

**Пояснювальна записка
до курсової роботи**

на тему: Інтелектуальний інтернет-магазин

КПІ.ІП-1522.045440.01.81

Київ – 2024

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	5
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
1.1 Загальні положення	6
1.2 Змістовний опис і аналіз предметної області	8
1.3 Аналіз існуючих технологій та успішних ІТ-проектів	9
1.3.1 Аналіз відомих алгоритмічних та технічних рішень	10
1.3.2 Аналіз допоміжних програмних засобів та засобів розробки	13
1.3.3 Аналіз відомих програмних продуктів	14
1.4 Аналіз вимог до програмного забезпечення	16
1.4.1 Розроблення функціональних вимог	21
1.4.2 Розроблення нефункціональних вимог	23
1.5 Постановка задачі	25
Висновки до розділу	26
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	27
2.1 Моделювання та аналіз програмного забезпечення	27
2.2 Архітектура програмного забезпечення	30
2.3 Конструювання програмного забезпечення	33
2.4 Аналіз безпеки даних	38
Висновки до розділу	38
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	40
3.1 Аналіз якості ПЗ	40
3.2 Опис процесів тестування	40
Висновки до розділу	45
4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	46
4.1 Розгортання програмного забезпечення	46
4.2 Підтримка програмного забезпечення	48
Висновки до розділу	49

ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CF	– Collaborative filtering – колаборативна фільтрація.
CBF	– Content-based filtering – фільтрація вмісту
IDE	– Integrated Development Environment – інтегроване середовище розробки.
ORM	– Object-Relational Mapping – об'єктно-реляційне відображення

ВСТУП

З появою та постійним розвитком інформаційних технологій, електронна комерція стає необхідною складовою сучасного бізнесу. Актуальність даної роботи обумовлена наростаючим попитом на інноваційні та ефективні інтернет-платформи, зокрема, інтелектуальних інтернет-магазинів. В даній пояснювальній записці розглядається процес розробки та реалізації інтелектуального інтернет-магазину SuperCoffee, спрямованого на оптимізацію процесів та покращення взаємодії з користувачами.

Світові тенденції підкреслюють важливість використання інтелектуальних рішень у сфері електронної комерції. Зараз більшість споживачів очікує персоналізованого підходу та інноваційних функцій при покупках онлайн. Розробка інтелектуального інтернет-магазину SuperCoffee відповідає світовим стандартам та вимогам сучасного електронного бізнесу.

Аналіз сучасного стану об'єкта розробки вказує на важливість створення продукту, який не лише задовольняє основні потреби споживачів у покупці кавових товарів, а й надає їм інноваційні можливості, такі як персоналізовані рекомендації та інтелектуальний аналіз вибору продуктів.

Робота також враховує практично розв'язані завдання провідними науковими установами та визнаними фахівцями у галузі розробки електронних торговельних платформ.

Можливі сфери застосування розробленого інтелектуального інтернет-магазину охоплюють не лише продаж кавових продуктів, але й встановлення нових стандартів у використанні інтелектуальних технологій у сфері електронної комерції.

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Предметна область даного дослідження охоплює розгляд сучасних тенденцій у електронній комерції, визначення ключових термінів та понять, а також огляд напрямків розробок у цій галузі.

Електронна комерція — це купівля та продаж товарів і послуг або передача коштів або даних через електронну мережу, переважно через Інтернет. Ці транзакції електронної комерції зазвичай діляться на чотири типи: бізнес-бізнес (B2B), бізнес-споживач (B2C), споживач-споживач або споживач-бізнес.

Терміни електронна комерція та електронний бізнес часто використовуються як синоніми. Термін e-tail також іноді використовується стосовно транзакційних процесів, які складають роздрібні покупки онлайн. За останні два десятиліття платформи електронної комерції, такі як Amazon і eBay, сприяли значному зростанню онлайн-торгівлі. Згідно з даними Бюро перепису населення США, у 2011 році електронна комерція становила 5% від загального обсягу роздрібних продажів. До 2 кварталу 2020 року, після початку пандемії COVID-19, електронна комерція становила 16,5% роздрібних продажів. Відтоді він дещо впав приблизно до 15%, оскільки фізичні магазини знову відкрилися.[1]

eCommerce Intelligence — це нова категорія програмного забезпечення, яке дозволяє мерчендайзерам і маркетологам максимізувати дохід від кожного сеансу відвідувача. Завдяки глибокому машинному навчанню EI автоматично персоналізує й оптимізує покупки на сайті для кожного сегмента відвідувачів. Результатом є покращене виявлення продукту, що, як відомо, підвищує дохід від відвідувача та рентабельність витрат на рекламу.

Сьогодні ваш клієнт вирішує робити покупки онлайн. А елітні команди електронної комерції знають, що не можна залишати коефіцієнт конверсії напризволяще, постійно приділяючи увагу кожній деталі покупки. Проблема в тому, що без найкращих технологій і інноваційних стратегій багато роздрібних торговців швидко відстають від очікувань відвідувачів.[2]

Результати щорічних досліджень Інтернет асоціації України Дослідницького холдингу Factum Group Ukraine на жовтень 2018 року відображають динаміку проникнення Інтернету в Україні. «Регулярні користувачі» заявляють, що використовують інтернет 1 раз на місяць і частіше. При цьому з аналізу статистики випливає, що поширення мобільних пристроїв – планшетів і смартфонів, серед населення України і покриття мобільними операторами практично всієї території України, призвело до того, що як по всій Україні, так і серед регулярних користувачів інтернету збільшився відсоток з низьким доходом при значному скороченні користувачів з середнім доходом та незначним скороченням користувачів з доходом вище середнього.

Чим більше розвинена економіка країни – тим більше стає обсяг ринку онлайн-торгівлі. Це простежується на прикладі країн Заходу, а в останні роки справедливо і для України. За даними сайту «Електронна комерція» в світовому масштабі ніша e-commerce за підсумками 2017 року показала вражаюче зростання +16%, при цьому досягнувши обсягу 1,5 трильйонів доларів. А ринок України в контексті зростання і перспектив є одним з найбільш привабливих. Але, на відміну від багатьох європейських країн, Уанету (визначають як частину Всесвітньої павутини, що належить до національного українського домену.ua) до рівня насичення ще дуже далеко.[3]

У період з 2014 по 2021 рік глобальні продажі електронної комерції зросли більш ніж на 269%. За даними Statista, до 2027 року світовий ринок електронної комерції становитиме майже 7,4 мільярда доларів США.

Ця цифра, швидше за все, зросла також через пандемію COVID - 19. Нещодавнє дослідження дослідницького центру Pew Research Center показало, що 90% громадян США сказали, що Інтернет був «важливим» для них під час пандемії.

У результаті індустрія кави також перейшла в Інтернет. Для виробників, трейдерів і обсмажувачів стає дедалі важливішим розвивати присутність в Інтернеті, щоб ефективно просувати та продавати свою каву через платформи електронної комерції. Немає сумніву, що за останні кілька десятиліть кавова

індустрія зросла в геометричній прогресії. Це включає ринок електронної комерції.

Відповідно до даних про продажі Amazon, кава була найпопулярнішою категорією продуктів харчування та напоїв у 2018 році. У цьому році лише продажі кави Amazon склали близько 140 мільйонів доларів США.

Враховуючи, що ці дані також були до пандемії, більш ніж імовірно, що з того часу зростання стрімко зросло.

Загалом це означає, що зараз, як ніколи, бренди кави повинні бути конкурентоспроможними під час продажу в Інтернеті. Споживачі кави все більше вимагають зручності, і електронна комерція є чудовим способом для компаній це запропонувати.

Для споживачів може бути важко знайти каву, яка б відповідала їхнім смаковим уподобанням, особливо якщо вони також шукають зерна з екологічно чистих та етичних джерел.

Платформи електронної комерції дозволяють споживачам купувати каву ефективніше, водночас забезпечуючи отримання всієї необхідної інформації.

Значна частина цього зосереджена на відстежуваності та прозорості, оскільки все більше споживачів прагнуть зрозуміти, як і де вирощується кава.

Ключовим є і той факт, що все більше людей купують товари з долоні. За оцінками Statista, до 2027 року майже 7,7 мільярдів людей у всьому світі матимуть смартфони, причому найбільша кількість користувачів у США, Китаї та Індії.

По суті, світ стає все більш цифровим. Продаж кави в Інтернеті тепер є необхідністю для будь-якого обсмажувача чи торговця, і це стає дедалі важливішим для деяких виробників.[4]

1.2 Змістовний опис і аналіз предметної області

У сучасному світі ІТ-технології є невід'ємною частиною багатьох галузей, включаючи торгівлю та роздрібну торгівлю. У зв'язку з цим, розробка

програмного забезпечення для інтелектуальних інтернет-магазинів стає актуальною задачею для підтримки та розвитку підприємств. У даному підрозділі розглядається процес використання знань предметної області у програмному забезпеченні на поточний момент розвитку ІТ-технологій.

На сьогоднішній день різні компанії в сфері роздрібної торгівлі кавою використовують різноманітні підходи до розвитку своїх інтернет-магазинів. Проте, часто спостерігаються недоліки у використанні знань предметної області. Наприклад, недостатня автоматизація управління запасами, неефективний аналіз покупців та їхніх уподобань, а також обмежений функціонал для взаємодії з клієнтами.

Зокрема, багато компаній не використовують повноцінно потенціал інтелектуальних технологій, таких як системи машинного навчання та аналіз даних. Це призводить до втрати можливостей для оптимізації пропозицій товарів, персоналізації обслуговування та підвищення лояльності клієнтів.

Можливі шляхи покращення ситуації включають в себе впровадження інтелектуальних алгоритмів для прогнозування попиту та рекомендаційних систем для клієнтів. Додатково, важливо розглядати питання кібербезпеки та захисту даних у зв'язку із зростанням кількості транзакцій та обробкою конфіденційної інформації.

У рамках курсової роботи обрано шлях імплементації інтелектуальних алгоритмів для аналізу покупців в інтернет-магазині кави SuperCoffee. Це дозволить надати персоналізовані пропозиції та обслуговування для кожного клієнта.

1.3 Аналіз існуючих технологій та успішних ІТ-проектів

У цьому підрозділі проведемо детальний аналіз алгоритмічного забезпечення та технічних рішень, що є відомими в сучасній області інтелектуальних інтернет-магазинів для кави. Зосередимося на технологіях, які можуть сприяти успішній реалізації нашого проекту, який має назву SuperCoffee.

1.3.1 Аналіз відомих алгоритмічних та технічних рішень

У контексті розвитку інтернет-магазину SuperCoffee, одним із ключових аспектів є реалізація рекомендаційних систем для забезпечення персоналізованого підходу до клієнтів. Існують різні підходи до розробки таких систем, серед них основні – Content-based filtering та Collaborative filtering.

