

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

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

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі системи та технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Платформа для створення програми лояльності для бізнесу
з використанням NFT»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІС-93

Коваленко Євген Олегович _____

Керівник:

Асистент кафедри ІСТ

Майер Ілля Сергійович _____

Рецензент:

Старший викладач

Алещенко Олексій Вадимович _____

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

Студент (-ка) _____

Київ – 2023 року

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

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

« ___ » _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту
Коваленку Євгену Олеговичу

1. Тема проєкту «Платформа для створення програми лояльності для бізнесу з використанням NFT», керівник проєкту Майер Ілля Сергійович асистент кафедри ІСТ затверджені наказом по університету від «31» травня 2023 р. № 2101-с
2. Термін подання студентом проєкту: «12» червня 2023 року
3. Вихідні дані до проєкту: Технічне завдання
4. Зміст пояснювальної записки:
 - 1) Задача побудови програми для відстеження задач і проєктів, аналіз якості та тестування програмного забезпечення
 - 2) Архітектура та бібліотеки для розробки веб додатку
 - 3) Розгортання програмного забезпечення
 - 4) Аналіз якості та тестування програмного забезпечення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

- 1) Діаграма використання
- 2) Схема бази даних
- 3) Діаграма класів серверної частини
- 4) Схема архітектури сервісу

6. Дата видачі завдання «14» листопада 2022 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
	<i>Затвердження теми проєкту</i>	15.03.2023-21.03.2023	
	<i>Вивчення та аналіз завдання</i>	22.03.2023-23.04.2023	
	<i>Розробка архітектури серверної частини та модуля з блокчейн технологіями та схемою бази даних</i>	14.04.2023-01.05.2023	
	<i>Розробка модулів архітектури клієнтської частини</i>	2.05.2023-12.05.2023	
	<i>Розробка модуля імпорту інформації через csv-файли</i>	13.05.2023-20.05.2023	
	<i>Оформлення пояснювальної записки</i>	20.05.2023-29.05.2023	
	<i>Захист програмного продукту</i>	11.06.2023	
	<i>Передзахист</i>	14.06.2023	
	<i>Захист</i>	20.06.2023	

Студент

Євген КОВАЛЕНКО

Керівник

Ілля МАЙЕР

АНОТАЦІЯ

Коваленко Є.О. Платформа для створення програми лояльності для бізнесу з використанням NFT. КПІ ім. Ігоря Сікорського, Київ, 2023.

Проект містить 66 с. тексту, 19 рисунків, 35 таблиці, посилання на 8 літературних джерел, додатки та 3 конструкторських документів.

Ключові слова: програма лояльності, NFT, smart-contract, оптимізація, покращення програми лояльності.

Об'єктом розробки є платформа для створення програми лояльності для бізнесу з використанням NFT. Мета розробки – покращення та вирішення проблем програм лояльності різних бізнесів. У дипломному проекті розроблено платформу для створення програми лояльності для бізнесу з використанням NFT, яка дає можливість: імпортувати продукти та користувачів бізнесу за допомогою csv-файлу, створити NFT в блокчейні NEAR Protocol, надати їй налаштувань щодо використання користувачем бізнесу, прив'язати NFT до продукту бізнесу, створити унікальну адресу користувачеві бізнесу в блокчейні NEAR та відправити NFT користувачеві за створеною блокчейн-адресою.

Розроблений програмний продукт був на мові TypeScript та Rust, який розділяється на серверну, клієнтську та блокчейн частини. Була описана архітектура проекту, визначені потрібні інструменти і бібліотеки для реалізації. Проведено моделювання та конструювання бази даних, модулів та методів ПЗ.

В результаті роботи було виконано всю розробку серверної, блокчейн та клієнтської частин, проведено тестування і аналіз розробленого функціоналу з переліком тест-кейсів, що допоможе бізнесам покращити свої програми лояльності

SUMMARY.

Kovalenko Y.O. Platform for creating a loyalty program for business using NFT. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2023.

The project contains 66 pages of text, 19 figures, 35 tables, references to 8 literary sources, appendices and 3 design documents.

Keywords: loyalty program, NFT, smart-contract, optimization, improvement of the loyalty program.

The object of development is a platform for creating a loyalty program for business using NFT. The purpose of the development is to improve and solve the problems of loyalty programs for various businesses. The thesis project developed a platform for creating a loyalty program for businesses using NFTs, which allows you to: import products and business users using a csv file, create an NFT in the NEAR Protocol blockchain, provide it with settings for use by a business user, link an NFT to a business product, create a unique address for a business user in the NEAR blockchain, and send an NFT to the user at the created blockchain address.

The software product was developed in TypeScript and Rust, which is divided into server, client, and blockchain parts. The project architecture was described, and the necessary tools and libraries for implementation were identified. We modeled and designed the database, software modules, and methods.

As a result of the work, the entire development of the server, blockchain, and client parts was completed, and the developed functionality was tested and analyzed with a list of test cases that will help businesses improve their loyalty programs.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер сістем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IC93.110БАК.004 ПЗ	Пояснювальна записка	66		
6	A3	IC93.110БАК.004 Д1	Платформа для створення програми лояльності для бізнесу з використанням NFT. Діаграма використання	1		
7						
8						
9						
10						
11	A3	IC93.110БАК.004 Д2	Платформа для створення програми лояльності для бізнесу з використанням NFT. Схема бази даних	1		
12						
13						
14						
15	A3	IC93.110БАК.004 Д3	Платформа для створення програми лояльності для бізнесу з використанням NFT. Діаграма класів серверної частини	1		
16						
17						
18						
19						
20	A3	IC93.110БАК.004 Д4	Платформа для створення програми лояльності для бізнесу з використанням NFT. Схема архітектури сервісу	1		
21						
22						
23						
24						
25						
26						
27						
28						

IC93.110БАК.004 ТП

Зм.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Коваленко Є.О.			Платформа для створення програми лояльності для бізнесу з використанням NFT. Відомість проекту	Літ.	Аркуш	Аркушів
Керівн.		Майер І.С.					1	1
						КПІ ім. Ігоря Сікорського Група IC-93		
Затв.								

Пояснювальна записка
до дипломного проєкту
на тему: «Платформа для створення програми
лояльності для бізнесу з використанням NFT»

Київ – 2023 року

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	7
1.1 Загальні положення.....	7
1.2 Змістовний опис і аналіз предметної області.....	9
1.3 Аналіз відомих програмних продуктів	9
1.4 Аналіз вимог до програмного забезпечення	13
1.4.1 Розроблення функціональних вимог	13
1.4.2 Розроблення нефункціональних вимог	21
1.4.3 Постановка завдання	21
Висновки до розділу	21
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	21
2.1 Вхідні дані.....	21
2.2 Вихідні дані.....	22
2.3 Опис таблиць, які використовуються для зберігання даних	22
Висновки до розділу	24
3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	25
3.1 Моделювання та аналіз програмного забезпечення	25
3.1.1 Серверна сторона	26
3.1.2 Використання бази даних	27
3.1.3 Середовище розробки.....	28
3.1.4 Блокчейн середовище	28
3.2 Архітектура програмного забезпечення	29
3.3 Конструювання програмного забезпечення	32
3.3.1 Модулі програмного забезпечення.....	32
3.3.2 Модулі програмного забезпечення.....	36
3.4 Діаграма послідовності.....	38
3.5 Діаграма потоків даних	39
Висновки до розділу	40
4 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	41
4.1 Керівництво користувача	41
4.2 Випробовування програмного продукту	48
4.2.1 Мета випробувань	48
4.2.2 Загальні положення.....	48

					ІС93.110БАК.004 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	Платформа для створення програм лояльності з використанням NFT. Пояснювальна записка						
Розроб.		Коваленко Є. О.							Літ.	Арк.	Акрушів
Перевір.		Майер І. С.							2	66	
									КПШ ім. Ігоря Сікорського		
Затверд.									Група ІС-93		

4.2.3 Результати випробувань	48
Висновки до розділу	63
ВИСНОВКИ	63
ПЕРЕЛІК ПОСИЛАНЬ	65

					<i>IC93.110БАК.004 ПЗ</i>	Арк.
						3
Зм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРО- ЧЕНЬ І ТЕРМІНІВ

API - application programming interface

БД - База даних

Клієнт - власник бізнесу, який використовує сервіс для створення програми лояльності з використанням NFT

NFT - (Non Fungible Token) - це унікальний цифровий актив, який побудований на основі блокчейну і не може бути замінений на інший актив однакової вартості. Кожен NFT має унікальну ідентифікаційну мітку, яка його відрізняє від інших

NFT-купон - створений за допомогою розробленого сервісу NFT, який може бути використаний як купон для знижок

Smart-contract - це програмний код, що запускається на блокчейні і містить правила та умови для автоматизованої взаємодії та виконання угод між сторонами без необхідності посередництва. Він є самовиконуючим та незмінним, що означає, що його логіка виконується автоматично без можливості зміни або втручання з боку третіх осіб.

Blockchain - це розподілена технологія збереження та передачі даних, що базується на децентралізованій мережі комп'ютерів, відомих як вузли. Вона використовує хронологічно впорядкований ланцюжок блоків для збереження інформації.

Крипто-адреса - це унікальний ідентифікатор, який використовується в криптовалютних системах для ідентифікації одержувача або відправника транзакції.

CRM-система - (Customer Relationship Management) - це система, що допомагає компаніям управляти та взаємодіяти зі своїми клієнтами. Вона об'єднує різні аспекти бізнесу, такі як продажі, маркетинг, обслуговування клієнтів, аналітика та інші, у єдину платформу для ефективного керування відносинами з клієнтами.

SDK - software development kit

					<i>IS93.110BAK.004 ПЗ</i>	Арк.
						4
Зм.	Лист	№ докум.	Підпис	Дата		

ВСТУП

Ми живемо в еру цифрових інновацій, де ми стикаємося зі зростаючою популярністю крипто-валют, а особливо NFT, які надають нам нові можливості у сфері мистецтва, музики, геймінгу та багатьох інших галузях. Також ми можемо застосувати цю потужну технологію для покращення практик програм лояльності для оф лайн бізнесів.

Уявіть, що бізнеси зможуть замінити традиційні паперові купони на унікальні цифрові активи, які можна зберігати та обмінювати в електронному вигляді.

За допомогою NFT, можливо створити цифрові купони з високою ступенем безпеки та унікальною ідентифікацією. Це відкриває широкі можливості для персоналізації та адаптації акцій та знижок до індивідуальних потреб наших клієнтів різних бізнесів, від кав'ярень, ресторанів та кафе до автозаправних станцій, спортзалів та перукарень.

Також круто було б мати можливість інтегрувати NFT-купони в власний мобільний застосунок, який уже використовують клієнти.

Крім того, використання NFT для купонів на знижки дає можливість бізнесам здійснювати прозору та надійну відстежування використання купонів.

Кожен токен має унікальний ідентифікатор, який можна перевірити в блокчейні, що гарантує його автентичність та непідробність.

Таким чином, уникнутья проблеми з підробками або подвійним використанням купонів.

Враховуючи всі ці плюси NFT був розроблений дипломний проєкт, який надає бізнесам створити такі NFT-купони.

Метою розробки є надання зручного, ефективного та простого методу створення NFT-купонів та прив'язкою їх до реальних продуктів

Практичним значенням дипломного проєкту є розроблений продукт реалізований у вигляді веб-сервісу, який надає можливості створення NFT-купонів, з прив'язкою їх до реальних продуктів, або послуг та з можливістю

					<i>IS93.110BAK.004 ПЗ</i>	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

відправки NFT клієнту бізнесу, попередньо створивши йому для цього крипто-адресу.

					<i>IC93.110БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Наразі багато бізнесів використовують програми лояльності для заохочення клієнтів користуватись їхніми послугами. Кожен бізнес робить це по різному, хтось друкує купони, хтось використовує картки для накопичення бонусів, хтось використовує телеграм ботів і отримує бонуси за номером телефону.

Програма лояльності - це стратегія, що використовується бізнесами для залучення, утримання та стимулювання вірних клієнтів. Вона базується на наданні певних переваг, винагород, знижок або привілеїв клієнтам, які регулярно користуються послугами або купують товари від даного бізнесу.

Основна мета програм лояльності полягає в підвищенні лояльності клієнтів, збільшенні повторних покупок, залученні нових клієнтів та підвищенні загального обсягу продажів.

Вона дозволяє бізнесам встановлювати довгострокові взаємовигідні відносини зі своїми клієнтами, створюючи позитивний досвід та відчуття цінності. Бізнеси використовують програми лояльності з різних причин.

Ось декілька головних:

- Залучення нових клієнтів: Програми лояльності можуть привертати увагу нових клієнтів та стимулювати їх перше придбання, що дозволяє розширити базу клієнтів;
- Підтримка повторних покупок: Завдяки програмам лояльності бізнеси можуть стимулювати клієнтів до повторних покупок та зберегти їхню вірність, що допомагає збільшити дохід і покращити рентабельність;
- Посилення зв'язку з клієнтами: Програми лояльності створюють можливість більш активної взаємодії з клієнтами, зокрема через розсилку ексклюзивних пропозицій, персоналізовані знижки та спеціальні події.
- Збільшення середнього чека: Відмінні привілеї та знижки, доступні через програму лояльності, можуть спонукати клієнтів до збільшення обсягу покупок.

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
						7
Зм.	Лист	№ докум.	Підпис	Дата		

– Збільшення середнього чека: Відмінні привілеї та знижки, доступні через програму лояльності, можуть спонукати клієнтів до збільшення обсягу покупок або використання додаткових послуг, що підвищує середній чек;

– Збільшення конкурентоспроможності: Наявність програми лояльності може виділити бізнес серед конкурентів, покращити його репутацію та привернути увагу споживачів.

Все це допомагає покращити стосунки з клієнтами, підвищити їх задоволеність та забезпечити стабільний потік продажів для бізнесу.

Однак, більшість цих програм лояльності застарілі і вже не викликають такого захоплення як раніше.

Через це багато бізнесів шукають рішення цієї проблеми, як створити успішну та популярну програму лояльності, якою будуть користуватись багато людей.

Цю проблему можна вирішити за допомогою сервісу, який дасть можливість створити програму лояльності з використанням NFT (Non Fungible Token) [2].

