваністю для середніх професійних шкіл. Також хочу зазначити, що з інших систем він має дуже сучасний дизайн інтерфейсу, ось наприклад сторінка профілю студента, зображена на рисунку 2.5.

Transaction Date	A/R Code	Description	Amount	Running Balance	Semester
01/01/2017	BF	Balance Forward		\$0.00	
07/23/2021	IN	TUITION FULL TIME	\$10,000.00	\$10,000.00	202110 - Fall 2020

Рисунок 2.5 – Профіль студента Campus Cafe

Campus Cafe має власні вбудовані фінансові модулі. Він пропонує інтеграцію з програмним забезпеченням, яке широко використовується в навчальних закладах, таких як Moodle, CollegeNet, CollegeBoard та Easy-League. Рішення базується на технології Java та сумісне з веб-браузерами різних платформ.

2.5 Порівняння систем

Проведення порівняльної характеристики розглянутих існуючих системних аналогів дало можливість виділити їх головні переваги та недоліки, що відображено в таблиці 2.1.

Таблиця 2.1 – Порівняльна характеристика програмних додатків

Назва програмного застосунку	«Alma»	«Classter»	«Fedena»	«Campus Cafe»
Користувацький інтерфейс	Інтерфейс є досить складним для користування	Зрозумілий та практичний інтерфейс.	Зрозумілий та легкий інтерфейс	Не складний інтерфейс, для роботи з певними функціями потрібна невелика підготовка
Допомога для користувача	Користувачам програми знадобиться постійна підтримка з боку експертів та технічного персоналу	Є вказівки для початку роботи з додатком	Є інструкції, вбудовані безпосередньо в інтерфейс користувача	Присутня інструкція користувача
Функціонал для студентів	Присутня можливість доступу та глядацький функціонал	Багато можливостей взаємодії з іншими представниками системи	Права доступу наявні лише у одного користувача (у викладача)	Доступ студентів до системи досить переконливий

Робота основних функцій без доступу до Інтернету	Неможливо, оскільки вам потрібно підключитися до зовнішньої бази даних на сервері	Це неможливо, оскільки це веб-система	Можливо, тому що створено локальне сховище даних	Можлива, тому що створено локальне сховище даних
Можливість взаємозв'язку зі студентами	Відсутня	Наявна	Відсутня	Відсутня

Висновки до розділу 2

Було розглянуто головні переваги, які можна наслідувати із кожного додатку та недоліки яких слід уникнути. Система Alma надає методи автоматичного оповіщення студентів про перевищення межі проходження, що має бути точно реалізовано в розробленій системі. Обов'язково уникайте основного недоліку Alma, який використовує занадто складні інтерфейси та дорогі компоненти. У програмі Classter немає очевидних недоліків, але це цікавий механізм статистики за допомогою фільтрів. Корисним рішенням, яке можна запозичити з програмного забезпечення відвідуваності, є докладні інструкції користувача.

Відстеження відвідуваності має спеціальну функцію, яка дозволяє вчителю надсилати сповіщення своїм учням. Основним недоліком двох згаданих вище програм є те, що розроблена система є неприйнятною: вони не використовують зовнішню базу даних та призначені для одного користувача.

3 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ

3.1 Мова програмування JavaScript

JavaScript – це мова програмування високого рівня , найпопулярніша у сфері розробки програмного забезпечення. Вона є впровадженням стандарту ES (ES2021 – остання версія , стандарт на момент написання статті). Крім JS, є й інші реалізації ES, такі як ActionScript, JScript і QtScript. Однак вони не досягнули такого успіху, тому майже всі вони вважаються застарілими. JS була створена як мова , що дозволяє додавати інтерактивність. Сторінки веб-сайту (анімації, динамічні зміни вмісту тощо). З часом ця мова набула популярності і зараз є домінуючою мовою для клієнтської веб-розробки. Її підтримують усі сучасні браузері. Історично, програми написані цєю мовою отримали назву «скрипти». JS інтерпретований, повний за Тюрінгом і схожий на Сі синтаксис. Це об'єктно-орієнтована мова, яка використовує прототипи.

До прибуття ES 6 у JS мові взагалі була відсутня концепція об'єктно-орієнтованих базових класів парадигми. Однак, незважаючи на те, що вони з'явилися, JS все ще існував орієнтованим на прототип. Класи є просто «синтаксичним цукром» для обробки прототипів. JS забезпечує автоматичне керування пам'яттю і емулює комп'ютер з необмеженою пам'яттю за допомогою системи «збору сміття». Оскільки JS є перш за все мовою для браузерів, то вона є в кожному браузері. Реалізує все необхідне для запуску сценаріїв JS. Може включати:

- ядро мови. Це частина, описана стандартом ES. Вона стосується інтерпретаторів, стандартних функцій, систем типів, операторів, тощо;
- об'єктна модель документа. Ця функція недоступна в ES, але вона характеризується стандартом , який описує та розробляє компанія W3C. Вона тут належить до здатності JS взаємодіяти з відтвореними сторінками веб-сайту у вигляді HTML-розмітки;

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		18

– об'єктна модель браузера. Ця функція також стандартизована W3C, а не ES. Сюди входять периферійні функції. Взаємодія браузер-користувач. Це геолокація, використання файлів cookie та локальне зберігання даних браузера, асинхронні запити (AJAX), керування історією перегляду веб-користувача, тощо.

Як бачите, JS сильно залежить від середовища в якому вона безпосередньо виконується. У зв'язку з цим також варто відзначити, що основні недоліки цього методу є мірою випуску нових версій стандартів ES або W3C . Розробники зображень також не можуть скористатися перевагами всіх нових функцій. Необхідно зачекати, поки основне середовище виконання (браузер) не оновить власне сам драйвер.

Драйвер – це термін, який замінив інтерпретатори, коли веб-додатки стали занадто складними для інтерпретації та розробників. Керування JS починало використовувати більш складні методи оптимізації виконання коду. Оновлення двигуна для підтримки нової стандартної версії не є швидким. Процес може зайняти роки. Для вирішення цієї проблеми розроблено кілька різних компіляцій коду, написаного в новіших версіях JS Таким способом можуть скористатися розробники, щоб реалізувати оновлені можливості в новому стандарті. Але вони усе ще є на стадії розробки , тому потрібно їх ще очікувати. Такі програми називають «перекладачами».

Мова JS відома своєю простою системою типів. Вона складається з трьох примітивів (число, логічне значення, рядок) і трьох типів об'єкти (об'єкти, масиви, функції). Мовний набір динамічний (тип змінної визначається під час виконання програми) та слабкий . Неявні змінні можуть бути зведені до різних типів, навіть якщо є втрати інформації , неявні змінні не потребують вказувати тип під час їх створення. Історично програмне забезпечення розроблялося однією мовою JS та завжди мало модульну архітектуру. «Модуль» у цьому сенсі є частина логіки, яка структурована та представлена в окремому файлі коду. Існують зв'язувальні модулі (імпортування модуля) в код іншого модуля. Імпорт, з можливістю використання явно наданих даних (експортується) у зв'язаний модуль за

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		19

класичним визначенням шаблонів. Дизайн можна описати як «фасад» шаблону оформлення. Окремо модуль може містити будь-яку кількість логіки і повинен експортувати один або більше об'єктів. У цьому випадку об'єкт може бути змінною, константою або функцією. Донедавна JS був не єдиним в екосистемі та архітектурі з використанням модулів. Це призвело до різних рішень сумісних та взаємовиключних. У стандарті мови JS поняття модулів в майбутньому будуть реалізовані в механізмі JS, який має замінити всі рішення, які не сумісні зі стандартом.

3.2 Веб фреймворки JavaScript

Фреймворки надають розробникам необхідну основу для створення програм JS. Це позбавляє розробників починати роботу з нуля, використовуючи функціональні фреймворки.

Для JS вона містить набір кодових баз. Код, скомпільований конкретною бібліотекою, забезпечує функціональність, специфічну для певного типу програми, над якою ви працюєте. По суті, фреймворк визначає структуру всього додатка.

Кожен фреймворк на основі JS розроблений для різних цілей та задач. JS є послідовним вибором для веб-розробки, і багато його фреймворків засновані на ньому. Розробка веб-додатків і веб-сайтів від початку до кінця може бути невеликим завданням. Веб-фреймворки — або, точніше, фреймворки JS — використовують той факт, що кожен веб-сайт і кожний веб-додаток мають спільні характеристики та параметри. Тому слово каркас можна сприймати буквально. Будинок також спирається на каркас як загальну опору для рихтувань і форму зовнішнього каркаса. Те ж саме стосується програмних фреймворків. Фреймворки JS відображають попередньо упакований його код, який забезпечує рутинні функції програмування для полегшення розробки.

Бібліотеки JS дозволяють розробникам викликати та використовувати певні функції з коду вищого рівня. Однак розробники не можуть викликати фреймворк JS. Натомість фреймворк викликає певний фрагмент коду і використовує його заздалегідь визначеним способом. Саму структуру визначає дизайн програми. Прийняття коду з фреймворка є більш комплексним, ніж використання бібліотеки. Фреймворк надає не тільки єдине рішення, а й повний шаблон розробки. Перевагами є краща організація та більша ефективність.

Ви можете використовувати багато фреймворків JS. Але пропоную розглянути найпопулярніші у світі з точки зору клієнтської взаємодії[2].

Отже першим треба розглянути Vue.js . Він оголошує себе як «прогресивний» фреймворк JS. Назва походить від його філософії поступового набуття. У Vue.js основна бібліотека зосереджена лише на шарі перегляду, тому будь-яку додаткову функціональність потрібно впроваджувати поступово. Фреймворк використовує шаблон архітектури Model-View-Viewmodel. Цей шаблон відокремлює графічний інтерфейс від бізнес-логіки або моделі програми. Рівень моделі перегляду є середовищем перетворення, яке синхронізує дані.

Другим у цьому гучному списку є AngularJS, що підтримується Google. Він усуває поширені труднощі створення односторінкових програм (SPA). Ця структура використовує словник HTML динамічного веб-сайту. Раніше HTML можна було використовувати лише для статичного вмісту. У SPA вміст завантажується динамічно з веб-сервера, а не з веб-браузера. Таким чином, SPA працює так само, як і мобільний додаток без перезавантаження.