1.3.1.1 Collaborative filtering

Колаборативна фільтрація є найбільш широко використовуваним підходом до проектування системи рекомендацій. Методи спільної фільтрації (CF) відіграють важливу роль у процесі рекомендацій, хоча колаборативна фільтрація часто використовується разом з іншими методами фільтрації, такими як фільтрація на основі вмісту, на основі знань. В основному методи спільної фільтрації засновані на зборі та дослідженні великої кількості інформації, яка ґрунтується на поведінці користувачів, діяльності чи вподобаннях і передбаченні смаків цього конкретного користувача, використовуючи їхню схожість з іншими користувачами. Він не залежить від повідомлень, що розкладаються машиною, і тому правильно рекомендує складені елементи, і через це є ключовою перевагою підходу до спільної фільтрації. У системі рекомендацій спільної фільтрації рекомендовані об'єкти вибираються на основі попередніх оцінок великої групи користувачів.

1.3.1.2 Content-based filtering

Фільтрування на основі вмісту (CBF) намагається рекомендувати елементи для активних користувачів на основі кількості подібностей, яку оцінює цей користувач позитивно в минулому. Наприклад, якщо користувачеві подобається веб-сторінка зі словами „мобільний“, „ручка“ та „RAM“, CBF рекомендуватиме сторінки, пов'язані з світ електроніки. Опис елемента та профіль орієнтації користувача відіграє важливу роль у фільтрації на основі вмісту. Алгоритми фільтрації на основі вмісту намагаються рекомендувати елементи на основі підрахунку подібності. Найкраще підходять предмети рекомендовано шляхом порівняння різних кандидатів із елементами, які раніше оцінив користувач.

Представлення $tf-idf$ є найбільш широко використовуваним алгоритмом (також називається представленням векторного простору). Для створення профілю користувача здебільшого система концентрується на двох типах інформації: 1. Модель переваг користувача. 2. Журнал взаємодія користувача із системою рекомендацій. По суті, профіль елемента використовується цими методами для (тобто набору різних розмірів і характеристик) кваліфікації елемента в системі. Створення контент-профілю користувачів виконується за допомогою зваженого вектору ознак предмета. Важливість кожної функції для користувач позначається вагами. Це можна розрахувати з індивідуально оціненими векторами вмісту з використанням різних навичок.

1.3.1.3 Hybrid filtering

Останні дослідження довели, що гібридний підхід у деяких випадках може бути більш ефективним. В основному підходи до колаборативної фільтрації та фільтрації на основі вмісту найбільш широко використовуються в програмах фільтрації інформації. Як ми знаємо, кожна монета має дві сторони, так само кожен підхід має свою винагороду та слабкі місця. Основним мотивом гібридного підходу є агрегування колаборативної фільтрації та фільтрації на основі вмісту для підвищення точності рекомендацій. Гібридні підходи можна реалізувати різними способами:

1. Індивідуально впроваджувати спільні методи та методи, що базуються на контенті, і агрегувати свої прогнози.
2. Інтегрувати деякі характеристики, що базуються на змісті, у спільний підхід.
3. Включити деякі характеристики співпраці в підхід, що базується на змісті.
4. Побудувати загальну консолідативну модель, яка об'єднує характеристики, що ґрунтуються на вмісті, і характеристики співпраці.

Холодний запуск і розрідженість є поширеними проблемами в системах рекомендацій, які вирішуються за допомогою цих методів. Хорошим прикладом гібридних систем рекомендацій є Netflix. Вони дають рекомендації, порівнюючи

звички спостереження та дослідження подібних експлуататорів (колаборативна фільтрація), а також надаючи фільми, які мають спільні риси з фільмами, які експлуататори високо оцінили (фільтрування на основі вмісту).[5]

1.3.1.4 Обране рішення

Порівнявши два основні підходи до рекомендаційних систем - Collaborative filtering та Content-based filtering, для інтернет-магазину SuperCoffee обрано Content-based filtering approach.

У магазині, який спеціалізується на продажу кави, характеристики продуктів є ключовим фактором визначення вибору користувача. Content-based filtering, який аналізує властивості товарів та рекомендує їх на основі схожості, ідеально підходить для такого контексту.

Content-based filtering дозволяє створити персоналізовані рекомендації, враховуючи конкретні вподобання та історію покупок кожного користувача. Це важливо для задоволення потреб клієнтів і створення індивідуального досвіду покупок.

Content-based filtering не потребує великої кількості даних про взаємодію між користувачами, тому це ефективний підхід для рекомендацій товарів новим користувачам, які ще не мали значного досвіду в магазині.

Collaborative filtering може стикатися з проблемами холодного старту та розрідженістю даних, особливо в нових магазинах. Content-based filtering в цьому випадку є менш залежним від подібності між користувачами.

Content-based filtering здатний надавати більш зрозумілі рекомендації, оскільки вони базуються на конкретних характеристиках товарів, що може бути важливим для споживачів.

Отже, враховуючи усі ці фактори, Content-based filtering є оптимальним вибором для реалізації рекомендаційної системи у SuperCoffee, де акцент робиться на персоналізованому та якісному обслуговуванні клієнтів.

1.3.2 Аналіз допоміжних програмних засобів та засобів розробки

У розробці системи рекомендацій для інтернет-магазину SuperCoffee, важливо вибрати підходящі інструменти та засоби розробки. Нижче подано аналіз допоміжних програмних засобів та рекомендації з їх використання.

У контексті розробки веб-додатків, вибір мови програмування природно падає на JavaScript. Використання JavaScript забезпечує високу ефективність та сумісність з потужними інструментами, спрямованими на створення інноваційних та високоякісних веб-застосунків. Для клієнтської сторони додатку була обрана бібліотека React, яка славиться своєю зручністю та ефективністю у розробці динамічних інтерфейсів. Для серверної частини обрано середовище виконання Node.js, яке забезпечує асинхронну обробку запитів та високу швидкість відгуку, що є критичним для забезпечення оптимального функціонування веб-додатку. Такий підхід дозволяє створити досконалий та ефективний веб-застосунок, що відповідає сучасним стандартам та вимогам ринку.[6]

React - це відома бібліотека для розробки інтерфейсів користувача. Вона дозволяє легко створювати ефективні та динамічні інтерфейси, що є важливим для забезпечення зручності користувача в інтернет-магазині. Зручність роботи з компонентами, швидкість та активна спільнота розробників роблять React популярним вибором для клієнтської частини додатків.[7]

Node.js є середовищем виконання JavaScript, що дозволяє розробникам використовувати JavaScript для серверної частини додатків. Його асинхронна природа і можливість швидко реагувати на запити роблять його ефективним вибором для реалізації серверної частини додатків, зокрема, для обробки запитів в інтернет-магазині.[8]

Бібліотека natural для Node.js є потужним інструментом для обробки текстових даних, таких як описи продуктів чи відгуки користувачів. Вона містить функціонал для токенізації, стемінгу, аналізу сентименту та інших завдань обробки природної мови. Застосування цієї бібліотеки допоможе

зрозуміти зміст текстових даних та використовувати цю інформацію для поліпшення системи рекомендацій.[9]

Visual Studio Code (VSCode) є популярною та потужною інтегрованою середовище розробки для JavaScript і Node.js. Вона надає розширення для React, автодоповнення коду, вбудовані інструменти для відлагодження, а також інші корисні функції. VSCode є легким, швидким та підтримується активною спільнотою розробників. [10]

- WebStorm: Інтегроване середовище для розробки JavaScript і Node.js. Має багатий функціонал, але може бути важчим для новачків та менш гнучким.[11]

- Atom: Легка та розширювана IDE, але може бути менш продуктивною в порівнянні з VSCode.[12]

Враховуючи зручність роботи з React і Node.js, а також потужність бібліотеки natural для обробки текстових даних, рекомендовано використовувати Visual Studio Code як інтегроване середовище розробки.

1.3.3 Аналіз відомих програмних продуктів

У сфері інтелектуальних інтернет-магазинів та систем рекомендацій розглянемо два визначальних інтернет-магазини кави[13] та світового гіганта в електронній комерції, Amazon.

1.3.3.1 Javy Coffee

Javy Coffee, один з найкращих інтернет-магазинів кави. Це неймовірно складний веб-сайт. У ньому використовуються дуже якісні фотографії продукту, навмисно вибрана палітра кольорів і деякі дуже вміло реалізовані анімації. На веб-сайті є ціла сторінка, присвячена публікації відгуків.

1.3.3.2 Hot and Cool Cafe

Hot and Cool Cafe вражає своєю унікальністю. Веб-сайт діє як канал для їхнього фізичного місцезнаходження, надаючи послуги самовивозу та доставки. Він відрізняється від типового веб-сайту електронної комерції, пропонуючи не

повноцінний інтернет-магазин, але акцентуючи увагу на особистому взаємодії з клієнтами.

1.3.3.3 Amazon

Amazon - глобальний лідер в електронній комерції. Цей інтернет-магазин пропонує безліч товарів, включаючи каву. Відмінності включають величезний каталог товарів, ефективну систему рекомендацій, аналіз покупців та високий рівень інтеграції з іншими сервісами Amazon.

Для здійснення порівняльного аналізу між курсовим проєктом "SuperCoffee" та аналогами, була розроблена таблиця 1.1, де порівнюються різні характеристики та функціонал.

Таблиця 1.1 – Порівняння з аналогом

Функціонал	Курсовий проєкт(назва)	Javy Coffee	Hot and Cool Cafe	Amazon	Пояснення
Персоналізована система рекомендацій	Так	Ні	Ні	Так	Наявність системи рекомендацій
Аналіз покупців	Так	Ні	Ні	Так	Можливість аналізу та вивчення поведінки покупців
Рейтингова система	Ні	Так	Ні	Так	Можливість оцінювати кожен товар
Список бестселерів	Ні	Так	Так	Так	Рекомендація продуктів, що більше всього продаються

Захист даних користувачів	Так	Так	Так	Так	Забезпечення конфіденційності та безпеки даних
---------------------------	-----	-----	-----	-----	--

Аналізуючи інтернет-магазини кави Javy Coffee, Hot and Cool Cafe та глобального гіганта в електронній комерції, Amazon, можна зазначити відмінності в підходах до систем рекомендацій та особливості їх функціоналу.

Javy Coffee вражає своєю різноманітністю та дбайливим ставленням до дизайну. Він пропонує персоналізовані рекомендації та систему відгуків, що створює унікальне споживацьке враження.

