Пояснювальна записка до курсової роботи

на тему: Веб-застосунок підтримки роботи пункту обміну валют

КПІ.ІП-1304.045440.02.81

3MICT

ПЕР	ЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
BCT	УП	5
1 Al	НАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
1.1	Загальні положення	6
1.2	Змістовний опис і аналіз предметної області	7
1.3	Аналіз існуючих технологій та успішних IT-проєктів	8
1.3.1	Аналіз відомих алгоритмічних та технічних рішень	9
1.3.2	Аналіз допоміжних програмних засобів та засобів розробки	.12
1.3.3	Аналіз відомих програмних продуктів	.14
1.4	Аналіз вимог до програмного забезпечення	. 15
1.4.1	Розроблення функціональних вимог	. 23
1.4.2	Розроблення нефункціональних вимог	. 27
1.5	Постановка задачі	. 28
Висн	новки до розділу	. 29
2 M	ОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО	
ЗАБ	ЕЗПЕЧЕННЯ	30
2.1	Моделювання та аналіз програмного забезпечення	30
2.2	Архітектура програмного забезпечення	.33
2.3	Конструювання програмного забезпечення	.35
2.4	Аналіз безпеки даних	.39
Висн	новки до розділу	40
3 Al	НАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕНН	Я41
3.1	Аналіз якості ПЗ	41
3.2	Опис процесів тестування	41
3.3	Опис контрольного прикладу	.47
Висн	новки до розділу	51
4 BI	ПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	.53
4.1	Розгортання програмного забезпечення	.53
4.2	Підтримка програмного забезпечення	.54

Висновки до розділу	54
ВИСНОВКИ	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IDE – Integrated Development Environment – інтегроване

середовище розробки.

API – Application programming interface, прикладний

програмний Інтерфейс

IT – Інформаційні технології

ER – Entity-Relation diagram

БД – База даних.

HTML – HyperText Markup Language

JSX – JavaScript XML

JS – JavaScript

VS Code – Visual Studio Code

BPMN – Business Process Model and Notation

XSS - Cross Site Scripting

MITM – Man in the middle

ВСТУП

У сучасному світі, де технологічний прогрес стає вирішальним чинником для розвитку різних галузей, надважливою стає роль інноваційних рішень у спрощенні та оптимізації буденних процесів. Враховуючи це, виникає актуальна потреба у створенні ефективних інструментів для обміну валют та взаємодії між клієнтом та фізичними пунктами обміну із застосуванням новітніх технологій.

Розробка веб-застосунку "CashFlow Exchange" ставить перед собою важливі цілі, спрямовані на вирішення актуальних проблем та вдосконалення валютних обмінних процесів. На тлі світових тенденцій у сфері фінансових технологій та сервісів, створення зручного інструменту для моніторингу курсів валют та ефективної взаємодії між клієнтами та обмінними пунктами стає важливим напрямом розвитку.

Сучасний стан об'єкта розробки дозволяє виявити прогресивні підходи та вже існуючі рішення, які використовують провідні наукові установи та організації. Детальний аналіз відомих розробок та наукових підходів розкриває сильні та слабкі сторони існуючих рішень, визначаючи можливості для подальшого вдосконалення та унікальні аспекти, які реалізуються в рамках даного проекту.

Ця курсова робота також зосереджена на розгляді сучасних глобальних тенденцій у сфері обміну валют та фінансових технологій, а також визначенні перспективних сфер застосування розробленого веб-застосунку.

Важливість інновацій фінансовому секторі необхідність У та процесів вдосконалення валютного обміну роблять цей проект високоактуальним та важливим для впровадження в практику фінансових послуг та економічної взаємодії.

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Фізичні пункти обміну валют^[1] — обмінні пункти, які належать до певних кредитно-фінансових установ або до суб'єктів підприємницької діяльності, що діють на підставі агентських угод з банками, та мають ліцензію Національного банку України на здійснення операцій з торгівлі іноземною валютою.

Торгівля валютою, як явище, існує дуже давно та є однією з основних операцій обміну товарів між людьми. В давні часи люди міняли просто між собою різні речі, в сучасному ж світі відбувається обмін валюти на потрібні речі, але по всьому світу поширені безліч валют, які належать різним країнам і безпосередньо залежать від статусу та стабільності в країна. Так, наприклад, американський долар ϵ однією з найстабільніших та головною валютою світу, якраз через статус країни володаря. Люди купують американську валюту, будучи впевненими, що США будуть існувати, як стабільний гарант цієї валюти ще багато років. Оскільки валюта повністю залежить від держави, якій належить, то нестабільна ситуація в Україні змушує українців проводити маніпуляції з валютою задля збереження власних коштів та уникнути залежності від інфляції, бо нікому не відомо, що може статись з гривнею завтра. Вже було безліч прикладів сильного знецінення, як, наприклад, гіперінфляція в Німеччині в 1923 р., де темп становив 3.25×10⁶ відсотки на місяць (тобто ціни подвоювались кожні два дні), а найбільша інфляція в історії в Угорщині 1945 р. мала темпи 1.295×10^{16} відсотки протягом місяця. $^{[2]}$ В сучасному ж світі такі країни, як Аргентина та Турція потерпають від безперервної інфляції, саме тому обмін валют – дуже важливий процес для людства

Виходячи з пояснень, наданих раніше, дуже важливо забезпечити, щоб обмін валюти залишався простим і ефективним процесом, не перевантажуючи людей своїми складнощами. Зважаючи на це, сучасна сфера обміну валюти

вимагає нових підходів для взаємодії клієнтів із фізичними пунктами обміну. Ініціатива «CashFlow Exchange» спрямована на інтеграцію технологій, які автоматизують облік операцій, пропонують онлайн-доступ до курсів у реальному часі та створюють ефективний інструмент для залучення клієнтів.

Предметна область передбачає підхід до процесу обміну валют, який є буденним для багатьох людей, з новими технологіями для оптимізації процесу обміну валют між людиною та сервісом. В еру сучасних валютних бірж та криптотехнологій пункти обміну валют мають відповідати планці, яку задали різні існуючі проекти, та відповідати всім можливим вимогам користувача, оскільки процес обміну валют не має здаватись складним для людей, які бажають здійснити процес, який має бути максимально простим.

Напрями розвитку в цій сфері фокусуються на розширенні можливостей веб-додатка, зокрема, розробці інтуїтивних інтерфейсів, максимальне спрощення інтерфейсу задля зручності та зміни ставлення людей до процесу обміну валют.

1.2 Змістовний опис і аналіз предметної області

На сучасному етапі розвитку ІТ-технологій пункти обміну валют використовують веб-застосунки для підтримки своєї роботи, проте існують деякі важливі аспекти, які потребують уваги та оптимізації.