Завершує цю трійцю React. Фреймворк ,що розроблений компанією Facebook, який спрощує створення інтерактивних інтерфейсів користувача. Це основа React Native, суміжного фреймворка для розробки мобільних додатків. Обидва фреймворки мають односторонній потік даних, який вважається більш інтуїтивним, ніж двосторонній з прив'язуванням даних. Гарячий перезапуск — це одна з популярних функцій фреймворка React, яка дозволяє розробникам бачити зміни під час їх застосування.

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		21

Чесно кажучи, фреймворки JS є чудовим вибором, коли справа доходить до вибору мови JS. Вони економлять час і гроші. Розуміння того, що насправді таке фреймворк JS, є важливим кроком в оптимізації процесу веб-розробки. Тепер, коли ознайомлення з основними клієнтськими фреймворками завершено, потрібно вирішити, який фреймворк JS найкраще підходить для нас.

Для цього проекту, дивлячись на потреби в інтерактивності та інтерфейсі користувача з відсутності складних розрахунків було прийняте рішення використати фреймворк Vue.js, оскільки він має найвищу швидкість розрахунків, найменші вимоги до пам'яті та найменшу вагу.

Основна ідея Vue.js полягає у використанні поняття «реактивності». Оперуючи термінологією можна зауважити, що реактивна розробка програмного забезпечення є концепцією програмування, яка поширена на основі асинхронних потоків даних. В наведеному випадку це послідовна робота функцій, як відповідь на їх власні події. Основний плюс такого підходу дає змогу писати великі частини коду та використовувати у будь-яких інших таких же частинах та незалежно їх конфігурувати.

Інтерфейси, що використовує Vue.js, отримали назву «компоненти». Це об'єкти, що були створені безпосередньо самими розробниками. Вони є основними «блоками» програми. Розробник інтерфейсу має змогу створювати такі компоненти необмежену кількість разів та використовувати у різних частинах його програми. Слід зазначити, що кожен такий екземпляр має свою внутрішню систему зберігання даних. Саме через це компоненти є основною функцією перевикористання у Vue.js. Крім компонентів фреймворк також має код, який можна використовувати повторно в залежності від задачі. Ось ще кілька переваг, за якими треба використовувати саме Vue.js:

- має вбудоване кешування даних для значно більшої швидкості та легкості розробки додатків на основі цього фреймворку;

– має офіційну бібліотеку, інтегровану з інструментами розробника в браузер і дозволяє контролювати стан даних та стан дерева компонентів безпосередньо з вікна програми в браузері;

Vue.js в основному призначений для створення однієї сторінки так званого Single page application. Це означає, що коли клієнт вперше надсилає запит на сервер, то генерується відповідь з усіма файлами, необхідними для відображення повного обсягу файлів сторінки програми, а також додаткові файли та дані, які потрібні лише деяким користувачам. Функції при необхідності завантажуються окремо за допомогою технології асинхронних запитів JS (AJAX). Це значно прискорює роботу програми і слугує вузьким місцем під час першого завантаження програми у браузері.

Даний фреймворк може використовувати в собі не лише компонентний підхід. Він має низку додаткових бібліотек та споміжних систем. Вони слугують для підтримки зручної розробки на основі Vue.js.

3.3 Css фреймворк JavaScript

Почнемо з того що таке фреймворки CSS. Вони у себе включають кілька таблиць стилів CSS, які можуть використовуватися веб-розробниками та дизайнерами. Таблиці стилів можна використовувати для стандартних функцій веб-дизайну: налаштування кольору, макета, шрифтів, панелей навігації тощо. Використовуючи фреймворк CSS, користувач має готову таблицю стилів CSS, і все, що йому потрібно зробити, це написати HTML з точним класом, структурою та ідентифікатором для створення веб-сторінки. У фреймворку вже передбачені вбудовані класи для найпоширеніших елементів веб-сайту — нижні колонтитули, повзунки, панелі навігації, меню гамбургерів, макети на основі стовпців тощо. Переваги використання фреймворків CSS:

– розробники та дизайнери можуть використовувати фреймворки CSS для реалізації різноманітних додаткових функцій та візуальних елементів на веб-сайт,

таких як форм, різних кнопок, панелей навігації, сухарів і навіть чистих симетричних макетів.

– фреймворки CSS дозволяють легко створювати сайти, сумісні з різними браузерами та версіями браузера. Це зменшує ймовірність помилок під час міжбраузерного тестування.

– оскільки ці фреймворки містять готові таблиці стилів, їх використання може прискорити та спростити розробку веб-сайту. Користувачам не потрібно вникати в код CSS для виконання бажаних завдань.

– розробники можуть швидко створювати зручні та візуально привабливі користувацькі інтерфейси, які можна змінювати з нуля протягом усього проекту.

Коли ми розібрали основні переваги використання саме фреймворків для стилізації веб застосунку пропоную розганути найбільш популярні[3] та привабливі з точки зору розробки їх варіанти.

3.3.1 Bootstrap

Bootstrap був розроблений в Twitter Джейкобом Торнтоном і Марком Отто як основа для внутрішньої узгодженості інструментів, Bootstrap — це фреймворк з відкритим вихідним кодом, який включає шаблони компонентів інтерфейсу на основі CSS і JS.Bootstrap відомий пропагуванням адаптивного дизайну серед веб-розробників. Він популяризував всюди суцю концепцію мобільного дизайну та надав необхідні інструменти для її легкої реалізації. Це пов'язано з введенням сіток - поділом екрана на колонки (не видимі для кінцевого користувача).

Завдяки Bootstrap розробникам не потрібно створювати окремі проекти, щоб адаптувати сайт до менших розмірів екрана. Досить включити необхідні класи Bootstrap, і дизайн налаштується сам. Bootstrap має величезну екосистему серед інтерфейсних фреймворків. Його екосистема не має собі рівних. Він надає величезну бібліотеку макетів, тем, елементів інтерфейсу користувача, панелей, модальвів, кнопок, повідомлень, карток тощо, з яких розробники та дизайнери

можуть вибирати та реалізовувати. Крім того, Bootstrap підтримується найкращою спільнотою в галузі. Прискорення створення прототипів: за допомогою Bootstrap дизайнери можуть просто написати HTML-код, увімкнути відповідні класи CSS і досягти швидкості реагування сайту. Їм не потрібно витрачати час на звикання до несумісності браузера, позиціонування CSS тощо. Також він має підтримку Twitter: не дивно, що коли великий бізнес гравець підтримує проект з відкритим кодом, користувачі можуть бути впевнені, що він тут, щоб залишитися і завоювати велику довіру серед тих, хто знає цю галузь. Той факт, що Bootstrap створений та підтримується компанією Twitter, підтверджує його ефективність

3.3.2 Vulma

Vulma — це адаптивний фреймворк CSS з відкритим кодом на основі Flexbox. Він має вражаючий набір вбудованих функцій, які допомагають прискорити та зменшити ручне кодування CSS. Він реалізує плитки для створення сітки в стилі Metro для плавного розташування сторінок. Крім того, користувачі можуть імпортувати лише ті проекти, які вони використовують, що ще більше спрощує процес.

Враховуючи, що вихідний код Vulma можна безкоштовно завантажити, користувачі можуть розширити його функціональність на власний розсуд. Він пропонує легковажний, чистий підхід CSS (без компонентів JS) і візуально привабливі параметри за замовчуванням для початку.

Vulma стала поширеною в спільноті Laravel, що сприяло її зростанню популярності. Модульний підхід Vulma до проектування і високий рівень налаштування роблять його найкращим інструментом для розробників і дизайнерів. Його адаптивні шаблони зменшують роботу над дизайном, і він має багатий каталог компонентів на вибір – панелі навігації, таблиці, панелі, спадні меню тощо. Vulma також поставляється з шаблонами для початківців та

інтерактивними підручниками. Крім того, він має потужну спільноту Stack Overflow, яка дуже корисна для відповідей на запитання.

Модульна природа Vulma розроблена для вирішення реальних проблем, з якими можуть зіткнутися окремі розробники. Vulma розроблена для легкої навігації, і люди можуть швидко зрозуміти, як вона працює. Незалежно від браузера: веб-сайти, розроблені на Vulma, сумісні з більшістю основних браузерів. Це полегшує життя тестувальникам, особливо під час міжбраузерного тестування. Vulma, можливо, була доступна нещодавно, але нові функції додаються майже постійно. Крім того, випускаються оновлення для усунення існуючих помилок або проблем, з якими можуть зіткнутися користувачі.

3.3.3 Tailwind

Офіційна документація Tailwind CSS описує його як «службово-орієнтований фреймворк CSS» із супутніми класами для створення дизайну інтерфейсу користувача безпосередньо в розмітці користувача. Він зручно реалізує вбудовані стилі для створення приголомшливих інтерфейсів користувача без написання CSS.

Одна з найпопулярніших бібліотек CSS, Tailwind CSS, надає значні переваги для веб-дизайну. Цікаво, що Адаму Ватану (творцю Tailwind) потрібно було переконати світ, що CSS на основі утиліти краще, ніж семантичний CSS. Але зрештою, достатньо розробників повірили, що він почав використовувати Tailwind, і тепер він використовує його досить часто, щоб бути в подібному списку. Tailwind CSS поставляється зі стандартною конфігурацією, але її можна змінити за допомогою файлу tailwind.config.js. Це дозволяє легко налаштовувати стилі, теми, розділювачі, кольорні палітри тощо. Використання утиліти Tailwind, дає змогу спростити управління проектами та створити веб-сайти, які подобаються. За допомогою Tailwind користувачі можуть зменшити потребу називати занадто багато класів. Він постачається із загальними шаблонами служб

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		26

для загальних завдань: визначення та організація класів, каскадних класів тощо. Коротше кажучи, створювати користувацькі компоненти стало простіше. Жорстко кодувати не потрібно, просто скористайтеся функцією `theme()`, щоб отримати значення конфігураційного файлу.