Hot and Cool Cafe, навпаки, використовує більш простий підхід, діючи як канал для фізичного магазину. Він не має системи персоналізованих рекомендацій, але акцентує увагу на особистому взаємодії з клієнтами та послугах самовивозу та доставки.

Amazon, як глобальний лідер, пропонує величезний каталог товарів та ефективну систему рекомендацій. Відмінність полягає в глобальному охопленні та величезному обсязі даних, що дозволяє створювати складні та персоналізовані рекомендації.

Узагальнюючи, можна встановити тенденцію: міжнародні корпорації, такі як Amazon, переважно розробляють складні персоналізовані системи рекомендацій, використовуючи обширні обсяги даних. З іншого боку, локальні компанії, наприклад, Hot and Cool Cafe, в основному використовують простіші системи, базуючись на популярних продуктах (бестселерах) та інколи рекомендаціях товарів, які клієнт вже придбав.

1.4 Аналіз вимог до програмного забезпечення

Веб-застосунок SuperCoffee спрямований на надання користувачам зручного та гнучкого інтерфейсу для ефективного використання інтернет-магазину. Основною метою розробки є забезпечення клієнтам можливості легко шукати товари, обирати їх, додавати до кошика та здійснювати замовлення

доставки. Програмне забезпечення SuperCoffee реалізує простий та інтуїтивний інтерфейс, що сприяє зручному користуванню.

Клієнти мають можливість взаємодії з інтернет-магазином як гість, дозволяючи їм шукати продукти, додавати їх у кошик та здійснювати оплату та доставку навіть без реєстрації. Однак реєстрація в системі SuperCoffee надає додаткові переваги користувачам, такі як можливість створення збереження персональних даних для аналізу та створення персональних рекомендацій. Основні функції програмного забезпечення SuperCoffee включають:

- Пошук та перегляд продуктів.
- Створення замовлення.
- Здійснення оплати та вибір методу доставки.
- Реєстрація та авторизація користувача для отримання персоналізованих функцій та рекомендацій.

Розглянемо діаграму варіантів використання для веб-застосунку SuperCoffee на рисунку 1.1.

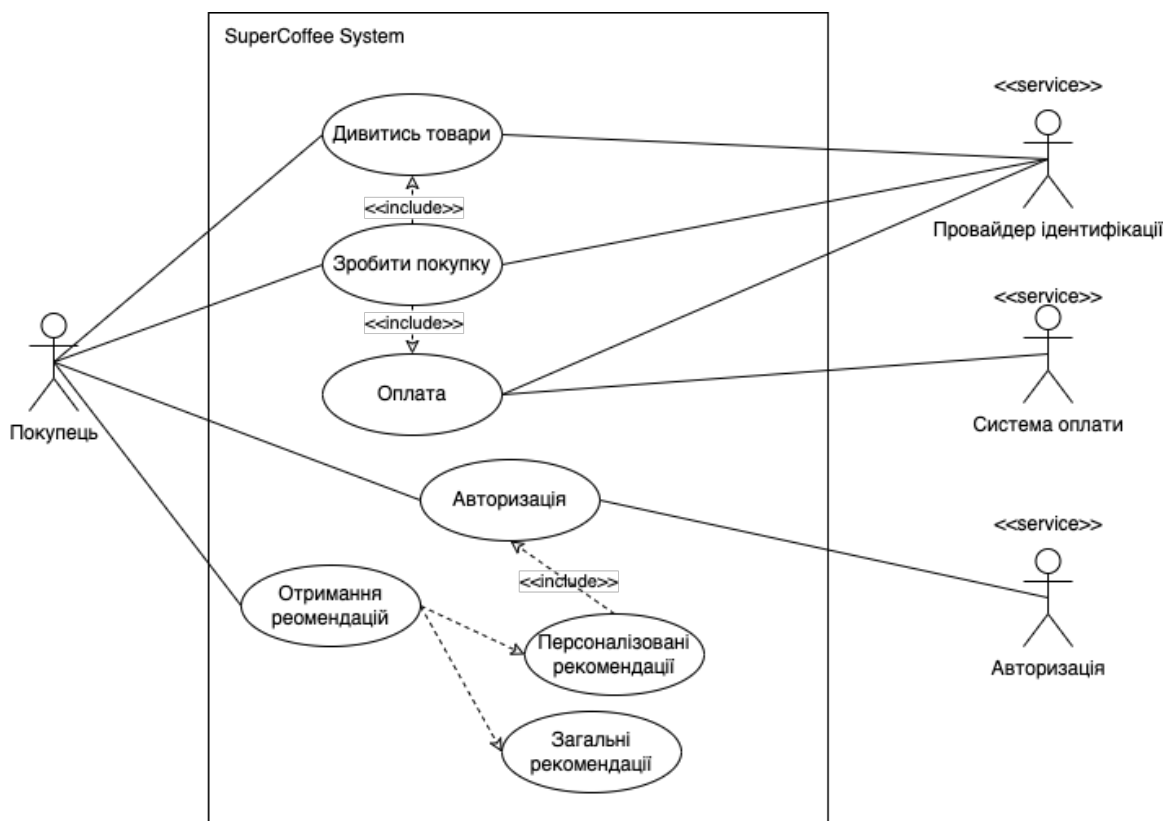


Рисунок 1.1 – Діаграма варіантів використання

В таблицях 1.2 - 1.6 наведені варіанти використання програмного забезпечення.

Таблиця 1.2 - Варіант використання UC-01

Use case name	Авторизація користувача
Use case ID	UC-01
Goals	Реєстрація/Авторизація нового користувача в системі
Actors	Гість/Зареєстрований користувач, Сервіс авторизації
Trigger	Користувач бажає зареєструватися/авторизуватися
Pre-conditions	-
Flow of Events	Користувач переходить на сторінку авторизації. В поля для реєстрації вводяться відповідні дані: пошта користувача, пароль в системі, та після заповнення даних користувача натискає кнопку реєстрації, або обирає авторизацію за допомогою системи гугл. Після цього користувач перенаправляється на сторінку входу.
Extension	Якщо якесь конкретне поле введено некоректно, то воно підсвічується червоним надписом.
Post-Condition	Авторизація користувача перехід на головну сторінку

Таблиця 1.3 - Варіант використання UC-02

Use case name	Перегляд товарів
Use case ID	UC-02
Goals	Користувач переглядає асортимент товарів
Actors	Користувач
Trigger	Користувач обрав опцій "Меню"
Pre-conditions	-
Flow of Events	Користувач обирає опцію " Меню " в головному меню. Система відображає доступний асортимент товарів. Користувач переглядає деталі товарів та їхні зображення. Користувач може застосовувати фільтри для сортування товарів за різними критеріями. Користувач обирає конкретний товар для отримання більш детальної інформації.

Extension	У випадку відсутності товарів за використаними фільтрами або загальною відсутністю товарів, сторінка буде пустою.
Post-Condition	Користувач має можливість переглядати та обирати товари з асортименту.

Таблиця 1.4 - Варіант використання UC-03

Use case name	Створення замовлення
Use case ID	UC-03
Goals	Здійснення покупки та створення замовлення на товари в інтернет-магазині.
Actors	Користувач, Провайдер ідентифікації
Trigger	Користувач обрав товари та готовий оформити покупку.
Pre-conditions	Користувач повинен мати товари у кошику.
Flow of Events	Користувач обирає товари та додає їх у корзину. Після вибору товарів, користувач переходить до корзини для перегляду та редагування замовлення. Користувач обирає адресу доставки та обирає спосіб оплати. Після введення необхідної інформації, користувач підтверджує та оформлює замовлення..
Extension	Якщо в корзині немає товарів, користувач не може здійснити замовлення. У випадку помилкового введення даних, користувач, некоректне поле підсвічується.
Post-Condition	Створення замовлення у системі та оновлення інформації про товари на складі.

Таблиця 1.5 - Варіант використання UC-04

Use case name	Оплата замовлення
Use case ID	UC-04
Goals	Оплатити доставку замовлення
Actors	Система оплати, Провайдер ідентифікації
Trigger	Натиснуто кнопку оплати та перехід на сервіс системи оплати
Pre-conditions	Перевірка замовлення з заповненими даними
Flow of Events	Користувач натискає кнопку "Оплатити". Система перевіряє дані замовлення та робить запит до провайдера ідентифікації. Користувач перенаправляється на сторінку платіжної системи,

	де вводить необхідні дані для оплати. Система оплати отримує підтвердження від провайдера ідентифікації та виконує оплату. Після успішної оплати, користувач отримує повідомлення про підтвердження оплати та підтвердження замовлення. Переходить на головну сторінку.
Extension	Якщо виникають труднощі при взаємодії з системою оплати, користувач отримує відповідне повідомлення та може спробувати інший спосіб оплати.
Post-Condition	Завершення процесу оплати та оновлення статусу замовлення у системі, очищення кошику.

Таблиця 1.6 - Варіант використання UC-05

Use case name	Отримання персоналізованих рекомендацій
Use case ID	UC-05
Goals	Забезпечити користувачеві персоналізовані рекомендації на основі його історії покупок та інтересів.
Actors	Авторизований користувач
Trigger	Користувач авторизований та перейшов на головну сторінку або на сторінку «Меню»
Pre-conditions	Користувач повинен бути авторизованим в системі.
Flow of Events	Користувач увійшов в систему та перейшов на головну сторінку або на сторінку «Меню». Система рекомендацій аналізує дії користувача. На основі аналізу система генерує персоналізовані рекомендації для користувача. Користувач переглядає та вибирає товари із запропонованих рекомендацій. Вибрані товари додаються до кошика або списку бажань користувача.
Extension	Якщо користувач не авторизований, він отримує загальні рекомендації, а його дії не зберігаються.
Post-Condition	Користувач отримує персоналізовані рекомендації, які пришвидшують пошук.

1.4.1 Розроблення функціональних вимог

Функціональні вимоги представляють собою конкретні функції або сервіси, які система чи програмне забезпечення повинні виконувати для вирішення конкретних завдань та відповіді на потреби користувачів. Ці вимоги детально описують очікувану функціональність та здатність системи виконувати певні операції. Формулюючись як конкретні задачі чи дії, які користувачі системи мають мати можливість виконати, функціональні вимоги визначають обсяг можливостей та функціональний спектр системи.

Нижче наведено опис функціональних вимог до програмного забезпечення у вигляді таблиць 1.7-1.14.

Таблиця 1.7 – Функціональна вимога FR-1

Назва	Авторизація користувача за допомогою електронної пошти
Опис	Система повинна надавати можливість авторизації користувачеві шляхом введення пошти, паролю.

Таблиця 1.8 – Функціональна вимога FR-2

Назва	Авторизація користувача за допомогою облікового запису Google
Опис	Система повинна надавати можливість авторизації користувачеві шляхом авторизації в системі Google.