На сьогоднішній день, веб-застосунки для підтримки роботи пунктів обміну валют забезпечують функціонал, пов'язаний з реєстрацією та авторизацією користувачів, переглядом актуальних курсів валют, розрахунком обміну, створенням запитів на обмін та веденням історії операцій. Однак, існують обмеження у зручності та ефективності використання, такі як складність інтерфейсу, відсутність автоматизованих підказок для користувачів та обмежені можливості аналітики.

Недоліки поточного стану речей:

- Складний користувацький інтерфейс: Багато веб-застосунків валютних обмінників мають неінтуїтивний та заплутаний інтерфейс, що може ускладнювати використання користувачами.
- Відсутність персоналізації: Більшість застосунків не надають можливості налаштування особистого кабінету та персональних уподобань користувачів з приводу відстежування певних валют.
- Недостатній функціонал аналітики: Відсутність розвинених засобів аналізу курсів валют та ринкових тенденцій може ускладнювати прийняття користувачем обґрунтованих рішень.

Можливі шляхи покращення ситуації в сфері IT:

- Удосконалення інтерфейсу: Розробка інтуїтивно зрозумілого та зручного інтерфейсу для користувачів, що сприятиме легкому використанню застосунку.
- Впровадження персоналізації: Розробка функцій налаштування та персоналізації особистого кабінету для зручності користувачів.
- Розвинення аналітичних засобів: Впровадження аналітичних інструментів, які дозволять користувачам отримувати детальну інформацію щодо курсів та ринкових тенденцій.

У межах курсової роботи покладено акцент на розробку веб-застосунку CashFlow Exchange, спрямованого на оптимізацію процесу обміну валют та полегшення комунікації між клієнтом та пунктом обміну. Проект передбачає впровадження інтуїтивного інтерфейсу, персоналізацію для користувачів та спрощення процесу обміну.

1.3 Аналіз існуючих технологій та успішних ІТ-проєктів

Проаналізуємо відоме на сьогодні алгоритмічне забезпечення у даній області та технічні рішення, що допоможуть у реалізації застосунку підтримки роботи пункту обміну валют. Далі будуть розглянуті допоміжні програмні засоби, засоби розробки та готові програмні рішення.

1.3.1 Аналіз відомих алгоритмічних та технічних рішень

У розробці застосунку підтримки роботи пункту обміну валют важливо обрати оптимальні алгоритми та технічні рішення, що відповідають особливостям предметної області та завданням застосунку. Далі наведено аналіз відомих алгоритмів та технічних рішень, зроблено порівняльний аналіз та обрано оптимальні елементи для використання у розробці.

У сфері обміну валют та фінансових операцій існує кілька відомих рішень та платформ, які надають послуги обміну валют та підтримки фінансових операцій. Приклади:

Revolut - це фінансова технологічна компанія, яка надає послуги обміну валют та багато інших фінансових опцій через їхній мобільний додаток (рисунок 1.3.1.1).

Особливості: Швидкий обмін валют з реальним курсом, можливість віртуальних та фізичних карток, розсилка грошей в будь-яку країну без комісії тощо.

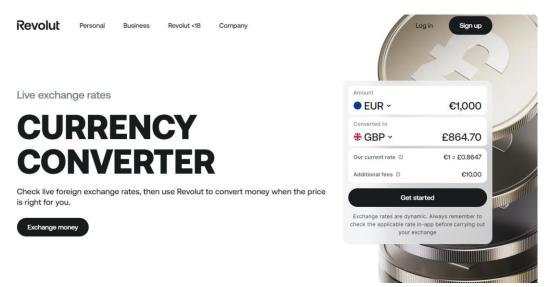


Рисунок 1.3.1.1 – конвертер валют Revolut.

Wise - це сервіс для переказу грошей та обміну валют, який дозволяє користувачам здійснювати міжнародні перекази за більш низькими комісіями та реальним курсом валют (рис.1.3.1.2).

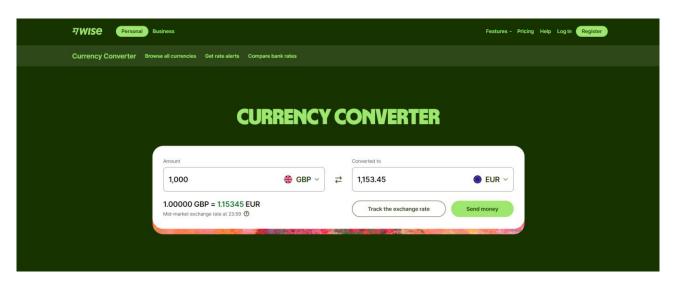


Рисунок 1.3.1.2 – конвертер валют Wise.

Особливості: Обмін валют без прихованих комісій, отримання місцевого банківського рахунку для зручних місцевих переказів тощо.

PayPal - популярна платіжна система, яка надає можливість здійснювати перекази грошей та обмін валют через свою платформу (рис. 1.3.1.3).

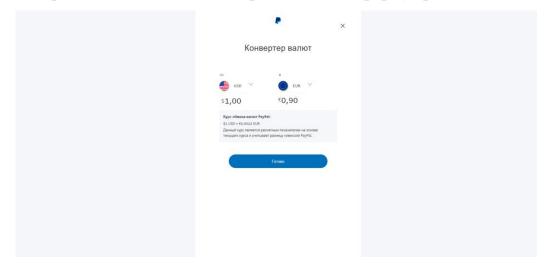


Рисунок 1.3.1.3 – конвертер валют PayPal.

Особливості: Легкість використання, міжнародний охоплення, різноманітні опції фінансових операцій тощо.

Coinbase - популярна платформа для обміну та зберігання криптовалют, таких як Bitcoin, Ethereum тощо.

Особливості: Обмін криптовалют, можливість виводу коштів на банківські рахунки, віртуальні гаманці для криптовалют тощо.

Firebase надає потужні інструменти для реалізації безпечної та ефективної системи реєстрації та авторизації користувачів. Firebase Auth використовує

алгоритм хешування для безпечного зберігання паролю в базі даних. Захист паролю за допомогою алгоритму хешування забезпечує конфіденційність особистих даних користувача. Після успішної реєстрації користувача, Firebase Auth використовує алгоритм для перевірки введеного пароля з збереженим хешем. У разі введення правильного пароля, користувач отримує унікальний токен, який використовується для авторизації при подальших запитаннях до сервера. Застосування токенів для авторизації забезпечує безпеку взаємодії із сервером.

Користувач виражає запит до сервера Firebase ^[3], вказуючи критерії пошуку або унікальний ідентифікатор запису. Firebase, використовуючи свою систему індексації, швидко знаходить та витягує необхідні дані. Переваги: швидкий доступ до даних завдяки ефективній системі індексації Firebase. Можливість синхронізації даних в реальному часі за допомогою Firestore.

