

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

«Платформа для розробки веб додатків Rocketi»

Виконав студент групи ІПЗ-21-4
ІГЛІНСЬКИЙ Іван Юрійович

Керівник роботи:
ЄФРЕМОВ Юрій Миколайович

Рецензент:
ЛОКТИКОВА Тамара Миколаївна

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна Вакалюк
«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ІГЛІНСЬКИЙ Іван Юрійович**

Керівник роботи: **ЄФРЕМОВ Юрій Миколайович**

Тема роботи: **«Платформа для розробки веб додатків Rocketi»,**
затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**
Вихідні дані для роботи: **платформа для розробки в контексті Web 3.0, який**
використовує сучасні технології для зберігання та розгортки додатків
Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів,
що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Єфремов Ю.М.	27.02.2025р.	27.02.2025р.
2	Єфремов Ю.М.	27.02.2025р.	27.02.2025р.
3	Єфремов Ю.М.	27.02.2025р.	27.02.2025р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з платформи для розробки веб додатків “Rocketi” та пояснювальної записки. Пояснювальна записка до випускної роботи містить 72 сторінок, 23 ілюстрацій, 11 таблиць.

Метою роботи є створення платформи для розробки веб додатків з використанням сучасних технологій та інтеграцією сучасних хмарних сервісів для розгортання веб додатків.

Архітектура платформи базується на хмарному сервісі Rocketbase, що забезпечує надійність, захист та швидкість майбутніх додатків.

Додаток має зручний інтерфейс, який дозволяє користувачам легко розгортати нові додатки, підтримувати, редагувати та адмініструвати старі додатки.

КЛЮЧОВІ СЛОВА: ВЕБПЛАТФОРМА, ХМАРА, ТЕХНОЛОГІЇ, ЗАХИСТ, РОЗРОБКА, ШВИДКІСТЬ, НАДІЙНІСТЬ.

		Іглінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

The bachelor's final qualification work consists of a platform for developing web applications “Pocketi” and an explanatory note. The explanatory note to the final work contains 72 pages, 23 illustrations and 11 tables.

The method of work is to create a platform for developing web applications using modern technologies and integrating modern cloud services for deploying web applications.

The platform architecture is based on the Pocketbase cloud service, which ensures reliability, security and speed of future applications.

The application has a user-friendly interface that allows users to easily deploy new applications, maintain, edit and administer old applications.

KEYWORDS: WEB PLATFORM, CLOUD, TECHNOLOGIES, SECURITY, DEVELOPMENT, SPEED, RELIABILITY.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ.....	2
ABSTRACT	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ПЛАТФОРМИ ДЛЯ РОЗРОБКИ ДОДАТКІВ.....	8
1.1 Постановка задачі	8
1.2 Аналіз аналогів програмного продукту	11
1.3 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення.....	17
Висновки до першого розділу	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПЛАТФОРМИ ДЛЯ РОЗРОБКИ КОДУ	21
2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи.....	21
2.2 Розробка бази даних	26
2.3 Проектування та реалізація алгоритмів роботи системи	28
2.4 Реалізація функціоналу платформи для розробки коду	30
Висновки до другого розділу.....	31
РОЗДІЛ 3. ПОРЯДОК РОБОТИ ТА ІНТЕРФЕЙС ДОДАТКУ ДЛЯ РОЗРОБКИ КОДУ	33
3.1. Порядок встановлення та налаштування параметрів платформи	33
3.2. Структура інтерфейсу платформи для розробки коду	34
3.3. Статистика та адміністрування платформи	47
Висновки до третього розділу	50
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API (Application Programming Interface) – інтерфейс програмування застосунків, що забезпечує взаємодію між клієнтською частиною платформи та сервером.

JWT (JSON Web Token) – токен у форматі JSON, що використовується для авторизації користувачів у системі.

CRUD (Create, Read, Update, Delete) – основні операції для роботи з даними: створення, читання, оновлення та видалення записів.

ID (Identifier) – унікальний ідентифікатор об'єкта в базі даних (користувача, проєкту, коментаря тощо).

UI (User Interface) – інтерфейс користувача, візуальна частина додатку для взаємодії з системою.

UX (User Experience) – досвід користувача під час роботи з платформою.

DB (Database) – база даних для збереження інформації про користувачів, проєкти, шаблони, коментарі та аналітичні дані.

		Іглінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Rocketi — це сучасна платформа для створення веб-додатків, яка допомагає розробникам ефективно та швидко реалізовувати свої ідеї, скорочуючи витрати часу та ресурсів. Вона поєднує зручний інтерфейс, модульну архітектуру та підтримку сучасних фреймворків. Завдяки цьому платформа значно спрощує процес розробки та розгортання веб-додатків навіть для користувачів без глибоких технічних знань.

Завдяки Rocketi, розробка стає більш доступною, а робочий процес — більш гнучким. Платформа підтримує інтеграцію з зовнішніми сервісами, адаптацію під мобільні пристрої та налаштування хостингу без потреби вручну керувати інфраструктурою. Це дозволяє командам зосередитися на функціональності продукту замість технічних аспектів реалізації.

Метою випускної кваліфікаційної роботи бакалавра є створення платформи для розробки веб-додатків Rocketi, яка дозволяє автоматизувати процеси проєктування, розробки та розгортання веб-додатків.

Для реалізації поставленої мети потрібно виконати наступні завдання: Спочатку потрібно вивчити сучасні інструменти та платформи для веб-розробки, щоб зрозуміти їхні сильні та слабкі сторони. Далі визначити, якою має бути архітектура і які функції повинна мати платформа Rocketi. Після цього розробити саму платформу з інтуїтивно зрозумілим інтерфейсом, що підтримує шаблони, модулі і можливість розміщення проєктів на хостингу.

Об'єктом дослідження є процес розробки веб-додатків за допомогою сучасних low-code та full-stack інструментів.

Предметом дослідження є функціональність платформи Rocketi як інструменту автоматизованої розробки веб-додатків, її можливості для індивідуальних розробників і команд, а також її переваги у порівнянні з іншими платформами.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

Дослідження також охоплює аналіз інтерфейсних рішень, архітектурної моделі платформи, принципів модульності та масштабованості, а також ефективність розробки проєктів на базі Rocketi у порівнянні з традиційними методами веб-розробки.

		Ієлінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ПЛАТФОРМИ ДЛЯ РОЗРОБКИ ДОДАТКІВ

1.1 Постановка задачі

Актуальність теми полягає у зростаючій потребі створення інструментів, які спрощують та прискорюють розробку сучасних веб-додатків. У добу стрімкого розвитку цифрових технологій, важливою є наявність доступних платформ, які дозволяють як досвідченим розробникам, так і початківцям ефективно створювати веб-застосунки без необхідності глибокого знання низькорівневих технологій. Платформа Rocketi створена з метою надати інтуїтивний, гнучкий та масштабований інструмент для побудови веб-додатків з використанням сучасних підходів і архітектур.

Метою випускної кваліфікаційної роботи бакалавра є розробка веб-платформи Rocketi для створення, редагування та публікації веб-додатків з використанням передових вебтехнологій.

Щоб досягти поставленої мети, необхідно виконати низку послідовних кроків. Спершу варто проаналізувати існуючі платформи для веб-розробки — дослідити їхні функціональні можливості, переваги, недоліки та архітектурні підходи. Це дозволить визначити найкращі практики та уникнути типових помилок. Далі слід обрати інструменти, фреймворки та бібліотеки, які будуть використовуватись під час створення платформи Rocketi. Вибір повинен базуватись на технічних вимогах, зручності підтримки, можливості масштабування та популярності серед розробників.

Наступним кроком є розробка архітектури системи з урахуванням її майбутнього зростання, модульного підходу та безпеки. Архітектура має бути гнучкою, щоб забезпечити легку інтеграцію нових компонентів та захищеною для безпечної роботи користувачів.

Після цього потрібно створити інтерфейс користувача та систему управління додатками. Особливу увагу слід приділити реалізації авторизації,

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

розподілу ролей і налаштуванню прав доступу для різних категорій користувачів.

Завершальним етапом стане тестування функціоналу платформи та оцінка її продуктивності. Це дозволить переконатися, що всі компоненти працюють коректно, система витримує навантаження і готова до використання в реальних умовах.

Платформа Rocketi включає кілька ключових модулів, кожен з яких виконує важливу функцію у процесі створення та керування веб-додатками.

Один з основних елементів — модуль створення додатків. Він надає користувачам можливість створювати власні веб-додатки з нуля або використовуючи готові шаблони. За допомогою цього модуля можна працювати з різними інтерфейсними компонентами, такими як форми, кнопки, таблиці та інші елементи, необхідні для побудови повноцінного інтерфейсу.

Модуль редагування та конфігурації відповідає за зручне налаштування поведінки додатку, зовнішнього вигляду його елементів і логічних зв'язків між ними. Це дозволяє користувачам змінювати структуру і функціональність додатків без потреби в програмуванні.

Збереження і публікація додатків відбувається через спеціальний модуль, який фіксує зміни в базі даних і забезпечує розгортання створених додатків, щоб вони стали доступними для використання в інтернеті.

За безпеку та обмеження доступу відповідає модуль авторизації. Він реалізує автентифікацію користувачів, керування ролями та правами доступу, що гарантує захист даних і контроль над тим, хто і які дії може виконувати в системі.

Нарешті, модуль адміністрування призначений для адміністраторів платформи. З його допомогою можна переглядати статистику та аналітику використання системи, керувати обліковими записами користувачів і контролювати технічний стан платформи.

		Гелінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Основні етапи розробки платформи Rocketi:

Розробка платформи Rocketi проходить через кілька ключових етапів, які забезпечують її якісну реалізацію та подальше ефективне функціонування.

На початковому етапі проводиться аналіз вимог до системи. Це включає вивчення потреб користувачів, визначення функціональних можливостей та створення прототипів інтерфейсу, які допомагають візуалізувати майбутній вигляд платформи.

Далі формується архітектура системи — як для фронтенду, так і для бекенду. Визначаються основні модулі, способи їх взаємодії, принципи масштабування та організації коду.

Після цього починається реалізація компонентів користувацького інтерфейсу. Створюються візуальні елементи, з якими взаємодіє користувач, і забезпечується їхня зручність та адаптивність до різних пристроїв.

Паралельно розробляється серверна частина, де реалізується бізнес-логіка платформи — обробка запитів, робота з базою даних, перевірка прав доступу тощо.

Для ефективного обміну даними між клієнтською та серверною частинами розробляється API, який забезпечує швидку та безпечну взаємодію між компонентами системи.

Особливу увагу приділяють безпеці: впроваджується надійна система аутентифікації користувачів та управління правами доступу, щоб захистити дані і запобігти несанкціонованому доступу.

Перед запуском платформи проводиться ретельне тестування: перевіряється функціональність, зручність використання та стабільність роботи під навантаженням.

Завершальним етапом є підготовка технічної документації, яка містить опис архітектури, інструкції з використання та інформацію для подальшого супроводу і розвитку платформи.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Результатом виконання роботи стане веб-платформа Rocketi, яка дозволяє швидко створювати та розгортати веб-застосунки, спираючись на сучасні вебтехнології, з інтуїтивно зрозумілим інтерфейсом і гнучкою системою конфігурацій.

1.2 Аналіз аналогів програмного продукту

Розробка платформи Rocketi потребує глибокого аналізу конкурентів на ринку. Оскільки створення інтегрованих середовищ для розробки веб-додатків (англ. web app builders) є популярною тенденцією, перед початком реалізації платформи доцільно провести огляд існуючих рішень, що працюють на основі технологій Web 2.0.

Однією з таких платформ є Bubble (рис. 1.1) — один із найвідоміших no-code конструкторів для створення веб-додатків. Платформа дозволяє створювати повнофункціональні веб-застосунки без написання коду, використовуючи візуальне середовище для компоновання інтерфейсу та налаштування логіки. [7]

Платформа Bubble пропонує широкий набір можливостей для створення веб-додатків без необхідності програмування. Однією з основних функцій є візуальний редактор інтерфейсу, що дозволяє розміщувати елементи на сторінці за допомогою drag-and-drop, значно спрощуючи процес розробки.

Bubble підтримує інтеграцію з зовнішніми сервісами через API — як за допомогою REST-запитів, так і через підключення плагінів. Крім того, для більш гнучких і кастомізованих рішень платформа дозволяє розширювати функціональність за допомогою власних плагінів і JavaScript-коду. Це дає змогу розробникам реалізовувати специфічну бізнес-логіку, налаштовувати обробку даних у реальному часі та взаємодіяти з нестандартними сторонніми сервісами.

Завдяки цьому Bubble стає потужним інструментом для створення повнофункціональних веб-застосунків без необхідності в повноцінному бекенді. [23]

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

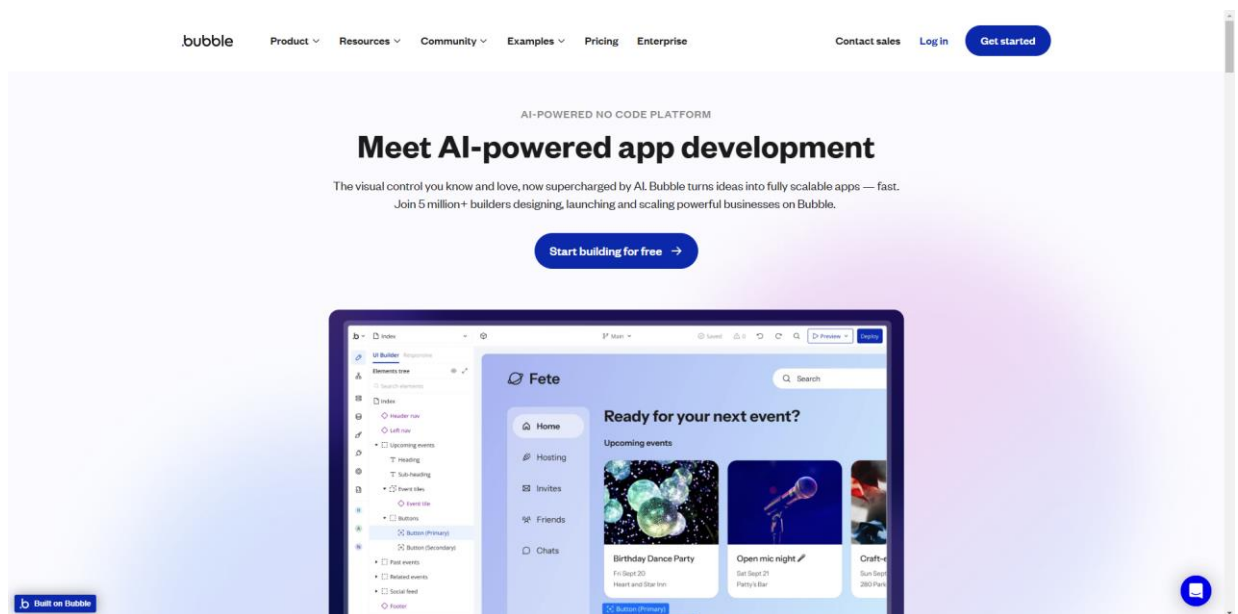


Рис. 1.1 Платформа Bubble.io

Bubble — одна з найпопулярніших no-code платформ, яка дозволяє створювати повноцінні вебдодатки без необхідності програмування. Вона надає користувачам візуальний інтерфейс для створення інтерфейсів, логіки додатку, роботи з базами даних та інтеграцією зовнішніх API.

Робота з базами даних також максимально спрощена — користувачі можуть створювати, зберігати та використовувати дані без потреби писати SQL-запити. Уся логіка роботи додатку реалізується за допомогою системи “work-flow”, яка дозволяє будувати послідовності дій і реакцій на події.

Серед основних переваг Bubble — можливість створювати повноцінні складні додатки, зокрема SaaS-сервіси, CRM-системи та маркетплейси. Платформа має активну спільноту користувачів і багатий маркетплейс готових плагінів. До того ж, хостинг і бекенд вже інтегровані у платформу, тож користувачеві не потрібно шукати окремі сервери чи налаштовувати інфраструктуру.

Втім, є й певні недоліки. Інтерфейс Bubble досить складний для новачків і потребує часу на освоєння. Крім того, розробники прив’язані до інфраструктури самої платформи, що ускладнює перенесення проєкту на інші сервіси та обмежує

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

масштабованість. Ще одне обмеження — це продуктивність: при великій кількості одночасних користувачів можуть виникати проблеми з оптимізацією швидкодії.

Таким чином, хоча Bubble і є потужним інструментом, Rocketi має на меті запропонувати простішу альтернативу, яка не потребує довгого навчання та дозволяє користувачам гнучко працювати з сучасними фреймворками.

FlutterFlow — платформа для створення мобільних і вебдодатків на базі фреймворку Flutter від Google (див. рис. 1.2). [4]

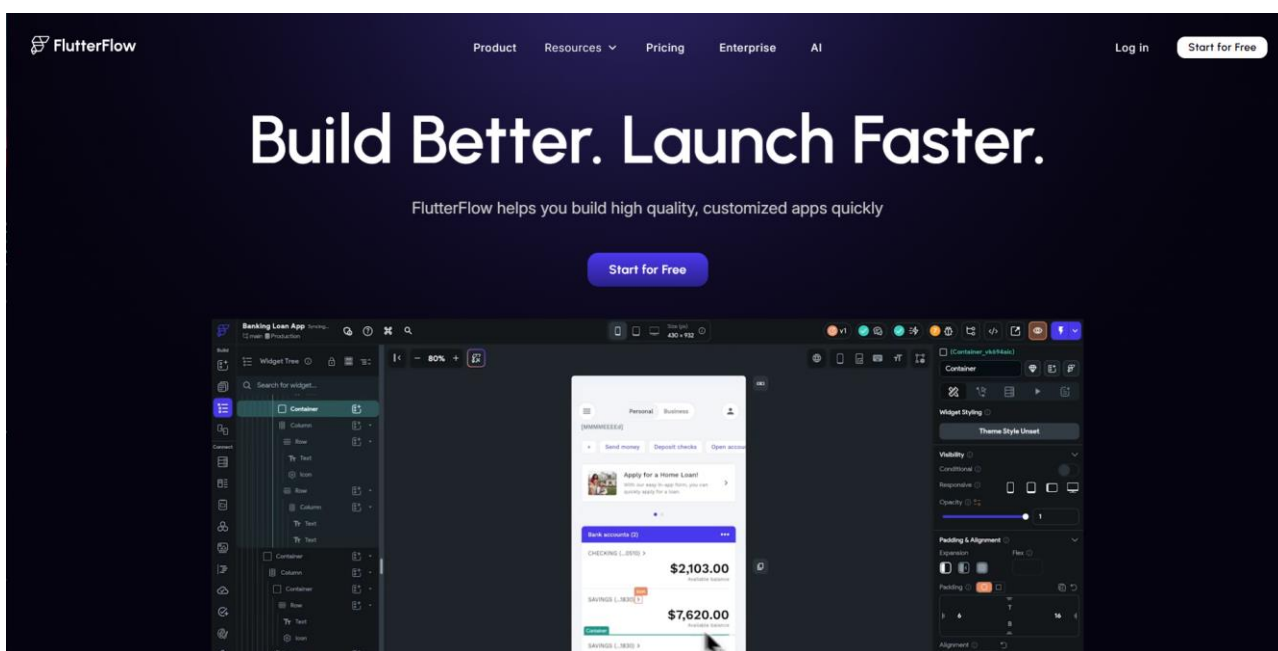


Рис.1.2 – платформа Flutterflow

FlutterFlow має ряд унікальних особливостей, які роблять його потужним інструментом для розробки мобільних та веб-додатків. Основою платформи є UI Builder, що базується на компонентах Flutter — це дозволяє створювати привабливі, сучасні та адаптивні інтерфейси без необхідності вручну писати код.