Tailwind CSS за замовчуванням підходить для мобільних пристроїв. Прочитуючи веб-сайт: «Кожен клас сервісу в Tailwind можна використовувати умовно на різних точках зупинки, що дозволяє створювати складні адаптивні інтерфейси, не продовжуючи використовувати HTML». Результат? Легко реалізуйте адаптивний дизайн.

Для цього проекту дивлячись на умови того що , наша система буде досить велика та масштабна , а кількість сторінок та різнопланових компонентів дуже велика , було прийняте рішення використати Tailwind. Він підходить під усі наші умови , його здатність масштабуватися та залишати код чистим дуже вражає.

3.4 Серверна частина. Firebase

У сучасному світі програмування важко уявити велику програмну систему без серверної частини. Безумовно наш додаток має більший напрямок на клієнтську частину , але питання про серверну частину є дуже важливим. Пропоную розглянуту на мою думку найбільший та найпопулярніший серверний сервіс.

Firebase[4] — серверна платформа для створення веб- та мобільних додатків. За своїм змістом це набір інструментів, на які розробники можуть покладатися при створенні програм і розширенні їх відповідно до своїх потреб. Firebase розроблено для вирішення багатьох серйозних проблем для розробників: швидкого створення програм, впевненої публікації та контролю програм, а також залучення користувачів. Розробники, які покладаються на цю платформу, мають доступ до послуг, які вони повинні розробити самостійно, що дозволяє їм зосередитися на створенні надійних додатків.

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		27

Функції платформи Google Firebase включають БД, аутентифікацію, push-повідомлення, аналітику, зберігання файлів тощо. Оскільки сервіс розміщений у хмарному зберіганні, розробники можуть легко виконувати масштабування на вимогу. На даний момент Firebase є однією з найкращих платформ розробки додатків, на яку покладаються розробники в усьому світі.

Популярні програми, які використовують Firebase, мають досить гучні імена. Серед них: Alibaba, The New York Times, Todoist, Le Figaro, eBay Motors. Тож, які типи програм можна розробляти за допомогою Firebase? Платформа дозволяє користувачам розробляти такі категорії додатків: Android, iOS та Web. Сфери застосування Firebase досить широкі і включають:

- mvr – мінімально життєздатні продукти
- програми реального часу
- чат та програми для обміну повідомленнями
- оптимізація реклами на основі поведінки користувача
- спільне використання фотографій
- оптимізація утримання користувачів за допомогою машинного навчання
- бізнес-додатки

3.4.1 Безкоштовний для початку

Firebase безкоштовний для початку роботи та дозволяє користувачам входити за допомогою своїх облікових записів Google. Плани Spark безкоштовні та пропонують багато функцій, які допоможуть розробникам розпочати роботу. Залежно від своїх зростаючих потреб вони можуть вибрати план Blaze. Можливість безкоштовно розпочати роботу — це, безумовно, чудова можливість, яку пропонує Firebase, і одна з причин, чому вона настільки популярна. Плани Spark мають значні обмеження щодо читання та запису в БД. Ще одним плюсом є те, що план Blaze враховує початковий безкоштовний ліміт. Наприклад, припустімо, що ви додали 20 ГБ пам'яті на свій план Blaze. Безкоштовний ліміт

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		28

становить 10 ГБ, що означає, що Firebase стягує лише додаткові 10 ГБ за використання.

3.4.2 Швидкість розробки

Firebase — це відповідний варіант розробки програмного забезпечення, який може допомогти розробникам значно скоротити час, необхідний для запуску програмного забезпечення. Як правило, кожному розробнику потрібен доступ до серверів і хостів для створення та підтримки БД і серверних служб. Отже, щоб створити програму, вам потрібно мати бекенд-розробника і фронтенд-розробника. Це справедливо навіть для невеликих програм, де бекенд і інтерфейс мають взаємодіяти на різних етапах. Наявність розробників front-end і back-end часто призводить до помилок і проблем, які викликають проблеми з програмою, збільшуючи вартість і складність розробки. Однак використання Firebase дозволяє розробникам інтерфейсу керувати всією роботою та скорочувати час, необхідний для виконання роботи. Це все завдяки готовим службам для зберігання даних, аутентифікації, обміну повідомленнями, аналітики тощо. Крім того, Firebase надає безліч готових послуг, і розробникам не потрібно створювати шаблонний код, винаходити щось усунюче та запускати бекенд з нуля. Завдяки високоякісному SDK, детальній документації та активній спільноті розробники інтерфейсу без проблем інтегрують Firebase у свою кодову базу.

3.4.3 Усе в одній платформі

Firebase також надає розробникам списки продуктів, щоб допомогти їм у процесі розробки. Існує два варіанти БД - Firestore і БД реального часу Firebase. Крім того, Firebase дозволяє легко зберігати хмарні медіафайли та пропонує безсерверну розробку програмного забезпечення з інтегрованими хмарними можливостями. Firebase охоплює весь цикл розробки програмного забезпечення,

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		29

а платформа включає функції для створення, публікації та моніторингу програм. Крім того, як завершальний етап циклу розробки програми, він надає інструменти для залучення користувачів і підтримки їх залучення.

3.4.4 Працює від Google

Firebase — це відповідний варіант розробки програмного забезпечення, який може допомогти розробникам значно скоротити час, необхідний для запуску програмного забезпечення. Як правило, кожному розробнику потрібен доступ до серверів і хостів для створення та підтримки БД і серверних служб. Отже, щоб створити програму, вам потрібно мати бекенд-розробника і фронтенд-розробника. Це справедливо навіть для невеликих програм, де бекенд і інтерфейс мають взаємодіяти на різних етапах. Наявність розробників front-end і back-end часто призводить до помилок і проблем, які викликають проблеми з програмою, збільшуючи вартість і складність розробки. Однак використання Firebase дозволяє розробникам інтерфейсу керувати всією роботою та скорочувати час, необхідний для виконання роботи. Це все завдяки готовим службам для зберігання даних, аутентифікації, обміну повідомленнями, аналітики тощо. Крім того, Firebase надає безліч готових послуг, і розробникам не потрібно створювати шаблонний код, винаходити щось усунюче та запускати бекенд з нуля. Завдяки високоякісному SDK, детальній документації та активній спільноті розробники інтерфейсу без проблем інтегрують Firebase у свою кодову базу.

3.4.5 Фокус на розробці фронтенду

Розробники по всьому світу віддають перевагу Firebase, оскільки це дозволяє їм зосередитися на створенні інтерфейсного коду для мобільних додатків. Firebase зменшує потребу в розробці коду бекенд-шаблону, відкладаючи терміни розробки програми. Firebase полегшує розробку додатків і знижує витрати. Використання

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		30

Firebase також дозволяє розробникам і компаніям стандартизувати бекенд-середовища в єдину технологію, просту в освоєнні. Backend-режим зменшує обсяг навчання, необхідний для його підтримки, і дозволяє розробникам інтерфейсу виконувати більшість операцій.

3.4.6 Безсерверність

Масштабування сервера – завдання не з легких! Зокрема, масштабування кластерів БД є проблемою, а оптимізація продуктивності для великих робочих навантажень вимагає досвідчених інженерів. Firebase вирішує цю проблему і забезпечує повністю безсерверне середовище. Firebase має власну безсерверну архітектуру, яка вимагає від користувачів платити по ходу, без необхідності керувати серверною інфраструктурою або турбуватися про неї. Це відрізняється від звичайного сервера, який повинен постійно працювати. Через безсерверну архітектуру Firebase користувачі платять лише тоді, коли використовується сервер. В результаті зменшується кількість проблем масштабування.

3.4.7 Машинне навчання

Машинне навчання знаходиться на порозі розвитку галузі інформаційних технологій. У незліченній кількості способів це вже почалося. За даними Gartner, 30% підприємств використовуватимуть ML в тій чи іншій частині своїх процесів. Перевагою Firebase є те, що він дає розробникам можливість покладатися на машинне навчання. Як розробники iOS, так і Android можуть використовувати ML, незалежно від рівня їхнього досвіду. Firebase постачається з пакетом ML із готовими API для різних функцій мобільної платформи, таких як розпізнавання тексту, розпізнавання обличчя, тегування зображень, сканування штрих-кодів тощо. Розробники можуть вибрати хмарний API або API пристрою відповідно до потреб розробки додатків.

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		31

3.4.8 Моніторинг помилок

Функція Crashlytics Firebase є чудовим інструментом для швидкого усунення несправностей. Firebase може відстежувати нефатальні та фатальні помилки та створювати звіти на основі їх впливу на продуктивність користувачів.

3.4.9 Резервне копіювання

Firebase забезпечує оптимальну безпеку та доступність ваших даних за допомогою регулярного резервного копіювання. Покладаючись на функцію автоматичного резервного копіювання платформи, програма захищена від будь-якої можливості втрати даних. Користувачі преміального плану Blaze можуть легко налаштувати БД реального часу для резервного копіювання. Ви можете отримати доступ до налаштувань автоматичного резервного копіювання, перейшовши на вкладку Резервні копії вашої БД Firebase.

Отже розглянувши усі важливі функції та переваги саме цього сервісу вирішив дати перевагу саме йому. Особливо подобається у Firebase вирішення питання надійності БД , її реактивність та можливість безкоштовно у рамках проекту використовувати хостинг.

Висновки до розділу 3

У цьому розділі вибираються та описуються всі програмні засоби, які будуть використовуватися при розробці основного веб-додатка , що взято за мету цього проекту. Усі вибрані технології є безкоштовними програмними забезпеченнями з відкритим вихідним кодом, що означає, що вони безкоштовні у використанні і навіть можуть використовуватися для розробки реальних комерційних проектів.

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		32

4 РОЗРОБКА АРХІТЕКТУРИ ВЕБ-ДОДАТКУ

Архітектура веб-додатків є ключовою причиною того, що одні веб-програми працюють погано, а інші - чудово. Це тому, що архітектура визначає, як різні компоненти програмного забезпечення взаємодіють друг з одним і з кінцевим користувачем. Пропоную розглянути роботу веб-додатків, та визначити як правильна архітектура додатків підвищує їхню продуктивність[5,6,7]. Дослідити новітні практики, які можуть допомогти нам у процесі веб-розробки.