Таблиця 1.9 – Функціональна вимога FR-3

Назва	Перегляд товару
Опис	Система повинна дозволяти користувачеві переходити на сторінку обраного товару.

Таблиця 1.10 – Функціональна вимога FR-4

Назва	Використання фільтрів
-------	-----------------------

Опис	Система повинна дозволяти користувачеві використовувати різноманітні фільтри для точного та зручного пошуку товарів, таких як фільтри за категорією, ціною, назвою.
------	---

Таблиця 1.11 – Функціональна вимога FR-5

Назва	Додавання товару у кошик
Опис	Система повинна забезпечувати можливість користувачеві додавати товари до кошика для подальшого оформлення замовлення.

Таблиця 1.12 – Функціональна вимога FR-6

Назва	Оформлення замовлення
Опис	Система повинна надавати можливість користувачеві оформляти замовлення після додавання товарів у кошик.

Таблиця 1.13 – Функціональна вимога FR-7

Назва	Оплата доставки
Опис	Система повинна надавати можливість користувачеві оплачувати послуги доставки при оформленні замовлення.

Таблиця 1.14 – Функціональна вимога FR-8

Назва	Створення рекомендація
Опис	Система повинна надавати можливість автоматичного створення персоналізованих рекомендацій для користувача на основі його попередніх виборів та покупок. Або загальні рекомендації для гостей.

Таблиця трасування вимог є інструментом системного аналізу та управління проектами, призначеним для моніторингу взаємозв'язків між різними елементами проекту. Головна мета цього інструменту полягає в установленні зв'язків між вихідними вимогами до системи та конкретними елементами, які реалізують ці вимоги.

У сфері розробки програмного забезпечення, таблиця трасування вимог зазвичай включає список вимог або функціональних можливостей, а також інші елементи, такі як сценарії використання, тестові випробування, компоненти системи та інші. Кожен елемент таблиці пов'язаний з конкретними вимогами, які він впроваджує чи тестує.

	FR-1	FR-2	FR-3	FR-4	FR-5	FR-6	FR-7	FR-8
UC-1	+	+						
UC-2			+	+	+			
UC-3						+		
UC-4							+	
UC-5								+

Рисунок 1.2 – Матриця трасування вимог

1.4.2 Розроблення нефункціональних вимог

Нефункціональні вимоги визначають характеристики системи чи програмного забезпечення, які не обмежуються конкретною функціональністю, але впливають на якісні аспекти їхньої роботи та характеристики. Ці вимоги націлені на визначення "якості" системи і охоплюють такі аспекти, як продуктивність, надійність, безпека та інші.

Нижче наведено опис нефункціональних вимог до програмного забезпечення у вигляді таблиць 1.15-1.22.

Таблиця 1.15 – Нефункціональна вимога NFR-1

Назва	Продуктивність
Опис	Система повинна забезпечувати швидке завантаження сторінок та ефективну роботу серверу під час навантаження.

Таблиця 1.16 – Нефункціональна вимога NFR-2

Назва	Надійність
Опис	Система повинна бути стійкою до можливих відмов та відновлювати роботу після перебоїв. Регулярне резервне копіювання даних обов'язкове.

Таблиця 1.17 – Нефункціональна вимога NFR-3

Назва	Безпека
Опис	Захист особистих даних користувачів.

Таблиця 1.18 – Нефункціональна вимога NFR-4

Назва	Сумісність
Опис	Сумісність з різними браузерами та мобільними пристроями.

Таблиця 1.19 – Нефункціональна вимога NFR-5

Назва	Інтерфейс
Опис	Розробка інтуїтивно зрозумілого та зручного інтерфейсу, а також адаптивного дизайну для різних розмірів екранів.

Таблиця 1.20 – Нефункціональна вимога NFR-6

Назва	Швидкість та ефективність
Опис	Оптимізація роботи алгоритмів та мінімізація часу обробки платіжних транзакцій.

Таблиця 1.21 – Нефункціональна вимога NFR-7

Назва	Доступність
Опис	Забезпечення доступності для людей з обмеженими можливостями шляхом дотримання стандартів доступності.

Таблиця 1.22 – Нефункціональна вимога NFR-8

Назва	Масштабованість
Опис	Забезпечення зручного масштабування системи в разі збільшення обсягів користувачів чи даних.

Вказані в таблиці вимоги сприяють врахуванню основних аспектів, що не лише забезпечують функціональну повноту, але й визначають якість програмного продукту.

1.5 Постановка задачі

Метою даної курсової роботи є розробка інтелектуального інтернет-магазину з продажу кави під назвою "SuperCoffee". Головною метою проєкту є створення платформи, що надасть користувачам можливість зручного та ефективного вибору, замовлення та отримання продуктів, пов'язаних із кавою.

Задачі проєкту включають наступне:

- Реалізація функціоналу для реєстрації та авторизації користувачів.
- Розробка інтерфейсу для пошуку та перегляду асортименту товарів, з можливістю застосування фільтрів для полегшення вибору.
- Створення кошика для збереження вибраних товарів та оформлення замовлення.
- Впровадження системи оплати та обробки замовлень.
- Розробка персоналізованої системи рекомендацій для збільшення задоволення від покупок та підтримки вірних клієнтів.
- Забезпечення безпеки та конфіденційності особистих даних користувачів.

Мета та завдання курсового проєкту спрямовані на створення функціонального та зручного інтернет-магазину, який відповідає потребам користувачів у сфері кавової продукції.

Висновки до розділу

У даному розділі проведено детальний аналіз предметної області і існуючих технологій для розробки інтелектуального інтернет-магазину "SuperCoffee". Розглянуті алгоритмічні та технічні рішення, допоміжні програмні засоби та програмні продукти, що використовуються у подібних проєктах.

Важливо відзначити, що при аналізі програмних продуктів виокремлено такі ключові аспекти, як персоналізовані системи рекомендацій, аналіз поведінки покупців, рейтингові системи, списки бестселерів та захист даних користувачів. Даний аналіз дозволив визначити основні функціональні та нефункціональні вимоги до розроблюваного програмного забезпечення.

В розділі також сформульовано завдання та мету розробки інтелектуального інтернет-магазину "SuperCoffee". Зокрема, визначено основні функції системи, такі як реєстрація та авторизація користувачів, пошук та перегляд товарів, оформлення замовлень, система оплати та реалізація персоналізованих систем рекомендацій.

Отже, результати аналізу визначають основні напрямки для подальшого проєктування та розробки програмного забезпечення "SuperCoffee".

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У сучасному інформаційному суспільстві створення програмного забезпечення є важливою ланкою для вирішення складних завдань та досягнення стратегічних цілей у різних галузях. Процес розробки програмного забезпечення визнається як складний та багатозадачний, вимагаючи глибокого розуміння вимог користувачів та застосування ефективних методів моделювання та конструювання.

Моделювання та конструювання програмного забезпечення становлять ключові етапи у життєвому циклі розробки програм, де визначаються концепції, архітектура та алгоритми, що становлять основу майбутнього продукту. Ефективне моделювання сприяє визначенню оптимальних рішень та забезпеченню гнучкості системи, а конструювання — перетворенню цих концепцій у функціональний та надійний програмний продукт.

Ця курсова робота ретельно розглядає процес моделювання та конструювання програмного забезпечення, аналізуючи основні підходи до створення моделей, методи та інструменти, які використовуються у цьому процесі. Зосереджуючи увагу на важливості етапів моделювання та конструювання, робота пропонує висвітлити ключові концепції та рішення, необхідні для успішної розробки програмного забезпечення в інформаційному середовищі.

2.1 Моделювання та аналіз програмного забезпечення

Проект програмного забезпечення SuperCoffee охоплює два ключові бізнес-процеси: авторизацію користувача та створення замовлення. Нижче розглядається детальний опис кожного процесу та їхній взаємозв'язок.

2.1.1 Авторизація користувача

Для опису бізнес процесу програмного забезпечення використовується BPMN модель (рисунок 2.1).

Опис авторизації користувача:

- Користувач запускає додаток SuperCoffee та переходить на сторінку авторизації, де йому надається можливість ввести облікові дані.
- Користувач вводить свою пошту та пароль для доступу до особистого кабінету.
- Система здійснює перевірку введених даних на відповідність. Якщо дані введено невірно, система виводить повідомлення про помилку та пропонує користувачеві спробувати знову.
- При успішній авторизації користувача система перенаправляє його на персональну сторінку, де він може здійснювати різноманітні операції.

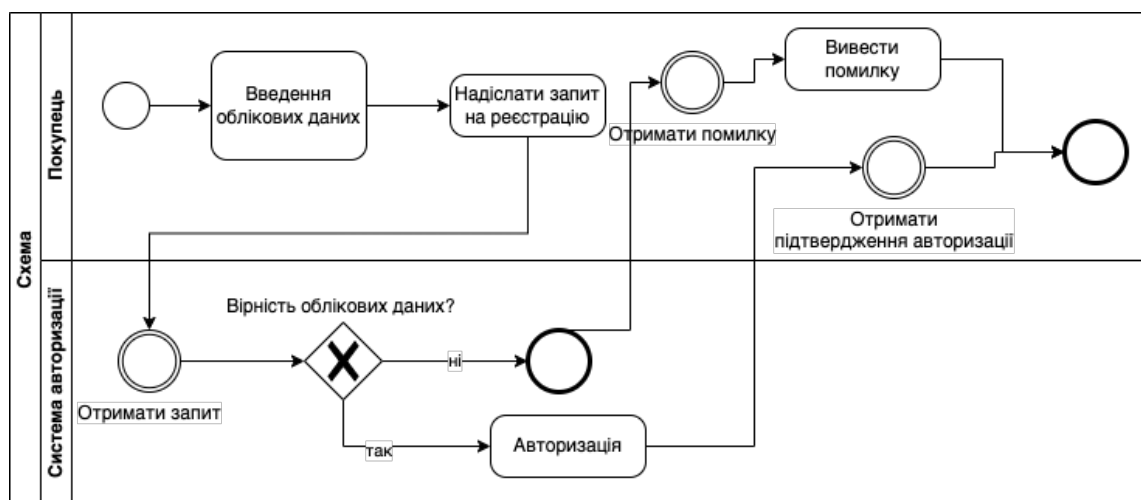


Рисунок 2.1 Схема бізнес-процесу авторизації

2.1.2 Створення замовлення

Для опису бізнес процесу програмного забезпечення використовується BPMN модель (рисунок 2.2).