Ще однією сильною стороною є можливість експорту повного Flutter-коду, що відкриває доступ до повного контролю над додатком. [9] Користувачі можуть редагувати код самостійно, коли стандартних інструментів стає недостатньо. Платформа має глибоку інтеграцію з Firebase — підтримується автентифікація, робота з базами даних, хмарні функції та інші сервіси бездодаткового

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

налаштування. Також у FlutterFlow є можливість створення власних функцій за допомогою мови Dart — це дозволяє реалізовувати унікальну логіку без виходу за межі середовища. [24]

Завдяки підтримці responsive design, розроблені інтерфейси автоматично адаптуються до різних розмірів екранів, що є критично важливим для кросплатформених проєктів.

Серед переваг варто відзначити високу продуктивність додатків, яка досягається завдяки використанню Flutter. Крім того, створені проєкти можна публікувати одразу на iOS, Android і навіть у веб-версії, що значно розширює можливості використання. Поєднання візуального редактора з можливістю вручну змінювати код — це оптимальне рішення для розробників з різним рівнем досвіду.

Проте, варто враховувати і деякі недоліки. Незважаючи на те, що FlutterFlow позиціонується як low-code платформа, для реалізації складніших рішень знання Dart буде необхідним. Інтерфейс може здатися складнішим у порівнянні з Bubble, особливо для новачків. Також платформа в першу чергу орієнтована на створення мобільних додатків, хоча підтримка веб-додатків також є.

Webflow — потужний інструмент для дизайнерів, які хочуть створювати вебсайти з точним контролем за візуальним виглядом без написання коду (Рис. 1.3). Однією з ключових функцій є CMS-система, яка дає змогу створювати структури даних (колекції), що легко інтегруються зі сторінками. Це зручно для створення блогів, портфоліо, каталогів та інших типів контенту, який оновлюється. [5]

Серед переваг варто виділити інтуїтивно зрозумілий інтерфейс, який дозволяє швидко адаптуватися навіть без досвіду програмування. Також Webflow пропонує велику гнучкість у стилізації та генерацію чистого, SEO-оптимізованого коду, що забезпечує швидке завантаження сторінок.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

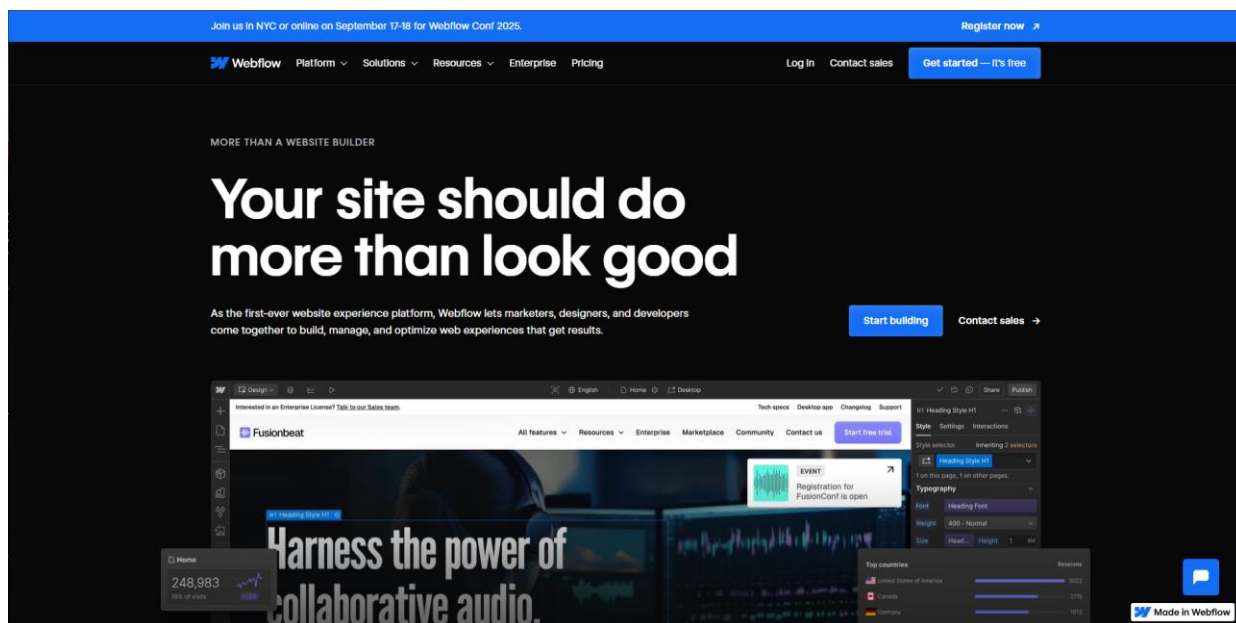


Рис.1.3- платформа Webflow

Webflow вирізняється з-поміж інших платформ своєю орієнтацією на професійний веб-дизайн, поєднуючи інструменти візуального редагування з точним контролем стилів, як у CSS та HTML. Його UI-редактор побудований таким чином, що кожен елемент має власні класи, стилі та позиціонування — це наближає досвід користування до знайомого дизайнерам середовища, як-от Figma чи Photoshop.

Платформа також підтримує створення анімацій та мікроваємодій без написання коду — це відкриває нові можливості для створення динамічних та привабливих інтерфейсів. Крім того, Webflow дозволяє експортувати готовий HTML, CSS та JavaScript-код, що робить його зручним для тих, хто хоче розміщувати сайт на власному сервері.

Втім, є й обмеження. Webflow не підходить для розробки додатків зі складною логікою, авторизацією або інтеграцією з повноцінними базами даних, як це можливо у Bubble.

OutSystems — це high-end low-code платформа, що орієнтована на корпоративний ринок. [6]

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

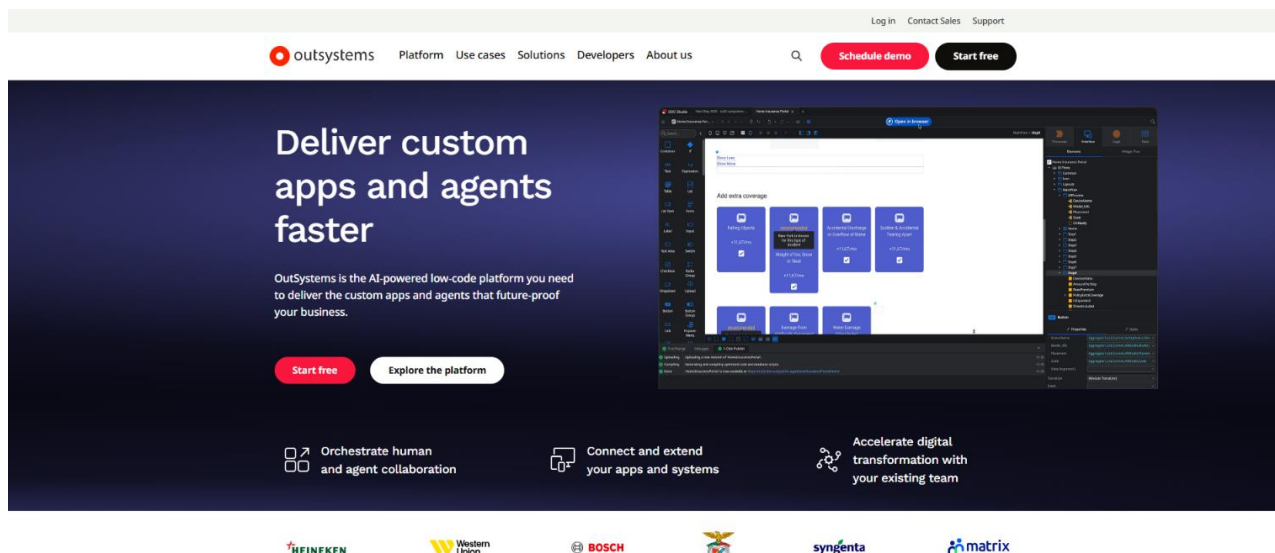


Рис. 1.4 – платформа OutSystems

Однією з ключових переваг є інтеграція з популярними корпоративними базами даних, такими як Oracle, SQL Server чи SAP.[10, 11, 12] Платформа підтримує повноцінний DevOps-підхід: забезпечує безперервну інтеграцію й розгортання (CI/CD), контроль версій і керування середовищами розробки та продакшену. [20]

Вбудований інструмент для побудови логіки — візуальний редактор — дозволяє створювати складні бізнес-процеси без написання великої кількості коду. Його функціональність нагадує workflow-систему Bubble, але з акцентом на масштабованість і складність процесів.

OutSystems також відповідає найвищим стандартам безпеки — підтримує вимоги GDPR, SOC 2 та HIPAA. Це дозволяє застосовувати платформу в галузях, де обробляються конфіденційні дані, як-от у медицині, банківській чи урядовій сферах.

До головних переваг платформи належать стабільна технічна підтримка для підприємств, здатність масштабуватись під навантаження великих організацій, а також гнучка інтеграція з зовнішніми системами й сервісами.

Однак, як і будь-яке професійне рішення, OutSystems має і недоліки. Зокрема, висока вартість ліцензії може бути бар'єром для невеликих команд. Попри концепцію low-code, для роботи з платформою все одно потрібна команда

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

розробників або архітекторів. Крім того, офіційна документація більше орієнтована на досвідчених бізнес-аналітиків та технічних спеціалістів, а не на новачків.

Виходячи з наявних переваг та недоліків, складемо порівняльну таблицю вищезгаданих програмних аналогів (табл. 1.1).

Таблиця 1.1

Таблиця порівняння платформ для розробки

Платформа	Рівень коду	Основна аудиторія	Сфера застосування	Експорт коду	Складність	Підтримка Web3
Bubble	No-code	Стартапи, фрілансери	SaaS, маркетплейси	✗	Висока	✗
Webflow	No-code	Дизайнери, маркетологи	Сайти, лендинги, CMS	✓	Середня	✗
FlutterFlow	Low-code	Розробники мобільних додатків	Мобільні додатки, вебдодатки	✓	Висока	✗
OutSystems	Low-code	Великі компанії	ERP, бізнес-процеси	✗	Висока	✗
Pocketi	No-code	Початківці, малі команди	Вебдодатки з Web 3.0	✓	Низька	✓

1.3 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Вебдодаток Pocketi буде побудований на основі клієнт-серверної архітектури.

Для розробки будуть використані сучасні технології, що відповідають вимогам високої продуктивності, безпеки, масштабованості та простоти підтримки.

Серверна частина (бекенд) платформи буде створена у вигляді REST API за допомогою фреймворку NestJS. Клієнтська частина (фронтенд) буде розроблена з використанням NextJS, який базується на ReactJS. Для зберігання даних

		Гелінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

про кампанії, користувачів і транзакції використовуватиметься реляційна база даних PostgreSQL. [1, 16, 3]

Таблиця 1.2

Таблиця технологічного стеку Pocketi

Компонент	Технологія	Призначення
Серверна частина	NestJS	Розробка API та логіки додатку
Клієнтська частина	NextJS (ReactJS)	Побудова інтерфейсу користувача (UI)
База даних	PostgreSQL	Сховище даних про збори і користувачів

Використані технології

Next.js — це один із найпопулярніших фреймворків для створення вебзастосунків на базі React, який чудово підходить для реалізації сучасних інтерфейсів, таких як Pocketi. Його головною перевагою є підтримка Server-Side Rendering (SSR), що дозволяє рендерити сторінки на сервері перед їх передачею користувачу. Це значно покращує швидкість завантаження та SEO-оптимізацію, що особливо важливо для публічних кампаній зборів коштів. [1]

Також Next.js підтримує Static Site Generation (SSG), тобто можливість попередньої генерації сторінок під час збірки. Це знижує навантаження на сервер і пришвидшує доступ до контенту для користувачів.

Фреймворк має вбудовану підтримку TypeScript, що забезпечує кращу організацію коду, статичну типізацію і підвищує надійність розробки. Завдяки гнучкій системі рендерингу — SSR, SSG і CSR (клієнтський рендеринг) — Next.js дозволяє створювати інтерфейс, який динамічно адаптується до різних сценаріїв використання. [2, 8]

Крім того, платформа легко розширюється: велика екосистема модулів, плагінів і готових рішень дозволяє швидко інтегрувати додаткову функціональність без зайвих зусиль. Це робить Next.js надійною основою для фронтенду Rocketi.

NextJS допоможе створити швидкий, інтуїтивно зрозумілий і доступний інтерфейс для користувачів Rocketi.

PostgreSQL (Postgres) — це потужна реляційна система управління базами даних з відкритим кодом. PostgreSQL є потужною системою управління базами даних, яка ідеально підходить для платформи Rocketi. Вона підтримує стандарт SQL, що дозволяє виконувати складні запити та працювати з реляційними зв'язками між таблицями, забезпечуючи гнучкість у роботі з даними. [16]

Однією з ключових переваг PostgreSQL є її надійність та безпека. Система підтримує шифрування даних, контроль доступу користувачів, а також має високу стійкість до збоїв, що гарантує захист важливої інформації.

Крім того, PostgreSQL забезпечує гнучкість завдяки підтримці зберігання даних у форматі JSON, а також можливості повнотекстового пошуку, що дозволяє швидко знаходити потрібну інформацію.

Важливою особливістю є розширюваність системи — можна створювати власні функції та типи даних, адаптуючи базу під специфічні потреби платформи. Загалом, PostgreSQL гарантує надійне зберігання інформації про користувачів, кампанії зі збору коштів та транзакції на Rocketi.

Технологічний стек платформи Rocketi орієнтований на забезпечення безпеки транзакцій, високу продуктивність, зручний і дружній користувачеві інтерфейс, а також гнучкість у масштабуванні системи.

Це дозволить створити сучасну децентралізовану платформу, яка відповідає всім вимогам епохи Web 3.0.

Висновки до першого розділу

У першому розділі було проведено аналіз актуальності розробки

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

платформи Rocketi для створення веб-додатків, визначено основні завдання, об'єкт і предмет дослідження. Було детально розглянуто існуючі аналогічні рішення на ринку — такі як Bubble, FlutterFlow, Webflow та OutSystems, їх сильні та слабкі сторони, що дозволило обґрунтувати унікальні переваги Rocketi: простоту використання, підтримку Web 3.0 та можливість експорту коду.

Також було обґрунтовано вибір технологічного стеку для реалізації платформи. Використання NestJS на серверній частині забезпечить побудову надійного, модульного і масштабованого API. Фреймворк NextJS дозволить створити сучасний, швидкий і SEO-оптимізований інтерфейс користувача. Для зберігання даних обрано реляційну базу даних PostgreSQL, що забезпечить надійність, гнучкість і продуктивність роботи з інформацією.

Таким чином, на основі проведеного аналізу було визначено концепцію, архітектуру та інструменти для подальшої розробки платформи Rocketi, яка має на меті забезпечити користувачам інтуїтивний процес створення сучасних веб-додатків.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПЛАТФОРМИ ДЛЯ РОЗРОБКИ КОДУ

2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

Платформа Pocketi надає вебінтерфейс, за допомогою якого користувачі можуть створювати власні веб-додатки без необхідності програмування. Функціональність системи залежить від активності користувачів, які будують, налаштовують і публікують свої проекти. Користувачі поділяються на дві основні групи: творці додатків, які розробляють нові продукти, та кінцеві користувачі, які взаємодіють із цими додатками. Проекти містять детальну інформацію про функціональність, налаштування інтерфейсу, інтеграції та інші параметри, що дозволяють створити повноцінний сучасний веб-додаток. Для наочного відображення взаємодії між основними елементами системи було побудовано Діаграму варіантів використання (див. рис. 2.1).