Усе описано вище залежить від швидкості та стабільності Інтернет- 3'єднання для ефективної роботи, але сьогодні Інтернет покриття не є проблемою для населення, особливо потребуючого регялурний обмін валют.

Інформація про актуальні курси валют надходить з зовнішньої АРІ, що гарантує надійність та актуальність інформації. Використання готової АРІ значно спрощує процес розробки, оскільки не потрібно витрачати час на складний та потенційно небезпечний процес скрапінгу або парсингу даних. Також, оскільки зовнішні АРІ призначені для ефективної взаємодії з віддаленими серверами, це дозволяє отримувати необхідні дані швидко та ефективно, зменшуючи навантаження на власний сервер. Також це дозволяє легко масштабувати функціонал застосунку, додаючи або змінюючи джерела даних без необхідності переробки всього збірника або скрапера.

На жаль, використання зовнішніх API робить застосунок залежним від інших сервісів, але існує безліч різних доступних сервісів, які зарекомендували себе, як надійні та стабільні.

Отже, використовуючи Firebase, можна забезпечити швидкий та ефективний доступ до БД, реалізувати аутентифікацію та безпеку даних,

зовнішні АРІ дозволяють отримувати актуальну інформацію про курси, Firebase дозволяє легко адаптувати систему до змінних потреб користувача через свою гнучкість та розширюваність, також пропонує готові рішення для обробки та зберігання даних без значних витрат на інфраструктуру, Firebase має вбудовані засоби для забезпечення безпеки та прискорення швидкості обробки даних.

1.3.2 Аналіз допоміжних програмних засобів та засобів розробки

У ході розробки веб-застосунку використовувалась низка допоміжних програмних засобів, які включають мови програмування, фреймворки та інструменти розробки. Нижче наведено опис та порівняльний аналіз цих засобів

JavaScript (JS)^[5]: Мова програмування, яка використовується для розробки динамічних веб-сайтів та веб-застосунків. Переваги: найпоширеніша мова для фронтенд-розробки. Має велику спільноту та екосистему, багато готових рішень, підтримується всіма сучасними браузерами.

 $React^{[4]}$: JavaScript-бібліотека для побудови інтерфейсів користувача, компонентах. Найвідомішим заснована аналогом React ϵ Vue.js. JSX у React дозволяє вписувати HTML-подібний Використання безпосередньо в JavaScript, що спрощує створення компонентів. Vue.js дозволяє використовувати HTML-подібний код у файлах шаблонів, роблячи його більш зрозумілим та читабельним. React використовує віртуальний DOM для оптимізації та прискорення процесу оновлення інтерфейсу. Vue.js вбудовує реактивність безпосередньо в систему, що спрощує відслідковування змін у даних та автоматичне оновлення інтерфейсу. React надає більше гнучкості у виборі та налаштуванні інструментів та бібліотек. React має широку та активну спільноту розробників та безліч корисних бібліотек. Хоча спільнота Vue.js менша, вона стрімко зростає, а фреймворк має добру екосистему.

Firebase: Облачна платформа від Google для розробки веб-застосунків, яка надає ряд послуг, таких як база даних, аутентифікація, зберігання та інше. Переваги: Firebase надає повний спектр інструментів для роботи з базою даних,

аутентифікацією, зберіганням файлів та іншими ключовими функціональностями. Вибір Firebase може прискорити розробку та спростити управління серверною частиною.

Visual Studio Code (VS Code): Легкий та потужний текстовий редактор від Microsoft з великою кількістю розширень для розробників. VS Code підтримує різні мови програмування та технології, що робить його універсальним інтерфейсом. Один із аналогів Visual Studio Code - Atom. VS Code використовує ефективний та швидкий двигун, що робить роботу з кодом ефективною та безперервною. Atom використовує JavaScript для свого інтерфейсу, що може призводити до меншої швидкодії та продуктивності, особливо в обробці великих файлів. VS Code має широкий магазин розширень, який дозволяє легко встановлювати та управляти розширеннями. Atom також має можливості розширення, але його екосистема може бути менш розвиненою порівняно із VS Code. VS Code добре працює навіть на менш потужних комп'ютерах і не вимагає великої кількості ресурсів. Атот може використовувати більше ресурсів, особливо при роботі з великими проектами та файлами. VS Code має вбудовану підтримку Git, що полегшує ведення версій та співпрацю з репозиторіями. В Atom для повноцінної підтримки Git може бути потрібно встановлювати деякі додаткові плагіни.

Vite^[6]: Швидкий та мінімалістичний інструмент для розробки вебзастосунків на базі JavaScript. Переваги: Дозволяє швидко розгортати проекти та надає широкий спектр можливостей. Один із аналогів - Webpack. Vite швидше за рахунок використання ESM, гарантує швидке перезавантаження браузера при внесенні змін, має простий конфігураційний файл, що робить його більше зрозумілим та легким.

Redux^[7]: Управління станом застосунків JavaScript, часто використовується з React. Переваги: Допомагає підтримувати прозорий та простий стан додатку, зручний для великих застосунків. Одним з аналогів Redux є MobX. Обидві бібліотеки застосовуються для управління станом додатку в JS-проектах, але кожна має свої особливості. Redux базується на

централізованому збереженні стану, що забезпечує структурованість та передбачуваність управління даними в додатку, в той же час МоЬХ дозволяє кожному об'єкту в додатку мати свій власний стан, що може призводити до менш очевидного контролю та більш складного управління станом у великих проектах. Redux використовує чисті функції (reducers) для зміни стану, що робить код більш декларативним та легким для тестування. Централізована структура Redux робить його більш підходящим для великих проектів, де важлива чіткість та контроль над станом. МоЬХ може бути зручним для менших проектів або тих, де не потрібна така велика централізована контрольованість.

1.3.3 Аналіз відомих програмних продуктів

Мопеу24 — веб-застосунок для підтримки роботи власних пунктів обміну валют по всій Україні, який має великий перелік доступних для обміну валют. Сайт містить інформацію про актуальні курси, калькулятор валют, можливість створювати запити на обмін у фізичному пункті без потреби реєстрації, а просто вказуючи номер телефону та ім'я.