Використання NFT для програм лояльності вирішує безліч проблем, а саме:

– Унікальність і необоротність: NFT - це цифрові активи, які можуть бути використані для представлення унікальних об'єктів, таких як мистецькі твори, відео, музика, або ж купон для знижки. Кожен NFT має свій власний цифровий слід, який гарантує його унікальність. Це дає можливість відслідковувати власності та походження цифрових активів, запобігаючи підробці та шахрайству.

– Контроль над налаштуваннями NFT: творці NFT мають можливість контролювати обмеження на розповсюдження і використання цих активів за допомогою криптовалютних технологій

– Відкритий ринок: власники NFT, вони можуть безпосередньо продавати свої NFT-купони та отримувати безпосередню винагороду без посередників. Таким чином, вони отримують більший контроль над своїми NFT. А бізнесу це дає великий маркетинговий потенціал

– Покращена прозорість та незмінність: Технологія блокчейн, на якій базуються NFT, забезпечує прозорість та незмінність у всіх операціях.

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
						8
Зм.	Лист	№ докум.	Підпис	Дата		

Інформація про транзакції з NFT зберігається на розподіленій мережі, що робить її відстеження та перевірку доступною для всіх зацікавлених сторін. Це може бути корисно при підтвердженні автентичності та походження цифрових активів.

Тому враховуючи ці плюси NFT, багато бізнесів хотіло б мати такий сервіс з використанням NFT для створення програм лояльності.

1.2 Змістовний опис і аналіз предметної області

Програмний продукт для вирішення цієї проблеми являє собою веб-сервіс в якому можна імпортувати продукти або послуги бізнесу та імпортування клієнтів бізнесу за допомогою csv-файлів, створити NFT-купон для знижки на вибраний продукт та відправити його своїм постійним клієнтам.

Такий продукт буде повністю автоматизований і не буде потребувати зовнішнього втручання в створення програми лояльності та NFT-купонів, окрім як власника бізнесу.

1.3 Аналіз відомих програмних продуктів

У результаті аналізу існуючих програмних продуктів були знайдені такі рішення:

- Hang
- Starbucks Odyssey
- Upfront

Оскільки проблема використанням NFT для створення програм лояльності виникла нещодавно, єдиним онлайн сервісом, який дає деякі можливість її вирішення є сервіс Hang. За допомогою сервісу Hang власник бізнес може створити програму лояльності у вигляді гри, де користувачі цієї програми будуть отримувати нагороди у вигляді колекційних NFT та звичайних купонів на знижки.

					<i>IS93.110BAK.004 ПЗ</i>	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

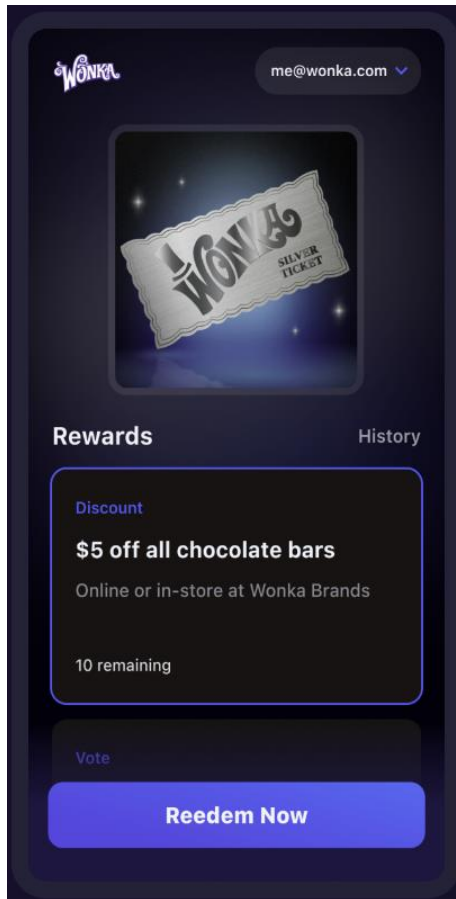


Рисунок 1.1 – Скріншот застосунку, який підключив програму лояльності від hang

Також у Hang реалізована адмін-панель, де можна налаштувати програму лояльності та слідкувати за активністю користувачів

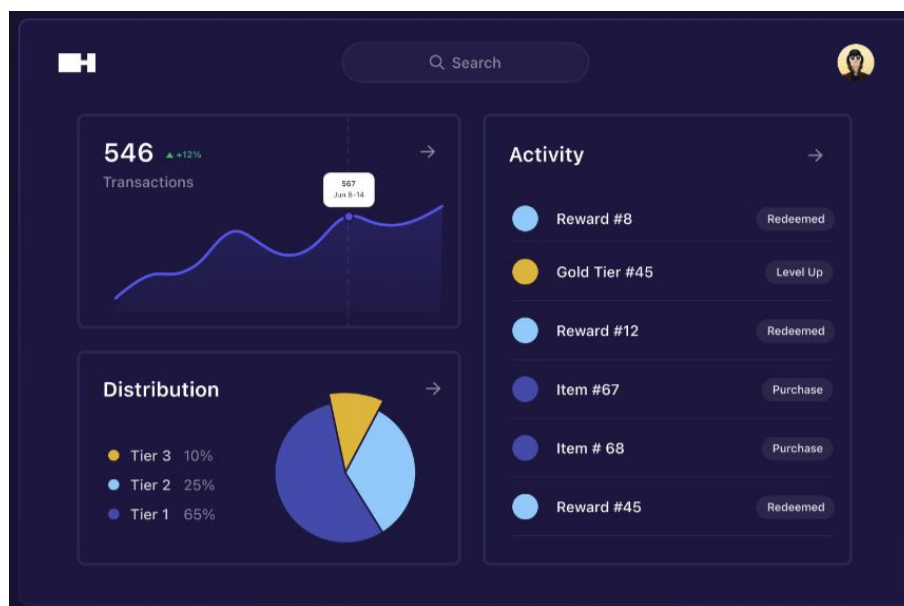


Рисунок 1.2 – Скріншот застосунку, який підключив програму лояльності від hang

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

Перевагами Hang є те що вони надають бізнесам широкий вибір як можна налаштувати програму лояльності у вигляді гри, але з цієї переваги випливає і проблема, що не всім бізнесам потрібна така складна система.

Також можна зазначити те, що сервіс Hang може бути інтегрованим в різні CRM-системи, що дає більше зручності, а ніж простий імпорт csv-файлів.

З недоліків можна зазначити, що в сервісі немає змоги обміну чи продажу колекційних NFT, продаж NFT-купонів на знижки, через що бізнес, який підключив цю програму лояльності втрачає інтерес користувачів та маркетингових потенціал.

Також вагомим недоліком є те що сервіс Hang, не дає можливості продажу відкритого NFT-купонів від безпосередніх власників бізнесів.

Існує ще один сервіс, який дає вирішення схожих проблем, але який заточений лише під компанію Starbucks - Starbucks Odyssey.

Поки що цей сервіс в бета-тесті, але вже можна сказати, що по функціоналу, він є трохи схожим на сервіс Hang, проте в сервісі Starbucks Odyssey у користувача вже є можливість продажу або обміну свого NFT-купону на різних сервісах для торгівлі NFT.

Також особливістю цього сервісу є те що, на відміну від Hang, він дає можливість створення колекційних NFT.

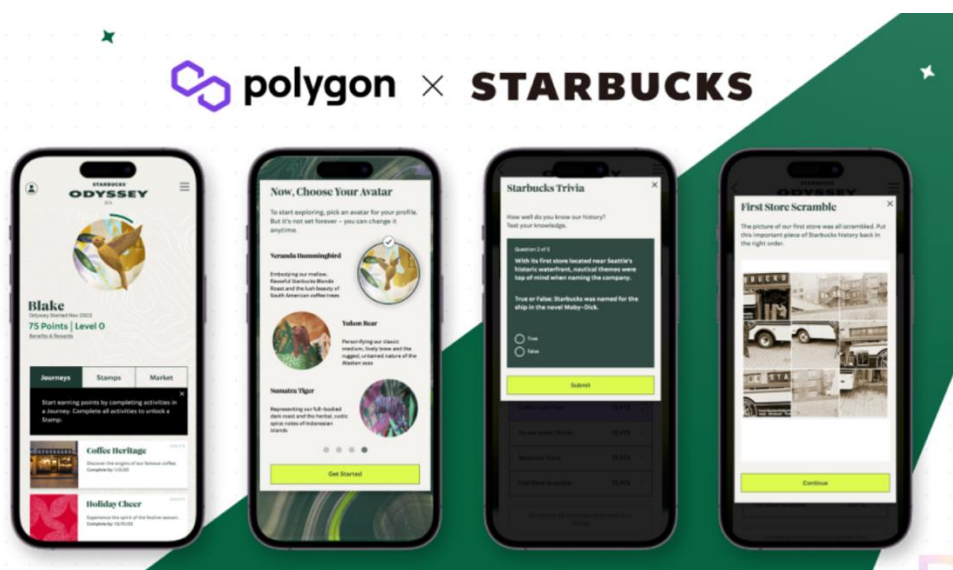


Рисунок 1.3 – Скріншот застосунку Starbucks Odyssey

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

З мінусів Starbucks Odyssey можна відзначити те, що він спеціалізований суто під Starbucks і ніякий бізнес не може собі інтегрувати таку ж систему.

Також, як і Hang, цей сервіс використовує блокчейн polygon, який є централізованим та побудований на базі іншого блокчейну Ethereum, який має безліч недоліків.

Також можна згадати український сервіс для створення сертифікатів для програм лояльності Urfront. Даний сервіс не є популярним, проте є достатньо технологічним і може створити конкуренцію схожим проектам на українському ринку.

З переваг Urfront можна відзначити практичність та легкість використання. За допомогою можна швидко створити електронні сертифікати для своєї програми лояльності та розіслати їх користувачам за e-mail адресою. У Urfront є можливість моніторингу всіх сертифікатів та їх використань.

Проте цей сервіс не ідеальний, він не використовує переваги блокчейнів та NFT, що зменшує маркетинговий потенціал для бізнесу, якщо він обере для використання саме цей сервіс.

Він не дає можливості імпорту продуктів та прив'язки їх до сертифікатів, а створює лише подарункові грошові сертифікати.

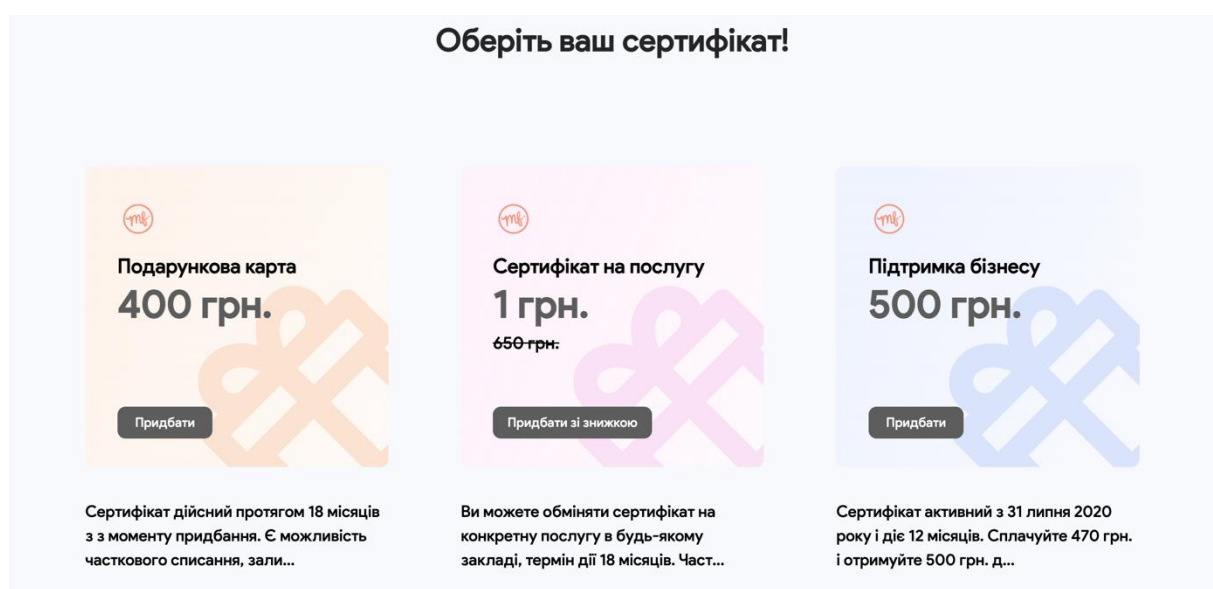


Рисунок 1.4 – Скріншот адмін-панелі застосунку Urfront

Як можна побачити недоліків в цього сервісу набагато більше ніж

					IC93.110BAK.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

переваг, тому він не користується популярністю в власників бізнесів для створення власних програм лояльності для своїх користувачів

Отже, як можна побачити з аналізу відомих програмних продуктів універсального рішення цієї проблеми майже немає, тому створення сервісу для створення програм лояльності з використанням NFT, з яким можна імпортувати продукти та користувачів бізнесу за допомогою csv-файлу, створити NFT в блокчейні NEAR Protocol, надати їй налаштувань щодо використання користувачем бізнесу, прив'язати NFT до продукту бізнесу, створити унікальну адресу користувачеві бізнесу в блокчейні NEAR та відправити NFT користувачеві за створеною блокчейн-адресою було б дуже доречним та унікальним рішенням, яке б привернуло б багато уваги як до програми лояльності, так і до самого бізнесу, який його інтегрував.