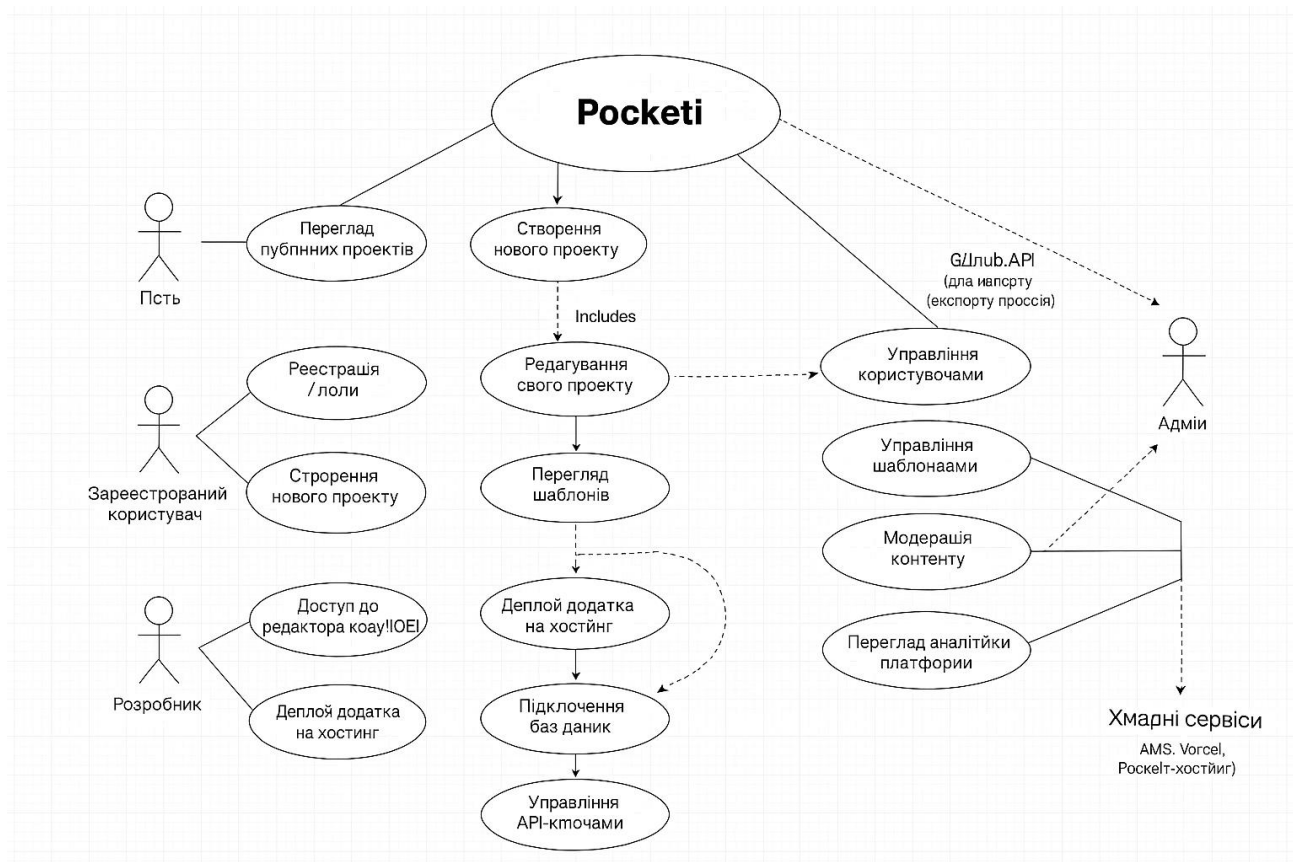


Рис. 2.1. Діаграма варіантів використання додатку

Перш за все, користувачі можуть зареєструватися або увійти в систему.

		Ієлінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

Ця функція доступна як для гостей, так і для вже зареєстрованих користувачів. Реєстрація та вхід дають змогу отримати доступ до основного функціоналу платформи — створення і управління веб-додатками.

Після авторизації користувачі мають можливість створювати нові проєкти. Для цього вони можуть скористатися готовими шаблонами або ж налаштувати проєкти під свої потреби самостійно. Такий підхід дозволяє швидко почати роботу або створити унікальний продукт з нуля.

Користувачі також можуть переглядати бібліотеку готових шаблонів веб-додатків. Це дає змогу обирати вже розроблені основи, які можна використовувати для власних проєктів, що значно спрощує і прискорює процес розробки.

Розробникам доступний вбудований редактор коду, який дозволяє безпосередньо змінювати програмний код проєкту в реальному часі. Окрім того, вони можуть розгортати свої проєкти на хостингу, одразу перевіряючи роботу своїх додатків у живому середовищі.

Також платформа дає змогу розробникам управляти API-ключами та базами даних, що необхідно для підключення сторонніх сервісів або збереження інформації у власних проєктах. Це забезпечує гнучкість у роботі з різними технологіями та ресурсами.

Адміністратори платформи мають можливість керувати обліковими записами користувачів, що включає редагування, блокування чи видалення акаунтів. Вони також контролюють набір доступних шаблонів, оновлюючи їх або додаючи нові, щоб підтримувати актуальність і якість ресурсів для користувачів. Окрім цього, адміністраторам доступна детальна аналітика та звітність по використанню платформи. Вони можуть відслідковувати статистику активності користувачів, популярність шаблонів, навантаження на систему, що дозволяє ефективно управляти ресурсами і планувати подальший розвиток платформи.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, кожен тип користувачів — від гостей і розробників до адміністраторів — має свої чітко визначені функції і можливості, що забезпечує зручність, безпеку та ефективність роботи в системі Rocketi.

Основна ціль розробки платформи Rocketi полягає у створенні універсального середовища для розробки веб-додатків, яке об'єднує готові шаблони, інструменти кодування, автоматизоване розгортання та керування ресурсами. Платформа орієнтована на забезпечення швидкого старту, зручного налаштування проєктів та підтримки як новачків, так і професійних розробників.

Основна мета розробки платформи Rocketi: Мета полягає в створенні інноваційної веб-платформи, яка забезпечить користувачів зручними інструментами для швидкої розробки, тестування, розгортання та управління веб-додатками з використанням готових шаблонів, інтеграцій та хмарних сервісів.

Для адміністратора платформи важливо мати можливість керувати користувачами, додавати і змінювати шаблони для веб-додатків, дивитися статистику використання системи та перевіряти проєкти, які публікують інші користувачі.

Розробники, які зареєстровані на платформі, можуть увійти через звичайний логін з електронною поштою або за допомогою Web 3.0. Вони створюють нові проєкти, базуючись на готових шаблонах або власному коді, можуть редагувати свої проєкти, публікувати їх для інших, а також переглядати статистику по відвідуваннях своїх додатків.

Гості, які не мають акаунту, можуть переглядати публічні проєкти, тестувати демонстрації без входу в систему. Щодо функціональності, платформа повинна підтримувати два способи входу: традиційний за допомогою email і пароля. Користувачі мають мати можливість створювати і зберігати свої веб-проєкти, користуватися редактором коду та публікувати додатки. Важливо, щоб вони могли спілкуватися через коментарі під шаблонами і проєктами, а також легко користуватися готовими шаблонами. Для адміністраторів і розробників доступна аналітика, що показує,

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

як використовується платформа. З точки зору зручності, платформа має бути простою і швидкою у освоєнні — новачки повинні розібратися за 3–5 хвилин, а досвідчені користувачі — навіть швидше. Відповідь на дії користувачів повинна бути миттєвою, за винятком важких операцій, де час очікування може сягати півтори хвилини. Інтерфейс повинен бути інтуїтивно зрозумілим і не вимагати спеціальної підготовки.

Особлива увага повинна приділятися UX/UI-дизайну: важливі елементи інтерфейсу мають бути візуально виділені, кнопки та навігаційні компоненти повинні мати логічне розташування, а форма подачі інформації — бути зрозумілою навіть для користувачів без досвіду роботи з подібними системами.

База даних, орієнтована на зберігання документів, наприклад PostgreSQL, дає змогу зберігати структури даних у форматі JSON, що забезпечує високу гнучкість і масштабованість при роботі з динамічними об'єктами, які можуть змінюватися залежно від логіки застосунку. Платформа повинна працювати безперервно — 24 години на добу, 7 днів на тиждень. Вона має витримувати одночасну роботу багатьох користувачів без збоїв і уповільнень.

Для комфортної роботи потрібен комп'ютер з мінімум 2 ГБ оперативної пам'яті, процесор з частотою не менше 1,4 GHz та стабільний інтернет. Сервер платформи повинен використовувати базу даних, орієнтовану на зберігання документів, підтримувати технології Web 3.0 для анонімної авторизації і працювати на Node.js для бекенд-логіки. [13]

Аналіз функціональних вимог дозволив виділити основні сутності, необхідні для реалізації платформи. На (рис 2.2) буде наведено діаграму класів, що використовується для роботи додатку.

Крім того, сервер має бути масштабованим для обробки великої кількості запитів і мати систему кешування для підвищення швидкодії. Усі компоненти платформи повинні бути інтегровані через API, що забезпечує гнучкість та модульність архітектури.

		Гелінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

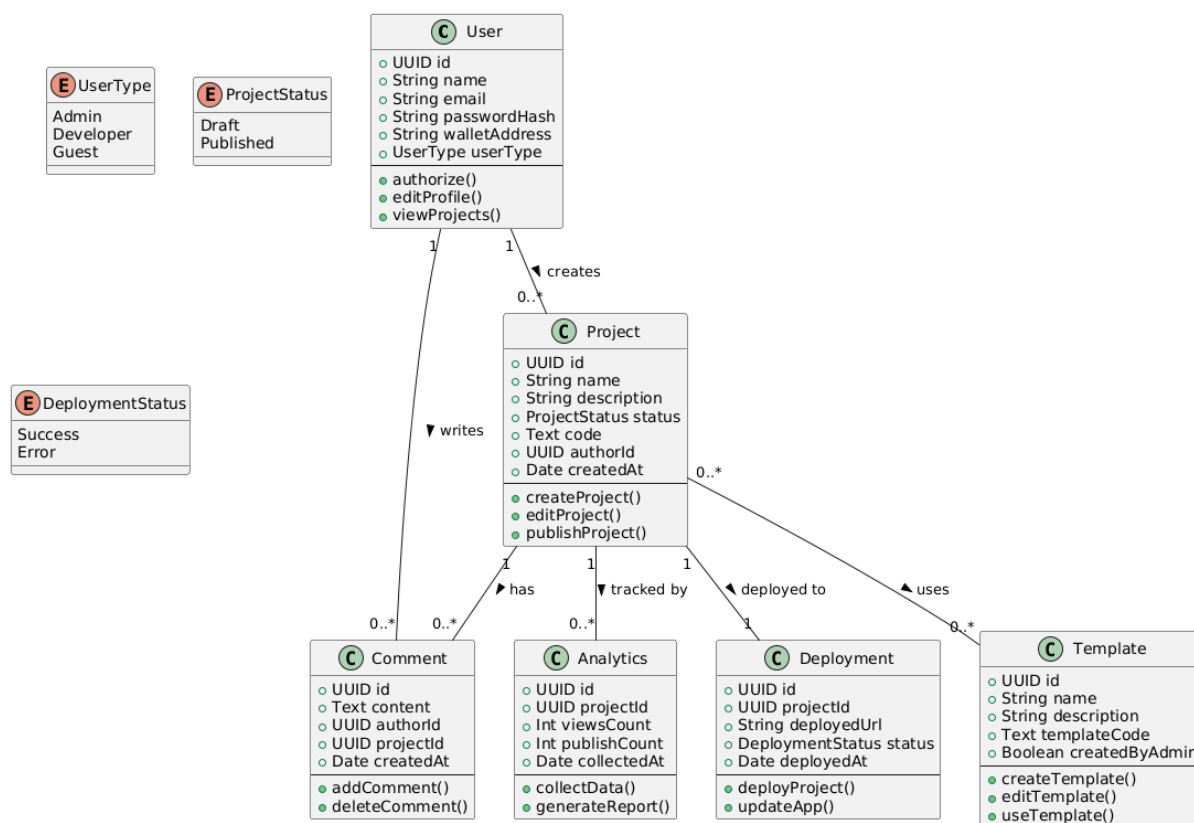


Рис.2.2 – діаграма класів

Опишемо діаграму класів (див. рис. 2.2)

Користувачі мають унікальний ідентифікатор, ім'я, електронну пошту та захищений пароль. Користувачі можуть бути різних типів: адміністратори, розробники або гості. Вони можуть входити в систему, редагувати свій профіль і переглядати свої проєкти.

Проєкти — це створені користувачами веб-додатки або продукти. У кожного проєкту є своя назва, опис, статус (наприклад, чернетка або опублікований), а також код, який може зберігатися в репозиторії або безпосередньо у системі. Кожен проєкт належить конкретному автору і має дату створення. Користувачі можуть створювати нові проєкти, редагувати їх і публікувати для загального доступу.

Шаблони — це готові заздалегідь створені структури або коди для проєктів, які додають адміністратори. Вони мають назву, опис і свій код.

Адміністратори можуть створювати, редагувати ці шаблони, а розробники використовують їх для своїх проєктів.

Коментарі — це тексти, які користувачі залишають під проєктами. Кожен коментар має автора, дату створення та належить певному проєкту. Користувачі можуть додавати свої коментарі і видаляти ті, що вони залишили раніше.

Аналітика збирає інформацію про те, скільки разів переглядають проєкти, скільки разів вони були опубліковані і коли відбувався збір цих даних. Ця інформація допомагає формувати звіти, які показують активність і популярність проєктів.

Розгортання — це процес запуску проєкту в інтернеті, щоб він став доступним користувачам. Для кожного розгортання зберігається посилання на додаток, статус (чи все пройшло успішно або сталася помилка) і дата розгортання. Користувачі можуть запускати свої проєкти в роботу і оновлювати вже запуснені додатки.

2.2 Розробка бази даних

Для збереження файлів та інформації у додатку використовується PostgreSQL — система управління базами даних, що базується на реляційній моделі. У системі є шість основних категорій даних. Перша — це користувачі, де зберігається вся інформація про тих, хто користується платформою. Друга — проєкти, де містяться відомості про створені користувачами веб-додатки. Третя — шаблони, які представляють собою готові заздалегідь підготовлені проєкти для швидкого використання. Четверта категорія — коментарі, де зберігаються всі відгуки та повідомлення, які користувачі залишають під проєктами. П'ята — аналітика, що збирає статистику про перегляди і використання проєктів. І нарешті, шоста — розгортання, яка фіксує інформацію про публікацію і запуск проєктів у роботу.

Опис кожної з таблиць бази даних наведений у таблицях 2.8 – 2.13.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.8

Опис таблиці Users

Field	Type	Description
id	string / UUID	Унікальний ідентифікатор
name	string	Ім'я користувача
email	string	Електронна пошта
passwordHash	string	Хеш пароля
walletAddress	string	Адреса Web3 гаманця
userType	enum	Тип користувача (admin, developer, guest)

Таблиця 2.9

Опис таблиці Project

Field	Type	Description
id	string / UUID	Унікальний ідентифікатор
name	string	Назва проєкту
description	string	Опис проєкту
status	enum	Статус (draft, published)
code	text / JSON	Код або репозиторій
authorId	string / UUID	ID автора (User)
createdAt	datetime	Дата створення

Таблиця 2.10

Опис таблиці Template

Field	Type	Description
id	string / UUID	Унікальний ідентифікатор
name	string	Назва шаблону
description	string	Опис шаблону
code	text / JSON	Код шаблону
createdByAdmin	boolean	Створено адміністратором

Таблиця 2.11

Опис таблиці Comment

Field	Type	Description
id	string / UUID	Унікальний ідентифікатор
content	string	Текст коментаря
authorId	string / UUID	ID автора (User)
projectId	string / UUID	ID проєкту
createdAt	datetime	Дата створення

Таблиця 2.12

Опис таблиці Analytics

Field	Type	Description
id	string / UUID	Унікальний ідентифікатор
projectId	string / UUID	ID проєкту
views	integer	Кількість переглядів
publications	integer	Кількість публікацій
collectedAt	datetime	Дата збору

Таблиця 2.13

Опис таблиці Deployment

Field	Type	Description
id	string / UUID	Унікальний ідентифікатор
projectId	string / UUID	ID проєкту
url	string	URL розгортання
status	enum	Статус (success, error)
deployedAt	datetime	Дата розгортання

2.3 Проектування та реалізація алгоритмів роботи системи

Додаток є простим у використанні, має всі основні функції для створення,

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

публікації та аналітики проєктів, що дозволяє користувачам легко взаємодіяти з платформою. Алгоритм роботи з додатком зображено на рисунку 2.3, а також детально описано його основні пункти.

Коли в систему заходить адміністратор, він потрапляє в спеціальну адміністративну панель, де може бачити всі створені проєкти та шаблони. Адміністратор має повноваження підтверджувати чи редагувати проєкти перед їх публікацією, створювати нові шаблони, редагувати або видаляти існуючі, а також змінювати ролі інших користувачів, наприклад, призначати нових адміністраторів.

Якщо користувач увійшов як розробник, він отримує можливість створювати нові проєкти, редагувати свої існуючі роботи і подавати їх на публікацію. Розробник також може користуватися готовими шаблонами для швидшої розробки, а ще переглядати статистику щодо переглядів і публікацій власних проєктів.

Гість, який не авторизувався або зайшов без реєстрації, може переглядати список вже опублікованих проєктів, залишати коментарі до них і дивитися статистику публічних проєктів. Крім того, гість може взаємодіяти з проєктами через публічний інтерфейс без необхідності реєстрації.

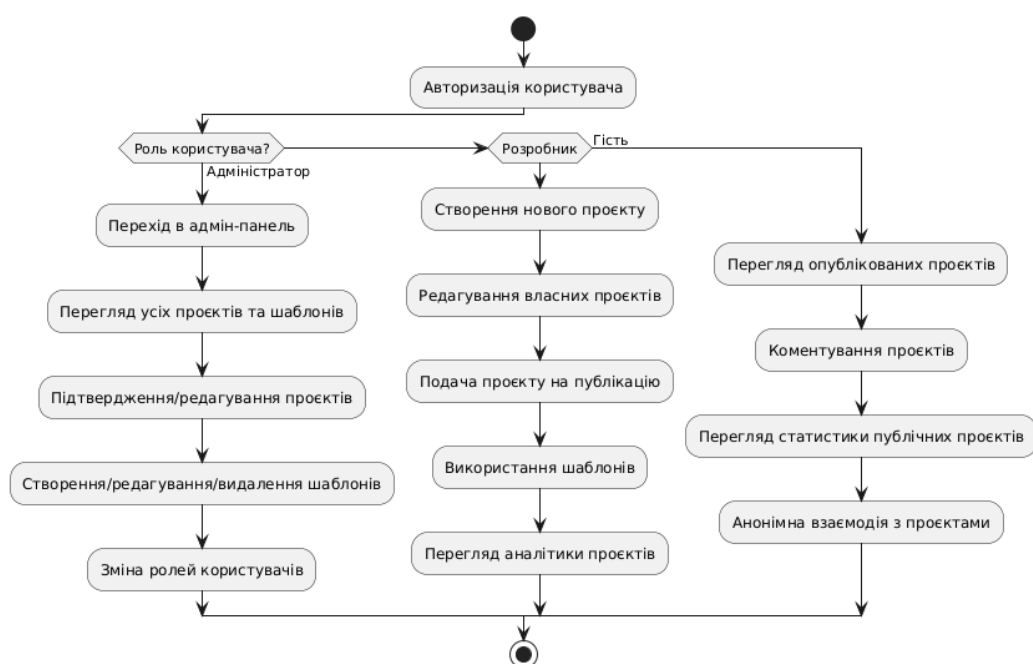


Рис. 2.3. Діаграма активності

2.4 Реалізація функціоналу платформи для розробки коду

У цьому розділі представлена реалізація основних методів системи, які забезпечують функціонування користувацького інтерфейсу, управління проєктами, шаблонами, коментарями, аналітикою та розгортанням застосунків.