Для порівняння курсової роботи з аналогом можна скористатись таблицею 1.3.3.1

Таблиця 1.3.3.1 – Порівняння з аналогом

Функціонал	Курсовий	Money24	Пояснення
	проєкт		
	(CashFlow		
	Exchange)		
Можливість	Так	Так	Обидва застосунки надають
перегляду			актуальні курси валют
актуальних			
курсів онлайн			
Можливість	Так	Так	Обидва застосунки мають
розрахунку			калькулятор, який надає

обміну валют в			можливість обраховувати обмін
калькуляторі			за актуальним курсом
Можливість	Так	Hi	Мопеу24 не передбачає
створення			створення акаунтів, що несе за
акаунта			собою певні наслідки
користувача			
Можливість	Так	Так	Функціонал реалізовано по-
створення			різному, але він наявних в обох
запитів на обмін			випадках. CashFlow Exchange
			потребує користувацького
			акаунту та верифікації пошти
			для цього, в той час як Мопеу24
			тільки номер телефона
Можливість	Так	Hi	Money24 не має можливості
переглядати			перегляду історії обмінів через
історію обмінів			відсутність особистих акаунтів
Можливість	Так	Hi	Через відсутність можливості
налаштування			створення акаунтів, даний
особистого			функціонал неможливий
акаунту			
Можливість	Так	Hi	Курсовий проєкт передбачає
відстежувати			вибір порогового значення, при
певні валюти та			зміні курсу на яке, на пошту
отримувати			користувачу приходить
сповіщення про			сповіщення, Мопеу24 не надає
їх зміни			подібного функціоналу

1.4 Аналіз вимог до програмного забезпечення

Головною функцією програмного забезпечення ϵ створення запитів на обмін валют.

Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

Користувач повинен мати можливість зареєструватися використовуючи внутрішню систему реєстрації. В результаті натиску кнопки "Sign Up" буде створено нового користувача, якщо всі дані валідні. Після процедури реєстрації користувача переадресовує на "Ноте" сторінку.

Користувач повинен мати можливість авторизуватись використовуючи систему. Кнопка Sign In відповідає за підтвердження введених даних і у випадку їх правильного вводу, користувач успішно авторизується та переадресовується на сторінку "Home".

Користувач повинен мати можливість переглядати актуальні курси валют на сторінці "Home".

Користувач повинен мати можливість обирати з якої та в яку валюту відбувається конвертація в калькуляторі натискаючи на кнопку вибору валют.

Неавторизований користувач повинен мати можливість перейти на сторінку реєстрації або на сторінку авторизації, якщо має існуючий обліковий запис.

Авторизований користувач повинен мати можливість перейти в налаштування власного облікового запису шляхом натискання "Profile".

Авторизований користувач повинен мати можливість перейти на сторінку створення угоди з обміну валют по кнопці "Start Exchange".

Користувач повинен мати можливість змінювати персональні дані ("First name", "Last name", "Email", "Password").

Користувач повинен мати можливість обирати значення зміни цікавого йому курсу, при якому про цю зміну буде йти сповіщення на пошту користувачу.

Користувач повинен мати можливість обирати валюти між якими буде проводитись обмін шляхом.

Користувач повинен мати можливість переглядати історію своїх обмінів (валюти, між якими було проведено, дату та час, статус заявки).

Користувач повинен мати можливість змінювати статус заявки на обмін з "In review" або "Take away" на "Rejected", якщо з якихось причин передумав проводити операцію.

Адміністратор системи повинен мати можливість переглядати усі створені запити та змінювати їх статус. Адміністратор може ставити 4 статуси ("Іп review" - статус ставиться по стандарту після створення та значить, що заявка очікує перевірки адміністратором. "Таке away" - статус ставиться, коли адміністратор одобрив заявку та клієнту треба прийти на фізичний пункт та виконати обмін. "Rejected" - статус ставиться за різних причин, коли обмін не може бути скоєним. "Done" - ставиться після успішного обміну вже на фізичному пункті.)

Більше функцій можна побачити на рисунку 1.4.1.

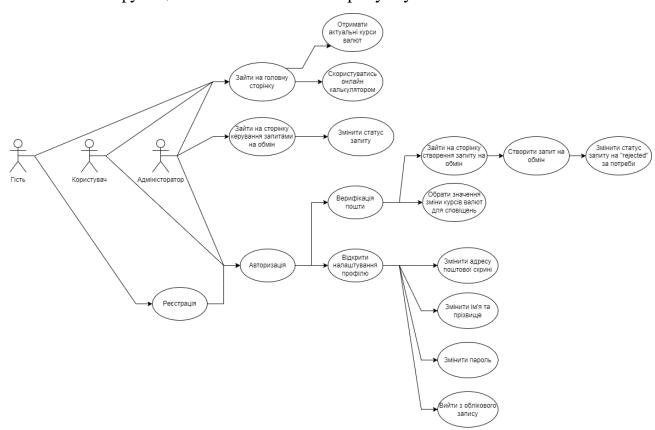


Рисунок 1.4.1 – Діаграма варіантів використання

В таблицях 1.4.1 – 1.4.11 наведені варіанти використання програмного забезпечення.

Таблиця 1.4.1 - Варіант використання UC-1

Use case name	Реєстрація користувача		
Use case ID	UC-01		
Goals	Реєстрація нового користувача в системі		
Actors	Гість (незареєстрований користувач)		
Trigger	Користувач бажає зареєструватися		
Pre-conditions	-		
Flow of Events	ts Користувач переходить на сторінку реєстрації. В поля для		
	реєстрації вводяться відповідні дані: ім'я, прізвище, пошта		
	користувача, пароль в системі. Після заповнення даних		
	користувач натискає кнопку реєстрації. Після цього користувача		
	авторизує та перенаправляє на домашню сторінку		
Extension	У випадку введення не коректних даних, з'являються помилки		
	біля відповідних полів та форма потребу ϵ виправлення		
	помилок, щоб бути підтвердженою		
Post-Condition	Створення сторінки користувача, користувач авторизований,		
	перехід на домашню сторінку		

Таблиця 1.4.2 - Варіант використання UC-2

Use case name	пате Авторизація користувача		
Use case ID	UC-02		
Goals	Авторизація користувача в існуючий обліковий запис		
Actors	Неавторизований користувач		
Trigger	Користувач бажає авторизуватись		
Pre-conditions	Обліковий запис користувача існує		
Flow of Events	Користувач переходить на сторінку авторизації. В поля для авторизації вводяться відповідні дані: пошта, пароль. Після заповнення даних користувач натискає кнопку авторизації. Після цього користувача авторизує та перенаправляє на домашню сторінку		
Extension	У випадку введення не коректних даних, з'являються помилки та форма потребує виправлення помилок, щоб бути підтвердженою.		
Post-Condition	Користувач авторизований у власний обліковий запис, перехід на домашню сторінку		

Таблиця 1.4.3 - Варіант використання UC-3

Use case name Вихід з облікового запису

Use case ID	UC-03		
Goals	Вийти з облікового запису		
Actors	Авторизований користувач		
Trigger	Користувач бажає вийти з облікового запису		
Pre-conditions	Обліковий запис користувача існує, користувач авторизований		
Flow of Events	Користувач переходить на сторінку "Profile". Користувач натискає кнопку "Sign Out". Після цього користувача викидає з облікового запису та перенаправляє на домашню сторінку		
Extension	-		
Post-Condition	Користувач не авторизований у власний обліковий запис,		
	перехід на домашню сторінку		