Основні етапи які слід дослідити в рамках розробки нашого додатку :

- що таке архітектура веб-застосунків?
- компоненти інфраструктури архітектури веб-додатків
- як працюють сучасні веб-програми
- типи архітектури веб-додатків
- найкращі практики розробки

4.1 Що взагалі таке архітектура веб-застосунків?

Архітектура веб-додатків описує взаємодію між програмами, базами даних та системами проміжного програмного забезпечення в Інтернеті. Вона забезпечує одночасну роботу кількох додатків. Зрозуміємо це на простому прикладі відкриття веб-сторінки. Як тільки користувач відкриває веб-сторінку, сервер надсилає браузеру певні дані у відповідь на запит користувача. Точніше кажучи, веб-клієнт (або агент користувача) може запитати веб-ресурси або відомі веб-документи (HTML, JSON, PDF тощо) через веб-сервер. Потім за допомогою цих мінімальних маніпуляцій запитувана інформація з'являється. Після цього починається взаємодія між користувачем та веб-сайтом. Найважливіше, на що слід звернути увагу, це код, розібраний браузером. Веб-програма працює аналогічним чином. (рисунок 4.1)

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		33

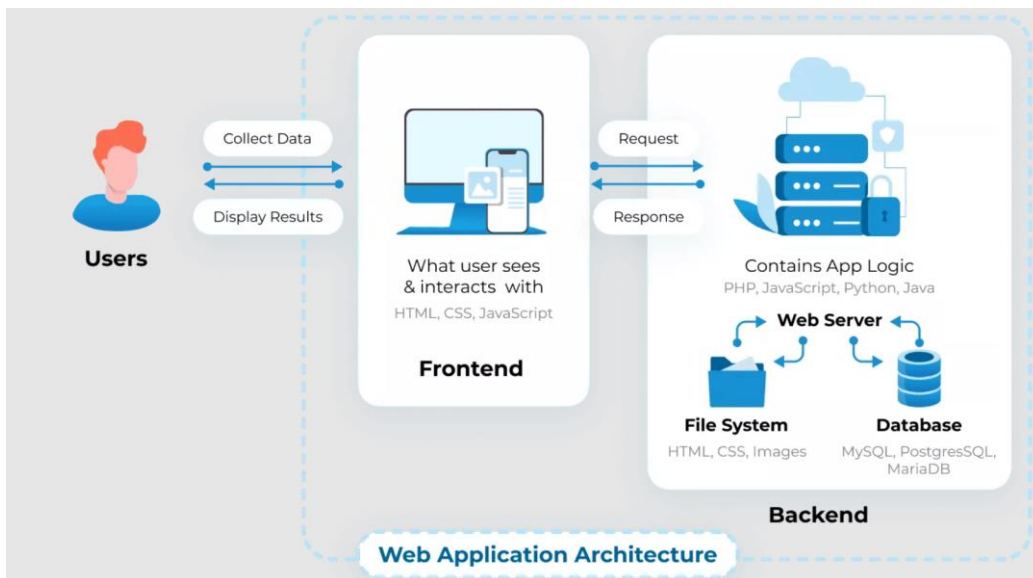


Рисунок 4.1 – Архітектура веб-застосунків

Цей код може містити або не містити конкретні інструкції, які вказують браузеру, як реагувати на різні типи поведінки користувача. Отже, архітектура веб-програми повинна включати всі підкомпоненти, а також зовнішні інтерфейси програм для всієї програмної програми, у вищезгаданому випадку - веб-сайту.

Архітектура веб-додатків незамінна в сучасному світі, оскільки більшість трафіку глобальної мережі, а також більшість програм і пристроїв використовують веб-комунікації. Архітектура веб-застосунків повинна забезпечувати не тільки ефективність, але й надійність, масштабованість, безпеку та стійкість.

4.2 Компоненти інфраструктури архітектури веб-додатків

Кожна програма (веб-програми, комп'ютерна або мобільна програма) має дві частини: фронтенд і бекенд. Фронтенд (сторона клієнта) – це код, який запускається у веб-браузері чи інтерфейсі користувача. Це частина програми, яку бачить користувач і з якою він взаємодіє. Бекенд (серверна частина) - сервери, бази даних та логіка програми, що працюють за завісою. Компоненти бекенда

зберігають дані, обробляють запити, надають інформацію, яку ви бачите у додатку.

Проведемо аналіз елементів архітектури веб-додатків та їх функції.

4.2.1 Інтерфейс користувача

Інтерфейс користувача (UI) - це візуальне представлення веб-програми, яке відображається у веб-браузері і дозволяє користувачам взаємодіяти з ним. Рідко можна зустріти UI як частину архітектури, але в сучасних додатках це є компонент, оскільки частина бізнес-логіки виконується в браузері за допомогою JavaScript. Вміст веб-сторінки включає: HTML, CSS, JavaScript.

Шар JavaScript знаходиться в браузері користувача та взаємодіє з бекендом для оновлення веб-сторінки в режимі реального часу. У сучасних моделях архітектури веб-сайтів це дозволяє вмісту сторінки динамічно змінюватися в міру того, як користувачі взаємодіють із елементами інтерфейсу.

4.2.2 Веб-сервер

Веб-сервери - це комп'ютери, які приймають запити, обробляють їх та надсилають дані до браузера користувача. Термін "сервер" застосовується до будь-якого комп'ютера, який надає послуги через Інтернет (або приватну мережу). Сервери поділяються на різні категорії, залежно від типу послуг, які на них виконуються. Наприклад, веб-сервери обробляють запити HTTP (порт 80) та HTTPS (порт 443).

4.2.3 Сервер бази даних

Сервери баз даних – це централізовані місця зберігання баз даних (організованих колекцій інформації). Ці сервери взаємодіють із веб-серверами та

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		35

надають дані авторизованим користувачам. Він використовує програму бази даних, яка надає послуги бази даних іншим комп'ютерним програмам або комп'ютерам відповідно до моделі клієнт-сервер. Системи управління базами даних часто надають функціональність сервера бази даних, а деякі системи управління базами даних (такі як MySQL) покладаються виключно на модель клієнт-сервер для доступу до бази даних (тоді як інші, такі як SQLite, призначені для використання як вбудована база даних).

Користувачі звертаються до сервера баз даних або через фронтенд, що працює на комп'ютері користувача - який відображає дані - або через бекенд, який працює на сервері і виконує такі завдання, як аналіз і зберігання даних.

У моделі "провідний-відомий" провідні сервери баз даних є центральними та основними місцями зберігання даних, а ведені сервери баз даних є синхронізованими резервними копіями провідних серверів, що діють як проксі-сервери. Більшість додатків баз даних відповідають мовою запитів. Кожна база даних розуміє свою мову запитів, перетворює кожен відправлений запит в форму, що читається сервером, і виконує його для отримання результатів.

4.2.4 DNS

Система доменних імен (DNS) - це інтернет-ієрархічна система іменування, яка переводить доменне ім'я (наприклад, www.google.com) в IP-адресу (8.8.8.8). Браузери потребують IP-адрес для зв'язку з веб-серверами. DNS просто допомагає користувачам встановлювати зв'язок, використовуючи імена, що читаються, замість складних чисел.

4.2.5 Проміжне програмне забезпечення для обміну повідомленнями

Веб-програми можуть складатися з багатьох сервісів, розміщених на декількох серверах. Посередницьке програмне забезпечення обміну

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		36

повідомленнями допомагає різним програмним модулям, мовам програмування та програмам взаємодіяти. Клієнтська та серверна частини веб-програми взаємодіють з використанням синхронних (HTTPS та TCP) протоколів, у той час як серверні компоненти програми можуть також використовувати асинхронний (AMQP) протокол зв'язку. Крім того, програмне забезпечення проміжного шару та інтерфейси прикладного програмування (API) забезпечують передачу даних між програмами та серверами.

4.2.6 Балансувальник навантаження

Кожен сервер має обмежений обсяг оперативної пам'яті та процесора, тому збільшення трафіку користувачів може призвести до проблем із продуктивністю. На допомогу приходять балансувальники навантаження. Балансувальник навантаження обробляє трафік і розподіляє його між веб-серверами, залежно від доступних обчислювальних ресурсів. Це оптимізує навантаження, забезпечуючи безперебійну роботу веб-програми.

4.2.7 Кеш

Кеш - це тимчасовий носій, який зберігає дані веб-сторінки і швидше надає їх за повторних запитів. У серверній архітектурі веб-застосунків кешування зазвичай відбувається в браузері. Наприклад, коли ви вперше завантажуєте веб-сторінку, її частина зберігається у локальному сховищі.

4.2.8 CDN

CDN - це система серверних вузлів, які зберігають та доставляють статичний контент (зображення, аудіо, відео, CSS та файли JavaScript). CDN можуть покращити затримку для користувачів, які живуть далеко від центрів обробки

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		37

даних, де розміщено ваш веб-додаток. Уявіть собі користувача з Каліфорнії, який звертається до веб-застосунку в центрі обробки даних у Нью-Йорку. Браузер може отримати інформацію з центру обробки даних, але сторінка завантажуватиметься довше через відстань. Тепер припустимо, що веб-додаток використовує CDN у Каліфорнії. Тоді браузер зможе завантажувати важкі файли з найближчого вузла набагато швидше.

4.3 Як працюють сучасні веб-програми?

Тепер, поговоримо що відбувається, коли користувач вводить URL-адресу в адресний рядок(рисунок 4.3).