Опис створення замовлення:

- Користувач переходить у каталог товарів, де він може переглядати та вибирати товари.
- Обравши необхідні товари, користувач додає їх до кошика для подальшого оформлення.
- Користувач переходить до оформлення замовлення, де заповнює важливу інформацію, таку як адреса доставки, спосіб оплати та інші деталі.
- Система перевіряє та підтверджує введену інформацію. У випадку невідповідності система сповіщає користувача та дозволяє внести необхідні зміни.
- Після успішного підтвердження система генерує унікальний номер замовлення та надсилає користувачеві підтвердження з інформацією про замовлення.

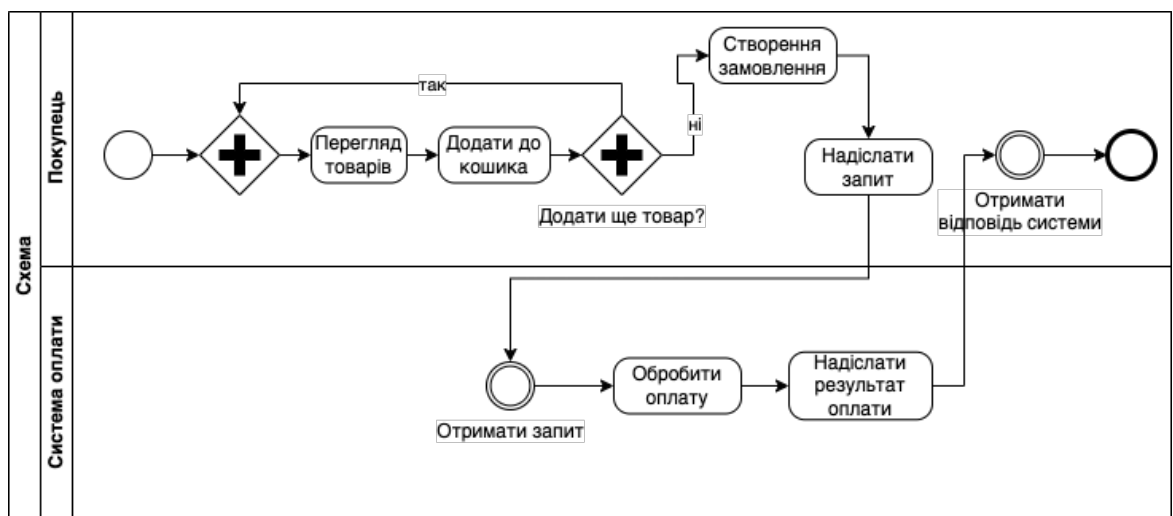


Рисунок 2.2 Схема бізнес-процесу створення замовлення

Кожен з цих процесів взаємодіє з користувачем та системою, забезпечуючи ефективне та зручне використання програмного забезпечення SuperCoffee.

2.2 Архітектура програмного забезпечення

В розробці інтернет-магазину SuperCoffee використовується архітектурний паттерн MVC (Model-View-Controller) для забезпечення ефективного розділення логіки додатку та його візуального представлення.

Використання архітектурного паттерну MVC (Model-View-Controller) у розробці інтернет-магазину SuperCoffee допомагає ефективно розділити логіку додатку та його візуальне представлення, надаючи кілька переваг.

Розділення відповідальностей: моделі управляють даними, контролери обробляють запити користувачів, а компоненти візуалізації відображають дані користувачу.

Підтримка масштабованості: розділення на моделі, контролери та компоненти візуалізації сприяє ефективному масштабуванню системи.

Легкість утримання та розширення: логічне розділення додатку на компоненти дозволяє легко утримувати та розширювати функціонал.

Підтримка тестування: кожен компонент може бути відокремлено протестований, що полегшує виявлення та усунення помилок в окремих частинах системи.

2.2.1 Backend (Серверна частина)

а. Моделі (Models):

Моделі взаємодіють з базами даних для збереження та отримання даних. Використовуються моделі для управління товарами та даними користувачів.

б. Контролери (Controllers):

Контролери відповідають за обробку HTTP-запитів та взаємодію з моделями та компонентами візуалізації. Реалізують бізнес-логіку та керують потоком додатку.

с. База Даних (MySQL):

MySQL використовується для зберігання даних про користувачів та дій для подальшого аналізу та створення рекомендацій.

2.2.2 Frontend (Клієнтська частина)

а. Компоненти Візуалізації (View Components):

Відповідають за представлення та візуалізацію даних користувачу. Забезпечують інтерфейс для пошуку товарів, авторизації та створення замовлень.

Використання React для створення компонентів дозволяє ефективно організувати код як для сторінок, так і для окремих частин додатку. При цьому внутрішній state management, використовуючи context, сприяє зручному та ефективному управлінню станом додатку.

Компоненти візуалізації у цьому контексті відіграють ключову роль у взаємодії з користувачем. Вони відповідають за створення інтерфейсу для пошуку товарів, забезпечення можливості авторизації та управління процесом створення замовлень. Застосування React дозволяє швидку та ефективну розробку зазначених компонентів, забезпечуючи високий рівень користувацької дружби та функціональності.

2.2.3 Бази Даних

а. CommerceJS:

У розробці інтернет-магазину SuperCoffee використовується CommerceJS для зберігання товарів та пов'язаної інформації про замовлення. CommerceJS володіє високою ефективністю у забезпеченні швидкого та надійного доступу до даних товарів, а також забезпечує зручний інтерфейс для роботи з інформацією про замовлення.

б. Firebase:

Для зберігання даних користувачів та забезпечення механізму авторизації використовується Firebase. Firebase дозволяє надійно та безпечно зберігати інформацію про користувачів, включаючи дані авторизації. Використання Firebase сприяє створенню безпечної та швидкодіючої системи авторизації користувачів.

Ці бази даних взаємодіють між собою для забезпечення повноцінного функціонування інтернет-магазину SuperCoffee та забезпечення користувачам зручного та надійного досвіду під час взаємодії з платформою.

2.2.4 Зв'язки між компонентами

a. Взаємодія frontend із сервісом CommerceJS:

Під час створення та обробки замовлень відбувається відправка даних про замовлення до сервісу CommerceJS. Цей сервіс відповідає за збереження товарів та деталей кожного замовлення, забезпечуючи централізовану систему управління товарами та їхніми характеристиками.

b. Використання Firebase для авторизації та збереження даних користувачів:

Сервіс Firebase використовується для реалізації механізму авторизації та збереження важливих даних про користувачів. Ця платформа забезпечує безпечне зберігання та обробку інформації, пов'язаної з аутентифікацією користувачів.

c. Взаємодія backend із базою даних MySQL:

Моделі та контролери backend взаємодіють із базою даних MySQL для ефективного зберігання та аналізу дій користувачів. Ця взаємодія дозволяє вести записи про взаємодію користувачів із системою, забезпечуючи аналіз та підтримку рекомендацій для користувачів.

d. Використання системи оплати Liqpay:

Для забезпечення зручної та безпечної оплати товарів в інтернет-магазині використовується система оплати Liqpay. Цей компонент забезпечує швидкі та надійні транзакції між покупцями та магазином, роблячи процес оплати максимально зручним для користувачів.

Зазначені зв'язки між компонентами створюють інтегровану систему, яка забезпечує зручний та ефективний функціонал інтернет-магазину SuperCoffee.

2.2.5 Аналіз та Рекомендації

а. Аналіз даних користувачів на MySQL базі даних використовується для створення рекомендацій, що підвищують персоналізацію та задоволення користувачів.

Архітектура програмного забезпечення SuperCoffee дозволяє ефективно використовувати ресурси та забезпечує чітке розділення відповідальностей між різними компонентами системи.

Розглянемо схему архітектури:

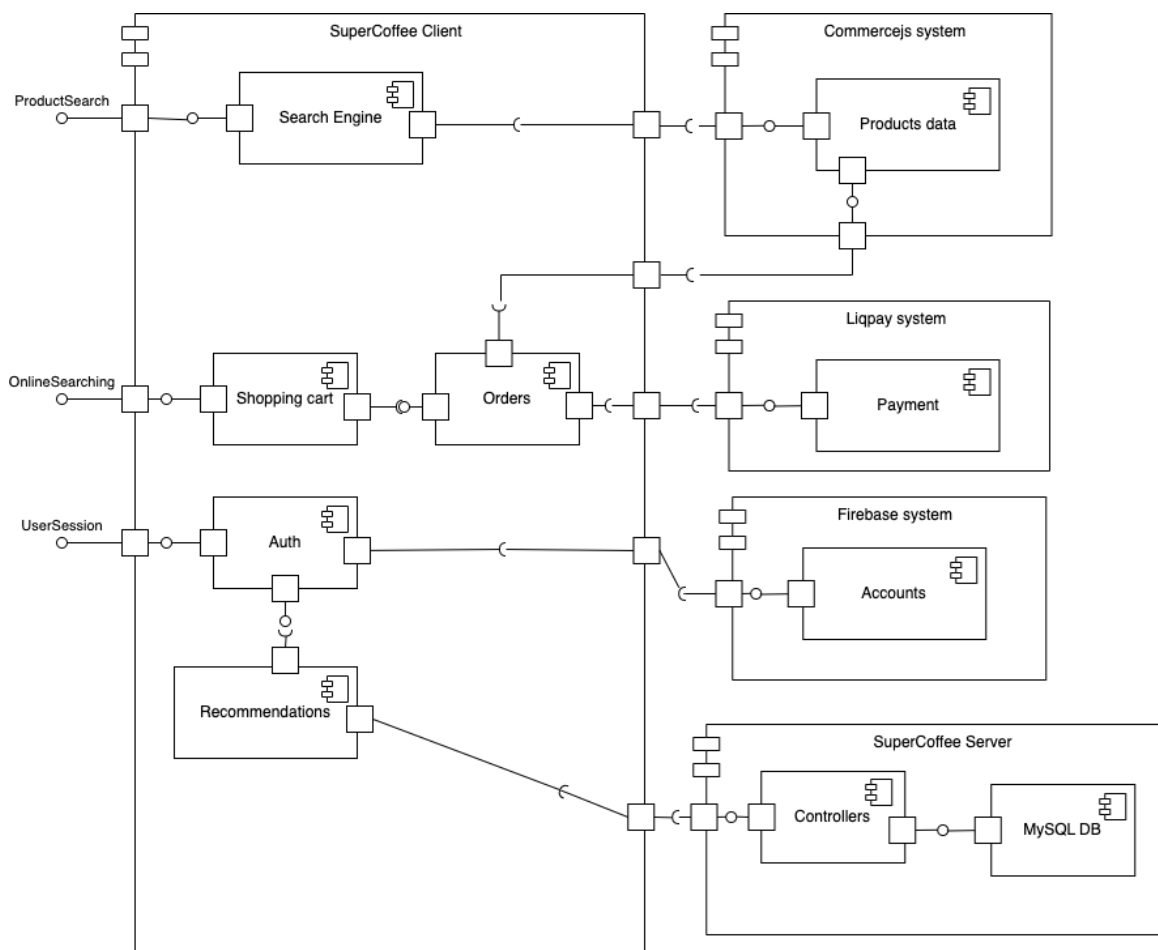


Рисунок 2.3 Діаграма компонентів

2.3 Конструювання програмного забезпечення

В розробці інтернет-магазину SuperCoffee використовується низка різноманітних технологій та бібліотек, які забезпечують ефективну та стабільну

роботу програмного забезпечення. Ось огляд основних використаних інструментів та бібліотек:

2.3.1 Frontend

React виступає як основна бібліотека для розробки користувацького інтерфейсу, надаючи зручний та гнучкий інструментарій для створення компонентів як для окремих сторінок, так і для відображення різних частин інтерфейсу.