Таблиця 1.4.4 - Варіант використання UC-4

Use case name	Розрахунок валюти		
Use case ID	UC-04		
Goals	Обрахувати кількість валюти при можливому обміні		
Actors	Гість або авторизований користувач		
Trigger	Гість бажає порахувати кількість грошей, які він отримає у		
	випадку обміну за актуальним курсом.		
Pre-conditions	Гість на сторінці "Ноте"		
Flow of Events	Гість обирає валюту, з якої буде проводить конвертація, та		
	валюту, в яку буде проводитись. В полі для вводу "Enter		
	amount" ввести потрібну кількість грошей. Поле "You receive"		
	виводить кількість грошей після конвертації		
Extension	Користувач не може ввести не коректні дані, поле приймає		
	тільки цифри без жодних інших символів		
Post-Condition	Користувач отримав інформацію про кількість грошей, які може		
	отримати при можливій конвертації		

Таблиця 1.4.5 - Варіант використання UC-5

Use case name	Створення запиту на обмін
Use case ID	UC-05
Goals	Створити запит на обмін валюти
Actors	Авторизований користувач
Trigger	Користувач бажає обміняти валюту та створити запит для цього
Pre-conditions	Обліковий запис користувача існу ϵ , користувач авторизований
Flow of Events	Користувач вводить необхідну кількість грошей для обміну в

	калькулятор та обирає валюти для операції. Далі натискає
	кнопку підтвердження та запит створюється та додається в
	історію запитів
Extension	Користувач не може ввести не коректні дані в поля вводу
	валюти. Якщо користувач залишив поля пустими, запит не
	створиться
Post-Condition	Запит створено та додано в історію запитів

Таблиця 1.4.6 - Варіант використання UC-6

Use case name	Зміна статусу обміну користувачем		
Use case ID	UC-06		
Goals	Відмінити створений запит на обмін		
Actors	Авторизований користувач		
Trigger	Користувач бажає відмінити виконання свого запиту на обмін		
Pre-conditions	Обліковий запис користувача існує, користувач авторизований,		
	користувач має створений запит зі статусом "pending" або "take		
	away"		
Flow of Events	На сторінці "Start Exchange" в історії власних запитів		
	користувач обирає потрібний запит, який хоче відмінити. Запит		
	має статус "pending" або "take away", користувач змінює статус		
	на "rejected"		
Extension	Користувач може змінити статус тільки з "pending" або "take		
	away" і тільки на "rejected", інші варіанти не приймаються		
Post-Condition	Статус запиту змінено		

Таблиця 1.4.7 - Варіант використання UC-7

Use case name	Зміна статусу обміну адміністратором		
Use case ID	UC-07		
Goals	Змінити створений користувачем запит на обмін		
Actors	Адміністратор застосунку		
Trigger	Адміністратор бажає розгледіти заявки на обмін		
Pre-conditions	Обліковий запис користувача існує, користувач авторизований,		
	користувач має створений запит зі статусом "pending" або "take		
	away"		
Flow of Events	На сторінці "Exchange Requests" в списку всіх запитів		
	адміністратор обирає потрібний запит, який хоче розгледіти.		
	Запит має статус "pending" або "take away", адміністратор		

	змінює на "completed", "rejected", "take away" в залежності від			
	ситуації			
Extension	Адміністратор не може міняти значення, "completed" та			
	"rejected", оскільки це вже відпрацьовані запити			
Post-Condition	Статус запиту змінено			

Таблиця 1.4.8 - Варіант використання UC-8

Use case name	Зміна персональних даних				
Use case ID	UC-08				
Goals	Зміна персональних даних облікового запису користувача				
Actors	Авторизований користувач				
Trigger	Користувач бажає змінити персональні дані (ім'я та прізвище)				
Pre-conditions	Обліковий запис користувача існує, користувач авторизований				
Flow of Events	Користувач переходить на сторінку "Profile", де змінює потрібні				
	йому дані (ім'я, прізвище). Натискає кнопку підтвердження				
Extension	У випадку введення не коректних даних, з'являються помилки				
	та форма потребу ϵ виправлення помилок, щоб бути				
	підтвердженою.				
Post-Condition	Персональні дані змінені				

Таблиця 1.4.9 - Варіант використання UC-9

Use case name	Зміна пошти			
Use case ID	UC-09			
Goals	Зміна даних облікового запису користувача (пошта)			
Actors	Авторизований користувач			
Trigger	Користувач бажає змінити пошту			
Pre-conditions	Авторизований користувач			
Flow of Events	Користувач переходить на сторінку "Profile", де змінює			
	потрібне йому поле (пошта). Для зміни пошти потрібно ввести			
	дійсний пароль для підтвердження зміни. Користувач натискає			
	кнопку підтвердження. Після чого його викидає з акаунта та			
	переадресову ϵ на сторінку авторизації. Нову пошту потрібно			
	верифікувати, щоб авторизуватись з неї.			
Extension	У випадку введення не коректних даних, з'являються помилки			
	та форма потребує виправлення помилок, щоб бути			
	підтвердженою.			
Post-Condition	Користувач не авторизований у власний обліковий запис,			

Таблиця 1.4.10 - Варіант використання UC-10

Use case name	Зміна паролю		
Use case ID	UC-10		
Goals	Зміна чутливої інформації (паролю)		
Actors	Авторизований користувач		
Trigger	Користувач бажає змінити пароль		
Pre-conditions	Авторизований користувач		
Flow of Events	Користувач переходить на сторінку "Profile", де змінює		
	потрібне йому поле (пароль). Для зміни паролю потрібно		
	ввести дійсний пароль для підтвердження зміни та новий		
	пароль. Користувач натискає кнопку підтвердження		
Extension	У випадку введення не коректних даних, з'являються помилки		
	та форма потребує виправлення помилок, щоб бути		
	підтвердженою		
Post-Condition	Користувач авторизований у власний обліковий запис з новим		
	паролем, перехід на сторінку авторизації		

Таблиця 1.4.11 - Варіант використання UC-11

Use case name	Зміна значення валют для сповіщення				
Use case ID	UC-11				
Goals	Зміна мінімальних значень валют, при зміні на які буде				
	надсилатись сповіщення при оновленні курсу				
Actors	Авторизований користувач				
Trigger	Користувач бажає отримувати сповіщення про зміну курсів				
Pre-conditions	Користувач авторизований, має верифіковану пошту.				
Flow of Events	Користувач переходить на сторінку "Profile", де значення біля				
	потрібних йому валют. Якщо при оновленні курсу, він				
	зміниться на таке значення чи більше по модулю, користувач				
	отримає сповіщення на верифіковану пошту. Користувач				
	натискає кнопку підтвердження				
Extension	У випадку введення не коректних даних (коректні від 0 до 10),				
	з'являються помилки та форма потребує виправлення помилок,				
	щоб бути підтвердженою. Якщо пошта користувача не				
	верифікована, він не матиме можливість ставити якісь значення				
Post-Condition	При оновленні курсу на пошту прийшло сповіщення відповідно				

1.4.1 Розроблення функціональних вимог

Програмне забезпечення розділене на модулі. Кожен модуль має свій певний набір функцій. На рисунку 1.4.1.1 наведено загальну модель вимог, а в таблицях 1.4.1.1 — 1.4.1.17 наведений опис функціональних вимог до програмного забезпечення. Матрицю трасування вимог можна побачити на рисунку 1.4.1.2.