Авторизація користувача: користувач вводить свою електронну пошту та пароль. Система перевіряє, чи такі дані існують і чи пароль співпадає. Якщо все вірно, система створює спеціальний токен, який підтверджує, що користувач увійшов, і дозволяє робити подальші дії в системі.

Редагування профілю користувача: користувач може оновити свої персональні дані, наприклад, змінити ім'я в профілі.

Перегляд проєктів: користувачі мають можливість переглядати всі проєкти, які вже опубліковані та доступні для загального огляду.

Створення нового проєкту: користувач може створити новий проєкт, який спочатку зберігається у статусі "чернетка", щоб потім до нього можна було повернутися і допрацювати.

Редагування проєкту: користувач може оновлювати інформацію про свій проєкт, змінювати назву, опис чи код.

Публікація проєкту: коли проєкт готовий, користувач може змінити його статус на "опублікований", щоб він став доступним для інших.

Створення шаблону: адміністратор має змогу створювати шаблони — готові заготовки для проєктів, які інші користувачі можуть використовувати для спрощення роботи.

Редагування шаблону: адміністратор може змінювати інформацію або вміст вже створених шаблонів.

Використання шаблону у проєкті: користувач може застосувати вибраний шаблон під час створення або редагування свого проєкту, щоб швидше наповнити його кодом.

Додавання коментаря: користувач може залишити коментар під будь-яким опублікованим проєктом.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

Видалення власного коментаря: користувач може видалити лише ті коментарі, які він сам додав.

Збір даних аналітики: система автоматично збирає статистику про проєкт, наприклад, скільки разів його переглянули або скільки разів він був опублікований.

Формування звітів: на основі зібраної статистики створюються звіти, які показують активність проєкту, як-от загальну кількість переглядів та публікацій.

Розгортання проєкту: проєкт можна розгорнути на віддаленому сервері, щоб він став доступним онлайн. Система зберігає інформацію про розгортання, зокрема адресу, де проєкт розміщений.

Оновлення додатку: якщо потрібно, розгорнутий додаток можна оновити, змінивши адресу або оновивши його версію, і система це зафіксує. Запропонований функціонал дозволяє реалізувати повний життєвий цикл роботи з проєктами: від їхнього створення до публікації, взаємодії з іншими користувачами через коментарі та аналізу ефективності через аналітику та звіти. Інтеграція шаблонів, а також можливість розгортання і оновлення проєктів робить систему зручною для постійного розвитку та масштабування.

Висновки до другого розділу

В процесі проектування платформи для розробки коду було проаналізовано всі можливі сценарії використання програмного продукту відповідно до вимог до програмного забезпечення. На основі отриманих результатів було спроектовано загальну архітектуру платформи. Проектування включало моделювання структури проєкту та визначення взаємодії між його основними складовими частинами.

Також була спроектована база даних для зберігання інформації про користувачів, проєкти, шаблони, коментарі, а також дані аналітики, що збираються у процесі використання додатку.

Було детально описано алгоритм роботи користувачів із системою, а також

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

сформульовані послідовні етапи їх взаємодії з функціоналом платформи. Крім того, було наведено опис основних функцій додатку — створення нового проєкту користувачем, його редагування, публікація, використання шаблонів для прискорення розробки, взаємодія через коментарі, а також збір аналітики і формування звітів. Було розглянуто можливості розгортання проєктів на сервері та їх подальшого оновлення.

		Іглінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ПОРЯДОК РОБОТИ ТА ІНТЕРФЕЙС ДОДАТКУ ДЛЯ РОЗРОБКИ КОДУ

3.1. Порядок встановлення та налаштування параметрів платформи

Встановлення та налаштування платформи вимагає розгортання всіх її компонентів та налаштування як кожного елементу окремо, так і загальної системи в цілому.

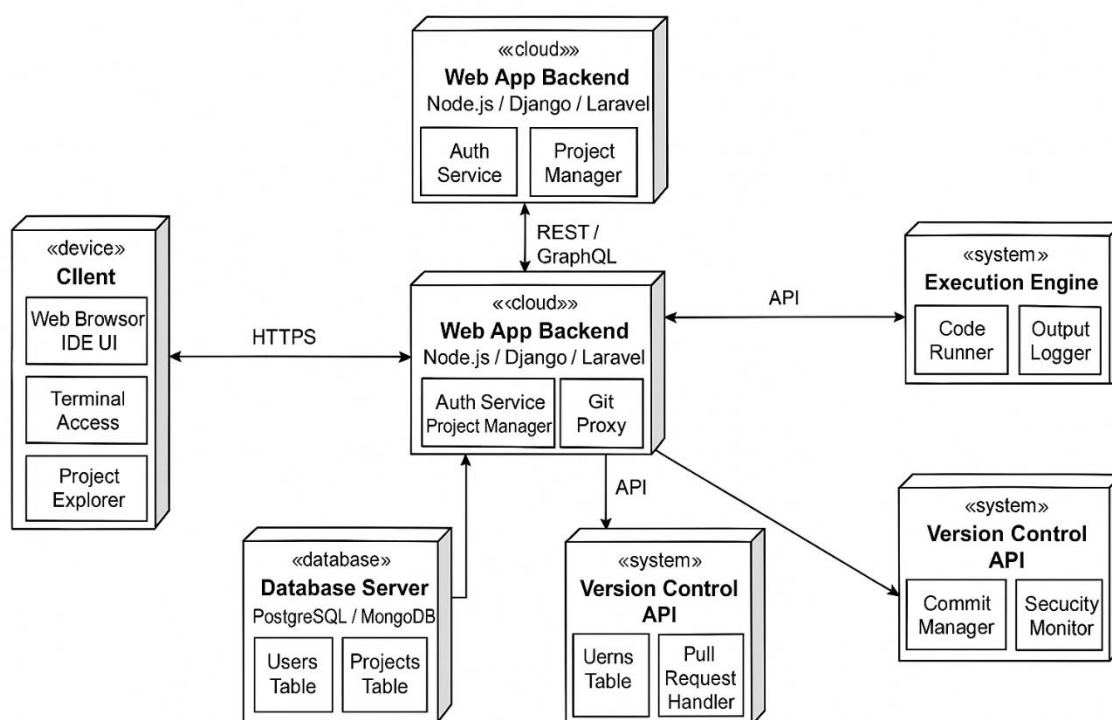


Рис.3.1- архітектура розгортання

Компонент під назвою Code Execution Engine відповідає за запуск і компіляцію коду користувача в безпечному середовищі — це може бути ізольований контейнер або віртуальна машина. Він підтримує різні мови програмування і слідкує, щоб код працював безпечно, не порушуючи систему.

Version Control API — це інструмент, який взаємодіє з системами контролю версій, наприклад, Git. Він допомагає створювати репозиторії, зберігати зміни, працювати з гілками і відстежувати історію змін у проєкті.

Метод Get project files дає можливість отримати файли проєкту

користувача, щоб їх можна було переглядати або редагувати через інтерфейс платформи. Метод Save changes дозволяє зберігати внесені користувачем зміни у файлах або налаштуваннях — ці зміни можуть зберігатися або у системі керування версіями, або у тимчасовому сховищі.

Метод Run tests запускає автоматичні тести, щоб перевірити, чи працює код правильно і якісно. Метод Collect statistics збирає різноманітну інформацію про проєкт — як часто роблять зміни, результати тестів, використання ресурсів і інші важливі показники.

WEB APP BACKEND — це серверна частина платформи, яка приймає запити від користувача, працює з базою даних, керує сесіями, дозволами та виконує операції створення, читання, оновлення і видалення даних, які не зберігаються у системі контролю версій.

WEB APP FRONTEND — це те, що бачить і з чим взаємодіє користувач у браузері чи додатку. Він поєднує дані з серверної частини і систем контролю версій, надаючи зручний інтерфейс для редагування коду, запуску проєктів, перегляду історії змін і статистики.

3.2. Структура інтерфейсу платформи для розробки коду

У додатку реалізована підтримка двох мов — української та англійської. Користувач може обрати потрібну мову у налаштуваннях, і інтерфейс автоматично адаптується, відображаючи всі підписи, кнопки, повідомлення та інші елементи на вибраній мові. Це робить роботу з додатком зручнішою для користувачів з різних мовних середовищ.

Після того, як користувач успішно авторизується або проходить реєстрацію, його автоматично перенаправляє на сторінку з особистими проєктами. Тут відображається список усіх проєктів, які належать цьому користувачу. Відсутність зайвих дій після входу дозволяє одразу приступити до роботи над власними завданнями, переглянути статус проєктів, редагувати їх або створювати нові. Це забезпечує швидкий і комфортний доступ до основного функціоналу платформи, підвищуючи продуктивність і зручність користування.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

Загалом, така організація інтерфейсу створює дружній та інтуїтивно зрозумілий досвід для користувачів, де все важливе — мова інтерфейсу та власні проєкти — завжди під рукою.

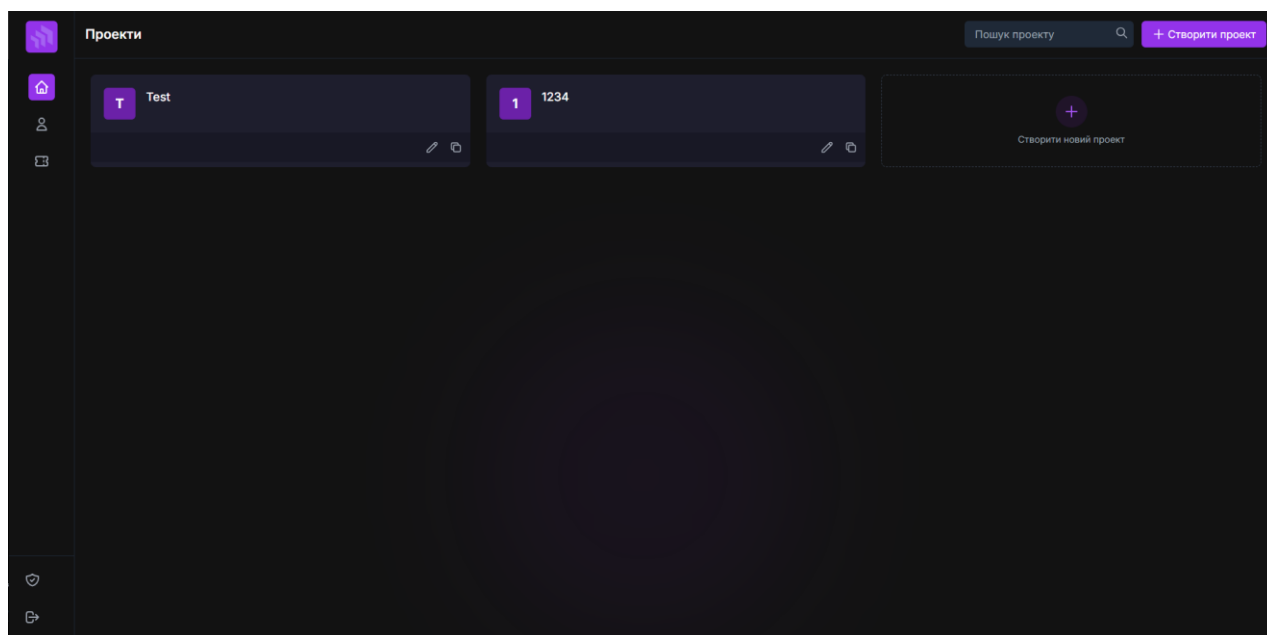


Рис.3.3- домашня сторінка

Оскільки ми вже увійшли в систему, у нас вже є створені проєкти, які зберігаються у моєму особистому кабінеті. Вони доступні для перегляду, редагування або публікації в будь-який момент. Щоб продемонструвати функціональність платформи, створимо новий проєкт.

Для цього потрібно перейти на відповідну сторінку або натиснути кнопку створення нового проєкту. Далі відкриється форма, де можна вказати основні деталі — назву проєкту, його опис та безпосередньо код або інші необхідні матеріали. Новий проєкт за замовчуванням створюється у статусі "чернетка", що дає можливість продовжити роботу над ним, вносити зміни і тестувати перед публікацією.

Такий підхід дозволяє гнучко керувати процесом розробки: ви можете створювати скільки завгодно проєктів, поступово їх удосконалювати, не турбуючись про негайну публікацію. Це особливо зручно для розробників, які хочуть працювати над кількома ідеями одночасно, планувати релізи та підтримувати якість продукту на високому рівні.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

Створення нового проекту — це перший крок у реалізації вашої ідеї на платформі, який відкриває доступ до всіх подальших можливостей: використання шаблонів, додавання коментарів, перегляду аналітики та розгортання проекту в робочому середовищі. Таким чином, процес розробки стає максимально простим, зручним і прозорим.

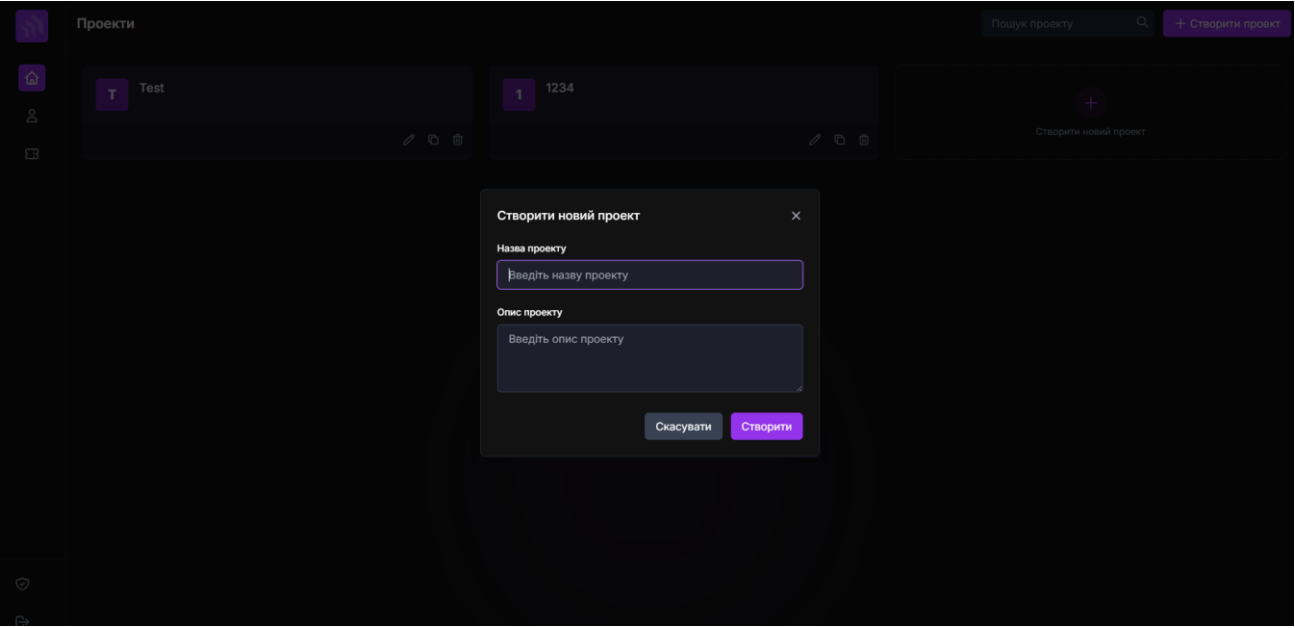


Рис.3.4 - форма для створення проекту

Відкривши наш новий проект, ми потрапляємо на сторінку, яка відображає всі ключові деталі та інструменти для роботи з ним. Тут можна побачити назву проекту, його опис, а також основний код або інші матеріали, що стосуються розробки. Цей інтерфейс створений так, щоб максимально спростити процес редагування і дозволити швидко вносити зміни.

У верхній частині сторінки зазвичай розміщені кнопки для збереження змін, публікації проекту або повернення до списку усіх проектів. Поруч можна знайти інформацію про статус проекту — чи він поки що чернетка, чи вже опублікований, а також дату останнього редагування.

Для більшої зручності користувача передбачена можливість використання шаблонів, які допомагають швидко заповнити структуру проекту або додати

стандартні елементи. Також доступна секція коментарів, де можна залишати власні нотатки або отримувати відгуки від інших користувачів.

Окрім цього, на сторінці проекту представлена аналітика — інформація про кількість переглядів, активність користувачів і статистика публікацій. Це дозволяє стежити за популярністю та ефективністю роботи над проектом.

Таким чином, відкриття проекту дає змогу не лише контролювати всі аспекти його створення та розвитку, але й зручно управляти робочим процесом, отримувати зворотний зв'язок і готувати проект до публікації чи розгортання. Це місце, де ідеї перетворюються на реальні результати.