CommerceJS – це потужний сервіс для управління товарами та замовленнями, який надає інструменти для ефективного ведення електронної комерції в інтернет-магазинах. Ця бібліотека взаємодіє з CommerceJS, надаючи розробникам зручний та досить гнучкий інтерфейс для взаємодії з продуктами та управління замовленнями. Основні можливості CommerceJS включають в себе можливість додавати, видаляти та редагувати товари, визначати їх властивості та категорії, а також ефективно обробляти та відстежувати замовлення. Завдяки цій технології розробники можуть легко інтегрувати функціонал інтернет-магазину, забезпечуючи користувачам зручний та надійний досвід покупок онлайн. CommerceJS надає розширені можливості для роботи з товарами, включаючи управління інвентарем, визначення знижок та акцій, а також деталізоване відстеження статусів замовлень. Використовуючи цю бібліотеку, розробники можуть ефективно організовувати роботу інтернет-магазину, забезпечуючи надійний та оптимальний функціонал для покупців.

Firebase використовується для реалізації авторизації користувачів та збереження важливих даних про них, забезпечуючи безпеку та зручність у взаємодії з основними функціями додатку.

Liqua – це інтегрована бібліотека, яка гарантує безпечні та ефективні процедури оплати в інтернет-магазині. Вона відповідає за створення надійної системи оплати та обробки транзакцій, забезпечуючи користувачам зручний та безпечний спосіб здійснення покупок. За допомогою Liqua, інтернет-магазин може легко і надійно імплементувати різні методи оплати, такі як кредитні картки, електронні гроші та інші. Бібліотека гарантує швидку та ефективну

обробку транзакцій, а також забезпечує високий ступінь безпеки, щоб захистити дані користувачів під час платежів. Окрім того, `Liqua` дозволяє інтегрувати різні функції, такі як повернення коштів, операції з підписанням та перевіркою статусу оплати. Ця бібліотека стає надійним інструментом для забезпечення безпечності та ефективності процесу оплати в інтернет-магазині.

`React Router Dom` – це бібліотека, яка використовується для впровадження системи навігації між різними сторінками додатку. Забезпечуючи зручний та інтуїтивно зрозумілий інтерфейс, вона дозволяє користувачам легко переміщатися в межах додатку, роблячи взаємодію з різними його частинами безпроблемною. `React Router Dom` допомагає визначити маршрути, які відповідають різним URL-адресам, та визначити, який компонент відображати при зміні URL. Це робить можливим використання динамічних адрес, таких як `/products`, `/orders` тощо, та автоматично відображає відповідний компонент для кожної сторінки. Таким чином, `React Router Dom` вирішує завдання організації навігації в додатку, що є важливим аспектом для створення зручного та інтуїтивно зрозумілого досвіду користувача.

Решта використаних бібліотек і утиліт, охоплюють обробку та взаємодію з даними, стилізацію інтерфейсу, тестування та інші важливі аспекти, сприяючи створенню повноцінного та ефективного додатку.

2.3.2 Backend

`Node.js` є потужним серверним середовищем, побудованим на мові програмування `JavaScript`. Ця технологія відкрита, ефективна та масштабована, що робить її ідеальним вибором для виконання серверного коду в інтернет-магазині `SuperCoffee`. Однією з ключових переваг `Node.js` є його можливість асинхронного виконання операцій, що дозволяє створювати високопродуктивні та швидкодіючі додатки. Це особливо важливо для обробки багатьох одночасних запитів, які часто виникають в онлайн-магазинах. `Node.js` активно використовується для розробки веб-серверів, API та інших серверних компонентів. Його ефективність та легкість використання роблять його популярним серед розробників, а широка спільнота та розгалуженість модулів

роблять його зручним вибором для будь-якого роду серверного розвитку. Враховуючи потужний інструментарій та можливості для масштабування, Node.js стає важливою складовою архітектури інтернет-магазину SuperCoffee, забезпечуючи швидкість та ефективність в обробці запитів користувачів.

MySQL2 та Sequelize виступають ключовими компонентами у взаємодії з базою даних MySQL в розробці інтернет-магазину SuperCoffee. MySQL2 є драйвером бази даних, який забезпечує швидке та ефективне виконання запитів до MySQL. Використовуючи асинхронний підхід з використанням обіцянок (promises), MySQL2 гарантує ефективну взаємодію з базою даних, забезпечуючи стабільність та продуктивність. Sequelize виступає як ORM (Object-Relational Mapping) для спрощення роботи з базою даних MySQL через Node.js. Sequelize дозволяє розробникам працювати з базою даних у вигляді об'єктів та класів, спрощуючи операції створення, читання, оновлення та видалення даних. Разом MySQL2 та Sequelize забезпечують потужні можливості для роботи з базою даних, сприяючи ефективній та надійній роботі інтернет-магазину SuperCoffee.

Бібліотека Natural використовується для обробки природної мови у розробці інтернет-магазину SuperCoffee. Вона надає інструменти для роботи з текстовою інформацією, такі як аналіз тексту, визначення частин мови, виявлення схожих слів та інші функції, що допомагають вдосконалити обробку та розуміння текстового контенту у додатку. Використання бібліотеки Natural сприяє поліпшенню здатностей інтернет-магазину у роботі з різноманітними формами текстового вводу та забезпечує ефективну обробку природної мови для користувачів.

Ці технології та бібліотеки взаємодіють між собою, створюючи зручну та ефективну інфраструктуру для інтернет-магазину SuperCoffee. Вони забезпечують

2.3.3 Збереження даних

Дані в інтернет-магазині SuperCoffee ретельно організовані та зберігаються в різних джерелах для ефективного управління інформацією:

Продукти та дані про замовлення утримуються в сервісі CommerceJS. Це забезпечує оптимальне управління товарами та процесами замовлення у межах магазину.

Інформація про користувачів та система авторизації розміщена в Firebase. Це забезпечує безпеку та зручність управління користувачами та їх авторизацією.

Дії користувачів, від перегляду товарів до реєстрації, фіксуються в MySQL базі даних. Наприклад, в таблиці "actions" зберігаються деталі про взаємодію користувачів з додатком. Це дозволяє аналізувати та вдосконалювати користувацький досвід.

Опис таблиці Actions наведено у таблиці 2.1.

Назва поля	Тип даних	Опис
action_id	INTEGER	Унікальний ідентифікатор дії (автоінкремент) користувача
user_id	STRING	Ідентифікатор користувача, який виконав дію
product_id	STRING	Ідентифікатор продукту, щодо якого була виконана дія
action_type	STRING	Тип дії (наприклад, "click", "cart", "payment")
timestamp	DATE	Дата та час виконання дії (за замовчуванням - поточний момент)

2.4 Аналіз безпеки даних

Аналіз безпеки даних у розробці інтернет-магазину SuperCoffee є ключовим аспектом, спрямованим на забезпечення конфіденційності, цілісності та доступності інформації.

Авторизація та аутентифікація здійснюються через механізми Firebase, що гарантує ефективний контроль доступу та високий рівень конфіденційності особистих даних користувачів.

Для захисту фінансових транзакцій та замовлень використовується система оплати Liqpay, яка забезпечує шифрування та безпеку фінансових операцій.

Захист даних в MySQL базі даних забезпечується за допомогою Sequelize, що дозволяє відповідати стандартам безпеки та зберігати дані надійно та конфіденційно.

Використання бібліотек, таких як Firebase, автоматично забезпечує шифрування даних для їхньої безпеки в процесі збереження та передачі.

Регулярний аналіз вразливостей та оновлення застосунків допомагають виявляти та усувати можливі ризики безпеки.

Механізми управління доступом в CommerceJS гарантують лише необхідний рівень доступу до даних та функціоналу.

Систематичне резервне копіювання баз даних та важливих ресурсів забезпечує можливість ефективного відновлення в разі виникнення ситуацій кризового управління.

Аналіз вразливостей та регулярні аудити безпеки гарантують стабільність та захищеність інтернет-магазину SuperCoffee від потенційних загроз.

Висновки до розділу

У даному розділі виконано комплексне моделювання, конструювання та аналіз програмного забезпечення для інтернет-магазину SuperCoffee. Нижче представлені висновки до кожного підрозділу.

Моделювання та аналіз програмного забезпечення дозволили визначити ключові функціональні та структурні вимоги до системи. Застосовано BPMN-моделювання для процесу авторизації користувача, що відобразило всі етапи та взаємодії у системі. Також була побудована архітектурна схема з використанням паттерну MVC для ефективного розділення логіки додатку та його візуального представлення.

Архітектура програмного забезпечення була ретельно описана, зокрема використання паттерну MVC у розробці інтернет-магазину SuperCoffee. Цей підхід дозволяє гнучко керувати різними компонентами додатку, забезпечуючи його ефективну масштабованість та легке розширення.

У підрозділі "Конструювання програмного забезпечення" надано огляд використаних технологій та бібліотек, таких як React, CommerceJS, Firebase, Lighthouse, і інших, зокрема для фронтенду та бекенду додатку. Зазначено їхні ключові функції та внесок у розробку.

Аналіз безпеки даних виявив, що система ретельно забезпечена за допомогою Firebase для авторизації, Lighthouse для безпечних транзакцій, та Sequelize для захисту від SQL-ін'єкцій. Регулярні аудити та аналіз вразливостей дозволяють ефективно виявляти та усувати можливі ризики безпеки.

Загальним висновком є те, що проведені етапи моделювання, конструювання та аналізу дозволили створити добре організовану та забезпечену систему для інтелектуального інтернет-магазину SuperCoffee, що відповідає вимогам сучасного електронного бізнесу.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Для докладного аналізу якості програмного забезпечення SuperCoffee був використаний інструмент `deepscan.io`. Цей аналіз дозволив виявити важливі аспекти, пов'язані з якістю коду та його надійністю.