Опис	Назва	Пріоритет	Ризик
1. Реєстрація	FR-1	1	Високий
2. Авторизація	FR-2	1	Високий
3. Перегляд головної сторінки	FR-3	1	Високий
3.1. Перегляд актуальних курсів	FR-4		
валют			
3.2. Обрахунок обміну в калькуляторі	FR-5		
валют			
4. Створення запиту на обмін	FR-6	1	Середній
4.1. Обрахунок обміну в калькуляторі	FR-5		
валют			
4.2. Перегляд історії обмінів	FR-7		
4.3. Відміна запиту на обмін	FR-8		
5. Керування запитами на обмін з боку	FR-9	2	Високий
адміністратора			
5.1. Перегляд усіх створених запитів.	FR-10		
5.2. Зміна статусу запиту.	FR-11		
6. Зміна персональних даних	FR-12	2	Високий
6.1 Зміна імені та прізвища	FR-13		
6.2 Зміна пошти	FR-14		

6.3 Зміна паролю	FR-15
7. Вибір порогу зміни валют для	FR-16 3 Низький
сповіщення	
8. Вихід з облікового запису	FR-17 1 Низький

Рисунок 1.4.1.1 – Модель вимог у загальному вигляді

Таблиця 1.4.1.1 — Функціональна вимога FR-1

Назва	Реєстраці	Я				
Опис	Система	повинна	надавати	можливість	реєстрації	користувачеві
	шляхом в	ведення іг	мені, прізві	ища, пошти, г	іаролю.	

Таблиця 1.4.1.2 – Функціональна вимога FR-2

Назва	Авторизація
Опис	Система повинна надавати можливість авторизуватись користувачеві
	шляхом введення пошти, паролю.

Таблиця 1.4.1.3 – Функціональна вимога FR-3

Назва	Перегляд	головної с	торінки			
Опис	Система	повинна	надавати	можливість	зображати	інформацію
	головної (сторінки у	сім користу	вачам, які зах	одять на сайт	Γ.

Таблиця 1.4.1.4 — Функціональна вимога FR-4

Назва	Перегляд актуальних курсів валют
Опис	Система повинна надавати можливість відображати актуальні курси
	валют на головній сторінці.

Таблиця 1.4.1.5 – Функціональна вимога FR-5

Назва	Обрахунок обміну в калькуляторі валют
Опис	Система повинна надавати можливість розраховувати конвертацію
	валюти згідно актуального курсу.

Таблиця 1.4.1.6 – Функціональна вимога FR-6

Назва	Створенн	я запиту н	іа обмін			
Опис	Система	повинна	надавати	можливість	користувачам	створювати
	запит на о	обмін.				

Таблиця 1.4.1.7 – Функціональна вимога FR-7

Назва	Перегляд історії обмінів
Опис	Система повинна надавати можливість користувачам переглядати
	історію власних обмінів.

Таблиця 1.4.1.8 – Функціональна вимога FR-8

Назва	Відміна запиту на обмін					
Опис	Система	повинна	надавати	можливість	користувачам	відміняти
	створені	запити на о	обмін.			

Таблиця 1.4.1.9 – Функціональна вимога FR-9

Назва	Керування запитами на обмін з боку адміністратора					
Опис	Система пог	Система повинна надавати можливість адміністраторам керувати				
	усіма створеними запитами.					

Таблиця 1.4.1.10 – Функціональна вимога FR-10

Назва	Перегляд усіх створених запитів
Опис	Система повинна надавати можливість адміністраторам переглядати
	усі створені запити.

Таблиця 1.4.1.11 — Функціональна вимога FR-11

Назва	Зміна статусу запиту
Опис	Система повинна надавати можливість адміністраторам змінювати
	статус запитів.

Таблиця 1.4.1.12 – Функціональна вимога FR-12

Назва	Зміна пер	Зміна персональних даних						
Опис	Система	повинна	надавати	можливість	користувачам	змінювати		
	персонал	ьні дані.						

Таблиця 1.4.1.13 – Функціональна вимога FR-13

Назва	Зміна імені та прізвища
Опис	Система повинна надавати можливість користувачам змінювати ім'я
	та прізвище.

Таблиця 1.4.1.14 – Функціональна вимога FR-14

Назва	Зміна пог	ШТИ				
Опис	Система	повинна	надавати	можливість	користувачам	змінювати
	пошту.					

Таблиця 1.4.1.15 – Функціональна вимога FR-15

Назва	Зміна паролю					
Опис	Система	повинна	надавати	можливість	користувачам	змінювати
	пароль.					

Таблиця 1.4.1.16 – Функціональна вимога FR-16

Назва	Вибір порогу зміни валют для сповіщення						
Опис	Система повинна надавати можливість користувачам обирати						
	порогові значення для валют, про зміну яких вони хочуть отримувати						
	сповіщення.						

Таблиця 1.4.1.17 — Функціональна вимога FR-17

Назва	Вихід з об	лікового	запису				
Опис	Система	повинна	надавати	можливість	користувачам	виходити	3

облікового запису.

	FR-1	FR-2	FR-3	FR-4	FR-5	FR-6	FR-7	FR-8	FR-9	FR-10	FR-11	FR-12	FR-13	FR-14	FR-15	FR-16	FR-17
UC-1	+																
UC-2		+															
UC-3																	+
UC-4			+	+	+												
UC-5					+	+											
UC-6							+	+									
UC-7									+	+	+						
UC-8												+	+				
UC-9												+		+			
UC-10												+			+		
UC-11																+	

Рисунок 1.4.1.2 – Матриця трасування вимог

1.4.2 Розроблення нефункціональних вимог

Нефункціональні вимоги ϵ важливим аспектом продукту, які визначають його якість та характеристики.

Таблиця 1.4.2.1 – Нефункціональні вимоги

Номер	Назва	Опис
NFR-1	Сумісність	Забезпечення сумісності із браузерами Google
		Chrome, Opera, Edge, Firefox та пристроями для
		зручного використання користувачами.
NFR-2	Надійність	Система повинна бути стійкою до помилок та
		забезпечувати контроль введення інформації та
		захист від некоректних дій користувача. При
		виникненні помилок, система повинна надавати
		зрозумілі та інформативні повідомлення
		користувачеві.
NFR-3	Безпека	Система повинна гарантувати конфіденційність
		інформації та захищати несанкціонованого
		доступу під час операцій.