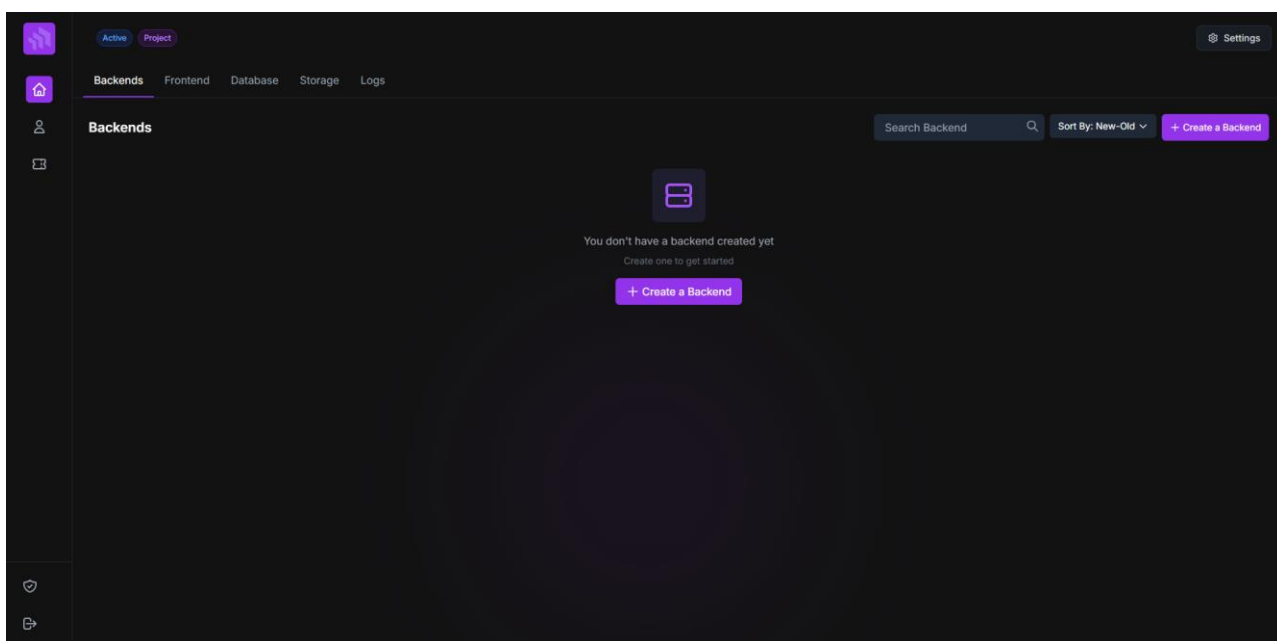


Рис.3.5 – вкладка для управління проектом

У нашому проекті передбачено п'ять основних вкладок, які допомагають повністю контролювати всі аспекти розробки та управління проектом. Кожна з них відповідає за окремий блок функціональності, що робить роботу більш структурованою та зручною.

Почнемо з першої вкладки — створення backend коду. Саме тут розробник може писати логіку, яка працює на серверній стороні. Наприклад, створення авторизації — це ключовий етап, який дозволяє користувачам безпечно входити

		Іглінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

в систему, підтверджувати свою особу за допомогою електронної пошти та пароля.

У цій вкладці ми можемо прописати весь необхідний код для обробки запитів на логін, реєстрацію, перевірку прав доступу та керування сесіями. Це дозволяє надійно захистити інформацію користувачів і контролювати, хто саме може виконувати певні дії в додатку.

Інтерфейс у цій вкладці інтуїтивний: можна легко додавати нові файли, редагувати існуючі, а також тестувати написаний код безпосередньо у платформі. Це значно пришвидшує розробку і дозволяє не виходити за межі проекту для перевірки функціональності.

Таким чином, перша вкладка — це основа, на якій будується вся серверна логіка, і від її якості залежить безпека, стабільність та зручність використання нашого проекту. Переходимо до створення необхідного backend функціоналу крок за кроком.

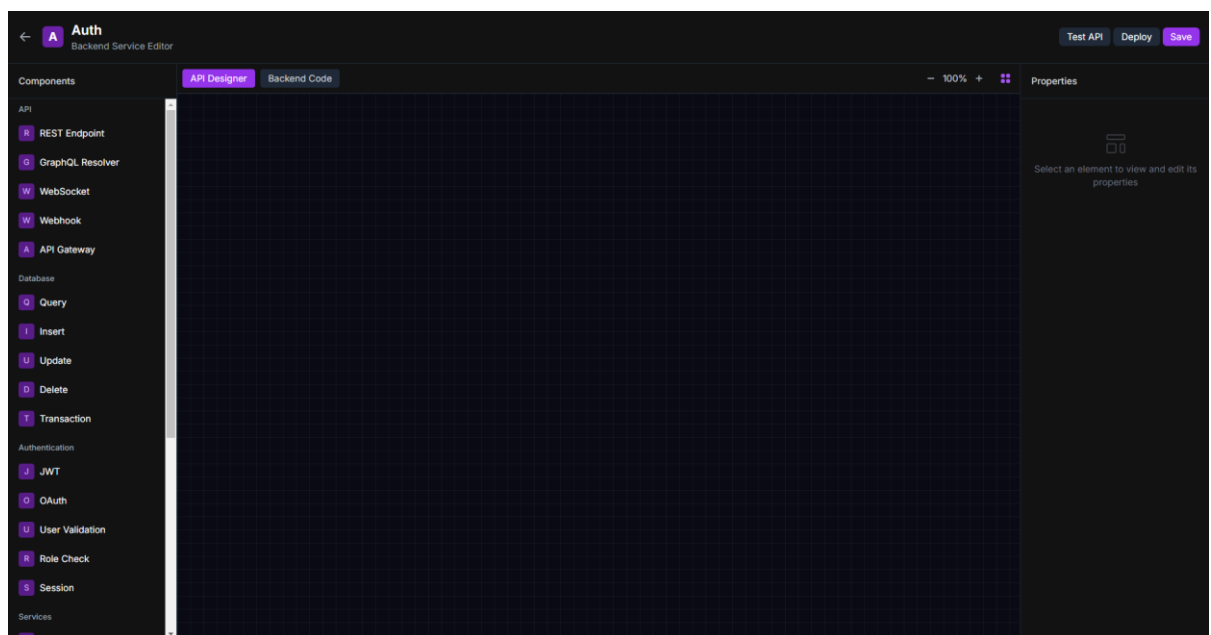


Рис.3.6 – вікно для створення функції авторизації

Зліва на екрані розташований набір готових компонентів, які можна легко використовувати і налаштовувати під свої потреби. Це дуже зручно, адже не потрібно писати все з нуля — можна швидко вибрати потрібний елемент і

		Іглінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

відредагувати його відповідно до вимог проекту.

Давайте створимо базову структуру для авторизації. Спочатку обираємо компоненти, які потрібні для форми входу: поля для введення електронної пошти та пароля, кнопку для підтвердження, а також, за бажанням, чекбокс “запам’ятати мене” чи посилання на відновлення пароля.

Після того, як ми розставимо всі елементи на сторінці, можна буде налаштувати їх зовнішній вигляд і поведінку — змінити назви, додати валідацію, щоб користувачі не могли відправити форму з порожніми або некоректними даними.

Цей підхід дозволяє швидко створювати потрібний функціонал і водночас підтримувати стиль та дизайн додатку в єдиному вигляді. Ми створюємо не просто форму, а зручний та інтуїтивно зрозумілий інтерфейс для користувачів, який допоможе їм легко увійти в систему.

Таким чином, використання готових компонентів і налаштування їх під свої потреби значно прискорює процес розробки і робить роботу комфортнішою. Тож переходимо до створення цієї базової структури для авторизації крок за кроком.

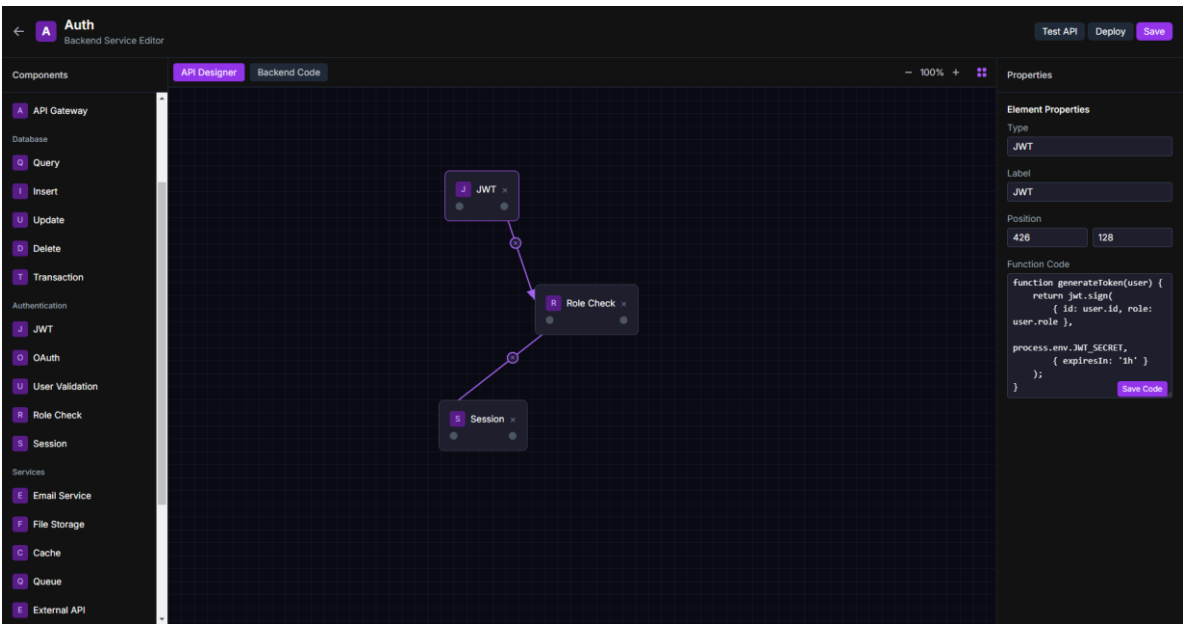


Рис.3.7- функція для авторизації

У правій частині екрану відкривається панель, де можна легко змінювати назву обраного компонента — це допомагає краще орієнтуватися у структурі проекту. Тут же можна відредагувати код функції, яка відповідає за роботу цього компонента. Наприклад, додати логіку авторизації, обробку помилок або інші потрібні дії.

Після внесення змін не забудьте натиснути кнопку Save, щоб зберегти оновлення. Це гарантує, що всі правки будуть застосовані і збережені у проекті.

Далі переходимо у вкладку Frontend — тут можна працювати з інтерфейсом користувача, налаштовувати вигляд та взаємодію елементів. Це місце, де ми створюємо те, що бачить і з чим взаємодіє користувач: кнопки, поля введення, списки, візуальні ефекти.

Тут можна швидко перевірити, як виглядає та працює наша форма авторизації, протестувати всі її елементи і, за потреби, внести корективи. Таке розділення між логікою і візуальною частиною робить розробку більш зрозумілою і зручною.

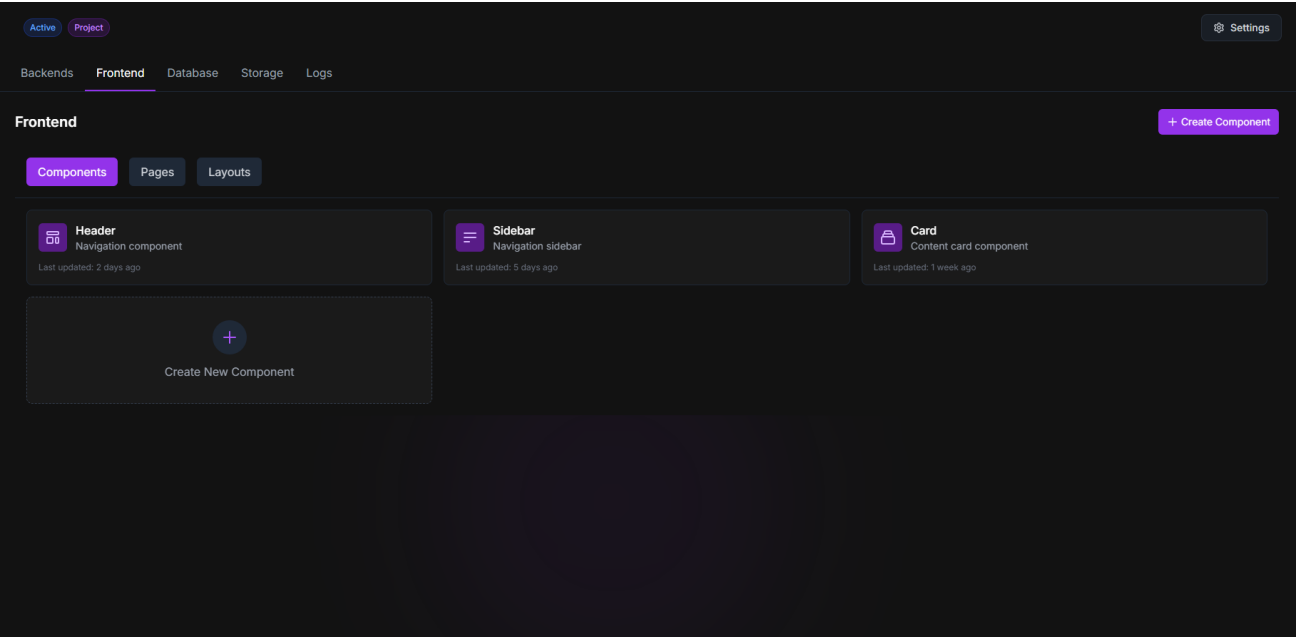


Рис.3.8 – вкладка frontend

Коли ми створюємо новий проект, система автоматично генерує набір готових компонентів, сторінок і макетів. Це допомагає швидко розпочати

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

розробку без необхідності створювати все з нуля. Завдяки цьому можна зосередитись на унікальній функціональності, не витрачаючи час на базові елементи.

Для прикладу відкриємо компонент Header — це верхня частина нашого інтерфейсу, яка зазвичай містить логотип, навігаційне меню, кнопки для входу чи налаштувань. Цей компонент відповідає за те, щоб користувачі могли легко орієнтуватися на сайті або в додатку.

Відкривши Header, ми можемо побачити його структуру, як він побудований, і при необхідності швидко відредагувати дизайн чи функціональність. Наприклад, можна змінити тексти кнопок, додати нові елементи меню або налаштувати адаптивність для різних пристроїв.

Такі готові компоненти значно прискорюють процес розробки, адже вони вже містять базову логіку і стиль, залишаючи простір для кастомізації під конкретні потреби проекту.

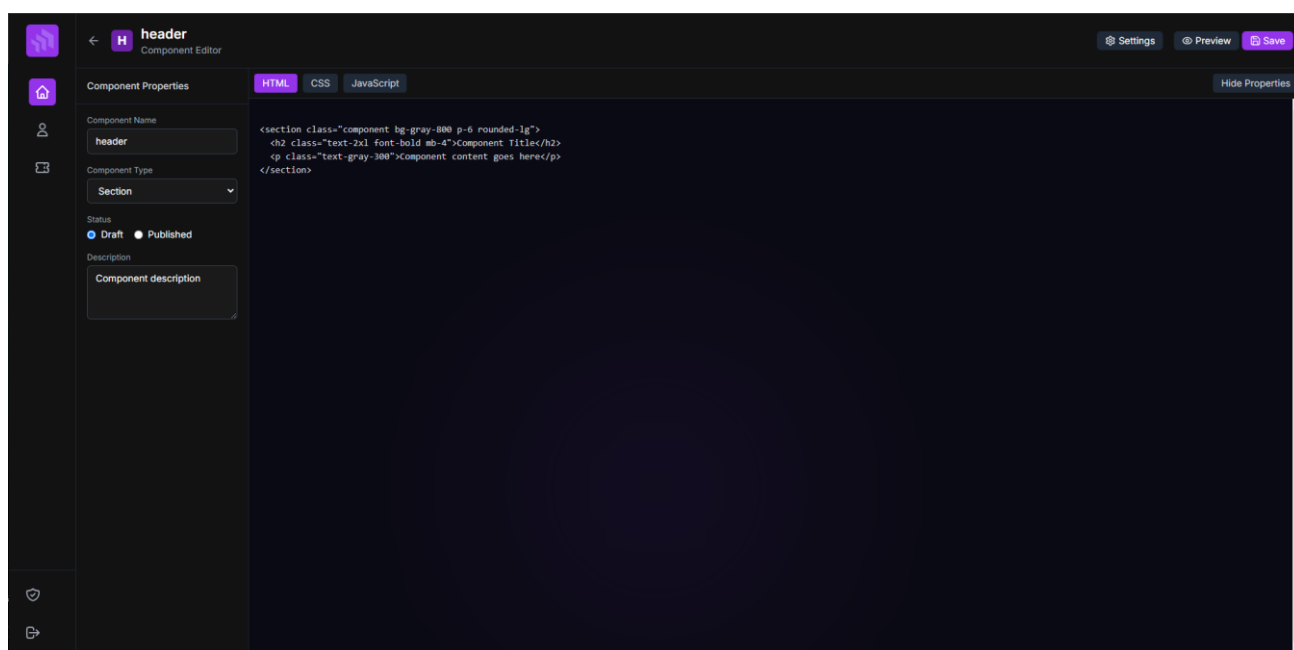


Рис.3.9- компонент Header

Для зручності розробки коду кожен компонент має кілька окремих вкладок: HTML, CSS і JavaScript. Це дозволяє працювати з кожним типом коду окремо, не змішуючи розмітку, стилі та логіку. [14, 15, 16]

		Гелінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

Такий підхід робить розробку більш зрозумілою і структурованою, адже кожна частина відповідає за свою задачу.

У вкладці HTML ми створюємо структуру компонента — його елементи, теги, розташування блоків. У CSS ми задаємо зовнішній вигляд — кольори, шрифти, відступи, анімації та інші стилі, що визначають, як саме буде виглядати компонент. В JavaScript додаємо інтерактивність — обробку подій, зміну стану, логіку роботи.

Під час розробки у нас завжди є можливість швидко подивитися, як виглядає і працює створений компонент. Це дає змогу одразу ж перевірити, чи все працює так, як задумано, і оперативно внести необхідні правки. Попередній перегляд оновлюється в реальному часі, тому ми бачимо всі зміни відразу після їх внесення.

Такий інтерактивний підхід до розробки значно прискорює роботу і допомагає уникнути помилок, бо замість довгого циклу «написав — перевірів — виправив» ми одразу бачимо результат і можемо швидко коригувати код. Це особливо важливо при створенні складних або динамічних інтерфейсів, де важлива точність і гнучкість.