1.4 Аналіз вимог до програмного забезпечення

1.4.1 Розроблення функціональних вимог

Для створення use-case діаграми потрібно визначити акторів, що взаємодітимуть з програмним забезпеченням. Оскільки застосунок є веб-сервісом, єдиним учасником системи є користувач. Додаток може бути використаний за нижче наведеними варіантами використання(use cases) у таблицях 1.1 – 1.5

Таблиця 1.1 – Варіант використання Реєстрації в системі

Номер	UC1
Назва	Реєстрація користувача в системі
Опис	Користувач реєструється в системі за допомогою сервісу google auth
Учасники	Користувач
Передумови	

Післяумови	Користувач заходить в адмін-панель
Основний сценарій	<ol style="list-style-type: none"> 1) Користувач відкриває браузер та заходить на сайт 2) Нажимає на кнопку реєстрації через сервіс google auth 3) Вибирає свій email для реєстрації 4) Нажимає кнопку sign in

Таблиця 1.2 – Варіант використання імпорт продуктів в систему

Номер	UC2
Назва	Імпорт продуктів користувача в системі
Опис	Користувач імпортує продукти за допомогою csv-файлу
Учасники	Користувач
Передумови	Користувач зареєстрований в системі
Післяумови	Користувач імпортував нові продукти в систему і може їх бачити, редагувати та видаляти в системі
Основний сценарій	<ol style="list-style-type: none"> 1) Користувач відкриває браузер та заходить на сайт

	<ul style="list-style-type: none"> 2) Переходить на сторінку контролю продуктів 3) Нажимає на кнопку імпорту продуктів 4) Вибирає файл для імпорту
--	---

Таблиця 1.3 – Варіант використання імпорту клієнтів в систему

Номер	УСЗ
Назва	Імпорт клієнтів користувача в системі
Опис	Користувач імпортує клієнтів за допомогою csv-файлу
Учасники	Користувач
Передумови	Користувач зареєстрований в системі
Післяумови	Користувач імпортував нові продукти в систему і може їх бачити, редагувати та видаляти в системі
Основний сценарій	<ul style="list-style-type: none"> 1) Користувач відкриває браузер та заходить на сайт 2) Переходить на сторінку контролю клієнтів 3) Нажимає на кнопку імпорту клієнтів 4) Вибирає файл для імпорту

Зм.	Лист	№ докум.	Підпис	Дата

--	--

Таблиця 1.4 – Варіант використання створення NFT-купона продукта

Номер	UC4
Назва	Створення NFT-купонів в системі
Опис	Користувач створює NFT-купони з прив'язаним продуктом
Учасники	Користувач
Передумови	Користувач зареєстрований в системі. Користувач має імпортовані продукти
Післяумови	Користувач створив NFT-купон в системі і може його бачити, редагувати та видаляти в системі
Основний сценарій	<ol style="list-style-type: none"> 1) Користувач відкриває браузер та заходить на сайт 2) Переходить на сторінку контролю NFT-купонів 3) Нажимає на кнопку створення NFT-купону 4) Вибирає продукт для якого хоче створити NFT-купон 5) Вибирає налаштування для NFT-купон 6) Нажимає кнопку зберегти

Зм.	Лист	№ докум.	Підпис	Дата

Таблиця 1.5 – Варіант використання відправка NFT-купона клієнту в системі

Номер	UC5
Назва	Відправка NFT-купона клієнту в системі
Опис	Користувач відправляє NFT-купон з прив'язаним продуктом на створений криптовалютний адрес клієнта
Учасники	Користувач
Передумови	Користувач зареєстрований в системі. Користувач має імпортовані продукти. Користувач має імпортованих клієнтів Користувач має створенні NFT-купони
Післяумови	Користувач відправив NFT-купон в системі клієнту
Основний сценарій	<ol style="list-style-type: none"> 1) Користувач відкриває браузер та заходить на сайт 2) Переходить на сторінку контролю NFT-купонів 3) Нажимає на кнопку відправки NFT-купону 4) Вибирає клієнта, якому хоче

Зм.	Лист	№ докум.	Підпис	Дата

	<p>відправити NFT-купон</p> <p>5) Нажимає кнопку відправити</p>
--	---

Виходячи з вище описаних вимог та аналізу аналогів можна сформуванати наступні функціональні вимоги представлені у таблицях 1.6 – 1.12:

Таблиця 1.6 – Опис функціональної вимоги “Обробка вхідних значень”

Номер	R1
Назва	Обробка вхідних значень
Опис	Система повинна приймати вхідні дані користувача зберігати їх та показувати на адмін-панелі

Таблиця 1.7 – Опис функціональної вимоги “Авторизація”

Номер	R2
Назва	Авторизація
Опис	Повинна бути реалізована авторизація користувача в систему

Таблиця 1.8 – Опис функціональної вимоги “Відображення даних про NFT-купони”

Номер	R3
-------	----

Назва	Відображення даних про NFT-купони
Опис	Система повинна видавати коректні дані про NFT-купони

Таблиця 1.9 – Опис функціональної вимоги “Відображення даних про використання NFT-купонів“

Номер	R4
Назва	Додавання продуктів в систему для подальшого використання
Опис	Система повинна надати можливість додавання коректних продуктів, для подальшого їх використання при створенні NFT-купону

Таблиця 1.10 – Опис функціональної вимоги “Відображення даних про використання NFT-купонів“

Номер	R5
Назва	Додавання користувачів в систему для подальшого використання
Опис	Система повинна надати можливість додавання коректних користувачів, для подальшого їх використання при надсиланні NFT-купону та створенні крипто-адреси

Таблиця 1.11 – Опис функціональної вимоги “Відображення даних про використання NFT-купонів“

Номер	R6
Назва	Створення NFT-купону
Опис	Система повинна надати можливість створити NFT з відповідними налаштуваннями для ефективної програми лояльності

Таблиця 1.12 – Опис функціональної вимоги “Відображення даних про використання NFT-купонів“

Номер	R7
Назва	Відправка NFT-купону користувачеві
Опис	Система повинна надавати можливість для створення крипто-адреси, та відправки NFT-купону для цієї адреси її

Таблиця 1.13 – Матриця трасування вимог

	R1	R2	R3	R4	R5	R6	R7
UC1							
UC2							
UC3							
UC4							
UC5							

1.4.2 Розроблення нефункціональних вимог

Програмне забезпечення повинне відповідати наступним нефункціональним вимогам:

- локалізація інтерфейсу
- англійська;
- підтримувані браузери Apple Safari, Google Chrome, Android Browser, Metamask Browser;
- доступ до мережі Інтернет.

1.4.3 Постановка завдання

Розроблене програмне забезпечення призначене для створення, редагування та моніторингу програм лояльності з використанням NFT.

Мета роботи – створення програмного комплексу для використання, створення, редагування та моніторингу програм лояльності з використанням NFT. Для досягнення поставленої мети необхідно вивчити особливості роботи з криптовалютами технологіями, та створити веб-сервіс для реалізації необхідного функціоналу.

Висновки до розділу

В даному розділі було описано загальні положення, ми зробили опис предметної області. Було поставлено комплекс вимог по розділу. Визначили функціональні та нефункціональні вимоги до програмного забезпечення. Також були представлені існуючі аналоги нашої системи та створено матрицю трасування вимог.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідними даними до даної системи буде:

- електронна пошта та пароль користувача системи

					<i>IS93.110BAK.004 ПЗ</i>	Арк.
						21
Зм.	Лист	№ докум.	Підпис	Дата		

- інформація про клієнтів бізнеса
- інформація про продукти бізнеса
- картинка для NFT-купону
- налаштування для NFT-купону

2.2 Вихідні дані

Вихідними даними даної системи буде створений NFT-купон, який прив'язаний до продукту

2.3 Опис таблиць, які використовуються для зберігання даних

Таблиця 2.1 – Таблиця бази даних сутності “Client”

Колонка	Опис	Тип
id	ідентифікатор	varchar
email	емейл користувача	varchar
password	пароль користувача	varchar
phone	номер телефону користувача	varchar
createdAt	дата створення запису користувача	date

Таблиця 2.2 – Таблиця бази даних сутності “ProjectUsers”

Колонка	Опис	Тип
id	ідентифікатор	varchar
clientId	ідентифікатор, який вказує на таблицю Clients	varchar
name	Ім'я клієнта бізнесу	varchar
email	емейл клієнта бізнесу	varchar

phone	номер телефону клієнта бізнесу	varchar
createdAt	Дата створення запису клієнта бізнесу	date

Таблиця 2.3 – Таблиця бази даних сутності “ProjectProducts”

Колонка	Опис	Тип
id	ідентифікатор	varchar
clientId	ідентифікатор, який вказує на таблицю Clients	varchar
title	назва продукту	varchar
description	опис продукту	varchar
image	посилання на картинку продукту	varchar
createdAt	Дата створення запису в таблицю ProjectProducts	date

Таблиця 2.4 – Таблиця бази даних сутності “Nfts”

Колонка	Опис	Тип
id	ідентифікатор	varchar
clientId	ідентифікатор, який вказує на таблицю Clients	varchar
title	назва NFT-купона	varchar
description	опис NFT-купонаFT	varchar
discountPercentage	сума знижки, який дає NFT-купон	decimal
minted	поле, яке показує чи	boolean

	був NFT-купон отриманий клієнтом	
blockchainLink	посилання на NFT-купон в блокчейні	varchar
coolDown	поле, яке вказує як часто NFT-купон може використовуватись	decimal
contractId	поле, яке вказує на назву контракту, за допомогою якого NFT-купон було створено в блокчейні	varchar
createAt	Дата створення запису в таблицю	date

Таблиця 2.5 – Таблиця бази даних сутності “BlockchainAddresses”

Колонка	Опис	Тип
address	ідентифікатор	varchar
clientId	ідентифікатор, який вказує на таблицю Clients	varchar
privateKey	Приватний ключ адреси, для використання	varchar
addressType	Тип адреси “smart-contract” або “user-address”	varchar
userId	Ідентифікатор, який вказує на таблицю ProjectUsers	Varchar or null
createdAt	Дата створення запису в таблицю BlockchainAddress	date

Висновки до розділу

В даному розділі було описано вхідні і вихідні дані до дипломного проєкту.
Було описано за допомогою таблиць всі сутності представлені в базі даних проєкту

3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Моделювання та аналіз програмного забезпечення

Під час створення програми буди використано такі засоби:

– Node.js;

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		25

- HTML, CSS;
- середовище розробки Visual Studio Code;
- база даних PostgreSQL;
- блокчейн NEAR PROTOCOL

3.1.1 Серверна сторона

Node.js [3] є потужним інструментом для використання на стороні серверу для створення веб-сервісів з кількох причин:

- Висока продуктивність: Node.js базується на швидкому та ефективному двигуні V8, розробленому Google [13] для виконання JavaScript. Це дозволяє Node.js виконувати операції швидше, забезпечуючи високу продуктивність для програм.

- Однопоточність та подієва модель: Node.js використовує однопоточну та подієву модель програмування. Це означає, що він може обробляти багато запитів одночасно без блокування процесу. Це сприяє високій масштабовності та забезпечує відмінну ефективність в операціях з введенням/виведенням, таких як мережеві взаємодії.

- Широке сприйняття та екосистема пакетів: Node.js має велику спільноту розробників, що призводить до багатофункціональної екосистеми пакетів npm (Node Package Manager). npm містить тисячі готових до використання пакетів, які полегшують розробку програм шляхом надання готових модулів та функціональності.

- Всебічність: Node.js може використовуватися для створення різноманітних типів програм, включаючи веб-сервери, мережеві додатки, мікросервіси, чат-боти та багато іншого. Він може бути використаний як основний рушій для повноцінного стеку веб-розробки, включаючи фронтенд та бекенд.

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
						26
Зм.	Лист	№ докум.	Підпис	Дата		

– Простота розробки: JavaScript є популярною мовою програмування, і використання Node.js дозволяє розробникам використовувати її як на фронтенді, так і на бекенді.

3.1.2 Використання бази даних

Як базу даних для даного проєкту було обрано PostgreSQL [5]. Використання бази даних PostgreSQL має безліч переваг:

– Надійність та стабільність: PostgreSQL є однією з найбільш надійних та стабільних реляційних баз даних. Вона володіє механізмами відновлення після збоїв, механізмами транзакцій та високою стійкістю до помилок, що робить її ідеальним вибором для критичних застосунків.

– Розширені можливості: PostgreSQL підтримує широкий набір розширених функцій, включаючи вирази, функції, операції з геоданими, повнотекстовий пошук, масштабування горизонтально та вертикально, резервне копіювання, реплікацію та багато іншого. Вона дозволяє розробникам ефективно працювати з різноманітними типами даних та виконувати складні операції.

– Висока продуктивність: PostgreSQL пропонує оптимізацію запитів, кешування та індексацію, що дозволяє досягти високої продуктивності бази даних. Вона ефективно оптимізує запити для швидкого виконання та може обробляти великі обсяги даних.

– Розширюваність та гнучкість: PostgreSQL дозволяє розширювати можливості бази даних шляхом створення власних типів даних, функцій та розширень. Вона також підтримує мови програмування, такі як SQL, Python, JavaScript і багато інших, що дає можливість розробникам використовувати їх у своїх застосунках.

– Відкрите джерело та активна спільнота: PostgreSQL є базою даних з відкритим вихідним кодом, що означає наявність великої спільноти розробників, які активно працюють над її розвитком та підтримкою. Це забезпечує швидку виправлення помилок, оновлення та підтримку нових функцій.

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
						27
Зм.	Лист	№ докум.	Підпис	Дата		

В цілому, PostgreSQL є потужною та надійною базою даних, яка надає багато можливостей для розробки складних та вимогливих застосунків.

3.1.3 Середовище розробки

Середовищем розробки в нашому випадку є Visual Studio Code. Ми обрали це середовище завдяки таким перевагам:

- безкоштовний;
- має зручний та зрозумілий інтерфейс;
- редактор підтримує багато мов програмування;
- VS Code підтримує роботу з системою керування версіями git;
- в VS Code присутня можливість налагодження коду.
- Ми можемо виділити певну частину нашого кода програми і дивитися за виконанням команд покроково. так набагато простіше виявити помилки;
- результат налагодження виводиться на вбудовану консоль;
- підтримує загрузку та подальшу установку різноманітних додатків;
- розширення присутні для більшості мов програмування.