NFR-4	Зручність	Забезпечення зручного, інтуїтивно зрозумілого
	інтерфейсу	та привабливого інтерфейсу для користувачів
		різних категорій.

1.5 Постановка задачі

В результаті розробки застосунку стоять цілі отримати гарно оптимізований застосунок з мінімалістичним та інтуїтивним інтерфейсом, який забезпечить найкращий досвід користування. Веб-застосунок повинен оптимізувати процес комунікації між потенційним клієнтом та пунктом обміну, а точніше максимально зменшити фізичний контакт між ними. Основна задача застосунку дати можливість людині передивлятись актуальні курси онлайн, створювати запити на обміни потрібних валют, відстежувати зміни потрібних курсів.

В ролі джерела даних буде використовуватись зовнішня АРІ, з якої щоденно автоматично будуть діставатись актуальні курси. В ролі технічного рішення для реалізації серверної частини використаємо хмарну платформу Firebase, яка надає весь потрібний функціонал у вигляді хмарного сховища Firestore, можливість написання власних функцій за допомогою Cloud Functions. Як інструменти для клієнтської частини використаємо JavaScript та його бібліотеку React, яка допомагає створювати функціональні компоненти для інтерфейсу користувачів.

Проект також має виконувати нефункціональні вимоги: сумісність, надійність, безпека, зручність інтерфейсу. Все це потрібно для забезпечення максимального задоволення користувача від процесу використання сайту.

Можливість отримувати сповіщення для обраних валют сильно допоможе користувачам, які серйозно займаються обміном валют, це допоможе їм першими дізнаватись про певні зміни в курсі та оперативно реагувати на них.

Висновки до розділу

В даному розділі було виконано детальний аналіз предметної області, визначено мету та ціль для розробки подібно додатку, обраного технології, які найкраще відповідають вимогам проекту та оптимізують час виконання завдання. Проведено аналіз аналогічних додатків та їх особливостей, звідси було отримано причини для розробки власного застосунку. Проведено детальний аналіз допоміжних технологій та їх аналогів, де описано, чому було обрано ту чи іншу технологію. Проведено детальне порівняння з локально відомим застосунок Мопеу24 та виявлено його недоліки.

Чітко визначено функціональні вимоги з допоміжними діаграмами та таблицями, розписано детальний процес виконання вимог. Також визначено нефункціональні вимоги, які гарно впливають на стабільність роботи застосунку та користувацький досвід.

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Для моделювання бізнес-процесів було вибрано BPMN ^[8], який є визнаним стандартом у цій області. BPMN надає уніфікований інструмент для загального розуміння, аналізу та оптимізації бізнес-процесів у корпоративному середовищі. Використання цього стандарту дозволяє застосовувати символи та правила для створення графічних представлень процесів, полегшуючи взаємодію між бізнес-аналітиками та розробниками програмного забезпечення.

Переваги використання ВРМN: визнаний як міжнародний стандарт для моделювання бізнес-процесів, що забезпечує єдиний набір термінів та концепцій. ВРМN використовує графічні символи та правила, що полегшує створення інтуїтивно зрозумілих графічних представлень бізнес-процесів. ВРМN є універсальним інструментом, придатним як для стратегічного моделювання, так і для деталей оперативних процесів. Застосування ВРМN дозволяє легше управляти і співпрацювати між бізнес-аналітиками та розробниками. Це сприяє створенню спільного розуміння між учасниками проєкту та узгодженню різних команд.

Отже, висунемо головні бізнес-процеси:

- реєстрація;
- авторизація;
- перегляд курсів валют та користування калькулятором валют;
- зміна персональних даних;
- створення запитів на обмін валют;
- керування запитами адміністратором;
- розсилання сповіщень про зміну курсу;

Для опису бізнес процесу програмного забезпечення використовуються BPMN моделі (рисунок 2.1.1 та рисунок 2.1.2).

Опис послідовності реєстрації користувача:

- користувач переходить на сторінку реєстрації;
- користувач заповнює поля реєстрації;
- якщо введені поля не відповідають шаблону заповнення на клієнтській стороні, відповідні поля підсвічуються помилкою.

Опис послідовності реєстрації користувача:

- користувач переходить на сторінку авторизації;
- користувач заповнює поля реєстрації;
- якщо введені поля не відповідають шаблону заповнення на клієнтській стороні, відповідні поля підсвічуються помилкою.

Опис послідовності перегляду курсів валют та користування калькулятором валют:

- користувач переходить на головну сторінку;
- користувач отримує інформацію про актуальні курси з таблички курсів;
 - користувач обирає потрібні валюти та вводить кількість грошей;
 - система вираховує та показує кількість грошей після обміну.

Опис послідовності зміни персональних даних:

- користувач переходить на сторінку профіля;
- користувач змінює дані в потрібних йому полях;
- користувач бачить змінені дані;
- якщо введені поля не відповідають шаблону заповнення на клієнтській стороні, відповідні поля підсвічуються помилкою.

Опис послідовності створення запитів на обмін валют:

- користувач переходить на сторінку початку обміну;
- користувач обирає валюти для обміну та вводить кількість грошей;

- користувач бачить кількість грошей після обміну та підтверджує обмін;
 - користувач бачить створений запит в історії своїх запитів.

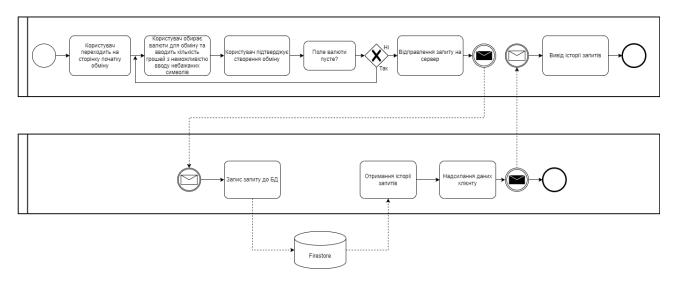


Рисунок 2.1.1 – Діаграма бізнес-процесу створення запитів на обмін.

Опис послідовності керування запитами адміністратором:

- адміністратор переходить на сторінку запитів на обмін;
- адміністратор бачить усі запити з їх детальною інформацією та станами;
 - адміністратор змінює статус заявки на потрібний.
 - адміністратор бачить заявку з оновленим статусом;

Опис послідовності розсилання сповіщень про зміну курсу:

- користувач переходить на сторінку профіля;
- користувач змінює мінімальне значення при якому прийде повідомлення на пошту по певним валютам;
 - користувач чекає на оновлення курсу в системі;
- при оновленні курсу користувач отримує сповіщення, якщо обрані валюти змінились на заданий курс.