Результати аналізу вказують на високу якість коду та його структуру, що робить додаток легко зрозумілим та модифікованим. Глибокий огляд кодової бази не виявив значущих помилок чи потенційних проблем, що свідчить про дбайливий підхід до програмування та відсутність серйозних технічних боргів.

Зокрема, аналіз `deepscan.io` дозволив ідентифікувати та виправити деякі невеликі покращення у коді, що призвело до збільшення його читабельності та оптимізації. Результати аналізу представлені на Рисунку 3.1.



Рисунок 3.1 Результати аналізу

3.2 Опис процесів тестування

Було проведено мануальне тестування програмного забезпечення SuperCoffee з метою перевірки його функціональностей та інтерфейсу. Під час тестування зверталася увага на коректність виконання основних операцій, відповідність вимогам та ефективність взаємодії з користувачем. Опис відповідних тестів наведено у таблицях 3.1 – 3.7.

Таблиця 3.1 – Тест 1

Тест	Реєстрація користувача
Номер тесту	1

Початковий стан системи	Користувач знаходиться на сторінці реєстрації
Вхідні данні	Нікнейм, Електронна пошта, пароль
Опис проведення тесту	У відповідні поля вводяться: Нікнейм, який може мати як літери так і цифри до 64 символів, коректна електронна пошта, яка до цього не була зареєстрована в системі, пароль від 8 до 50 символів, який містить хоча б з одну англійську літеру, одне число і один спеціальний символ. Після цього натискається кнопка Submit
Очікуваний результат	Реєстрація проходить успішно, користувач додається у систему, одразу авторизується і переходить на головну сторінку.
Фактичний результат	Реєстрація проходить успішно, користувач додається у систему, одразу авторизується і переходить на головну сторінку.

Таблиця 3.2 – Тест 2

Тест	Перегляд продуктів
Номер тесту	2
Початковий стан системи	Користувач перебуває сторінці меню.
Вхідні данні	Назва продукту
Опис проведення тесту	Користувач переглядає список продуктів, використовуючи інтерфейс користувача. Переконається, що всі продукти вірно відображаються, інформація про кожен продукт повна і коректна. Також використовується фільтри: сортування за ціною, вибір категорії товару чи у пошук в водиться назва продукту

Очікуваний результат	Користувач може безперешкодно переглядати всі доступні продукти, інформація про них вірно відображається.
Фактичний результат	Користувач може безперешкодно переглядати всі доступні продукти, інформація про них вірно відображається.

Таблиця 3.3 – Тест 3

Тест	Перегляд деталей продукту
Номер тесту	3
Початковий стан системи	Користувач вже переглянув список продуктів і вибрав конкретний продукт для детального перегляду.
Вхідні данні	-
Опис проведення тесту	Користувач натискає на обраний продукт для переходу на сторінку з його деталями. Переконується, що всі деталі продукту вірно відображаються: назва, опис, ціна, зображення тощо.
Очікуваний результат	Користувач переходить на сторінку з деталями обраного продукту, і усі деталі вірно відображаються.
Фактичний результат	Користувач переходить на сторінку з деталями обраного продукту, і усі деталі вірно відображаються.

Таблиця 3.4– Тест 4

Тест	Перегляд деталей продукту
Номер тесту	4
Початковий стан системи	Користувач вже переглянув список продуктів і вибрав конкретний продукт для детального перегляду.
Вхідні данні	-

Опис проведення тесту	Користувач натискає на обраний продукт для переходу на сторінку з його деталями. Переконається, що всі деталі продукту вірно відображаються: назва, опис, ціна, зображення тощо.
Очікуваний результат	Користувач переходить на сторінку з деталями обраного продукту, і усі деталі вірно відображаються.
Фактичний результат	Користувач переходить на сторінку з деталями обраного продукту, і усі деталі вірно відображаються.

Таблиця 3.5— Тест 5

Тест	Додавання товарів до кошика
Номер тесту	5
Початковий стан системи	Користувач знаходиться на сторінці обраного продукту і готовий додати його до кошика.
Вхідні данні	-
Опис проведення тесту	Користувач може обрати кількість. Користувач натискає на кнопку "Додати до кошика" на сторінці продукту. Впевнюється, що обраний продукт додається до кошика, на іконці кошика в головному меню є цифра кількості продуктів.
Очікуваний результат	Обраний продукт додається до кошика користувача, і користувач отримує підтвердження додавання товару.
Фактичний результат	Обраний продукт додається до кошика користувача, і користувач отримує підтвердження додавання товару.

Таблиця 3.6— Тест 6

Тест	Оформлення замовлення
Номер тесту	6

Початковий стан системи	Користувач має обраний товар в кошику і готовий оформити замовлення. Натискає кнопку "Оформити замовлення" і переходить на сторінку перевірки.
Вхідні данні	Ім'я, прізвище, email та телефон отримувача, а також адреса доставки (вулиця, будинок, квартира).
Опис проведення тесту	Користувач вводить всі необхідні дані для доставки. Перевіряє правильність введених даних. Після цього натискає кнопку "Оплатити".
Очікуваний результат	Користувач після натискання кнопки "Підтвердити замовлення" переходить на сторінку сервісу оплати. Там він обирає зручний спосіб оплати, підтверджує оплату і автоматично повертається на головну сторінку із очищеним кошиком.
Фактичний результат	Користувач після натискання кнопки "Підтвердити замовлення" переходить на сторінку сервісу оплати. Там він обирає зручний спосіб оплати, підтверджує оплату і автоматично повертається на головну сторінку із очищеним кошиком.

Таблиця 3.7– Тест 7

Тест	Рекомендації для користувача
Номер тесту	7
Початковий стан системи	Користувач знаходиться на головній сторінці інтернет-магазину "SuperCoffee" чи на сторінці меню.
Вхідні данні	-
Опис проведення тесту	Користувач ознайомлюється з рекомендаціями, які система пропонує для нього. Він перевіряє, чи рекомендації відображаються на основі його попередніх виборів та інтересів.

Очікуваний результат	Користувач повинен бачити персоналізовані рекомендації, які враховують його попередні покупки та перегляди.
Фактичний результат	Користувач повинен бачити персоналізовані рекомендації, які враховують його попередні покупки та перегляди.

Висновки до розділу

У данному розділі було проведено детальний аналіз якості програмного забезпечення "SuperCoffee" та визначено етапи та результати тестування.

Було використано сервіс deepscan.io для проведення аналізу якості програмного коду. Результати аналізу показали високу якість коду та відсутність критичних помилок чи вразливостей.

Мануальне тестування було проведено для впевненості у функціональності та безпеці програмного забезпечення. Було проведено різні тести, такі як реєстрація користувача, перегляд продуктів, додавання товарів до кошика, оформлення замовлення та отримання рекомендацій. Усі тести пройшли успішно, що підтверджує надійність та стабільність роботи системи.

Аналіз якості та тестування програмного забезпечення підтверджують, що "SuperCoffee" є надійним та безпечним інтернет-магазином. Висока якість коду, відсутність критичних помилок та успішно пройдені тести свідчать про готовність програмного забезпечення до впровадження та використання користувачами.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

4.1.1 Розгортання клієнтської частини на Netlify

Зареєструвати обліковий запис на [Netlify](https://www.netlify.com/) та увійти в систему.

Перейти до панелі керування Netlify і створити новий сайт, пов'язаний із репозиторієм фронтенду на GitHub чи перемістити папку build.

Вказати основні параметри проекту, такі як відділення Git, команда для будівництва (зазвичай `npm run build`), та папка, в якій знаходиться збудований код.

У вкладці "Deploys" налаштувати автоматичне розгортання для гілки `main`. Кожен новий коміт в цю гілку буде тригерити автоматичне оновлення на Netlify.

Зберегти API ключ Netlify, який знадобиться для налаштування зв'язку між фронтендом і бекендом.

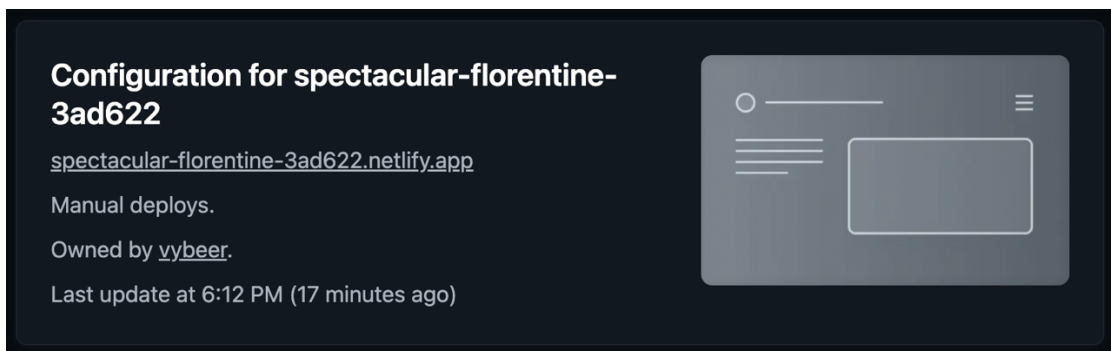


Рисунок 4.1 Розгорнутий фронтенд

4.1.2 Розгортання серверної частини на Render

Зареєструвати обліковий запис на [Render](https://render.com/) та увійти в систему.

Створити новий проект і вибрати "Web Service" як тип.

Додати новий сервіс, вибрати середовище виконання (Node.js) та вказати команду для запуску сервера.

В налаштуваннях проєкту вказати GitHub репозиторій та гілку для автоматичного розгортання. Render буде автоматично розгортати новий код.

Зберегти URL серверу, який буде використовуватися фронтендом для з'єднання з бекендом.