3.1.4 Блокчейн середовище

Блокчейн середовищем в нашому випадку було обрано Near Protocol

Near Protocol є платформою блокчейн та криптовалютою, яка має декілька переваг:

- Масштабованість: Near Protocol розроблений з орієнтацією на масштабованість, забезпечуючи високу швидкість та продуктивність мережі. Вона використовує консенсусний алгоритм Proof of Stake (PoS) та механізми шарування, що дозволяє обробляти тисячі транзакцій за секунду.
- Низькі витрати на транзакції: Near Protocol пропонує низькі комісії за

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
						28
Зм.	Лист	№ докум.	Підпис	Дата		

транзакції порівняно з іншими блокчейн платформами. Це робить його привабливим вибором для мікроплатежів та інших ситуацій, де важлива економічна ефективність. Широкі можливості розробки: Near Protocol надає розробникам широкі можливості для створення розумних контрактів та децентралізованих додатків (DApps). Вона підтримує мову програмування Rust, яка відома своєю швидкістю та безпекою. Також доступні SDK та інструменти для спрощення розробки на платформі.

– Екосистема та спільнота: Near Protocol має активну та зростаючу спільноту розробників та користувачів. Це створює благоприятне середовище для обміну ідеями, співпраці та розвитку проєктів. Також вона має підтримку від великих інвестиційних фондів та партнерів, що сприяє її прийняттю та росту.

– Складність і безпека: Near Protocol намагається зробити розробку та використання блокчейн-технологій більш простими та доступними. Вона надає інструменти для зручного управління гаманцями, доступом до додатків та безпекою активів. Крім того, Near Protocol працює над забезпеченням високого рівня безпеки мережі та захисту від шкідливих атак.

Загалом, Near Protocol пропонує масштабовану та швидку платформу блокчейн з низькими витратами на транзакції, що відкриває широкі можливості для розробки децентралізованих додатків та розумних контрактів

3.2 Архітектура програмного забезпечення

Програмне забезпечення реалізовано за допомогою SPA [8] архітектури на стороні клієнта і за допомогою MVC архітектури на серверній стороні. Архітектура проєкту складається з серверної частини та клієнтської частини. Серверна частина (т. зв. back-end) написана мовою JavaScript [11] на бібліотеці Node.js. Клієнтська частина (т. зв. front-end) розроблена за допомогою фреймворка React.js. Відповідає за обробку даних, які надходять з сервера. Дані, якими обмінюється клієнт з сервером надсилаються в форматі json.

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
						29
Зм.	Лист	№ докум.	Підпис	Дата		

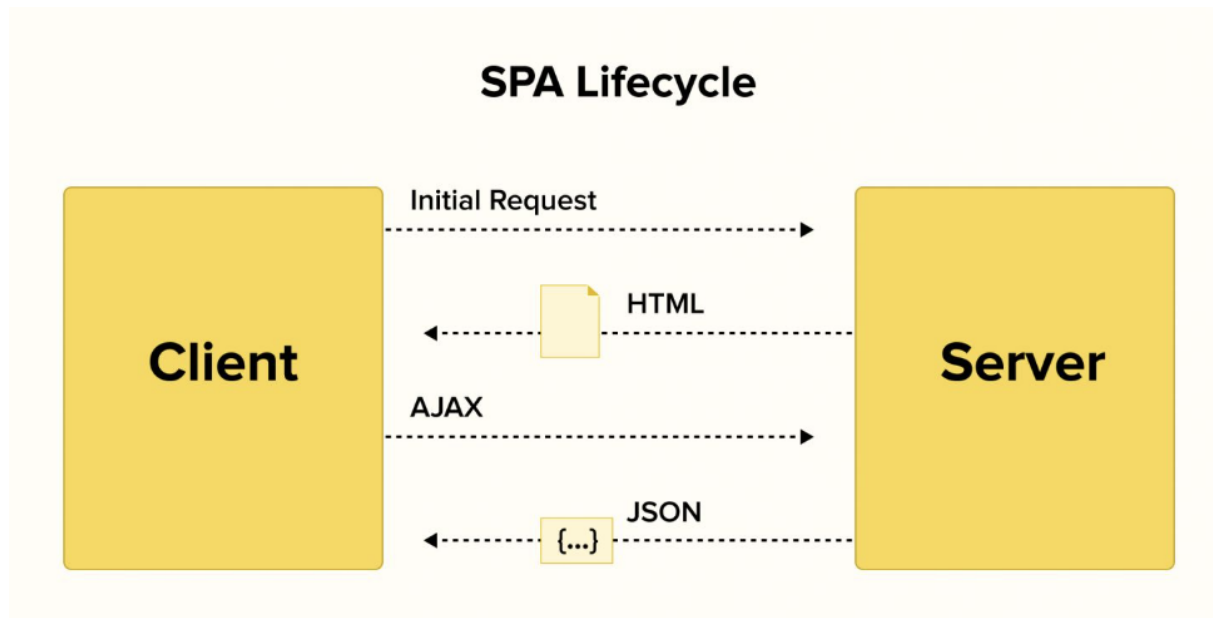


Рисунок 3.1 – Компоненти SPA-архітектури

SPA (Single-Page Application) - це тип архітектури проектування веб-додатків, де весь контент завантажується один раз під час початкового завантаження сторінки, і подальша навігація та взаємодія з додатком відбувається без повторного завантаження сторінки.

Основні переваги архітектури SPA:

- Зручний користувацький інтерфейс: SPA надає більш інтерактивний та зручний користувацький інтерфейс, оскільки взаємодія з додатком відбувається без перезавантаження сторінки. Всі взаємодії відбуваються швидко та плавно, що поліпшує загальний досвід користувача.

- Швидкість та продуктивність: Оскільки сторінка завантажується лише один раз, SPA дозволяє зменшити навантаження на сервер та мережу, що призводить до швидшого відгуку та завантаження додатку. SPA також використовує асинхронне завантаження даних, що дозволяє оновлювати лише необхідну частину сторінки замість повного перезавантаження.

- Багаторазове використання коду: Код, написаний для SPA, може бути багаторазовим, оскільки він використовується на всіх сторінках додатку. Це полегшує розробку, тестування та підтримку проекту, а також сприяє швидкому впровадженню змін та додаванню нового функціоналу.

– Менша потреба в серверних ресурсах: Оскільки SPA завантажує лише один раз, використовуючи API для отримання даних, вона вимагає менше серверних ресурсів порівняно з традиційними багатосторінковими додатками. Це може знизити витрати на хостинг та підтримку серверної інфраструктури.

– Підтримка мобільних платформ: SPA добре підходить для розробки мобільних додатків, оскільки вона може бути запущена у веб-браузері мобільного пристрою без необхідності встановлення додатка. Це спрощує розробку та розгортання на різних платформах.

Незважаючи на ці переваги, SPA також має деякі недоліки, такі як більшу складність в розробці та більший обсяг першого завантаження сторінки.

Також, через використання JavaScript для взаємодії з додатком, підтримка JavaScript-обов'язкова для користувачів.

MVC (Model-View-Controller) - це архітектурний патерн, який широко використовується для розробки серверної сторони веб-застосунків. Основні компоненти цього патерну - це модель, представлення та контролер. Ось п'ять переваг, які надає MVC порівняно з іншими архітектурами:

– Розділення обов'язків: MVC дозволяє чітко визначити ролі та обов'язки кожного компонента. Модель відповідає за обробку бізнес-логіки та доступ до даних, представлення відповідає за відображення даних користувачу, а контролер керує потоком даних між ними. Це розділення сприяє покращенню читабельності коду та зручності його обслуговування.

– Повторне використання коду: Завдяки розділенню обов'язків, різні компоненти можуть бути повторно використані. Модель може залишатися незмінною, незалежно від змін у представленні або контролері, що спрощує розширення та модифікацію системи.

– Гнучкість та масштабованість: Завдяки розділенню на компоненти, ви можете легко замінювати або змінювати один компонент без впливу на решту системи. Це дозволяє легко масштабувати або змінювати функціонал системи відповідно до потреб.

					<i>IS93.110BAK.004 ПЗ</i>	Арк.
						31
Зм.	Лист	№ докум.	Підпис	Дата		

– Тестованість: Кожен компонент - модель, представлення та контролер - може бути тестований окремо. Це полегшує написання автоматичних тестів для перевірки правильності роботи окремих частин системи та забезпечує більшу надійність та якість коду.

– Можливість паралельної роботи: Завдяки розділенню обов'язків та незалежності компонентів, розробники можуть працювати над різними частинами системи паралельно, що сприяє підвищенню продуктивності та швидкості розробки.

Ці переваги роблять MVC привабливим вибором для серверної сторони веб-застосунків, забезпечуючи легкість розробки, підтримку кодування та покращення розширюваності та підтримки.

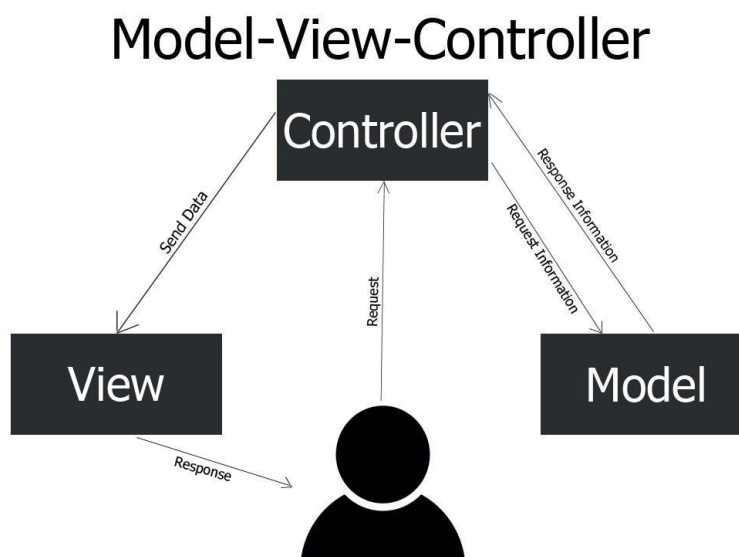


Рисунок 3.2 – Діаграма використання MVC-архітектури

3.3 Конструювання програмного забезпечення

3.3.1 Модулі програмного забезпечення

Програмне забезпечення було розроблене з використанням наступних технологій, мов програмування, фреймворків та бібліотек:

- Nest.js
- Sequelize.js
- React.js

– near-api-js

Nest.js - це прогресивний фреймворк для розробки серверних застосунків на мові JavaScript/TypeScript. Він побудований на основі Node.js та використовує концепції та засади архітектури Angular, що робить його потужним та гнучким інструментом для створення розширюваних та масштабованих серверних додатків. [4]
Основні переваги Nest.js:

– Модульність: Nest.js пропонує модульну структуру розробки, що дозволяє організувати ваш додаток на незалежні функціональні модулі. Це полегшує розподіл роботи між розробниками та сприяє повторному використанню коду.

– Використання TypeScript: Nest.js повністю підтримує TypeScript, що дозволяє вам використовувати переваги сильно типізованої мови програмування. TypeScript пропонує багатий набір інструментів для розробки, таких як статична типізація, автодоповнення коду та зручну рефакторізацію.

– Архітектурний шаблон MVC [6]: Nest.js використовує архітектурний шаблон Model-View-Controller (MVC) для організації додатку. Це дозволяє чітко розділити логіку бізнес-логіки, обробки запитів та представлення даних.

– Вбудована підтримка HTTP і WebSocket: Nest.js надає потужний механізм для обробки HTTP-запитів з вбудованою підтримкою маршрутизації, middleware, валідації даних та багато іншого. Крім того, він також підтримує WebSocket, що дозволяє створювати багатофункціональні додатки в реальному часі.

Узагальнюючи, Nest.js є потужним фреймворком, який поєднує переваги TypeScript, архітектурного шаблону MVC та модульної структури. Він дозволяє швидко розробляти масштабовані та добре організовані серверні додатки з підтримкою багатьох функціональних можливостей.

React.js - це фреймворк для створення frontend.[7]

React.js є потужним інструментом для створення клієнтських застосунків з кількох причин:

– Компонентна архітектура: React.js використовує компонентну модель

					<i>IC93.110БАК.004 ПЗ</i>	Арк.
						33
Зм.	Лист	№ докум.	Підпис	Дата		

розробки, що дозволяє розбити користувацький інтерфейс на невеликі, самодостатні та повторно використовувані компоненти. Це полегшує розробку, тестування та підтримку застосунків, спрощує організацію коду та забезпечує більшу переви-користовуваність.

– Віртуальний DOM: React.js використовує віртуальний DOM, що дозволяє ефективно оновлювати та відображати лише змінені частини інтерфейсу. Це покращує продуктивність застосунку та забезпечує швидку відповідь на дії користу-вача.

– Одностороннє зв'язування даних: React.js використовує одностороннє зв'язування даних, що означає, що зміни в даних автоматично відображаються в інте-рфейсі. Це спрощує синхронізацію даних та стану компонентів, покращує керованість та прогнозованість застосунку.

– Розширюваність: React.js дозволяє використовувати різні додаткові бібліо-теки та інструменти для розширення можливостей розробки. Наприклад, Redux для керування станом застосунку, React Router для навігації, або Material-UI для вико-ристання готових компонентів.

– Активна спільнота та екосистема: React.js має велику та активну спільноту розробників, що призводить до широкої екосистеми з багатьма розширеннями, компонентами та інструментами. Це забезпечує доступ до різноманітних ресурсів, документації та підтримки для розробки.

Ці переваги роблять React.js популярним вибором для створення клієнтських застосунків, незалежно від їх складності та розмірів.

Sequelize.js - це об'єктно-реляційний мапер (ORM) для Node.js, який дозволяє зручно та ефективно взаємодіяти з реляційними базами даних, такими як MySQL, PostgreSQL, SQLite та інші. [9]

Він надає високорівневий інтерфейс для створення, запиту та зміни даних у базі даних, спрощуючи розробку серверних додатків.

Основні переваги Sequelize.js:

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
						34
Зм.	Лист	№ докум.	Підпис	Дата		

– Підтримка багатьох СУБД: Sequelize.js підтримує широкий спектр реляційних баз даних, включаючи MySQL, PostgreSQL, SQLite, MSSQL та Oracle. Це дозволяє розробникам використовувати улюблену базу даних та переключатися між ними з легкістю, не залежно від специфіки проекту.

– Моделювання даних: Sequelize.js надає можливість визначати моделі даних з використанням JavaScript-об'єктів, які відповідають таблицям у базі даних. Це спрощує роботу з даними, оскільки моделі можна використовувати для створення, читання, оновлення та видалення записів.