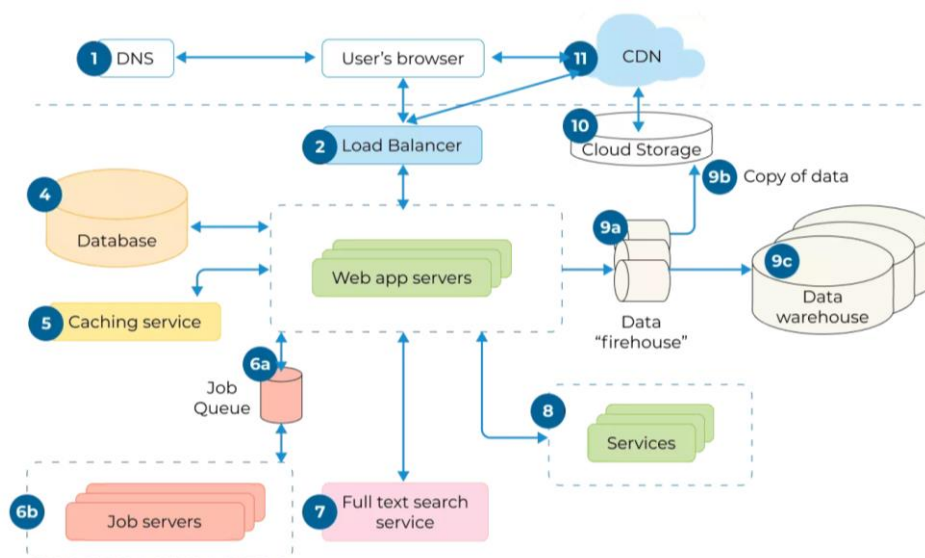


Рисунок 4.3 – Робота сучасного веб-додатку

Клієнт зв'язується з DNS-сервером, який надає IP-адресу. Потім браузер надсилає запит на веб-документи на веб-сервер. Веб-сервер направляє запити до прикладної логіки (бізнес-шар програми), яка визначає порядок дій залежно від рівня доступу користувача та типу запиту. Якщо запит відповідає критеріям доступу, логіка програми завантажує дані з бази даних та надсилає їх у браузер. Браузер отримує веб-документ із даними, аналізує його та завантажує будь-які

статичні файли (JavaScript, CSS, зображення, відео тощо) з CDN. У сучасних веб-додатках логіка програми може мати шар JavaScript, який потім взаємодіє з бекендом та різними API з браузера. Браузер рендерить інформацію та відображає її у вигляді сторінки сайту. Всі ці процеси відбуваються на різних рівнях веб-програми.

4.4 Типи архітектури веб-додатків

Архітектура визначає високорівневі залежності між компонентами фронтенду та бекенду. Розглянемо поширені типи сучасних архітектур окремо для клієнтської та серверної сторін програми.

4.4.1 Архітектури фронтенду

SSR - архітектура з рендерингом на стороні сервера веб-сервер, генерує всю сторінку кожного запиту. Це означає, що браузеру доводиться перезавантажуватися щоразу, коли виникає потреба оновити сторінку. Хоча це добре, коли час рендерингу невеликий, UX може зіпсуватися, якщо це займе занадто багато часу.

SPA - Односторінкова програма - це сучасна архітектура веб-додатків, яка обслуговує веб-сторінки з вмістом, що динамічно оновлюється. Ось як відбувається процес:

- веб-сервер генерує одну веб-сторінку з мінімальною кількістю HTML-даних та шаром JavaScript.
- шар JavaScript взаємодіє з бекендом для отримання даних і відображення HTML, що залишився. Він також забезпечує оновлення у режимі реального часу.

Архітектура SPA дозволяє веб-застосунку динамічно переписувати вміст сторінки без її перезавантаження. Це робить досвід користувача динамічним і знижує навантаження на ресурси бекенда.

PWA - архітектура прогресивних веб-програм, які є гібридом між SPA і мобільними програмами. Тобто вони використовують прогресивні покращення для забезпечення користувачам більш надійного досвіду, використовують нові можливості для забезпечення більш інтегрованого досвіду та можуть бути встановлені. А оскільки це веб-програма, вона може бути доступна будь-якій людині, в будь-якому місці, на будь-якому пристрої, і все це з єдиною кодовою базою. Після встановлення PWA виглядає як будь-яка інша програма, наприклад:

- має значок на головному екрані, програмі запуску програм та в панелі запуску або в меню "Пуск".
- з'являється під час пошуку програм на пристрої.
- відкривається в окремому вікні, повністю відокремленому від інтерфейсу браузера.
- має доступ до вищих рівнів інтеграції з ОС, наприклад, до обробки URL-адрес або налаштування рядка заголовка.
- може працювати в автономному режимі.

4.4.1 Архітектури бекенду

Мікросервісна архітектура - бекенд програми розділений на кілька одноцільових слабозв'язаних сервісів. Ці послуги працюють окремо, взаємодіють через мережу і масштабуються незалежно друг від друга. Крім того, мікросервіси взаємодіють синхронно (RESTful API, TCP) або асинхронно (обмін повідомленнями). Наприклад, інтернет-магазин може мати мікросервіси для реєстрації клієнтів, завантаження файлів, інтерфейсу користувача, каталогу товарів, кошика і процесу оплати.

Ось що робить ізольовану структуру цієї архітектури привабливою:

- мікросервіс може бути побудований незалежно різними командами, що підвищує швидкість розробки.

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		40

- код або технічний стек для кожного сервісу можна змінювати, не чіпаючи всі програми, що дозволяє тестувати різні частини системи
- сервіси можуть бути додані в програму на будь-якому етапі життєвого циклу програми.

Проте ця архітектура має й недоліки. Тестування та моніторинг відмово утруднені, оскільки доводиться працювати з окремими модулями на декількох серверах. Крім того, ваша команда має документувати комунікації між кожним мікросервісом.

Безсерверна архітектура - хмарна інфраструктура, розміщена у сторонніх провайдерів, таких як Amazon Web Services (AWS), Microsoft Azure та Google Cloud. Безсерверні рішення можуть похвалитися низкою переваг:

- провайдер бере на себе управління апаратним та програмним забезпеченням, дозволяючи вашій команді зосередитися на створенні та керуванні програмою.
- ви можете швидко масштабувати інфраструктуру вгору та вниз залежно від ваших потреб
- більшість безсерверних планів мають гнучку модель тарифікації, тому ви платите лише за використовувані ресурси та сховища.
- безпека інфраструктури та доступні ресурси залежать від стороннього постачальника, що може турбувати деякі підприємства (розум приходять FinTech і охорона здоров'я).

4.5 Архітектура веб-додатків: Найкращі практики розробки

Неможливо передбачити всі проблеми розробки з якими зустрічається будь-який розробник. Але можна використовувати деякі популярні підходи, які можуть їх пом'якшити.

CI/CD конвеєр для розробки в архітектурі мікросервісу веб-додатків. Безперервна інтеграція (CI) та безперервна доставка (CD) були створені для

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		41

поділу розробки веб-додатків на незалежно керовані та компоненти, що доставляються. Простіше кажучи, інженерні команди кодують, тестують та інтегрують програму поетапно. Компанії можуть використовувати CI/CD для покращення інтеграційного тестування та стабільності коду. Наприклад, можна впровадити модуль, що тестується, який компілює і автоматично тестує код від різних розробників. Отже, можна дізнатися про помилки та несумісність відразу після їх виникнення.

Методологія API-first починається з планування та розробки макету API. Після того, як зацікавлені сторони узгодять напрям і вимоги до інтерфейсу, решта програми може розроблятися ітераціями у вигляді мікросервісів. API-first є протилежністю методології code-first, у якій інженери будують API навколо коду. В результаті інтерфейс у проектах code-first може здаватися погано оптимізованим.

JAMstack - це динамічний підхід до попереднього рендерингу, заснований на JavaScript, попередньо створеної гіпертекстової розмітки, багаторазово використовуваних API та CDN. Веб-програма повністю відображається заздалегідь і розміщується в CDN. За запитом користувача веб-програма вчасно доставляється з CDN до браузера користувача. За допомогою JavaScript у браузері програма може розширити свою функціональність, використовуючи API сторонніх розробників.

Блокчейн - це однорангова система обліку транзакцій, у якій дані розподілені мережі комп'ютерів. Вибраний учасник створює новий блок даних, який містить хеш (цифровий відбиток) попереднього блоку та набір транзакцій. У мережах блокчейн також використовується механізм доказу роботи, який обмежує кількість блоків даних, які можуть з'являтися через задані часові проміжки (наприклад, кожні десять хвилин). Щоб підробити один блок, необхідно змінити підтвердження роботи та хеші наступних блоків. Таким чином, створюється ланцюжок даних, який практично неможливо змінити непоміченим. Крім того, ви

можете використовувати мережу блокчейн для зберігання частини ваших даних для забезпечення додаткової безпеки та стійкості.

Контейнеризація робить ваш веб-додаток переносимим. Ця технологія полягає в упаковці програми разом із супутніми бібліотеками, конфігураційними файлами та допоміжними залежностями у єдиний програмний пакет (контейнер). Контейнер не залежить від операційної системи хоста, що дозволяє працювати практично на будь-якій платформі. Контейнери мають сенс, якщо у вас немає чіткого плану проектування архітектури веб-додатків. Ви також можете вибрати контейнеризацію, якщо не бажаєте прив'язувати себе до одного середовища.

Висновки до розділу 4

У даному розділі було ознайомлено з усіма видами архітектур сучасних веб-додатків. Розглянуті архітектури дали нам змогу краще розуміти необхідні модулі та системи для розробки . Для нашого додатку щодо клієнтської частини (фронтенду) вирішено вибрати архітектуру SPA , атже вона відмінно підходить для нашого проекту, в якому важлива швидкість та якість усієї системи , а також її безпеність. Також загальною структурою було вибрано API-first.

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		43

5 РОЗРОБКА ВЕБ-ДОДАТКУ

5.1 Створення проекту

Будь-який проект на Vue.js починається з створення його за допомогою команди `vue create app` [8]. Після написання цієї команди у нас є можливість вибрати та конфігурувати необхідні бібліотеки та допоміжні засоби розробки. Початковим вибором технологій, які рекомендує програма `vue create` є :

- `babel` - безкоштовний JavaScript компілятор[9] з відкритим вихідним кодом, який в основному використовується для перетворення коду ECMAScript 2015+ (ES6+) у зворотно-сумісну версію JavaScript, яка може працювати на старих механізмах JavaScript. Він є популярним інструментом для використання нових можливостей мови програмування JavaScript. Розробники можуть використовувати нові функції мови JavaScript через Babel, щоб перетворити свій вихідний код у версію JavaScript, з якою можуть працювати веб-переглядачі.

- `vueRouter` - додаткова бібліотека для Vue.js[10], яка слугує для задання маршрутів у браузері для веб-додатку та відповідно відображення певних компонентів за цими маршрутами.