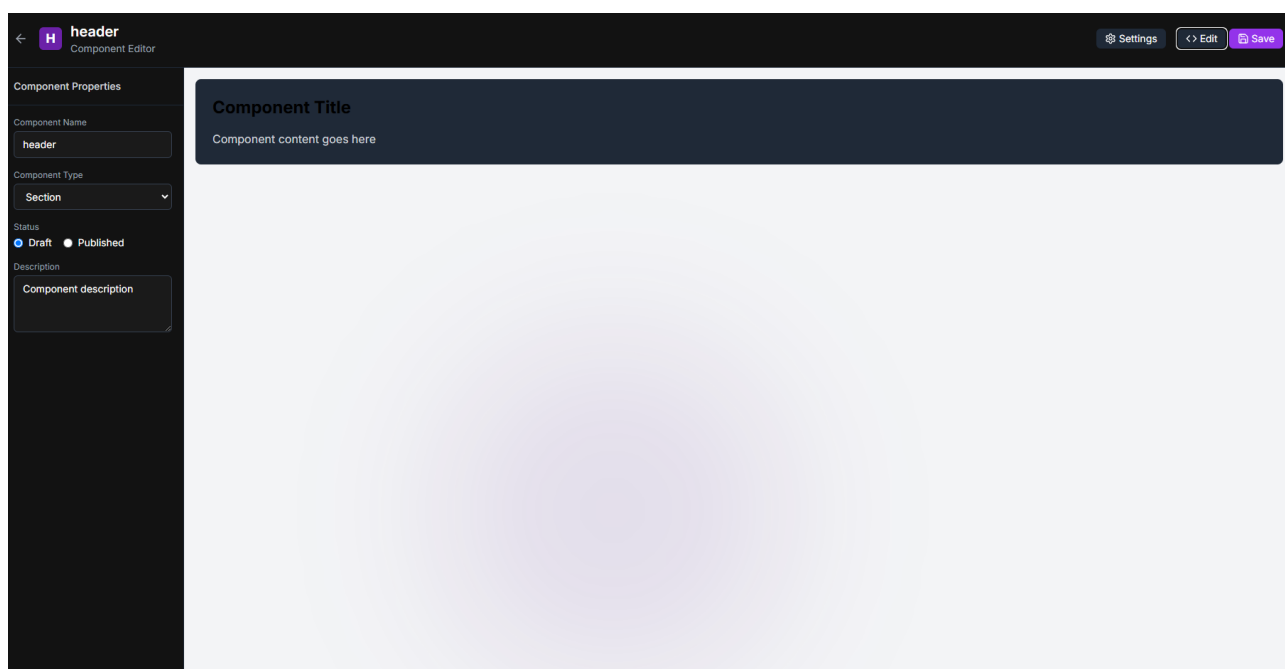


Рис.3.10- демонстрація компоненту

		Іглінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдяки функції live preview ми можемо миттєво бачити, як зміни у коді впливають на вигляд і роботу компонента. Це дає змогу швидко переключатися між вкладкою для редагування і вкладкою для перегляду результату, не втрачаючи темпу роботи та контролю над процесом. Такий підхід допомагає одразу помічати помилки або недоліки і виправляти їх на місці, що значно спрощує розробку.

Тепер перейдемо до вкладки Database. Тут ми маємо можливість керувати всіма даними, пов'язаними з нашим проектом. У цій вкладці можна створювати та редагувати таблиці, додавати записи, налаштовувати зв'язки між різними об'єктами. Це дуже зручно, адже вся інформація зберігається централізовано і доступна для використання у проекті.

Вкладка Database допомагає налаштувати структуру даних без необхідності писати складні SQL-запити вручну. Можна просто вказати поля, типи даних, обмеження, а також переглядати та оновлювати інформацію у зручному інтерфейсі. Це особливо корисно, коли потрібно швидко додати нові властивості або змінити вже існуючі.

Особливо цінною є можливість створювати зв'язки між таблицями у візуальному форматі. Це дозволяє не лише уникнути помилок у логіці зв'язків, але й краще розуміти загальну архітектуру даних. Наприклад, можна легко налаштувати зв'язок між користувачем і його публікаціями, або між товарами і категоріями, без необхідності писати складні SQL-запити або опрацьовувати зовнішні ключі вручну.

Для більш досвідчених користувачів інтерфейс дозволяє переглядати структуру запитів, які відправляються до бази даних, та (у деяких випадках) оптимізувати їх вручну. Це корисно у складних проектах, де необхідно контролювати продуктивність запитів або налаштовувати індекси. Також у цій вкладці можна слідкувати за зв'язками між таблицями, що дає змогу будувати складніші й більш логічні структури даних, які легко інтегруються у фронтенд і бекенд частини проекту.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

Це спрощує роботу розробника і робить проект більш гнучким і масштабованим.

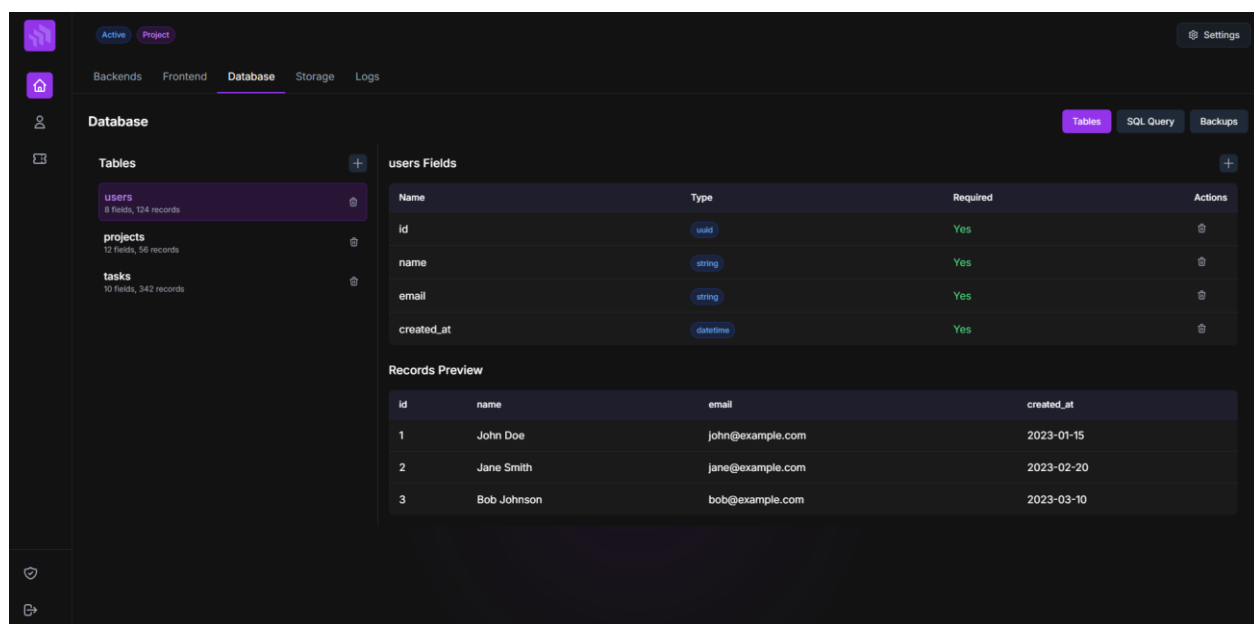


Рис.3.11 – вкладка для управління базою даних

У цій частині платформи ми можемо не лише створювати нові таблиці і налаштовувати їх структуру, а й безпосередньо писати SQL-запити для вибірки, оновлення або видалення даних. Це дає більше контролю і гнучкості тим, хто добре знається на базах даних і хоче робити складні операції вручну. Крім того, тут можна створювати резервні копії бази даних — це дуже важливо для збереження інформації і безпеки проекту, адже в разі проблем чи помилок можна швидко відновити дані.

Тепер перейдемо до сторінки управління власним акаунтом. Тут користувач може налаштовувати свої персональні дані, змінювати пароль, оновлювати контактну інформацію та керувати налаштуваннями безпеки. Ця сторінка створена для того, щоб кожен міг контролювати свої дані та персоналізувати свій досвід роботи з платформою.

Окрім основної інформації, на цій сторінці також можна вибрати мову інтерфейсу, налаштувати сповіщення і підключити додаткові сервіси, якщо це передбачено. Все зроблено максимально просто і зрозуміло, щоб навіть

користувач без технічних знань міг легко керувати своїм акаунтом та почуватися впевнено під час роботи.

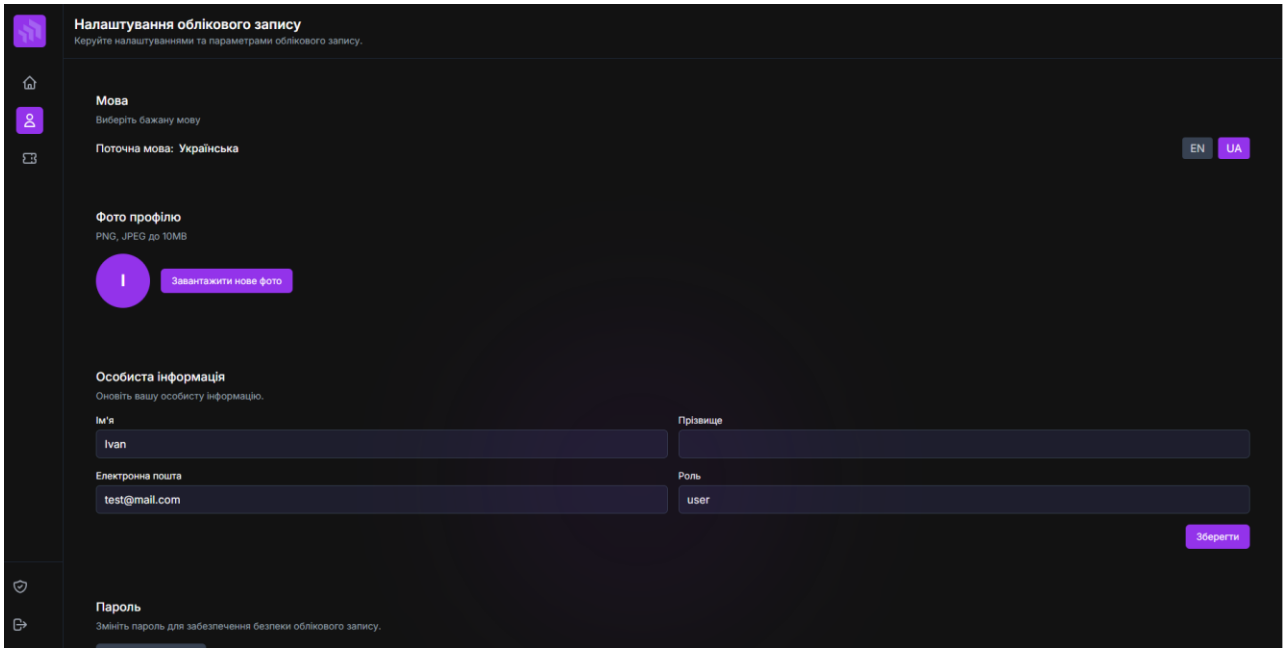


Рис.3.12 – сторінка управління акаунтом(ч.1)

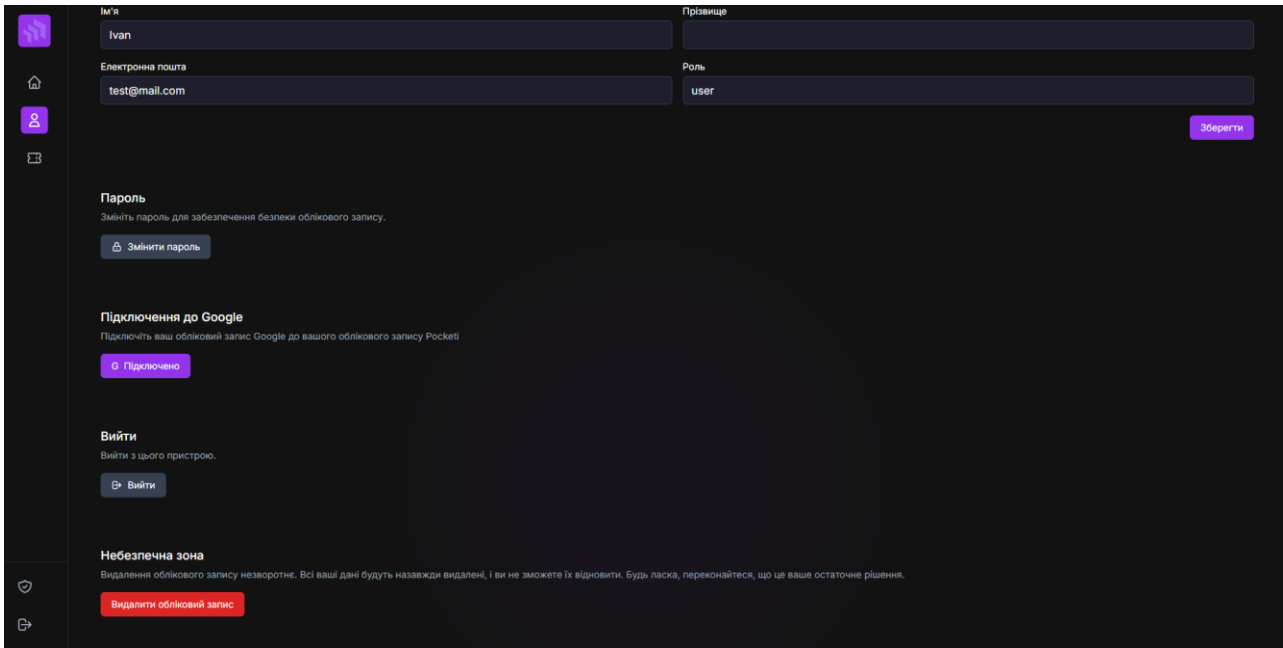


Рис.3.13 – сторінка управління акаунтом(ч.2)

На цій сторінці можна легко оновити свої особисті дані — наприклад, ім'я, прізвище, електронну пошту та іншу контактну інформацію. Якщо потрібно,

можна змінити мову інтерфейсу — обрати між українською та англійською, щоб робота в додатку була ще зручнішою.

Також тут можна керувати налаштуваннями свого акаунту: поміняти пароль для безпеки, вийти з системи або навіть повністю видалити свій профіль, якщо більше не плануєте користуватися платформою. Щоб підтримувати розвиток і забезпечувати якісний сервіс, у додатку реалізована система підписок. Вона дає змогу отримувати додаткові можливості або розширений функціонал, а також допомагає платформі приносити прибуток для подальшого розвитку та вдосконалення сервісів.

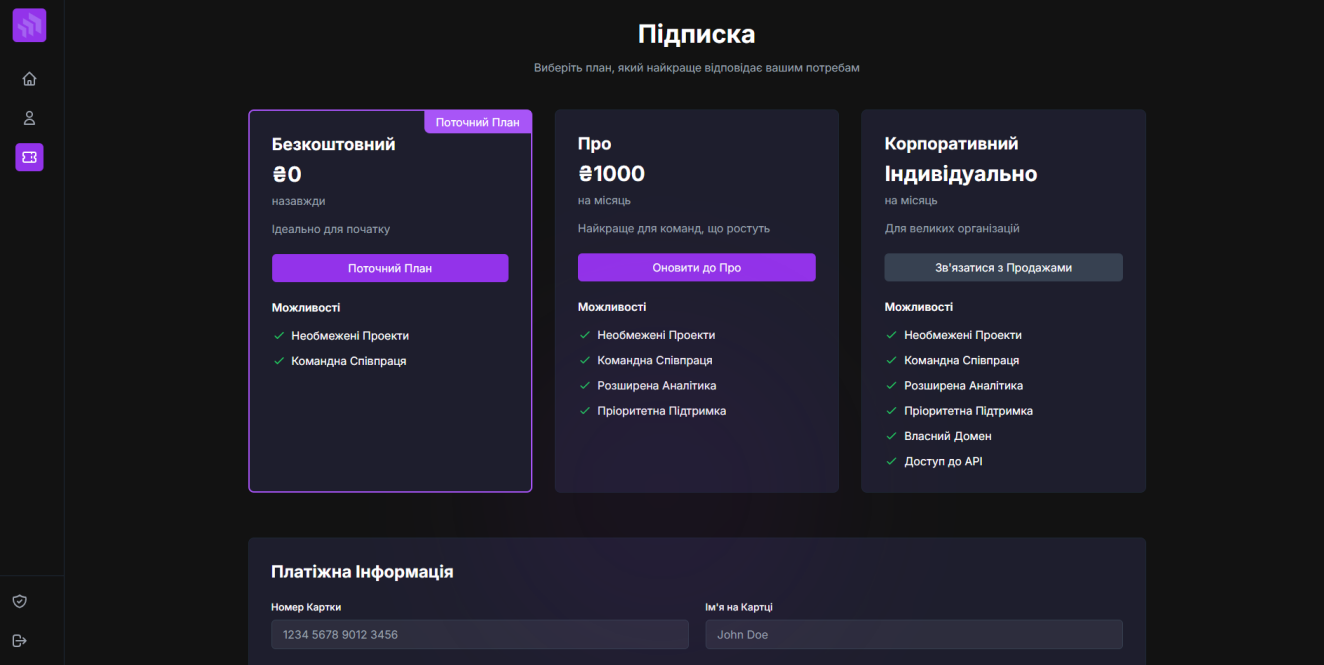


Рис.3.14 – сторінка з підписками

У додатку передбачено три основні варіанти підписки, щоб кожен міг обрати той, який йому найбільше підходить. Безкоштовний план дозволяє користуватися базовим набором функцій — цього достатньо для ознайомлення з платформою та виконання простих завдань, але деякі просунуті можливості будуть недоступні. Якщо ж ви хочете отримати повний доступ до майже всіх інструментів, а також користуватися підтримкою від адміністраторів, то план

Про стане ідеальним вибором. Для компаній і організацій, які потребують глибшого контролю, більш тонких налаштувань та корпоративної підтримки, існує спеціальний Корпоративний план. Він розрахований на бізнес-користувачів і забезпечує максимальні можливості адаптації платформи під їхні потреби. Внизу сторінки розташована форма для введення платіжної інформації, що дозволяє зручно оформити підписку безпосередньо в додатку. Це спрощує процес оплати і дозволяє швидко перейти до користування розширеними функціями без зайвих кроків.