– Міграції бази даних: Sequelize.js дозволяє використовувати міграції для керування схемою бази даних. Це дозволяє легко змінювати структуру бази даних та розгорнути оновлення без необхідності вручну втручатися в базу даних.

– Запити до бази даних: Sequelize.js надає потужну систему запитів, що дозволяє легко створювати складні SQL-запити до бази даних. Він підтримує вибірку, фільтрацію, сортування, агрегацію та інші операції над даними.

Загалом, Sequelize.js є потужним ORM-фреймворком, який спрощує взаємодію з реляційними базами даних у Node.js додатках. Він надає багатий набір інструментів для моделювання даних, запитів до бази даних, міграцій та багато іншого, що полегшує розробку та підтримку бази даних у вашому проекті.

near-api-js - це офіційний JavaScript-фреймворк для розробки децентралізованих додатків (DApps) на платформі Near Protocol [1]. Він надає потужні інструменти та бібліотеки для взаємодії з блокчейном Near, дозволяючи розробникам створювати розумні контракти, виконувати транзакції та працювати зі станом блокчейну. [14]

Основні переваги фреймворку near-api-js:

– Простота використання: near-api-js надає простий та зрозумілий API, який дозволяє легко взаємодіяти з блокчейном Near. Він має документацію, приклади коду та ресурси, що допомагають розробникам швидко освоїти його та почати розробку DApps.

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		35

– Повний функціонал блокчейну Near: фреймворк надає доступ до всіх можливостей блокчейну Near, включаючи виконання транзакцій, читання та запис даних у блокчейн, створення та виконання розумних контрактів. Він також підтримує мультипідписи, аутентифікацію та інші функції блокчейну Near.

– Розширені можливості: near-api-js дозволяє використовувати розширені можливості платформи Near, такі як робота з NEP-141 токенами, розумний стягнення комісій, управління ключами, експорт/імпорт гаманця та багато іншого. Він надає гнучкість та контроль над взаємодією з блокчейном Near.

– Розробка DApps у середовищі JavaScript: фреймворк побудований на мові JavaScript, що дозволяє розробникам використовувати знайомі інструменти та бібліотеки для розробки DApps. Він підтримується у середовищах Node.js та веб-браузерах, що дає можливість розробляти DApps на будь-якій платформі.

Взагалі, фреймворк near-api-js є потужним інструментом для розробки децентралізованих додатків на платформі Near Protocol. Він надає простоту використання, повний функціонал блокчейну Near, розширені можливості та підтримку спільноти, що робить його відмінним вибором для розробки DApps.

3.3.2 Модулі програмного забезпечення

Модулі програмного забезпечення на стороні backend були виконані у вигляді окремих менших модулів, кожен з яких відповідає за свою частину бізнес логіки:

– Auth module - модуль, який відповідає за авторизацію користувача за допомогою google auth сервісу та збереження інформації про користувача, при запиті на контролери цього модулю сервіс валідує дані, які приходять з клієнтської сторони веб-застосунку і зберігає їх в базу даних в таблицю “Clients”.

– Clients module - модуль, який відповідає за контроль клієнтами користувача, їхнім редагуванням та видаленням. При запиті на контролери цього модулю сервіс валідує дані, які приходять з клієнтської сторони веб-застосунку, обробляє їх та вносить зміни в таблицю “Clients”.

					<i>IS93.110BAK.004 ПЗ</i>	Арк.
						36
Зм.	Лист	№ докум.	Підпис	Дата		

– Products module - модуль, який відповідає за контроль продуктами користувача, їхнім редагуванням та видаленням в таблиці “ProjectProducts”. В модулі реалізований імпорт інформації про продукти з csv-файлу та збереження її в базу даних

– Users module - модуль, який відповідає за контроль клієнтами користувача, їхнім редагуванням та видаленням в таблиці “ProjectUsers”. В модулі реалізований імпорт інформації про клієнтів користувача з csv-файлу та збереження її в базу даних

– Nfts module - модуль, який відповідає за створення NFT-купонів, прив'язкою їх до продуктів, редагування налаштувань NFT-купонів, на їх видалення. Для цього в модулі використовується таблиця “Nfts”. При створенні та редагуванні налаштувань NFT-купону також використовується Blockchain module.

– Blockchain module - модуль, який відповідає створення для користувача smart-contract в блокчейні NEAR Protocol за допомогою бібліотеки near-api-js, при реєстрації користувача на сервісі. Також модуль відповідає за створення крипто-адрес в блокчейні NEAR Protocol для клієнтів користувача, що дає йому правом називатись мозком серверної сторони веб-застосунку, для всього цього Blockchain module використовує таблицю “BlockchainAddresses”

– ImageUploader module – модуль, який відповідає за завантаження будь-яких картинок в системі, для продуктів або для NFT-купонів, а також збереження їх на сторонньому сервісі Amazon S3 [15]

На стороні frontend модулі програмного забезпечення були реалізовані у вигляді html-сторінок у форматі jsx, які генеруються у html за допомогою фреймворку react.js:

– Головна сторінка – головна сторінка клієнтської сторони, на якій можна побачити статистику по використанню NFT-купонів, доходи користувача, які йому приносять NFT-купони та витрати користувача в системі

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
						37
Зм.	Лист	№ докум.	Підпис	Дата		

– Сторінка продуктів – сторінка на якій можна побачити створені або імпортовані користувачем продукти. На цій сторінці продукти можна створювати, імпортувати за допомогою csv-файлів, редагувати та видаляти

– Сторінка редагування та створення продуктів – сторінка на якій можна ввести потрібні дані створення користувачем продукту бізнесу. На цій сторінці клієнтів бізнесу можна вписати ім'я, опис, вставити картинку продукту, який редагується або створюється

– Сторінка клієнтів – сторінка на якій можна побачити створені або імпортовані користувачем клієнти бізнесу. На цій сторінці клієнтів бізнесу можна створювати, імпортувати за допомогою csv-файлів, редагувати та видаляти.

– Сторінка редагування та створення клієнтів – сторінка на якій можна ввести потрібні дані створення користувачем клієнту бізнесу. На цій сторінці клієнтів бізнесу можна вписати ім'я, пошту та мобільний телефон клієнта, який редагується або створюється. Ця сторінка важлива для контролю над клієнтом бізнесу

– Сторінка NFT-купонів - сторінка на якій можна побачити створені користувачем NFT-купони. На цій сторінці NFT-купони можна створювати, редагувати, видаляти, прив'язати їх до імпортованих або створених продуктів, за допомогою інтерфейсу та відправити NFT-купон до клієнта бізнесу, створивши йому крипто-адресу у блокчейні NEAR Protocol

3.4 Діаграма послідовності

Діаграма послідовності - використовується для покрокового зображення дій системи від початку до кінця. Її можна вважати більш абстрактною формою блок-схеми системи.

Діаграму послідовності для платформи для створення програм лояльності з використанням NFT на рис. 3.3.

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
						38
Зм.	Лист	№ докум.	Підпис	Дата		

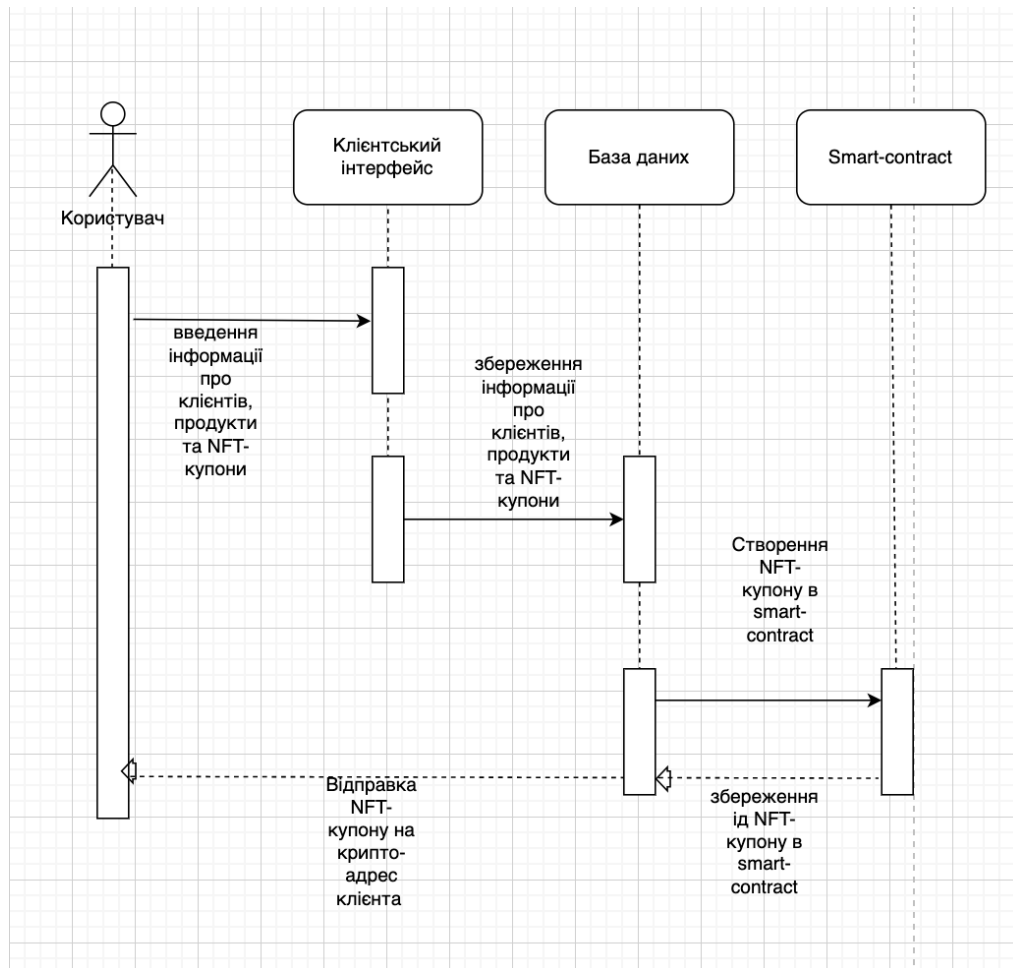


Рисунок 3.3 – Діаграма послідовності

Діаграма показує послідовність дій системи для введення даних, від користувача зі сторінки, до системи у форматі необхідних даних для збереження у базі даних та подальше надання цих даних для обробки в smart-contract [10].

3.5 Діаграма потоків даних

На даній діаграмі потоків ми можемо побачити, як дані оброблюються платформою для створення програм лояльності з використанням NFT, як вони надаються на вхід та на вихід, разом із тим, як саме вони зберігаються. Діаграму потоків даних для платформи для створення програм лояльності з використанням NFT надано на рисунку 3.5.

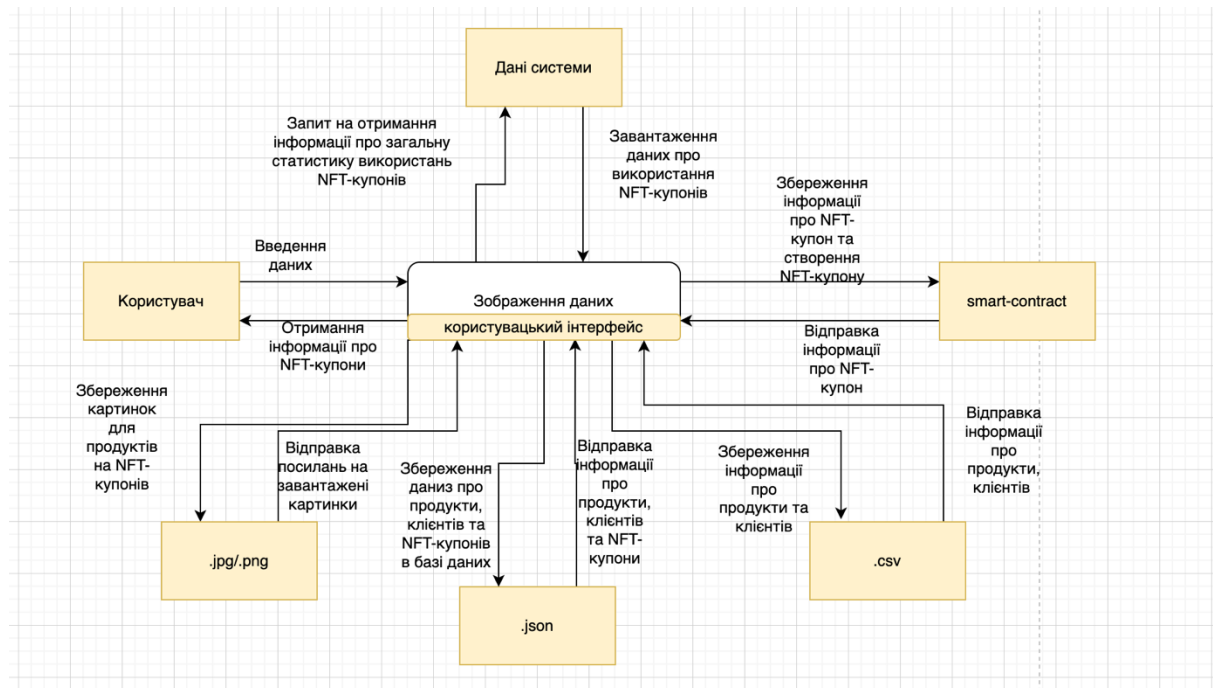


Рисунок 3.4 – Діаграма потоку даних

У цій створеній діаграмі розглянуто передачу даних між різними етапами роботи системи. Для кожної дії системи було зображено взаємозв'язок між користувачем, який може вводити дані, функціями системи, які обробляють отриману інформацію, та роботу з різними типами файлів для отримання і обробки інформації в платформі.

Висновки до розділу

В даному розділі було описано моделювання та аналіз програмного застосування, досліджено архітектуру програмного застосування, а саме SPA -на клієнтській стороні та MVC – на серверній стороні. Описано конструювання та моделювання програмного забезпечення до дипломного проекту.