- `vuex` - патерн керування станом та бібліотека для програм Vue. Js[11]. Він служить централізованим сховищем для всіх компонентів програми, при цьому правила гарантують, що стан може бути змінено лише передбачуваним чином.

Також за допомогою консольної команди `pnpm i` (далі пишемо необхідну бібліотеку) ми повинні встановити усі бібліотеки про які йшла мова у попередніх розділах та додаткові. Увесь список необхідностей, що ми будемо використовувати можна знайти у спеціальному файлі, що має назву `package.json`(рисунок 5.1).

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		44

```

"dependencies": {
  "@kyvg/vue3-notification": "^2.3.4",
  "axios": "^0.27.2",
  "core-js": "^3.6.5",
  "firebase": "^9.8.1",
  "lodash": "^4.17.21",
  "node": "^14.18.0",
  "nprogress": "^0.2.0",
  "object-to-formdata": "^4.4.2",
  "vee-validate": "^4.5.11",
  "vue": "^3.0.0",
  "vue-router": "^4.0.0-0",
  "vuex": "^4.0.0-0",
  "webpack": "^4.46.0",
  "webpack-cli": "^4.9.2",
  "yup": "^0.32.11"
},
"devDependencies": {
  "@vue/cli-plugin-babel": "~4.5.15",
  "@vue/cli-plugin-router": "~4.5.15",
  "@vue/cli-plugin-vuex": "~4.5.15",
  "@vue/cli-service": "~4.5.15",
  "@vue/compiler-sfc": "^3.0.0",
  "autoprefixer": "^9.8.8",
  "node-sass": "^4.12.0",
  "postcss": "^7.0.39",
  "sass-loader": "^8.0.2",
  "tailwindcss": "npm:@tailwindcss/postcss7-compat@^2.2.17"
},

```

Рисунок 5.1 Package.json

Файл package.json – це свого роду маніфест нашого проекту. Він може робити багато речей, зовсім не пов'язаних між собою. Наприклад, це центральне конфігураційне сховище для інструментів. Тут же npm та yarn зберігають імена та версії всіх встановлених пакетів. Загально він містить усі залежності які ми будемо використовувати.

5.2 Налаштування Vue Router

Ми створюємо досить велику веб-систему , тому вона матиме різні під-системи для студентів , вчителів та самої адміністрації. Нам потрібно зробити маршрути у додатку , щоб вони відображалися лише для певних сутностей , та

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		45

користувач з іншою роллю не міг відвідати ту сторінку , яка не була створена та сконфігурована для нього. Так виділивши три підсистеми ми за допомогою middleware конструкції створюємо файли формату js : isAuth, isAdmin, isTeacher, isStudent де отримуємо статус користувача, перевіряємо чи підходить він нам та захищаємо ці дані. Об'єднуємо ці файли за допомогою конструкції middlewarePipeline , яка допомагає нам зібрати ці middleware`s у самому VueRouter, передавши їх відповідне поле.

Також потрібно створити публічні шляхи та приватні. До публичних ми відносимо сторінки реєстрації та входу. Щодо приватних , до них належать: особисті кабінети , публічні профілі , сторінки груп та одної групи та інші залежно від типу сутності активного користувача. Потім імпортуємо усе в main.js (головний файл додатку) та налаштування vue router на цьому можна вважати успішним.

5.3 Налаштування Firebase

Увесь вище описаний чудовий функціонал та інтерфейс не працював би без бази даних та міцного серверного підтримання. Ми у розділах вище описували усі переваги вибору саме Firebase, тому хочемо у цьому загальному розділі про розробку поговорити про налаштування та підключення його до нашого проекту.

Створення проекту на firebase відбувається досить легко через їх консоль ми починаємо створювати проект , спочатку вибираємо назву , потім розташування серверу територіально , та інші налаштування. Після реєстрації ми вибрали тип нашого додатку , тобто веб-застосунок , нам запропонували встановити сам firebase у коді проекту[12] , але ми вже це зробили , потім нам дали спеціальні приватні ключі , щоб додаток зміг працювати з Firebase, ми їх прописали у файлі з налаштуваннями, це зображено на рисунку 5.2.

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		46

```
const firebaseConfig = {
  apiKey: "AIzaSyDABSSTOWTEEx12HPnga0Vz33jhi_s2w8lw",
  authDomain: "camp-43d57.firebaseio.com",
  projectId: "camp-43d57",
  storageBucket: "camp-43d57.appspot.com",
  messagingSenderId: "87422008141",
  appId: "1:87422008141:web:cfa4a1cd5d12aa55341504",
  measurementId: "G-2MXH9BYFVB"
};

export default firebaseConfig
```

Рисунок 5.2 Налаштування Firebase

Далі ми ці ключі експортуємо та підключаємо у головному файлі проекту. Тепер слід зазначити, що працюючи з базою даних ми робимо певні запити до неї у корні проекту, та не використовуємо стороні бібліотеки, таку змогу надає сам сервіс. Саме через це ми створили окрему папку де винесли усю логіку по роботі з базою даних. У цій папці в головному файлі index.js є усі типи запитів які ми використали у ході розробки[13]. Це такий тип функцій:

- set слугує для основних операцій запису. Будь-які існуючі дані можна замінити таким чином за допомогою set() для збереження даних за вказаним посиланням.
- get використовується коли дані потрібно прочитати лише один раз, щоб отримати знімок даних з бази даних. Якщо з якихось причин get() не може повернути значення сервера, клієнт перевірить кеш локального сховища і поверне помилку, якщо значення все ще не знайдено.
- update – використовується для одночасного запису на деякі дочірні вузли, не перезаписуючи інші дочірні вузли. Під час виклику update() базові підзначення можна оновити, вказавши шлях ключа. Якщо дані зберігаються в кількох місцях для кращого масштабування, можна розділити дані, щоб оновити всі екземпляри даних.

Сервіс firebase ми використали не лише для бази даних , а й для реєстрації та входу в систему університету. Ми вибрали тип аутенфікації по емейлу та паролю. Поговоримо трохи детальніше про доступні методи[14].

CreateUserWithEmailAndPassword – цей метод надає змогу реєструвати користувача використовуючи лише пароль та адресу електронної пошти , введеною за допомогою користувацького інтерфейсу.

SignInWithEmailAndPassword - цей метод слугує для входу до вже уснуючого профіля, коли користувач входить у програму, необхідно лише передати адресу електронної пошти та пароль. Якщо дані не співпадають з тими що є у базі , то сервер видасть помилку та відповідне повідомлення.

Проте виникає проблема , коли користувач забув свій пароль. Сервіс надає ще один метод sendPasswordResetEmail. Він досить простий , потрібно лише передати свій емейл , та Firebase надішле вам лист на електронну скриньку , де ви перенаправитесь на веб-додаток та зможете створити новий пароль.

5.4 Налаштування Vuex

Vuex – це бібліотека управління станом розробки для Vue.js. Вона дає нам змогу централізовано зберігати дані та стан усього додатку та вирішити проблему пропс-дринлінгу , коли якась частина даних пересувається постійно через усе дерево компонентів, що не є дуже гарною практикою , адже це впливає на загальну оптимізацію та продуктивність програми. Так як у нас буде багато запитів на сервер та отриману інформацію краще зберігати в одному місці. Також , будуть певні внутрішні компоненти програми, які нам потрібно буде використати у будь-якій частині програми , незалежно від їх розташування, вирішено створити на цей випадок vuex-модулі. Кожен модуль може містити свій власний стан, мутації, дії, гетери та навіть вкладені модулі - це фрактал по всій довжині. Об'єднавши їх у один загальний модуль він і буде слугувати нам тим самим станом програми.

Auth модуль потрібен нам для зберігання інформації користувача. Він містить у собі функції отримання даних з бази даних , встановлення цих даних , а також їх очищення , це зображено на рисунку 5.3.

```
import {getDatabase, ref, child, get , onValue} from "firebase/database";

const state = {
  authType: '',
  user: {},
};

const actions = {
  getUserRole(context, uid) { ...
  },
  getUserData(context, payload) { ...
  },
};

const mutations = {
  setUser: (state, value) => { ...
  },
  setUserId: (state, value) => { ...
  },
  setAuthType: (state, value) => { ...
  },
  logout: (state , router) => { ...
  },
}

const getters = {
  getAuthType: (state) => state.authType,
};

export default {
  state,
  getters,
  actions,
  mutations,
};
```

Рисунок 5.3 Auth модуль

Loader модуль для встановлення етапу завантаження та блокування інтерфейсу додатку на цей момент. Це дозволяє зручно відстежувати та візуально розуміти звичайному користувачу розуміти етап навантаження. Блокування робиться задля того, щоб користувач системи не міг зробити додаткових дій під час запиту на сервер, та обробки тої інформації яка надсилається на нього. Завдяки vuex ми можемо викликати його з будь-якої частини додатку та назначати статус завантаження в залежності від запиту на сервер та його відповіді. Цей модуль досить простий у конфігурації (рисунок 5.4).

```

const state = {
  loading: false
};

const actions = {
};

const mutations = {
  isLoading: (state, value) => {
    state.loading = value
  },
}
const getters = {
  getLoadingStatus: (state) => state.loading,
};

export default {
  state,
  getters,
  actions,
  mutations,
};

```

Рисунок 5.4 Loader модуль

5.5 Створення компонентів

Однією з найважливіших особливостей Vue є можливість використання компонентів. Компоненти - це багаторазово використовувані екземпляри Vue з елементами HTML. Компоненти можна повторно використовувати скільки завгодно разів або використовувати в іншому компоненті, перетворюючи його на дочірній компонент. У компоненті Vue можна використовувати дані, обчислення, спостерігачі та методи. Щоб створити компонент Vue.js потрібно лише написати назву файлу та через крапку вказати розширення vue. Загальна структура кожного компонента представлена на рисунку 5.5 .

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		50


```
<template>
  <div class="wrapper">
    some html code
  </div>
</template>

<script setup>

//some js code

</script>

<style>
.wrapper{
  /* some styles */
}
</style>
```

Рисунок 5.5 Структура Vue.js компонента

Отже будь-який компонент складається з трьох блоків : template , script, style. Поговоримо детальніше про кожний.