3.3. Статистика та адміністрування платформи

В платформі для розробки коду rocketi, розроблена адмін панель в якій можна управляти всією платформою:

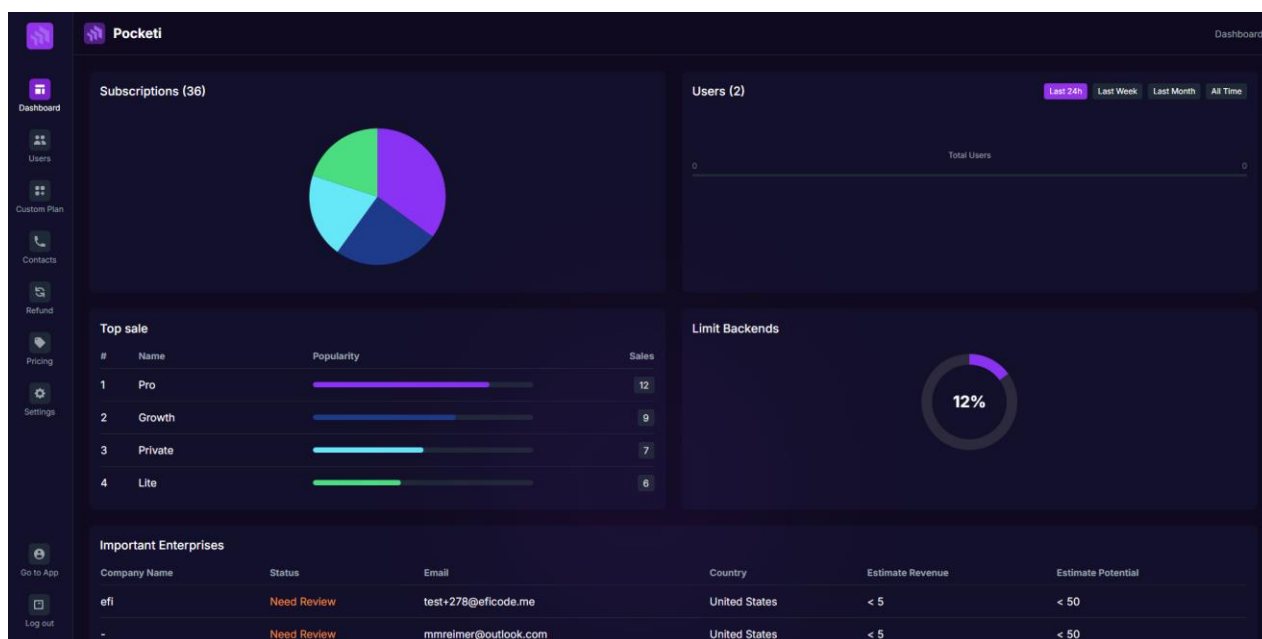


Рис.3.15- головна сторінка з адмін панелі

Адмін панель має свій унікальний, відмінний від основного додатку дизайн, який орієнтований на зручність управління та швидкий доступ до важливої інформації. Головна сторінка адмінки (рис. 3.17) містить різноманітні показники, діаграми та графіки, що допомагають відслідковувати стан платформи в режимі реального часу. Тут відображається статистика активності

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

користувачів, стан проектів, результати аналітики та інші ключові метрики, які необхідні для прийняття оперативних рішень.

Далі ми переходимо до сторінки з управління користувачами. На цій сторінці адміністратори можуть переглядати список всіх зареєстрованих користувачів, їхні основні дані та статуси. Також тут доступні функції редагування інформації, зміни ролей, блокування або видалення акаунтів у разі потреби. Все це дозволяє ефективно контролювати користувацьку базу та підтримувати порядок на платформі, забезпечуючи безпеку і комфорт для всіх учасників.

EMAIL	FIRST NAME	LAST NAME	PLAN	SUBSCRIPTION START DATE	SUBSCRIPTION END DATE	STATUS	ACTIONS
bryan@printdeck.io	-	-	Pro	Apr 28, 2023 11:05 pm	May 28, 2023 11:05 pm	Active	Edit Delete
alexis1@alexis.app	alexis1	-	Pro	Apr 28, 2023 11:43 pm	May 28, 2023 11:43 pm	Active	Edit Delete
test+24@alexxz.me	test+24	-	Pro	Apr 25, 2023 6:14 pm	May 25, 2023 6:14 pm	Active	Edit Delete
test+23@alexxz.me	test+23	-	Pro	Apr 19, 2023 5:48 pm	May 19, 2023 5:48 pm	Active	Edit Delete
test+22@alexxz.me	test+22	-	Pro	Apr 19, 2023 4:57 pm	May 19, 2023 4:57 pm	Active	Edit Delete
test+21@alexxz.me	test+21	-	Pro	Apr 18, 2023 4:41 pm	May 18, 2023 4:41 pm	Active	Edit Delete
test+20@alexxz.me	test+20	-	Pro	Apr 18, 2023 4:37 pm	May 18, 2023 4:37 pm	Active	Edit Delete
test+19@alexxz.me	test+19	-	Pro	Apr 18, 2023 2:00 pm	May 18, 2023 2:00 pm	Active	Edit Delete
test+17@alexxz.me	test+17	-	Pro	Apr 17, 2023 5:48 pm	May 17, 2023 5:48 pm	Active	Edit Delete
test+16@alexxz.me	test+16	-	Pro	Apr 10, 2023 3:48 pm	May 10, 2023 3:48 pm	Inactive	Edit Delete
test+15@alexxz.me	test+15	-	Pro	Apr 5, 2023 2:48 pm	May 5, 2023 2:48 pm	Inactive	Edit Delete

Рис.3.16 – вкладка для управління користувачами всієї платформи

На цій сторінці ми маємо змогу переглядати детальну інформацію про кожного користувача додатку — їхні особисті дані, активність, підписки та інші важливі параметри. Окрім перегляду, тут доступні функції редагування профілів, змін ролей або прав доступу, а також можливість видаляти користувачів у разі потреби. Це допомагає підтримувати порядок у системі, оперативно реагувати на проблеми та забезпечувати безпеку платформи.

Тепер перейдемо до сторінки, присвяченої індивідуальним планам. Тут адміністратори можуть створювати та налаштовувати спеціальні тарифні пакети

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

для конкретних клієнтів або компаній. Можна визначати набір функцій, обмеження, цінову

політику та терміни дії таких планів. Це дає змогу гнучко підходити до потреб різних користувачів і пропонувати їм оптимальні умови співпраці, які відповідають їхнім вимогам та бюджету.

FIRST NAME	LAST NAME	STATUS	EMAIL	PHONE	COUNTRY	COMPANY NAME	ESTIMATE BACKENDS	ESTIMATE STORAGE	ACTIONS
test+23	fgh	Need Review	test+23@alexz.me	-	United States	ifk	< 5	< 50	View
-	-	Need Review	roma+user1@alexz.app	-	United States	-	< 5	< 50	View
Roman	USDEK	Need Review	roma@alexz.app	5	United States	-	< 5	< 50	View
Ivan	Uslex	Need Review	ivan@alexz.app	55555	United States	-	< 5	< 50	View
Ivan	Uslex	Need Review	ivan@alexz.app	55555	United States	-	< 5	< 50	View
Ivan	Uslex	Need Review	ivan@alexz.app	55555	United States	-	< 5	< 50	View
Ivan	Uslex	Need Review	ivan@alexz.app	5555555	United States	-	< 5	< 50	View
Roman	USDEK	Need Review	roma@alexz.app	5	United States	-	< 5	< 50	View
Roman	USDEK	Need Review	roma@alexz.app	5	United States	-	< 5	> 2,000	View
Ivan	Uslex	Need Review	ivan@alexz.app	55555555	United States	-	< 5	< 50	View

Рис.3.17 – сторінка з управлінням індивідуальними планами

На цій сторінці ми можемо переглядати інформацію про запити на створення індивідуальних планів. В залежності від потреб адмін створює індивідуальну підписку з індивідуальним прайсом.

В залежності від отриманого запиту, адміністратор може створити індивідуальну підписку, в якій враховуються побажання користувача, включаючи обсяг доступних функцій, термін дії плану, рівень технічної підтримки тощо. Для кожного такого плану можна задати індивідуальний прайс, що дає змогу формувати гнучку цінову політику та адаптувати продукт під різні категорії користувачів — від стартапів до корпоративних клієнтів.

Окрема увага приділяється вкладці загальних налаштувань — це одна з ключових частин адміністративної панелі, яка дозволяє конфігурувати базову поведінку всієї платформи. У цій секції доступні наступні можливості:

		Гелінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

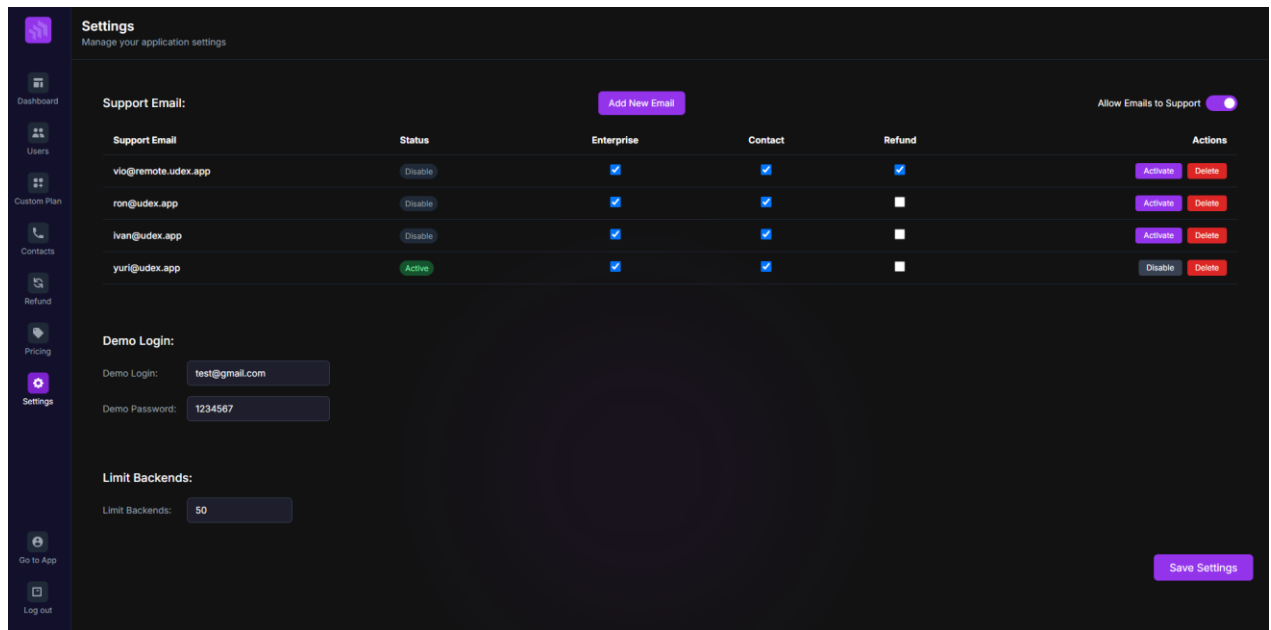


Рис.3.18 – загальні налаштування платформи

На цій сторінці адмін може, змінювати електроні пошти користувачів підтримки, має доступ до тестового проекту через демо логін, та встановлювати загальний ліміт створення бекенд запитів

Висновки до третього розділу

У цьому розділі ми детально описали, як працює і розгортається платформа Rocketi — система для онлайн-розробки коду. Кожна частина системи була пояснена, а також показано, як вони взаємодіють між собою. Завдяки цьому можна повністю зрозуміти, як побудоване програмне забезпечення і як його основні компоненти працюють разом.

Ми розглянули клієнтську частину (Web IDE), серверну логіку, механізми запуску коду, базу даних, де зберігаються проекти та історія змін. Спільна робота цих модулів забезпечує надійну, масштабовану і безпечну роботу платформи.

Також ми описали інтерфейс користувача та всі основні можливості додатку, залежно від ролі користувача. Звичайні користувачі, тобто розробники, мають доступ до редактора коду, терміналу, можуть запускати свій код, дивитися

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

результати та керувати проєктами. Адміністратори ж отримують додаткові інструменти для контролю системи: моніторинг ресурсів, перегляд журналів, управління користувачами та контролювання запуску контейнерів. Це дозволяє кожному користувачу легко орієнтуватися у тому, що він може робити, і використовувати платформу максимально ефективно відповідно до своїх завдань.

Окремо показано, як працює статистика в системі. Для розробників відображається інформація про кількість їхніх проєктів та їх стан. Адміністратори ж бачать розширену аналітику — скільки сесій зараз активні, як навантажуються сервери, стан контейнерів, історію помилок і як розподіляється робота між користувачами. Такий підхід робить Rocketi зручною платформою як для навчання, так і для професійної командної розробки.

		Іглінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було створено платформу Rosketi — сучасне онлайн-середовище для написання, тестування та збереження програмного коду. На початку було проведено аналіз існуючих рішень, таких як Flutterflow, Bubble та CodeSandbox, що допомогло визначити, які функції і можливості має мати наша система.

Головними завданнями платформи стали: підтримка роботи кількох користувачів одночасно, безпечне ізольоване виконання коду, збереження проєктів, робота з різними мовами програмування і інтеграція з системами контролю версій.

Було розроблено архітектуру системи, обрано потрібні технології і спроектовано основні частини вебдодатку. В процесі проектування врахували, як користувачі будуть взаємодіяти з клієнтською частиною, сервером і середовищем для запуску коду. Також створили структуру бази даних для зберігання інформації про користувачів, проєкти, сесії, історію запусків та активність розробників.

Описи роботи системи враховують різні ролі користувачів. Основний функціонал платформи включає створення проєктів, редагування коду в режимі реального часу, запуск і тестування програм, а також ведення історії змін.

У роботі показано, як розгортається платформа — це Web IDE, серверна частина, середовище запуску в контейнерах і взаємодія з базою даних. Наведено приклади двох типів користувачів — розробника і адміністратора — і які можливості їм доступні. Розробники бачать свою активність і статистику по проєктах, а адміністратори мають доступ до розширеної аналітики про навантаження системи, активні сесії та використання ресурсів.

Архітектура платформи створена з урахуванням масштабованості, безпеки і розподіленої роботи. Ізоляція середовища запуску коду гарантує стабільність і захист при виконанні будь-яких програм.

		Гелінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтерфейс платформи зручний і зрозумілий, що дозволяє легко створювати нові проєкти, запускати код і ділитися результатами. Rocketi підходить як для навчання, так і для професійної розробки завдяки потужному функціоналу та прозорому управлінню ресурсами.

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Next.js – The React Framework for the Web [Електронний ресурс] – Режим доступу: [<https://nextjs.org>]
2. Tailwind CSS – Rapidly build modern websites [Електронний ресурс] – Режим доступу: [<https://tailwindcss.com>]
3. React – A JavaScript library for building user interfaces [Електронний ресурс] – Режим доступу: [<https://react.dev>]
4. FlutterFlow – Build apps visually [Електронний ресурс] – Режим доступу: [<https://flutterflow.io>]
5. Webflow – Build better business websites, faster [Електронний ресурс] – Режим доступу: [<https://webflow.com>]
6. OutSystems – High-performance low-code platform [Електронний ресурс] – Режим доступу: [<https://www.outsystems.com>]
7. Bubble – Visual web development without code [Електронний ресурс] – Режим доступу: [<https://bubble.io>]
8. TypeScript – JavaScript with syntax for types [Електронний ресурс] – Режим доступу: [<https://www.typescriptlang.org>]
9. Flutter – Build apps for any screen [Електронний ресурс] – Режим доступу: [<https://flutter.dev>]
10. Oracle – Data management and cloud services [Електронний ресурс] – Режим доступу: [<https://www.oracle.com>]
11. Microsoft SQL Server – Data platform for modern applications [Електронний ресурс] – Режим доступу: [<https://www.microsoft.com/en-us/sql-server>]
12. SAP – Enterprise software to manage business operations [Електронний ресурс] – Режим доступу: [<https://www.sap.com>]
13. Node.js – JavaScript runtime built on Chrome's V8 engine [Електронний ресурс] – Режим доступу: [<https://nodejs.org>]
14. HTML – HyperText Markup Language [Електронний ресурс] – Режим доступу: [<https://developer.mozilla.org/en-US/docs/Web/HTML>]

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

15. CSS – Cascading Style Sheets [Електронний ресурс] – Режим доступу: [\[https://developer.mozilla.org/en-US/docs/Web/CSS\]](https://developer.mozilla.org/en-US/docs/Web/CSS)
16. JavaScript – Programming language of the Web [Електронний ресурс] – Режим доступу: [\[https://developer.mozilla.org/en-US/docs/Web/JavaScript\]](https://developer.mozilla.org/en-US/docs/Web/JavaScript)
17. Dart – A client-optimized language for fast apps on any platform [Електронний ресурс] – Режим доступу: [\[https://dart.dev\]](https://dart.dev)
18. PostgreSQL – The world’s most advanced open source database [Електронний ресурс] – Режим доступу: [\[https://www.postgresql.org\]](https://www.postgresql.org)
19. JWT (JSON Web Token) – Open standard for secure transmission of information [Електронний ресурс] – Режим доступу: [\[https://jwt.io\]](https://jwt.io)
20. CI/CD – Continuous Integration and Continuous Deployment [Електронний ресурс] – Режим доступу: [\[https://www.redhat.com/en/topics/devops/what-is-ci-cd\]](https://www.redhat.com/en/topics/devops/what-is-ci-cd)
21. Addison Wesley. Next.js Quick Start Guide: Server-side rendering done right. – Birmingham: Packt Publishing, 2018.
22. Cedar Greg. Fullstack React with TypeScript. – Independently published, 2021.
23. Фрімен Ерік, Робсон Елізабет. JavaScript. Детальне керівництво. – Київ: Наш Формат, 2022.
24. Дакетт Джон. HTML і CSS. Розробка та дизайн веб-сайтів. – Львів: Видавництво Старого Лева, 2020.
25. Дакетт Джон. JavaScript та jQuery. Інтерактивна веброботка. – Київ: Видавництво "Наш Формат", 2021.
26. Мартін Роберт С. Чистий код. Створення, аналіз і рефакторинг. – Київ: Видавництво Старого Лева, 2021.
27. Бендер Філіп. TypeScript для професіоналів. – Харків: Віват, 2023.
28. Seligman Greg. Learning PostgreSQL: A beginner's guide to building high-performance PostgreSQL database solutions. – Packt Publishing, 2020.
29. Kramer Mark. React Design Patterns and Best Practices. – Packt Publishing, 2019.

		Гелінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

30. Фаулер Мартін. Рефакторинг. Поліпшення існуючого коду. – Київ: Наш
Формат, 2020

ДОДАТКИ

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

Головна сторінка home:

```
'use client';

import { useState, useEffect } from 'react';
import './animations.css'; // We'll create this file next
import { useRouter } from 'next/navigation';
import { useSession } from 'next-auth/react';
import { useLanguage } from './i18n/LanguageContext';

interface Project {
  id: string;
  name: string;
  description: string;
  createdAt: Date;
  updatedAt: Date;
}

export default function Page() {
  const router = useRouter();
  const { data: session, status } = useSession();
  const { t } = useLanguage();
  const [searchQuery, setSearchQuery] = useState('');
  const [projects, setProjects] = useState<Project[]>([]);
  const [isCreateModalOpen, setIsCreateModalOpen] = useState(false);
  const [isEditModalOpen, setIsEditModalOpen] = useState(false);
  const [newProjectName, setNewProjectName] = useState('');
  const [newProjectDescription, setNewProjectDescription] = useState('');
  const [editingProject, setEditingProject] = useState<Project | null>(null);
  const [copyMessage, setCopyMessage] = useState({ show: false, text: '' });
  const [isLoading, setIsLoading] = useState(false);
  const [error, setError] = useState('');
  const [isDeleteModalOpen, setIsDeleteModalOpen] = useState(false);
  const [projectToDelete, setProjectToDelete] = useState<Project | null>(null);

  useEffect(() => {
    fetchProjects();
  }, []);

  const fetchProjects = async () => {
    try {
      const response = await fetch('/api/projects');
      const data = await response.json();
      setProjects(data);
    } catch (error) {
      console.error('Error fetching projects:', error);
    }
  };
};
```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const openCreateModal = () => {
  setIsCreateModalOpen(true);
};

const closeCreateModal = () => {
  setIsCreateModalOpen(false);
  setNewProjectName('');
  setNewProjectDescription('');
};

const handleCreateProject = async () => {
  if (status !== 'authenticated') {
    router.push('/auth/signin');
    return;
  }

  if (!newProjectName.trim()) {
    setError(t('projects.errors.nameRequired'));
    return;
  }

  setIsLoading(true);
  setError('');

  try {
    const response = await fetch('/api/projects', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        name: newProjectName,
        description: newProjectDescription
      }),
      credentials: 'include'
    });

    if (!response.ok) {
      const data = await response.json();
      throw new Error(data.error || t('projects.errors.createFailed'));
    }

    const data = await response.json();
    setProjects([...projects, data]);
    closeCreateModal();
  } catch (error) {
    console.error('Error creating project:', error);
    setError(error instanceof Error ? error.message : t('projects.errors.createFailed'));
  }
}

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    } finally {
        setIsLoading(false);
    }
};

const handleUpdateProject = async () => {
    if (editingProject && newProjectName.trim()) {
        try {
            const response = await fetch('/api/projects', {
                method: 'PUT',
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify({
                    id: editingProject.id,
                    name: newProjectName,
                    description: newProjectDescription || editingProject.de-
description,
                }),
            });

            if (response.ok) {
                const updatedProject = await response.json();
                setProjects(
                    projects.map((p) => (p.id === updatedProject.id ? updat-
edProject : p)),
                );
                setIsEditModalOpen(false);
                setEditingProject(null);
                setNewProjectName('');
                setNewProjectDescription('');
            }
        } catch (error) {
            console.error('Error updating project:', error);
        }
    }
};

const handleDeleteProject = async (id: string) => {
    try {
        const response = await fetch('/api/projects', {
            method: 'DELETE',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify({ id }),
        });

        if (response.ok) {

```

		Іглінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        setProjects(projects.filter((p) => p.id !== id));
    }
} catch (error) {
    console.error('Error deleting project:', error);
}
};

if (status === 'loading') {
    return (
        <div className="min-h-screen flex items-center justify-center bg-
[#121212]">
            <div className="text-white">{t('common.loading')}

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <button
            onClick={openCreateModal}
            className="bg-purple-600 hover:bg-purple-700 transition-
colors text-white px-4 py-2 rounded-md flex items-center"
        >
            <svg
                xmlns="http://www.w3.org/2000/svg"
                className="h-5 w-5 mr-1"
                fill="none"
                viewBox="0 0 24 24"
                stroke="currentColor"
            >
                <path
                    strokeLinecap="round"
                    strokeLinejoin="round"
                    strokeWidth={2}
                    d="M12 4v16m8-8H4"
                />
            </svg>
            {t('projects.create.button')}
        </button>

    </div>
</header>

<main className="flex-1 p-6">
    <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-
6">
        {projects
            .filter((project) =>
                project.name.toLowerCase().includes(searchQuery.toLow-
erCase()),
            )
            .map((project) => (
                <div
                    key={project.id}
                    className="bg-[#1e1e2d] rounded-lg overflow-hidden
hover:ring-1 hover:ring-purple-500 transition-all cursor-pointer"
                    onClick={() => (window.location.href = `/pro-
ject/${project.id}`)}
                >
                    <div className="p-5 flex items-start">
                        <div className="h-12 w-12 bg-purple-800
rounded-md flex items-center justify-center text-xl font-bold mr-4">
                            {project.name.charAt(0)}
                        </div>
                        <div>
                            <h3 className="text-lg font-medium">{pro-
ject.name}</h3>

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <p className="text-gray-400 text-sm">
            {project.description}
        </p>
    </div>
</div>
<div className="bg-[#181825] p-2 flex justify-end
space-x-2">

    <button
        className="p-1 hover:bg-gray-700 rounded
text-gray-400 hover:text-white transition-colors"
        onClick={(e) => {
            e.stopPropagation();
            setEditingProject(project);
            setIsEditModalOpen(true);
        }}
    >
        <svg
            xmlns="http://www.w3.org/2000/svg"
            className="h-5 w-5"
            fill="none"
            viewBox="0 0 24 24"
            stroke="currentColor"
        >
            <path
                strokeLinecap="round"
                strokeLinejoin="round"
                strokeWidth={2}
                d="M15.232 5.232 13.536 3.536m-
2.036-5.036a2.5 2.5 0 113.536 3.536L6.5 21.036H3v-3.572L16.732 3.732z"
            />
        </svg>
    </button>
    <button
        className="p-1 hover:bg-gray-700 rounded
text-gray-400 hover:text-white transition-colors"
        onClick={(e) => {
            e.stopPropagation();
            navigator.clipboard
                .writeText(project.name)
                .then(() => {
                    setCopyMessage({
                        show: true,
                        text: t('projects.copy.suc-
cess') + ' "' + project.name + '"',

false, text: '' });

                }, 3000);

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    })
    .catch((err) => {
        console.error('Failed to copy:

', err);

        setCopyMessage({
            show: true,
            text: t('projects.copy.er-
ror'),

        });
        setTimeout(() => {
            setCopyMessage({ show:

false, text: '' });

        }, 3000);
    });
}}
>
<svg
  xmlns="http://www.w3.org/2000/svg"
  className="h-5 w-5"
  fill="none"
  viewBox="0 0 24 24"
  stroke="currentColor"
>
  <path
    strokeLinecap="round"
    strokeLinejoin="round"
    strokeWidth={2}
    d="M8 16H6a2 2 0 01-2-2V6a2 2 0
012-2h8a2 2 0 02-2v-8a2 2 0 00-2-2v8a2 2 0
002 2z"
  />
</svg>
</button>
<button
  className="p-1 hover:bg-red-700 rounded
text-gray-400 hover:text-white transition-colors"
  onClick={(e) => {
    e.stopPropagation();
    setProjectToDelete(project);
    setIsDeleteModalOpen(true);
  }}
>
  <svg
    xmlns="http://www.w3.org/2000/svg"
    className="h-5 w-5"
    fill="none"
    viewBox="0 0 24 24"
    stroke="currentColor"
  >

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<path
                                strokeLinecap="round"
                                strokeLinejoin="round"
                                strokeWidth={2}
                                d="M19 71-.867 12.142A2 2 0
0116.138 21H7.862a2 2 0 01-1.995-1.858L5 7m5 4v6m4-6v6m1-10V4a1 1 0 00-1-1h-4a1 1 0
00-1 1v3M4 7h16"
                                />
                            </svg>
                        </button>
                    </div>
                </div>
            )}}

            { /* Show empty state if no projects match search */ }
            { projects.filter((project) =>
                project.name.toLowerCase().includes(searchQuery.toLower-
Case()),
                ).length === 0 &&
                searchQuery !== '' && (
                    <div className="col-span-full flex flex-col items-cen-
ter justify-center py-10">
                        <svg
                            xmlns="http://www.w3.org/2000/svg"
                            className="h-12 w-12 text-gray-500 mb-4"
                            fill="none"
                            viewBox="0 0 24 24"
                            stroke="currentColor"
                        >
                            <path
                                strokeLinecap="round"
                                strokeLinejoin="round"
                                strokeWidth={2}
                                d="M21 21l-6-6m2-5a7 7 0 11-14 0 7 7 0 0114
0z"
                                />
                            </svg>
                            <p className="text-gray-400 text-lg">
                                {t('projects.search.noResults')} + ' ' +
searchQuery + ' ' + ' '
                            </p>
                            <button
                                onClick={() => setSearchQuery('')}
                                className="mt-4 text-purple-500 hover:text-pur-
ple-400"
                            >
                                {t('projects.search.clearSearch')}
                            </button>
                        </div>
                    )
                )
            }
        </div>
    </div>

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Єфремов Ю.М.				64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    })

    {/* Empty project card - only show when not searching */}
    {searchQuery === '' && (
      <div
        onClick={openCreateModal}
        className="border border-dashed border-gray-700
rounded-lg flex items-center justify-center h-[140px] cursor-pointer hover:border-
purple-500 transition-colors"
      >
        <div className="text-center">
          <div className="mx-auto h-12 w-12 rounded-full bg-
purple-900 bg-opacity-20 flex items-center justify-center mb-2">
            <svg
              xmlns="http://www.w3.org/2000/svg"
              className="h-6 w-6 text-purple-500"
              fill="none"
              viewBox="0 0 24 24"
              stroke="currentColor"
            >
              <path
                strokeLinecap="round"
                strokeLinejoin="round"
                strokeWidth={2}
                d="M12 4v16m8-8H4"
              />
            </svg>
          </div>
          <p className="text-gray-400 text-sm">{t('pro-
jects.create.emptyState')}/* Create Project Modal */
{isCreateModalOpen && (
  <div className="fixed inset-0 bg-black bg-opacity-70 flex items-
center justify-center z-50">
    <div className="bg-[#121212] rounded-lg w-[500px] overflow-hid-
den border border-gray-800 shadow-xl">
      <div className="p-6">
        <div className="flex justify-between items-center mb-
6">
          <h3 className="text-lg font-medium">{t('pro-
jects.create.title')}

```

```

sition-colors"
        >
        <svg
            xmlns="http://www.w3.org/2000/svg"
            className="h-5 w-5"
            viewBox="0 0 20 20"
            fill="currentColor"
        >
            <path
                fillRule="evenodd"
                d="M4.293 4.293a1 1 0 011.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414 1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z"
                clipRule="evenodd"
            />
        </svg>
    </button>
</div>
<div className="mb-6">
    <label className="block text-sm font-medium mb-2">
        {t('projects.create.name.label')}
    </label>
    <input
        type="text"
        placeholder={t('projects.create.name.placeholder')}
        className="w-full bg-[#1e1e2d] border border-gray-700 rounded-md px-4 py-2 text-white focus:outline-none focus:ring-1 focus:ring-purple-500"
        value={newProjectName}
        onChange={(e) => setNewProjectName(e.target.value)}
        autoFocus
    />
</div>
<div className="mb-6">
    <label className="block text-sm font-medium mb-2">
        {t('projects.create.description.label')}
    </label>
    <textarea
        placeholder={t('projects.create.description.placeholder')}
        className="w-full bg-[#1e1e2d] border border-gray-700 rounded-md px-4 py-2 text-white focus:outline-none focus:ring-1 focus:ring-purple-500 min-h-[100px]"
        value={newProjectDescription}
        onChange={(e) => setNewProjectDescription(e.target.value)}
    />
</div>

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        />
      </div>
      <div className="flex justify-end space-x-3">
        <button
          onClick={closeCreateModal}
          className="px-4 py-2 bg-gray-700 text-white
rounded-md hover:bg-gray-600 transition-colors"
        >
          {t('common.cancel')}
        </button>
        <button
          onClick={handleCreateProject}
          className="px-4 py-2 bg-purple-600 text-white
rounded-md hover:bg-purple-700 transition-colors"
        >
          {t('common.create')}
        </button>
      </div>
    </div>
  </div>
</div>
)}

{/* Edit Project Modal */}
{isEditModalOpen && editingProject && (
  <div className="fixed inset-0 bg-black bg-opacity-70 flex items-
center justify-center z-50">
    <div className="bg-[#121212] rounded-lg w-[500px] overflow-hid-
den border border-gray-800 shadow-xl">
      <div className="p-6">
        <div className="flex justify-between items-center mb-
6">
          <h3 className="text-lg font-medium">{t('pro-
jects.edit.title')}</h3>
          <button
            onClick={() => setIsEditModalOpen(false)}
            className="text-gray-400 hover:text-white tran-
sition-colors"
          >
            <svg
              xmlns="http://www.w3.org/2000/svg"
              className="h-5 w-5"
              viewBox="0 0 20 20"
              fill="currentColor"
            >
              <path
                fillRule="evenodd"
                d="M4.293 4.293a1 1 0 011.414 0L10

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

8.58614.293-4.293a1 1 0 111.414 1.414L11.414 1014.293 4.293a1 1 0 01-1.414 1.414L10
11.4141-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z"
clipRule="evenodd"
/>
</svg>
</button>
</div>
<div className="space-y-4">
  <div>
    <label className="block text-sm font-medium mb-
2">
      {t('projects.edit.name.label')}
    </label>
    <input
      type="text"
      placeholder={t('projects.edit.name.place-
holder')}}
      className="w-full bg-[#1e1e2d] border bor-
der-gray-700 rounded-md px-4 py-2 text-white focus:outline-none focus:ring-1 fo-
cus:ring-purple-500"
      value={newProjectName}
      onChange={(e) => {
        setNewProjectName(e.target.value);
        setEditingProject({
          ...editingProject,
          name: e.target.value,
        });
      }}
      autoFocus
    />
  </div>
  <div>
    <label className="block text-sm font-medium mb-
2">
      {t('projects.edit.description.label')}
    </label>
    <textarea
      placeholder={t('projects.edit.descrip-
tion.placeholder')}}
      className="w-full bg-[#1e1e2d] border bor-
der-gray-700 rounded-md px-4 py-2 text-white focus:outline-none focus:ring-1 fo-
cus:ring-purple-500 min-h-[100px]"
      value={newProjectDescription}
      onChange={(e) => {
        setNewProjectDescription(e.tar-
get.value);
        setEditingProject({
          ...editingProject,
          description: e.target.value,

```

		Ізлінський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    });
  }}
  />
</div>
</div>
<div className="flex justify-end space-x-3 mt-6">
  <button
    onClick={() => setIsEditModalOpen(false)}
    className="px-4 py-2 bg-gray-700 text-white
rounded-md hover:bg-gray-600 transition-colors"
  >
    {t('common.cancel')}
  </button>
  <button
    onClick={handleUpdateProject}
    className="px-4 py-2 bg-purple-600 text-white
rounded-md hover:bg-purple-700 transition-colors"
  >
    {t('projects.edit.saveChanges')}
  </button>
</div>
</div>
</div>
</div>
)}

{/* Copy to clipboard notification */}
{copyMessage.show && (
  <div className="fixed bottom-4 right-4 bg-gray-800 text-white px-4
py-2 rounded-md shadow-lg border border-gray-700 z-50 animate-fade-in-up">
    <div className="flex items-center">
      <svg
        xmlns="http://www.w3.org/2000/svg"
        className="h-5 w-5 mr-2 text-green-500"
        viewBox="0 0 20 20"
        fill="currentColor"
      >
        <path
          fillRule="evenodd"
          d="M10 18a8 8 0 100-16 8 8 0 0016zm3.707-9.293a1
1 0 00-1.414-1.414L9 10.586 7.707 9.293a1 1 0 00-1.414 1.414L2 11 0 14-
4z"
          clipRule="evenodd"
        />
      </svg>
      {copyMessage.text}
    </div>
  </div>
)}

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

```
{/* Delete Project Modal */}
{isDeleteModalOpen && projectToDelete && (
  <div className="fixed inset-0 bg-black bg-opacity-70 flex items-center justify-center z-50">
    <div className="bg-[#121212] rounded-lg w-[400px] overflow-hidden border border-gray-800 shadow-xl">
      <div className="p-6">
        <div className="flex justify-between items-center mb-6">
          <h3 className="text-lg font-medium">{t('projects.delete.title')}</h3>
          <button
            onClick={() => setIsDeleteModalOpen(false)}
            className="text-gray-400 hover:text-white transition-colors"
          >
            <svg
              xmlns="http://www.w3.org/2000/svg"
              className="h-5 w-5"
              viewBox="0 0 20 20"
              fill="currentColor"
            >
              <path
                fillRule="evenodd"
                d="M4.293 4.293a1 1 0 011.414 1.414L11.414 10l1.414 1.414a1 1 0 01-1.414 1.414L8.586 11.414a1 1 0 01-1.414 1.414L4.293 11.414a1 1 0 01-1.414 1.414L1 11.414a1 1 0 011.414 1.414L4.293 14.293a1 1 0 01-1.414 1.414L1 14.293a1 1 0 01-1.414 1.414z"
                clipRule="evenodd"
              />
            </svg>
          </button>
        </div>
        <p className="text-gray-300 mb-6">
          {t('projects.delete.confirmation', { name: projectToDelete.name })}
        </p>
        <div className="flex justify-end space-x-3">
          <button
            onClick={() => setIsDeleteModalOpen(false)}
            className="px-4 py-2 bg-gray-700 text-white rounded-md hover:bg-gray-600 transition-colors"
          >
            {t('common.cancel')}
          </button>
          <button
            onClick={() => {
              handleDeleteProject(projectToDelete.id);
              setIsDeleteModalOpen(false);
              setProjectToDelete(null);
            }}
          >
```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Єфремов Ю.М.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        }}
        className="px-4 py-2 bg-red-600 text-white
rounded-md hover:bg-red-700 transition-colors"
        >
        {t('common.delete')}
    </button>
</div>
</div>
</div>
</div>
    )}
</div>
    );
}

```

		Ігліньський І.Ю.			ІПЗ.КР.Б – 121 – 25 -ПЗ	Арк.
		Сфремов Ю.М.				71
Змн.	Арк.	№ докум.	Підпис	Дата		