4 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

4.1 Керівництво користувача

Для початку роботи з сервісом користувачеві потрібно перейти на головну сторінку, де йому потрібно зареєструватись за допомогою сервісу Google Auth. Якщо у користувача немає аккаунту google, то йому потрібно створити такий аккаунт ввівши бажану email адресу та номер телефону. Після чого користувач буде мати власний google аккаунт і зможе зареєструватись в сервісі

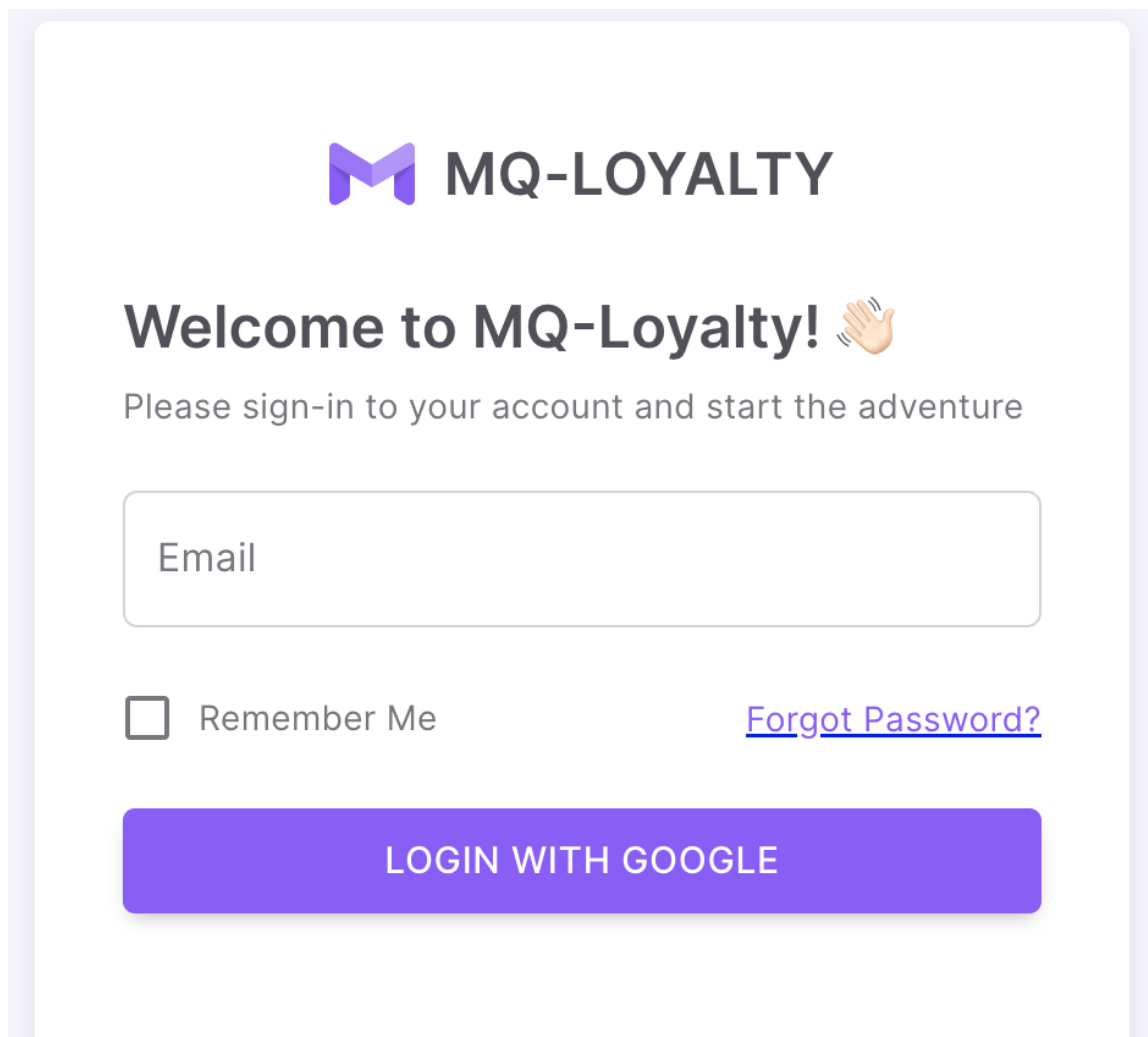


Рисунок 4.1 – Компонент клієнтської сторони для авторизації користувача через сервіс google auth

Після введення користувачем своєї електронної пошти, в браузері

					<i>IS93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		41

користувача відкриється вікно для авторизації через гугл, яке надасть список доступних аккаунтів для використання в платформі. Якщо в користувача не буде таких аккаунтів, можна буде створити новий.

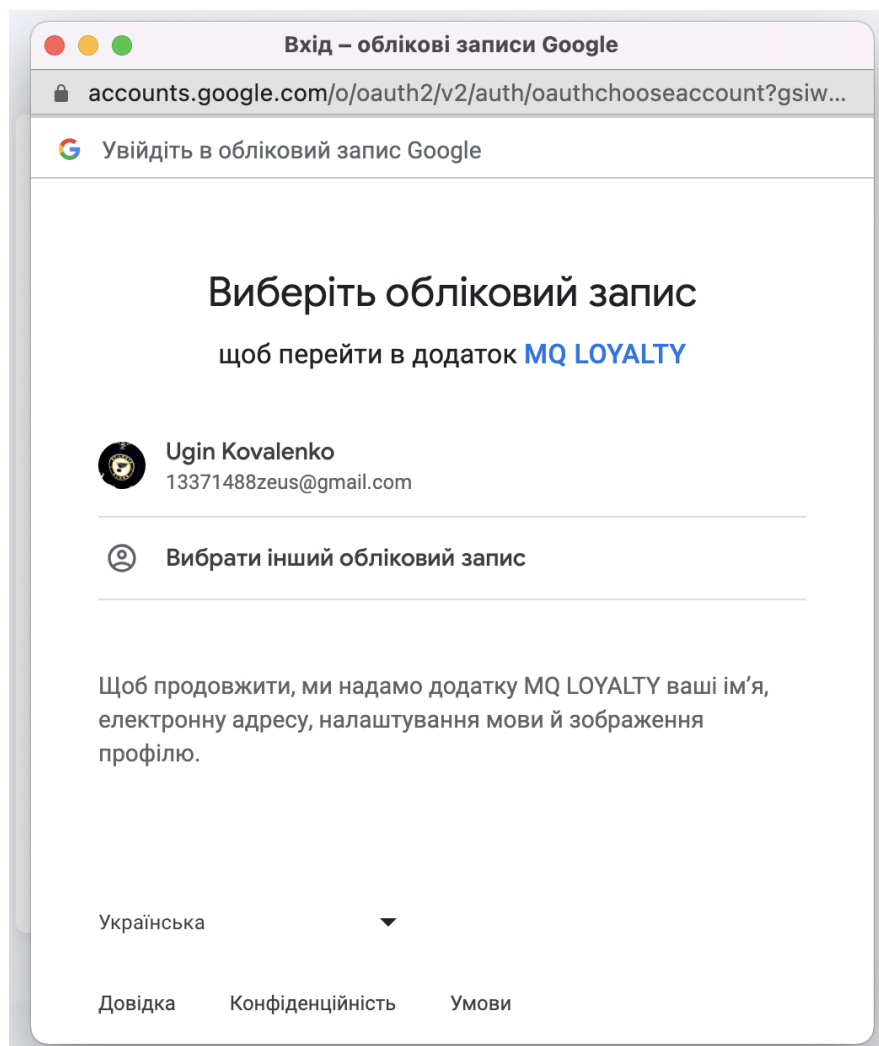


Рисунок 4.2 – Вікно для вибору google аккаунту користувача через сервіс google auth

Вибравши google account користувач опиняється на головній сторінці сервісу, де він може побачити статистику використань NFT-купонів, статистику колекції NFT-купонів, статистику по прибуткам та витратами сервісі. Така звітність дає змогу для швидкої роботи з сервісом, орієнтування в роботі сервісу та прийняття важливих рішень в продовженні, покращенні та зміні векторів програми лояльності.

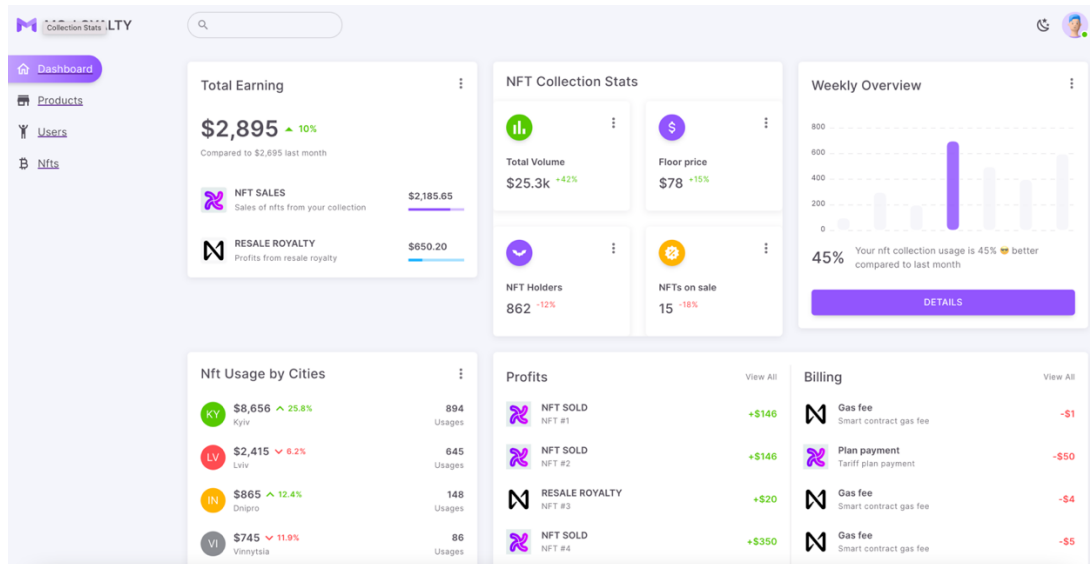


Рисунок 4.3 – Головна сторінка сервісу

Перейдемо до головних функцій сервісу. На сторінці всіх продуктів користувач може створити продукт за допомогою кнопки “Create product”, або імпортувати продукти за допомогою кнопки “Upload new products with csv file”. Також користувач може переглянути всі створені раніше продукти, редагувати їх за допомогою кнопки “Edit” та видаляти за допомогою кнопки “Delete”.

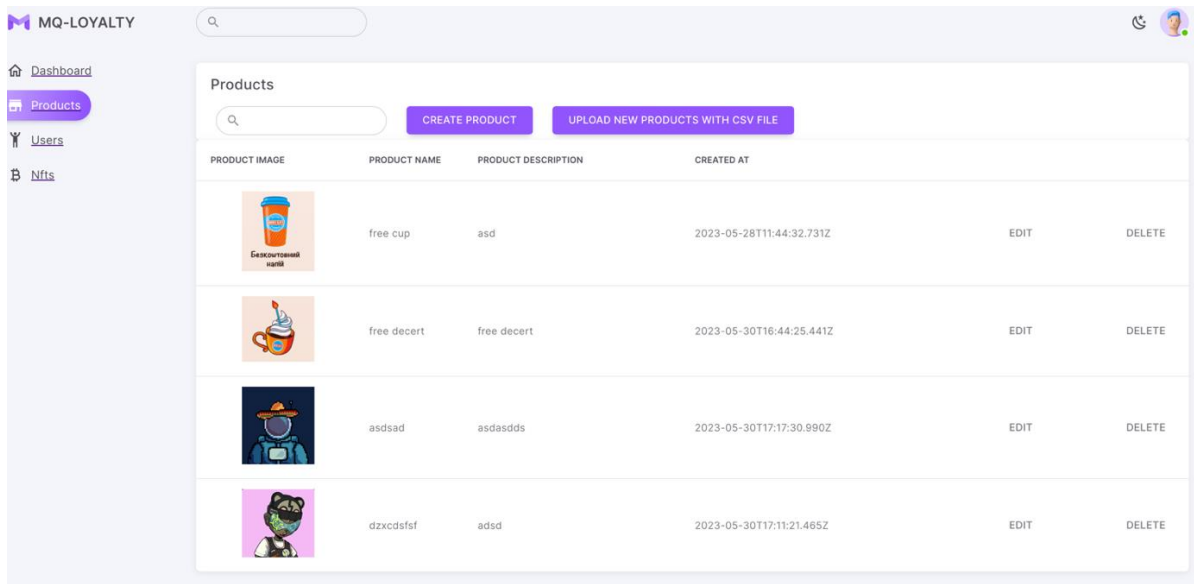


Рисунок 4.4 – Сторінка продуктів

Для створення продукту користувачеві потрібно завантажити фото продукту,

вписати назву, опис продукту та натиснути кнопку “Save Changes”. Аналогічно для редагування продукту

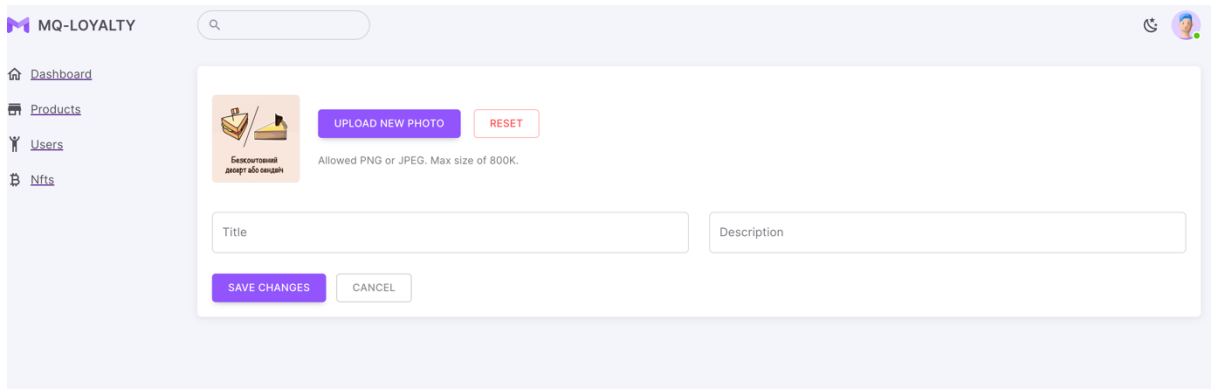


Рисунок 4.5 – Сторінка створення та редагування продуктів

На сторінці всіх клієнтів користувач може створити клієнта за допомогою кнопки “Create User”, або імпортувати клієнтів за допомогою кнопки “Upload new users with csv file”. Також користувач може переглянути всі створених раніше клієнтів, редагувати їх за допомогою кнопки “Edit” та видаляти за допомогою кнопки “Delete”.