Template – використовує синтаксис шаблонів на основі HTML, який дозволяє декларативно пов'язувати візуалізується DOM з даними базового екземпляра компонента. Всі шаблони Vue – це синтаксично коректний HTML, який може бути розібраний браузером та HTML-парсерами, що відповідають специфікації. Під капотом Vue компілює шаблони у високооптимізований код JavaScript. У поєднанні із системою реактивності Vue здатний інтелектуально визначити мінімальну кількість компонентів для повторного відображення та застосувати мінімальну кількість маніпуляцій із DOM при зміні стану програми.

Script setup – використовує код усередині який компілюється як вміст функції setup() компонента. Це означає, що на відміну від звичайного тега script, який виконується лише один раз при першому імпорті компонента, код усередині script setup буде виконуватися щоразу під час створення екземпляра компонента. При

використанні `<script setup>` будь-які прив'язки верхнього рівня (включаючи змінні, оголошення функцій та імпорт), оголошені в його середині, можна безпосередньо використовувати у шаблоні. Через нього також можна імпортувати інші модулі з проекту, або компоненти. Існує можливість отримання та відправки даних через дочірні та батьківські компоненти.

`Style` – тег, який використовується для написання стилів `css`. Слід зазначити, що один файл з розширенням `vue` може містити кілька тегів `style`. Тег `style` може мати атрибути `scoped` або `module`, щоб допомогти інкапсулювати стилі у поточний компонент. Декілька тегів `style` з різними режимами інкапсуляції можуть бути змішані в одному компоненті. Також цей тег має можливість обрати написання стилів через `css-препроцесори`.

5.6 Реєстрація та вхід до кабінету

На публічному роуті `/login` ми створюємо сторінку реєстрації для вчителів та студентів. Як бачимо на рисунку 5.6 ця сторінка містить галочку для реєстрації де можна вибрати, ким ви є у системі університету.

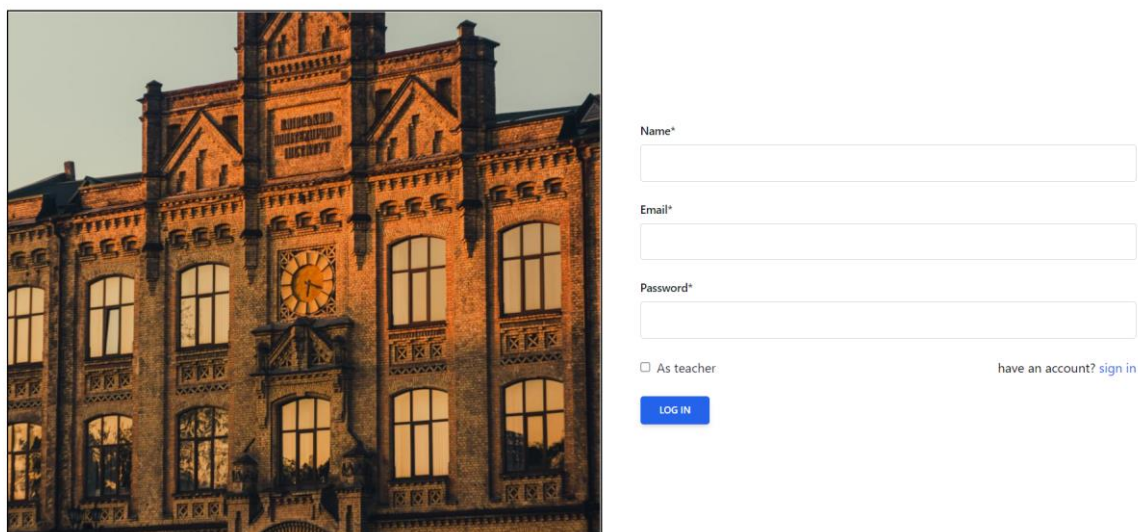


Рисунок 5.6 Сторінка реєстрації

Також ця сторінка містить посилання на сторінку входу , якщо у вас вже є акаунт(рисунок 5.7)

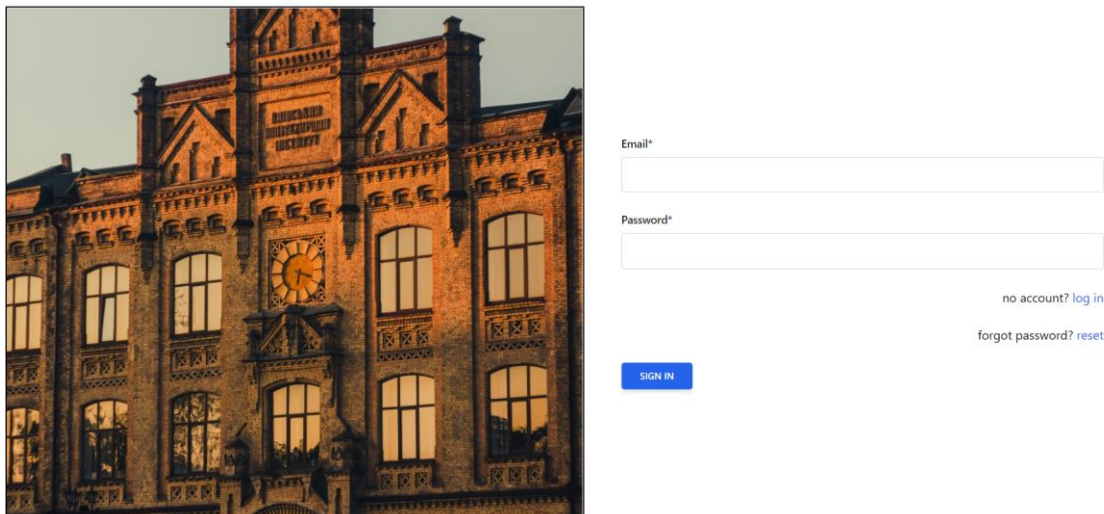


Рисунок 5.7 Сторінка входу

На цій сторінці міститься також можливість перейти на сторінку відновлення пароля , якщо ви його не пам'ятаєте.

Функціонал входу та реєстрації ми створили за допомогою кастомних елементів форми , їх валідації за допомогою vee-validate та можливостей аутентифікації платформи firebase . Після створення акаунту кожен користувач отримує свій унікальний токен, за яким у бд , можна отримати усю його інформацію та роль у цій системі. Також для нашої платформи ми створили сторінку для входу адміністрації , адже це зовсім інша сутність та вона потребує окремого інтерфейсу . Її можна знайти за посиланням /admin , проте вона має лише одну закріплену за нею інформацію щодо входу.

5.7 Користувачі

Ми створюємо додаток та піклуємося у першу чергу про легкий , зрозумілий та сучасний користувацький інтерфейс. Тому розділили веб-додаток на умовні три

екосистеми : адміністрація університету , вчителі та студенти . Та подбали про кожну з них , отримавши різні інтерфейси , про які поговоримо далі.

5.7.1 Адміністрація

На нашу думку , адміністрація університету є головною сутністю , адже саме вона піклується про усі навчальні процеси , слідкує за своїми вчителями та студентами. Тому ми створили персональний кабінет для неї (рисунок 5.8).

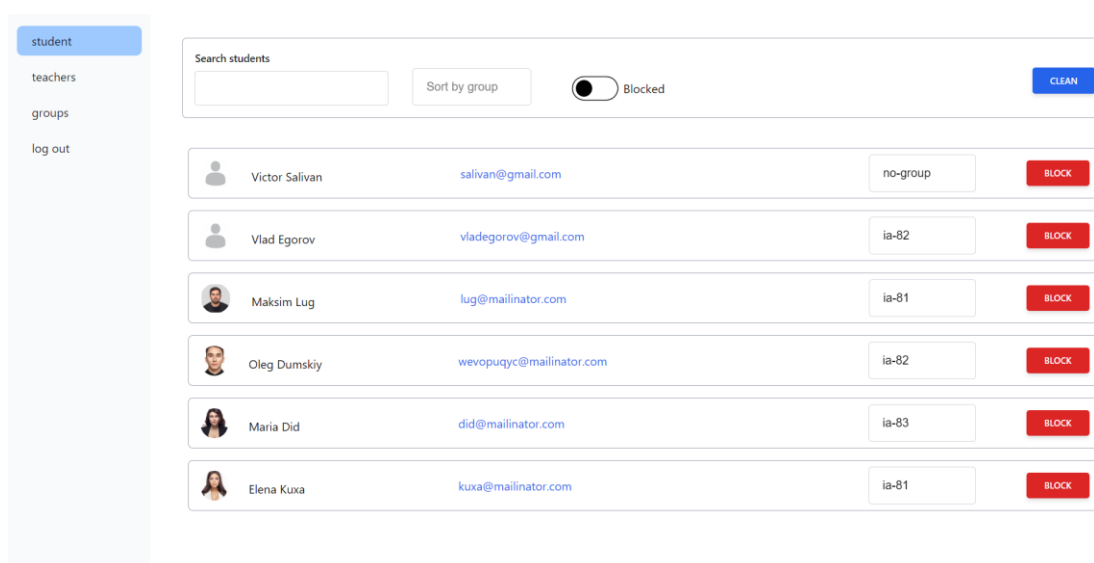
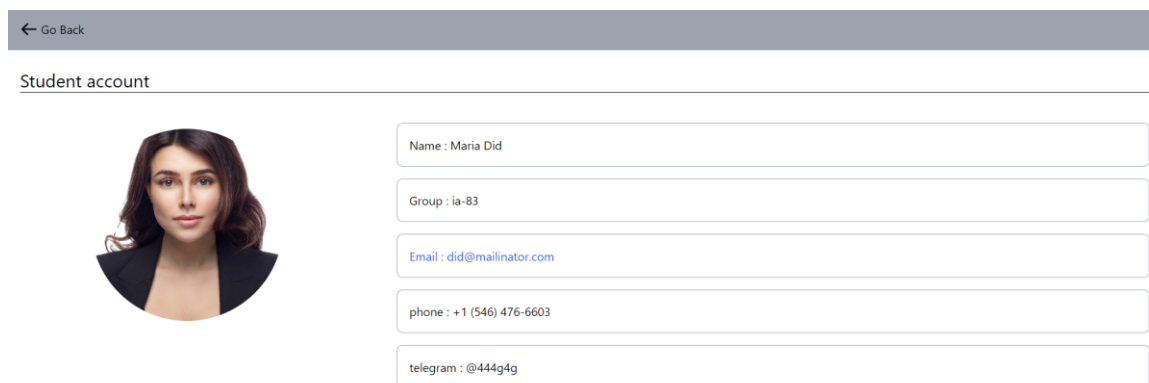