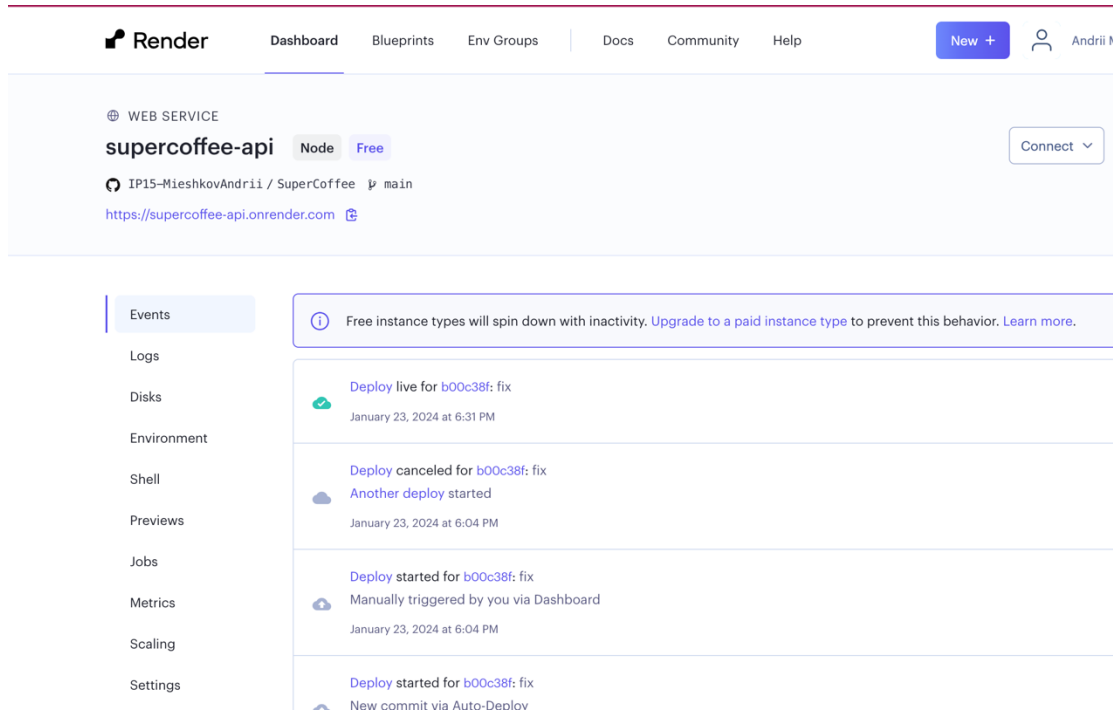


Рисунок 4.2 Розгорнутий бекенд

4.1.3 Розгортання бази даних MySQL на Clever Cloud

Зареєструвати обліковий запис на [Clever Cloud](https://www.clever-cloud.com/) та увійти в систему.

Створити новий додаток і вибрати MySQL як сервіс бази даних.

Вказати параметри для бази даних, такі як ім'я, користувач, пароль і т. д.

Зберегти дані для доступу до бази даних, а також змінити конфігурацію свого сервера для використання цих даних.

Зберегти URL бази даних, який буде використовуватися сервером для з'єднання з базою даних.

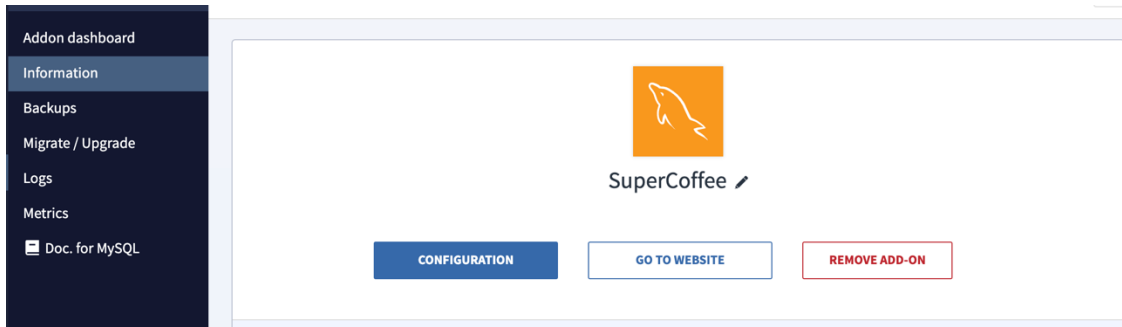


Рисунок 4.3 Розгорнута база даних

Зараз програмне забезпечення повністю розгорнуто. Фронтенд працює на Netlify, серверні компоненти розміщені на Render, а база даних MySQL запущена на Clever Cloud.

4.2 Підтримка програмного забезпечення

Підтримка програмного забезпечення є ключовою складовою для забезпечення безперебійної роботи та вдосконалення функціональності. У цьому розділі ми розглянемо стратегії підтримки для клієнтської і серверної частин програмного забезпечення, а також бази даних.

4.2.1 Підтримка клієнтської частини на Netlify

Використання інструментів моніторингу, які надає Netlify, для відстеження використання та виявлення можливих проблем.

Використання можливостей автоматичного розгортання Netlify для безпечного і швидкого впровадження змін у продакшн.

Забезпечення регулярного створення запасних копій фронтенду та використання систем версіонування, таких як Git, для ведення історії змін.

4.2.2 Підтримка серверної частини на Render

Встановлення моніторингу серверної частини за допомогою інструментів, таких як Prometheus або New Relic, для виявлення проблем та аналізу продуктивності.

Використання системи журналування для реєстрації подій та помилок, що дозволяє швидко реагувати на проблеми та аналізувати причини неполадок.

Налаштування Render для автоматичного розгортання нових версій серверної частини при кожному коміті в гілку `main`.

Регулярне вивчення питання безпеки та оновлення залежності, щоб уникнути вразливостей.

4.2.3 Підтримка бази даних MySQL на Clever Cloud

Планування регулярних резервних копій бази даних та випробовування процедури відновлення для забезпечення безперебійної доступності даних.

Використання інструментів моніторингу для визначення продуктивності бази даних та виявлення можливих проблем.

Аналіз та оптимізація запитів до бази даних для забезпечення ефективності роботи.

Підтримка програмного забезпечення – це постійний процес, який включає в себе моніторинг, тестування, резервне копіювання та постійне оновлення. Зазначені стратегії допоможуть утримати програмне забезпечення в робочому стані та готовим до зустрічі нових викликів.

Висновки до розділу

У даному розділі було висвітлено ключові аспекти розгортання та підтримки програмного забезпечення.

Використано сучасні технології для розгортання клієнтської та серверної частини. Зокрема, для фронтенду обрано платформу Netlify, що дозволяє автоматизувати процеси розгортання через інтеграцію з GitHub Actions. Для серверної частини використано Render, який також використовує автоматичне розгортання при кожному коміті.

Розкрито стратегії для підтримки роботи фронтенду, серверної частини та бази даних. Зокрема, використано моніторинг, резервне копіювання та заходи безпеки для забезпечення надійності та продуктивності системи.

Загальною метою даного розділу було забезпечити не лише ефективне впровадження розробленого програмного продукту, але й засіб його стійкої та довгострокової роботи за умов реального експлуатаційного середовища.

Розгортання та підтримка програмного забезпечення є ключовим етапом в життєвому циклі розробки. Використання сучасних сервісів для розгортання, таких як Netlify та Render, а також впровадження стратегій підтримки дозволяє забезпечити стабільну та ефективну роботу програмного продукту. Постійний моніторинг та регулярне оновлення гарантують високу якість та безпеку використання програмного забезпечення у реальних умовах експлуатації.

ВИСНОВКИ

В ході виконання курсової роботи було розроблено інтелектуальний інтернет-магазин, який включає в себе рекомендаційну систему, засновану на стратегії фільтрації контенту. Проект використовує технологію Full-Stack, де клієнтська частина побудована з використанням бібліотеки React, а серверна частина використовує NodeJS. Клієнтська програма надає візуальний інтерфейс для взаємодії з покупцем, використовуючи запити для збереження різноманітних даних у відповідних сховищах. З метою забезпечення безпеки клієнтської інформації використовується Firebase. Для ефективного управління даними про продукти та швидкої обробки замовлень використовується CommerceJS, що сприяє практичному застосуванню веб-застосунку у реальному бізнесі.

У якості середовища розробки обрано Visual Studio Code за його зручний інтерфейс та вбудовані бібліотеки.

В ході реалізації основного алгоритму, який перетворює інтернет-магазин у інтелектуальний, була використана рекомендаційна система, що базується на діях користувачів та використовує базу даних MySQL для зберігання інформації. Цей алгоритм робить проект більш конкурентоспроможним у сфері бізнесу порівняно з іншими інтернет-комерційними рішеннями. Розроблена система виявляється привабливою для клієнтів та прискорює процес замовлення, що становить значущий плюс для споживачів.

Оцінка одержаних результатів свідчить про високий рівень застосування сучасних наукових і технічних знань у процесі розробки. Проект відзначається ступенем впровадження та обширним спектром можливих галузей застосування у реальному бізнесі. Наукова, науково-технічна та соціально-економічна значущість роботи виявляються у створенні ефективного інтернет-магазину та його впливу на покращення процесів електронної комерції. Зазначається доцільність подальших досліджень у даній тематиці для ще глибшого розуміння та вдосконалення розробленого інтернет-магазину.

В ході оцінки отриманих результатів слід відзначити, що даний проєкт, незважаючи на свою ефективність та конкурентоспроможність, не претендує на

позиції глобальних корпорацій у сфері електронної комерції. Тем не менше, він розглядається як важливий інструмент для покращення умов для малих бізнесів та вузьконаправлених магазинів.

Застосування інтелектуального підходу та рекомендаційної системи може допомогти вдосконалити малі та середні підприємства, надаючи їм можливість конкурувати у своїх нішах. Проєкт спрямований на поліпшення конкурентоспроможності в цих сегментах ринку, забезпечуючи їм засоби для ефективного взаємодії з клієнтами та оптимізації процесів управління.

Окрім того, даний проєкт може слугувати як приклад та вихідна точка для інших малих бізнесів, які мають намір розвивати свою присутність у сфері електронної комерції. Враховуючи сучасні тенденції та технічні можливості, він надає відмінну можливість для малого бізнесу покращити свою ефективність та залучити нових клієнтів, тим самим сприяючи розвитку даного сектору економіки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) E-commerce. *The TechTarget*. URL: <https://www.techtarget.com/searchcio/definition/e-commerce>
- 2) What is eCommerce Intelligence? *Obviyo*. URL: <https://obviyo.com/ecommerce-intelligence/>
- 3) Артемов, В., Бахчеван, Е., Данько, Т. Тенденції розвитку електронної комерції в Україні. *Науковий вісник Одеського національного економічного університету*. 2018. №10. С.13-35.
- 4) The rise of ecommerce in the coffee industry. *Perfect Daily Grind*. URL: <https://perfectdailygrind.com/2023/03/the-rise-of-ecommerce-in-the-coffee-industry/>
- 5) Poonam B. Thorat, R. M. Goudar, Sunita Barve. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*. 2015. №4. С.32-34.
- 6) JavaScript. *MDN Web Docs*. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- 7) React. URL: <https://react.dev/>
- 8) Node.js. URL: <https://nodejs.org/>
- 9) Бібліотека natural. URL: <https://github.com/NaturalNode/natural>
- 10) Visual Studio Code IDE. URL: <https://code.visualstudio.com/docs>
- 11) WebStorm IDE. URL: <https://www.jetbrains.com/webstorm/?var=1>
- 12) Atom IDE. URL: <https://atom-editor.cc/docs/>
- 13) Selling Coffee Online. URL: <https://www.sellingcoffeeonline.com/the-best-coffee-shop-websites/>