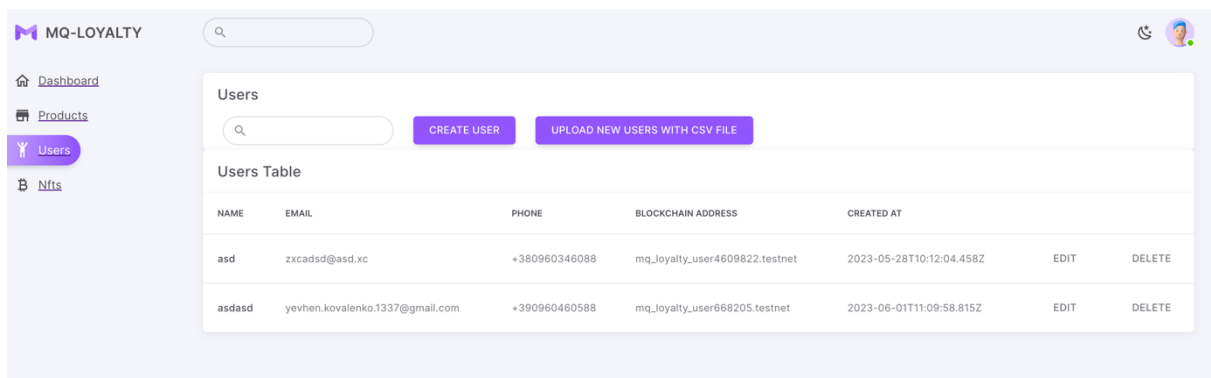


Рисунок 4.6 – Сторінка перегляду клієнтів

Для створення клієнту користувачеві потрібно вписати ім’я, номер телефону на електронну пошту клієнта та натиснути кнопку “Save” .

Аналогічно до редагування продукта ми можемо редагувати і клієнта, для цього нам потрібно ввести ім’я електронну пошту та телефон клієнта.

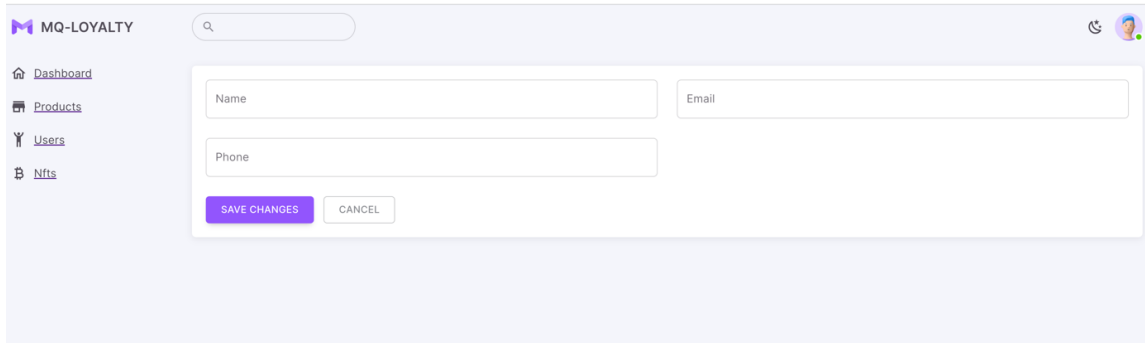


Рисунок 4.7 – Сторінка створення та редагування клієнтів

На сторіці “Nfts” користувач може створити NFT-купон за допомогою кнопки “Create NFT”. Також користувач може переглянути всі створені раніше NFT-купони, переглянути їх налаштування, продукт до якого привязаний до NFT-купону та номер телефону клієнта якому був відправлений NFT-купон, якщо NFT-купон не був відправлений клієнту, у рядку таблиці такого NFT-купону з’явиться кнопка “MINT FOR USER”, яка дасть можливість відправити клієнту NFT-купон, попередньо створивши для нього крипто-адресу в блокчейні NEAR Protocol. Також редагувати їх за допомогою кнопки “Edit” та видаляти за допомогою кнопки “Delete”.

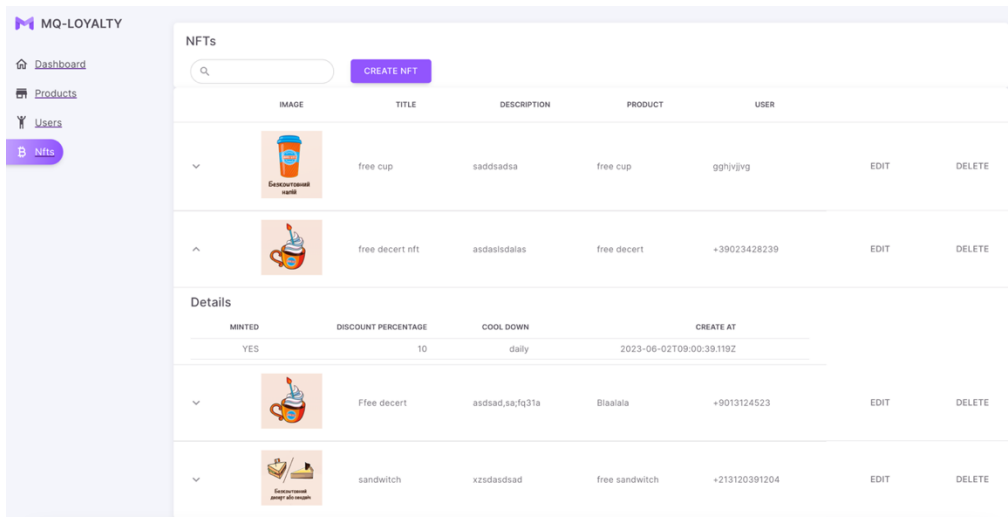


Рисунок 4.8 – Сторінка перегляду NFT-купонів

Для створення NFT-купону користувачеві потрібно вписати назву купону, опис купон, вибрати відсоток знижки на продукт, який можна вибрати за допомогою кнопки “Link to Product” та вибрати частоту використання купону, наразі наявні наступні опції:

- Використання купону раз в день
- Використання купону раз в три дні
- Використання купону раз в тиждень
- Використання купону раз в два тижні
- Використання купону раз в місяць
- Використання купону раз на 3 місяці
- Використання купону раз на пів-року
- Використання купону раз на 10 місяців
- Використання купону раз в рік

Після цього натиснути кнопку “Save Changes”. Аналогічно всі ці поля потрібно вибрати для редагування NFT-купону. Також для редагування купону може бути вибраний новий продукт, який може бути прив’язаним до купону. Також для редагування доступна картинка NFT-купону, що дозволить змінити картинку, якщо буде потрібно.

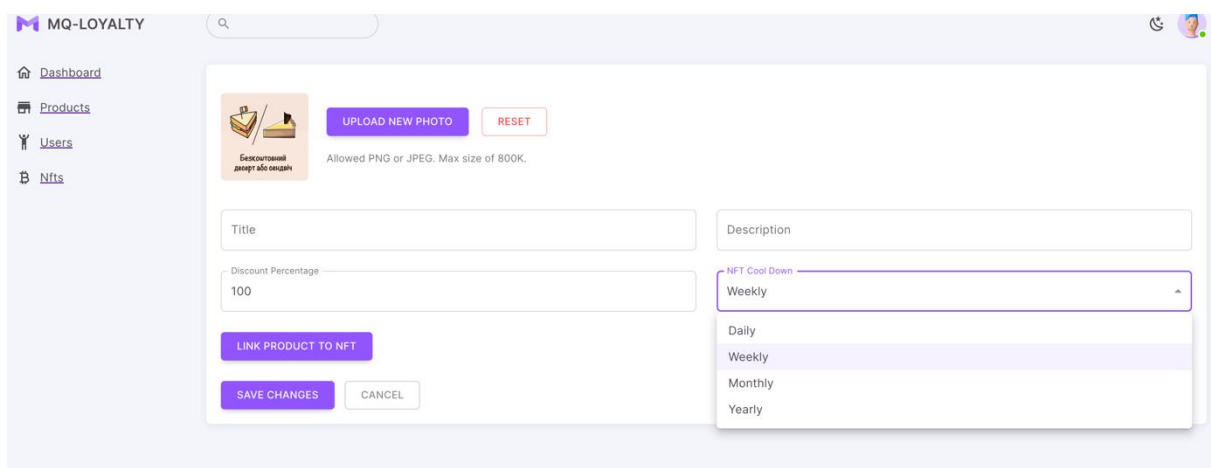


Рисунок 4.9 – Сторінка створення та редагування NFT-купонів

Під час прив'язки продукту до NFT-купону у користувача відкриється наступне вікно, яке надасть доступні продукти до вибору

Select Product

CLOSE

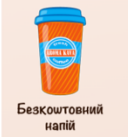

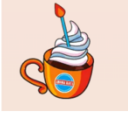

PRODUCT IMAGE	PRODUCT NAME	PRODUCT DESCRIPTION	
 Безкоштовний напій	free cup	ghgh	SELECT
 Безкоштовний десерт	free decert	blab blal	SELECT
 Безкоштовний десерт або овендч	Blaalala	zcxzxcxz	SELECT
 Безкоштовний десерт або овендч	free sandwich	balsdlsd	SELECT


Рисунок 4.10 – Вікно вибору продукту для прив'язки до NFT-купону

Після того як продукт до прив'язки був обраним, на сторінці створення купону з'явиться блок прив'язаного продукту.

MQ-LOYALTY

⌵


- Dashboard
- Products
- Users
- Nfts



UPLOAD NEW PHOTO
USE SAME IMAGE AS PRODUCT
RESET

Allowed PNG or JPEG. Max size of 800K.

NFT Cool Down
 Weekly

PRODUCT IMAGE	PRODUCT NAME	PRODUCT DESCRIPTION	
 Безкоштовний десерт або овендч	free sandwich	balsdlsd	SELECT ANOTHER PRODUCT

SAVE CHANGES
CANCEL

Рисунок 4.11 – Вікно вибору продукту для прив'язки до NFT-купону

4.2 Випробовування програмного продукту

4.2.1 Мета випробувань

Метою випробувань програмного продукту є перевірка функціоналу платформи для створення програми лояльності за допомогою використання NFT для визначення коректності роботи сервісу.

4.2.2 Загальні положення

У процесі тестування був перевірений увесь функціонал розробленої платформи для створення програми лояльності за допомогою використання NFT мануальним чином, та продемонстровано усе у вигляді таблиці.

4.2.3 Результати випробувань

Усі варіанти використання були протестовані вручну, а результатом перевірки стали таблиці, що будуть описані нижче:

- авторизація в системі (таблиця 4.1);
- вихід із системи (таблиця 4.2);
- перегляд клієнтів у вигляді таблиці (таблиця 4.3)
- створення клієнта (таблиця 4.4);
- редагування клієнта (таблиця 4.5);
- видалення клієнта (таблиця 4.6);
- імпорт клієнта за допомогою csv-файлу (таблиця 4.7);
- перегляд продуктів у вигляді таблиці (таблиця 4.8)
- створення продукту (таблиця 4.9);
- редагування продукту (таблиця 4.10);
- видалення продукту (таблиця 4.11);
- імпорт продуктів за допомогою csv-файлу (таблиця 4.12);

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		48

- перегляд NFT-купонів у вигляді таблиці (таблиця 4.13);
- створення NFT-купону (таблиця 4.14);
- редагування NFT-купону (таблиця 4.15);
- видалення NFT-купону (таблиця 4.16);
- прив’язка NFT-купону до раніше створеного продукту (таблиця 4.17);
- відправка NFT-купону до раніше створеного клієнту (таблиця 4.18);
- створення крипто-адреси в блокчейні NEAR PROTOCOL для створеного раніше клієнта (таблиця 4.19);

В таблиці 4.1 наведене тестування авторизації користувача

Таблиця 4.1 – Тестування працездатності функціоналу авторизації користувача

Назва процесу	Тестування працездатності функціоналу авторизації користувача
Тест	При на кнопку “Login with google” повинне відкритись вікно вибору google аккаунту
Початковий стан системи	Відкрита сторінку логіну в систему
Дія	Натискаємо на кнопку “Login with google”
Очікуваний результат	Буде відкрите вікно вибору google аккаунту
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.2 наведене тестування виходу користувача з систему

Таблиця 4.2 – Тестування працездатності функціоналу виходу користувача з систему

Назва процесу	Тестування працездатності функціоналу виходу користувача з систему
Тест	При кнопки “logout” користувач повинно переслати на сторінку авторизації в систему і не давати можливості повернутись назад
Початковий стан системи	Відкрита головна сторінка платформи
Дія	Натискаємо на кнопку “logout” вгорі в профілі користувача
Очікуваний результат	Користувача повинно викинуто на сторінку авторизації в систему
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.3 наведене тестування перегляду клієнтів у вигляді таблиці

Таблиця 4.3 – Тестування працездатності функціоналу в перегляду клієнтів у вигляді таблиці

Назва процесу	Тестування працездатності функціоналу в перегляду клієнтів у вигляді таблиці
Тест	При натисканні на сторінку клієнтів користувачу повинна показитись таблиця всіх клієнтів, які наявні в системі
Початковий стан системи	Відкрита головна сторінка платформи

Дія	Натискаємо на сторінку клієнтів у лівому меню
Очікуваний результат	Користувачу повинна показатись таблиця всіх клієнтів, які наявні в системі
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.4 наведене тестування створення клієнта

Таблиця 4.4 – Тестування працездатності функціоналу створення клієнта

Назва процесу	Тестування працездатності функціоналу створення клієнта
Тест	При натисканні на кнопку “CREATE USER” користувача повинно перекинути на сторінку створення клієнта, де він зможе створити користувача
Початковий стан системи	Відкрита сторінка клієнтів платформи
Дія	Натискаємо на кнопку “CREATE USER”, яка знаходиться зверху таблиці клієнтів
Очікуваний результат	Клієнта повинно перекинути на сторінку створення клієнта, де він зможе створити користувача
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.5 наведене тестування редагування клієнта