Рисунок 5.8 Кабінет адміністрації

Почнемо з розгляду лівої панелі , вона містить посилання на керування студентами , вчителями та групами які вже , або будуть у системі. Це є зручно, адже уся навігація міститься в одному місці і важко в ній заплутатися. Також далі ми бачимо активну вкладку студентів . На ній ми можемо бачити увесь список студентів які є в учбовому місці . За допомогою верхньої панелі , яка доречі є і на сторінці усіх вчителів , можливо виконувати ряд важливих функцій. Коли студент тільки зареєструвався адміністрація може додавати його до певної групи , тим самим верифікувати акаунт , або якщо цей студент , за якихось причин був відрахован , його можна заблокувати , та він отримає відповідне повідомлення у

своєму акаунті. Також , якщо студент поновив навчання , його можна за допомогою відповідної кнопки поновити. Корисним є також пошук певного студента , або сортування за групами. Список усіх студентів є досить зручним , знайшовши потрібного студента ми одразу бачим головну інформацію про нього , але якщо її не вистачає , ми можемо натиснути на студента , та перейти на його сторінку , де буде повна контактна інформація. Це зображено на рисунку 5.9.



← Go Back

Student account

Name : Maria Did

Group : ia-83

Email : did@mailinator.com

phone : +1 (546) 476-6603

telegram : @444g4g

Рисунок 5.9 Профіль окремого студента

Доречі такий профіль має кожен вчитель у системі , знайти його досить просто , потрібно лише натиснути на певного вчителя , на вкладці де увесь їх список. Ми розуміли , що поняття групи повино бути окремим. Попіклувавшись про це у лівій панелі є посилання на список усіх груп. Там зручно можна подивитися список усіх можливих існуючих груп , нажавши на будь-яку ми побачимо склад , до якого входять усі вчителі та лектори та поточний список студентів , якщо студента відрахували, то його вже не буде в цьому списку.

5.7.2 Вчителі

Невід’ємною частиною будь-якого навчального закладу є вчителі. Отже одразу після реєстрації вчитель потрапляє на сторінку свого профіля. Тут він може завантажити свою фотографію , щоб студенти та адміністрація мала більший

візуальний ефект та для більшої зручності, проте це не обов'язково, адже одразу він має стандартну фотографію, яку змінює за бажанням (рисунок 5.10).

Account

Personal account

Groups

log out

Name : Boris Turnev

Email : turnev@mailinator.com

Teacher groups

1. ia-83

2. ia-93

Lesson name

Math

Lesson description

Mathematics is the science that deals with the logic of shape, quantity and arrangement. Iv

About myself

Рисунок 5.10 Персональний профіль вчителя

Його профіль є важливим для студентів, вони можуть бачити усю інформацію про його предмет, сутність цього предмету та усі необхідні посилання на зум, гугл міт чи іншу систему наприклад для консультації та онлайн лекції чи практики. Профіль містить контактну інформацію для усіх. Щодо питання назначення груп та лекцій, це робить адміністрація, вона може зайти на профіль конкретного вчителя, та через поле пошуку знайти якусь групу та назначити цього вчителя викладати у цій групі, або видалити вже існуючу. Вчителі також можуть бачити список тих груп та занять де вони викладають, ця інформація є у розділі груп у лівій панелі.

5.7.3 Студенти

Студенти у цій системі є сутністю яка виконує роль читача. Вони також можуть налаштовувати свій персональний профіль, який має схожість з профілем

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		56

вчителя , де можна заповнити дані та обрати аватарку. Додаткові функції для студентів це перегляд своїх лекцій які були назначені адміністрацією , детальний огляд своєї групи та легка комунікація з усіма сутностями які їй доступні.

5.8 Адаптивність

Кожен веб-додаток чи то простий сайт опублікований у мережі інтернету, має коло відвідувачі , які переглядають це через різні девайси. Існує багато прикладів коли адаптивної (мобільної) версії просто немає. Це дуже псує враження користувача , та його можливості . Адже за статистикою у сучасному світі більшість з користувачів використовують свої мобільні пристрої для перегляду сайтів в інтернеті , не кожен має змогу сидіти через комп'ютер. Тому ми вирішили зробити нашу систему відповідно адаптивною та кросбраузерною , щоб будь-хто та з будь-чого міг використати наш веб-додаток.

Будь яка адаптивність робиться за допомогою медіа запитів , методів які нам надає Cсс . Тобто потрібно прописати певні точки, за якими програма буде розуміти, на яких екранах змінюється початково задані стилі веб додатку. Наприклад програмною мовою ми задаємо мобільний екран та прописуємо зміну стилів. Усі точки адаптивності ми беремо за статистикою девайсів , які найчастіше використовують мережу інтернет. Такий підхід є найбільш відомим та використовуваним. Проте ми вирішили усі наші стилі закласти та використовувати через спеціальний фреймворк tailwind , в попередніх розділах обгрунтувавши причини вибору. Так-от tailwind використовує стилі класовим підходом , тобто ви пишете певний клас(за офіційною документацією) та логічно використовуєте , такий же підхід ми беремо і для адаптивів. Розглянемо приклад з офіційної документації[15] , де показано як писати адаптивну верстку для веб-додатку(рисунок 5.11).

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		57

Breakpoint prefix	Minimum width	CSS
`sm`	640px	`@media (min-width: 640px) { ... }`
`md`	768px	`@media (min-width: 768px) { ... }`
`lg`	1024px	`@media (min-width: 1024px) { ... }`
`xl`	1280px	`@media (min-width: 1280px) { ... }`
`2xl`	1536px	`@media (min-width: 1536px) { ... }`


```
<!-- Width of 16 by default, 32 on medium screens, and 48 on large screens -->

```

Рисунок 5.11 Таблиця адаптивності та приклад використання

По-перше, ми бачимо таблицю точок зупинки (breakpoint), які якраз слугують для створення адаптивності, за принципом спочатку створюється мобільна версія, а потім вже по наростаючій дописуються інші стилі для певного класу. Одразу після цієї таблиці є приклад, як взагалі це використовувати, ми просто прописуємо необхідну точку зупинки, а далі tailwind клас який хочемо використати на цій точці та за цим розширенням. За таким прикладом ми робили увесь адаптив нашого веб-додатку.

Висновки до розділу 5

У даному розділі було розроблено сучасний веб-додаток на основі фреймворку Vue.js та мови JavaScript. Додаток враховував недоліки існуючих систем у мережі інтернету. Також було показано використання передових технологій, які зробили розробку швидкою та зручною. Додаток вийшов кросбраузерним, адаптивним та оптимізованим під різні пристрої.

ВИСНОВКИ

У результаті розробки індивідуального дослідницького проєкту був створений веб-застосунок, що дозволяє підтримувати та вести навчальний процес у вищих навчальних закладах. Однією із головних переваг розробленої програми є те, що вона враховує недоліки своїх існуючих конкурентів. Це об'єднання дає змогу вирішити одну із багатьох проблем навчальних закладів, а саме проблему автоматизації усього навчального процесу для його представників .

На початку розробки індивідуального дослідницького проєкту встановлюються та описуються технічні характеристики початкової заявки. У процесі розробки існуючі рішення були переглянуті, щоб визначити їх сильні та слабкі сторони та розробити необхідну функціональність для самого додатка.

В розділі про вибір мови програмування було проведено огляд можливих мов для реалізації поставленої задачі. В результаті було обрано високорівневу мову програмування JavaScript, тому що саме вона містить в собі, всі необхідні для реалізації сучасного веб-додатку бібліотеки та методи.

Після цього було розроблено діаграми: класів, варіантів використання та прецедентів. Вони надали змогу побачити загальну картину функціонування програмного забезпечення.

Результатом виконання індивідуального дослідницького проєкту є розроблене програмне забезпечення, що має можливість майже повністю автоматизувати процес навчання та допомогу у освоєнні усім особам які беруть участь у навчанні. Розроблений застосунок повністю відповідає поставленим технічним умовам, та може бути використаний у вищих навчальних закладах за задуманим призначенням.

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		59

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. School Management Software / Software advice. URL: <https://www.softwareadvice.com/school-management/>
2. JavaScript frameworks / Trio. URL: <https://www.trio.dev/blog/javascript-framework>
3. Css frameworks / Browser stack. URL: <https://www.browserstack.com/guide/top-css-frameworks>
4. What is Firebase / Medium. URL: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>
5. Modern Web Application Architecture to Build a High-Performance App / Acropolium Blog. URL: <https://acropolium.com/blog/modern-web-app-architecture/>
6. Web Application Architecture / Litslink. URL: <https://litslink.com/blog/web-application-architecture>
7. What is Web Application Architecture / Hackr.io. URL: <https://hackr.io/blog/web-application-architecture-definition-models-types-and-more#:~:text=A%20web%20application%20architecture%20is,the%20client%20and%20server%20sides.>
8. Creating a Project/ Vue cli. URL: <https://cli.vuejs.org/guide/creating-a-project.html>
9. Babel / Документація. URL: <https://babeljs.io/>
10. Vue Router / Vue router. URL: <https://router.vuejs.org/>
11. Vuex / Vuex. URL: <https://vuex.vuejs.org/>
12. Add Firebase to your JavaScript project / Документація . URL: <https://firebase.google.com/docs/web/setup>
13. Read and Write Data on the Web / Firebase. URL: <https://firebase.google.com/docs/database/web/read-and-write>
14. Authenticate with Firebase / Firebase. URL: <https://firebase.google.com/docs/auth/web/password-auth>

					ІА82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		60

15. Responsive Design / Tailwind. URL: <https://tailwindcss.com/docs/responsive-design>

					IA82.020БАК.003 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		61