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		51

Таблиця 4.5 – Тестування працездатності функціоналу редагування клієнту

Назва процесу	Тестування працездатності функціоналу редагування клієнту
Тест	При натисканні на кнопку “EDIT” в рядку таблиці клієнтів, користувача повинно перекинуто на сторінку редагування клієнта
Початковий стан системи	Відкрита сторінка клієнтів платформи
Дія	Натискаємо на кнопку “EDIT” в рядку таблиці клієнтів
Очікуваний результат	Користувача повинно перекинуто на сторінку редагування клієнта, де він може внести правки до клієнта та зберегти оновлений результат
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.6 наведене тестування видалення клієнта

Таблиця 4.6 – Тестування працездатності функціоналу видалення клієнта

Назва процесу	Тестування працездатності функціоналу видалення клієнта
Тест	При натисканні на кнопку “DELETE” в рядку таблиці клієнтів, вибраний клієнт повинний бути видалений з системи, а сторінка повинна бути перезавантаженою

Початковий стан системи	Вікрита сторінка клієнтів платформи
Дія	Натискаємо на кнопку “DELETE” в рядку таблиці клієнтів
Очікуваний результат	Вибраний клієнт повинний бути видалений з системи, а сторінка повинна бути перезавантаженою
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.7 наведене тестування імпорту клієнтів за допомогою csv-файлу

Таблиця 4.7 – Тестування працездатності функціоналу імпорту клієнтів за допомогою csv-файлу

Назва процесу	Тестування працездатності функціоналу імпорту клієнтів за допомогою csv-файлу
Тест	При натисканні на кнопку “UPLOAD USERS WITH CSV FILE” повинно відкритись вікно для вибору csv-файлу, перезавантажена сторінка, де вже будуть імпортовані клієнти
Початковий стан системи	Відкрита сторінка клієнтів платформи
Дія	Натискаємо на на кнопку “UPLOAD USERS WITH CSV FILE”
Очікуваний результат	Повинно відкритись вікно для вибору csv-файлу, перезавантажена

Зм.	Лист	№ докум.	Підпис	Дата

	сторінка, де вже будуть імпортовані клієнти
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.8 наведене тестування перегляду продуктів у вигляді таблиці

Таблиця 4.8 – Тестування працездатності функціоналу в перегляду продуктів у вигляді таблиці

Назва процесу	Тестування працездатності функціоналу в перегляду продуктів у вигляді таблиці
Тест	При натисканні на сторінку продуктів користувачу повинна показатись таблиця всіх продуктів, які наявні в системі
Початковий стан системи	Відкрита головна сторінка платформи
Дія	Натискаємо на сторінку продуктів у лівому меню
Очікуваний результат	Користувачу повинна показатись таблиця всіх продуктів, які наявні в системі
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.9 наведене тестування створення продукту

Таблиця 4.9 – Тестування працездатності функціоналу створення продукту

Назва процесу	Тестування працездатності функціоналу створення продукту
Тест	При натисканні на кнопку “CREATE PRODUCT” користувача повинно перекинути на сторінку створення продукту, де він зможе створити продукт
Початковий стан системи	Відкрита сторінка продуктів платформи
Дія	Натискаємо на кнопку “CREATE PRODUCT”, яка знаходиться зверху таблиці продуктів
Очікуваний результат	Користувача повинно перекинути на сторінку створення продукту, де він зможе створити продукту
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.10 наведене тестування редагування продукту

Таблиця 4.10 – Тестування працездатності функціоналу редагування продукту

Назва процесу	Тестування працездатності функціоналу редагування продукту
Тест	При натисканні на кнопку “EDIT” в рядку таблиці продуктів, користувача повинно перекинуто на сторінку редагування продукту

Початковий стан системи	Відкрита сторінка продуктів платформи
Дія	Натискаємо на кнопку “EDIT” в рядку таблиці продуктів
Очікуваний результат	Користувача повинно перекинуто на сторінку редагування продукту, де він може внести правки до продукту та зберегти оновлений результат
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.11 наведене тестування видалення продукту

Таблиця 4.11 – Тестування працездатності функціоналу видалення продукту

Назва процесу	Тестування працездатності функціоналу видалення продукту
Тест	При натисканні на кнопку “DELETE” в рядку таблиці продуктів, вибраний продукт повинний бути видалений з системи, а сторінка повинна бути перезавантаженою
Початковий стан системи	Вікрита сторінка продуктів платформи
Дія	Натискаємо на кнопку “DELETE” в рядку таблиці продуктів

Очікуваний результат	Вибраний продукт повинний бути видалений з системи, а сторінка повинна бути перезавантаженою
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.12 наведене тестування імпорту продуктів за допомогою csv-файлу

Таблиця 4.12 – Тестування працездатності функціоналу імпорту продуктів за допомогою csv-файлу

Назва процесу	Тестування працездатності функціоналу імпорту продуктів за допомогою csv-файлу
Тест	При натисканні на кнопку “UPLOAD PRODUCTS WITH CSV FILE” повинно відкритись вікно для вибору csv-файлу, перезавантажена сторінка, де вже будуть імпортовані продукти
Початковий стан системи	Відкрита сторінка продуктів платформи
Дія	Натискаємо на на кнопку “UPLOAD PRODUCTS WITH CSV FILE”
Очікуваний результат	Повинно відкритись вікно для вибору csv-файлу, перезавантажена сторінка, де вже будуть імпортовані продукти

Зм.	Лист	№ докум.	Підпис	Дата

Фактичний результат співпадає з очікуваним	Так
--	-----

В таблиці 4.13 наведене тестування перегляду NFT-купонів у вигляді таблиці

Таблиця 4.13 – Тестування працездатності функціоналу в перегляду NFT-купонів у вигляді таблиці

Назва процесу	Тестування працездатності функціоналу в перегляду NFT-купонів у вигляді таблиці
Тест	При натисканні на сторінку NFT-купонів користувачу повинна показатись таблиця всіх NFT-купонів, які наявні в системі
Початковий стан системи	Відкрита головна сторінка платформи
Дія	Натискаємо на сторінку NFT-купонів у лівому меню
Очікуваний результат	Користувачу повинна показатись таблиця всіх NFT-купонів, які наявні в системі
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.14 наведене тестування створення NFT-купону

Таблиця 4.14 – Тестування працездатності функціоналу створення NFT-купону

Назва процесу	Тестування працездатності функціоналу створення NFT-купону
Тест	При натисканні на кнопку “CREATE NFT” користувача повинно перекинути на сторінку створення NFT-купону, де він зможе створити NFT-купон
Початковий стан системи	Відкрита сторінка клієнтів платформи
Дія	Натискаємо на кнопку “CREATE NFT”, яка знаходиться зверху таблиці NFT-купонів
Очікуваний результат	Користувача повинно перекинути на сторінку створення NFT-купону, де він зможе створити NFT-купон
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.15 наведене тестування редагування NFT-купону

Таблиця 4.15 – Тестування працездатності функціоналу редагування NFT-купону

Назва процесу	Тестування працездатності функціоналу редагування NFT-купону
Тест	При натисканні на кнопку “EDIT” в рядку таблиці NFT-купонів, користувача повинно перекинуто на сторінку редагування NFT-купону

Початковий стан системи	Відкрита сторінка NFT-купонів платформи
Дія	Натискаємо на кнопку “EDIT” в рядку таблиці NFT-купонів
Очікуваний результат	Користувача повинно перекинуто на сторінку редагування NFT-купонів, де він може внести правки до NFT-купону та зберегти оновлений результат
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.16 наведене тестування видалення NFT-купону

Таблиця 4.16 – Тестування працездатності функціоналу видалення NFT-купону

Назва процесу	Тестування працездатності функціоналу видалення NFT-купону
Тест	При натисканні на кнопку “DELETE” в рядку таблиці NFT-купонів, вибраний NFT-купон повинний бути видалений з системи, а сторінка повинна бути перезавантаженою
Початковий стан системи	Вікрита сторінка NFT-купонів платформи
Дія	Натискаємо на кнопку “DELETE” в рядку таблиці NFT-купонів

Очікуваний результат	Вибраний NFT-купон повинний бути видалений з системи, а сторінка повинна бути перезавантаженою
Фактичний результат співпадає з очікуваним	Так

В таблиці 4.17 наведено тестування прив'язки NFT-купону до раніше створеного продукту

Таблиця 4.17 – Тестування працездатності функціоналу прив'язки NFT-купону до раніше створеного продукту

Назва процесу	Тестування працездатності функціоналу прив'язки NFT-купону до раніше створеного продукту
Тест	При натисканні на кнопку “LINK WITH PRODUCT” повинно відкритись вікно для вибору наявних продуктів в системі, які користувач може вибрати для привязки до NFT-купону
Початковий стан системи	Відкрита сторінка створення NFT-купону платформи
Дія	Натискаємо на кнопку “LINK WITH PRODUCT”
Очікуваний результат	Повинно відкритись вікно для вибору наявних продуктів в системі, які користувач може вибрати для привязки до NFT-купону

Фактичний результат співпадає з очікуваним	Так
--	-----

В таблиці 4.18 наведене тестування відправки NFT-купону до раніше створеного клієнту

Таблиця 4.18 – Тестування працездатності функціоналу відправки NFT-купону до раніше створеного клієнту

Назва процесу	Тестування працездатності функціоналу відправки NFT-купону до раніше створеного клієнту
Тест	При натисканні на кнопку “MINT FOR USER” в рядку таблиці NFT-купонів, повинно відкритись вікно зі всіма клієнтами наявними в системі, користувач повинний мати змогу відправити NFT-купон клієнту, за попередньо створеним йому крипто-адресою у блокчейні NEAR PROTOCOL
Початковий стан системи	Вікрита сторінка NFT-купонів платформи
Дія	Натискаємо на кнопку “MINT FOR USER” в рядку таблиці NFT
Очікуваний результат	Повинно відкритись вікно зі всіма клієнтами наявними в системі, користувач повинний мати змогу відправити NFT-купон клієнту, за попередньо створеним йому крипто-

Зм.	Лист	№ докум.	Підпис	Дата

IC93.110BAK.004 ПЗ

Арк.

62

	адресою у блокчейні NEAR PROTOCOL
Фактичний результат співпадає з очікуваним	Так

Висновки до розділу

Отже, у даному розділі було виконано та проаналізовано тестування всіх функцій платформи для створення програм лояльності за допомогою NFT. Всі можливі тестові використання сервісу були показані у таблицях 4.1-4.18, у яких можна побачити опис тестування, сенс тесту, початковий стан системи до тестування, дії під час тесту та очікуваний результат тесту. Згідно до проведеного аналізу можна зробити висновок, що система працює коректно

ВИСНОВКИ

В даному дипломному проєкті виконано основну задачу – створено систему для створення програм лояльності з використанням NFT.

					<i>IS93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		63

Ця платформа вирішує проблеми бізнесу , як створення унікальної системи лояльності з використанням NFT, яка допоможе залучити, втримати та спонукати вірних клієнтів. Адже звичайні бонусні програми такі як бонусний рахунок, або паперові купони або промокоди вже не такі цікаві для звичайних покупців. В роботі використовувались такі архітектурні рішення як SPA та MVC.

Завдяки цим архітектурним рішенням ми отримали зручний в підтримці та використанні веб-сервіс. Система отримала зручний інтефейс, який спрощує взаємодію з програмою для користувача.

Для виконання поставленої задачі на серверній стороні сервісу було використано мову програмування JavaScript, бібліотеку Node.js, базу даних Postgres, фреймворк Nest.js та Typescript. На клієнтській стороні сервісу було використано мову програмування JavaScript, фреймворк Next.js [12], який написаний на основі react.js, typescript, HTML та CSS.

Як середовище розробки було обрано Visual Studio Code. Для створення редагування та передавання NFT-купонів був використаний блокчейн NEAR protocol, який є децентралізованим та масштабованим, що чудово підходить під проекти такого типу.

Для зручності використання блокчейну NEAR protocol був використаний sdk, який написаний на мові програмування Rust та спеціальний sdk для JavaScript near-api-js.

При розробці системи було підготовлено різноманітні діаграми, використання яких спростило побудову програмного забезпечення. Такі як схема бази даних, діаграма використання сервісу, діаграма класів серверної частини та схема архітектури сервісу.

Діаграми демонструють, як взаємодіють компоненти між собою. Після розробки було проведено випробування системи через різноманітні тести – завдяки цьому ми визначили, що наша програма відповідає функціональним вимогам та технічному завданню та працює задовільно.

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		64

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. NEAR Protocol. NEAR Documentation <https://near.org/>
2. NFT for loyalty program. <https://algobitz.com/nft-loyalty-program/>
3. Node.js documentation. <https://nodejs.org/en/>

					<i>IC93.110БАК.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		65

4. Nest.js - a progressive Node.js framework. <https://nestjs.com/>
5. PostgreSQL -The World's Most Advanced Open Source Relational Database .
<https://www.postgresql.org/>
6. Model-View-Controller documentation.
<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
7. Next.js documentation. <https://nextjs.org/docs>
8. Single page application documentation. https://en.wikipedia.org/wiki/Single-page_application
9. Sequelize is a modern TypeScript and Node.js ORM for Oracle, Postgres, MySQL, MariaDB, SQLite and SQL Server, and more. <https://sequelize.org/>
10. HOW TO CREATE AND DEPLOY SMART CONTRACTS ON THE NEAR PROTOCOL. <https://www.leewayhertz.com/create-smart-contracts-on-near-protocol/>
11. Basic Javascript. <https://developer.mozilla.org/ru/docs/Web/JavaScript>
12. Easy Next.js frontend. <https://codevoweb.com/build-a-simple-frontend-app-in-nextjs13-app-directory/>
13. How to Implement Login with Google in Nest JS.
<https://dev.to/imichaelwolabi/how-to-implement-login-with-google-in-nest-js-2a0a>
14. Near-api-js documentation. <https://www.npmjs.com/package/near-api-js>
15. S3 with digital ocean and nest js. <https://wanago.io/2020/08/03/api-nestjs-uploading-public-files-to-amazon-s3>

					<i>IC93.110BAK.004 ПЗ</i>	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		66