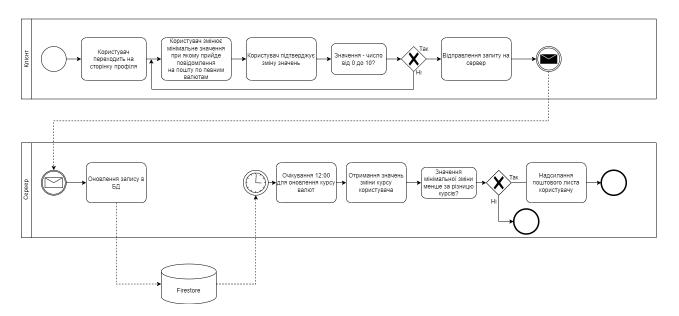


Рисунок 2.1.2 – Діаграма бізнес-процесу розсилання сповіщень про зміну курсу.

2.2 Архітектура програмного забезпечення

Клієнт-серверна архітектура^[9] — це підхід до побудови програмних систем, при якому функціонал системи розподіляється між клієнтською стороною та серверною. Клієнтський сторона в свою чергу відповідає за користувацький інтерфейс, а серверний компонент виконує обробку даних та надає потрібні ресурси. Цей підхід сприяє зменшенню завантаження на клієнтські пристрої та централізованому управлінню ресурсами на сервері.

В нашому випадку клієнт-серверна архітектура представлена у вигляді клієнтської частини, яка використовує такі технології, як React, який є ефективною бібліотекою для побудови користувацьких інтерфейсів, його роль полягає в наданні інтерфейсу для взаємодії з додатком та відправленні запитів на серверну частину для отримання даних для оновлення інтерфейсу, а сама серверна частина представлена у вигляді Firebase сервісів та зовнішньої АРІ для отримання курсів валют, тому даний веб-застосунок можна ще класифікувати як мікросервісний. Наприклад, клієнтська частина потребує відображення актуальних курсів валют, для чого посилається запит до зовнішньої АРІ, яка вже повертає потрібні дані. Також як приклад, клієнт хоче створити запит на обмін валют, для цього після створення запиту, його дані мають записатись в БД, знову ж доводиться звертатись до серверу.

3 переваг даного підходу можна відмітити можливість централізовано керувати та забезпечувати консистентність даних, також важливим пунктом ϵ зниження навантаження на клієнт, тому клієнтська частина може фокусуватись на інтерфейсі та користувацькому досвіді.

Представлення архітектури даного проекту можна побачити на рис. 2.2.1.

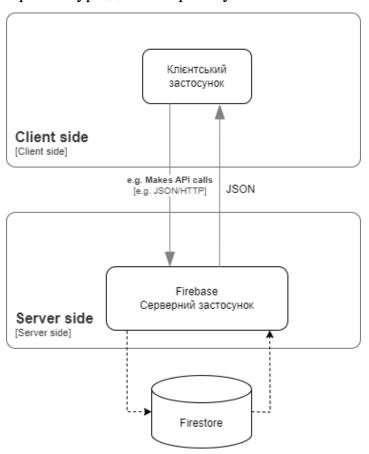


Рисунок 2.2.1 – Загальна архітектура веб-застосунку

Мікросервісна архітектура^[10] — це підхід до розробки програмного забезпечення, при якому програмна система розбивається на невеликі і незалежні компоненти під назвою мікросервіси. Кожен мікросервіс виконує певний функціонал та взаємодіє з іншими мікросервісами через АРІ. Такий підхід сприяє гнучкості, масштабованості та легкості розвитку програмних систем.

В даному застосунку мікросервісна архітектура представлена у вигляді сервісів Firebase Functions, незалежних всередині (Firestore, Cloud Authentication) зовнішньої API вигляді мікросервісу. та V використовується для зберігання даних у вигляді документо-орієнтованої БД,

Сloud Functions створює можливість створювати власні функції на Node.js^[11], завдяки яким реалізовано процес збору даних з зовнішньої API та розсилки сповіщень на пошту. Authentication використовується як сервіс для авторизації та збереження даних авторизації, які незалежні від даних користувача в Firestore. Клієнтська частина звертається окремо до різних мікросервісів за потреб, якщо потрібно оновити курси валют, потрібно звертатись до зовнішньої API, якщо потрібно авторизуватись, то потрібно звертатись до мікросервісу аутентифікації.

3 переваг даної архітектури ε гнучкість та незалежність, де кожен мікросервіс розглядається як окремий компонент, який працю ε незалежно від інших мікросервісів, також це спрощу ε розвиток та розгортання додатку через декомпозицію.

Бізнес-логіка реалізована в серверній частині, його компоненти можна побачити на рис. 2.2.2.

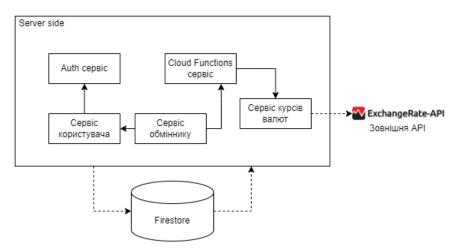


Рисунок 2.2.2 – діаграма бізнес логіки застосунку.

2.3 Конструювання програмного забезпечення

Веб-застосунок, який реалізується у межах даної курсової роботи, не вимагає впровадження конкретних алгоритмів чи специфічних технічних рішень. Було обрано $NoSQL^{[12]}$ базу даних, оскільки дані не потребують складної структури, вона реалізує простий дизайн схеми БД, значно спрощує горизонтальне масштабування на кластери машин і має тонкий контроль над доступністю, через відмінну структуру даних потрібні операції виконуються

швидше, ніж було б на реляційній БД. База даних призначена для зберігання користувачів, їх запитів на обмін та актуальних курсів валют.

Опис таблиць бази даних наведено у таблицях 2.3.1-2.3.3. ER модель бази даних наведена на рисунку 2.3.1.

Таблиця 2.3.1 – Опис колекції users

Таблиця	Назва поля	Тип даних	Опис			
users	id	Serial	id документу, ідентифікаційний номер користувача			
	firstName	String	ім'я користувача			
	lastName	String	прізвище користувача			
	email	String	електрона адреса користувача			
	isAdmin	Boolean	наявність особливих прав у користувача			
	currencyDiff	Object	об'єкт, який зберігає інформацію про зміну валют для трігеру сповіщень користувачу			
	currencyDiff. EUR	Number	значення зміни курсу євро для приходу сповіщення на пошту			
	currencyDiff. GBP	Number	значення зміни курсу фунта для приходу сповіщення на пошту			
	currencyDiff. PLN	Number	значення зміни курсу злотих для приходу сповіщення на пошту			
	currencyDiff. USD	Number	значення зміни курсу долара для приходу сповіщення на пошту			

Таблиця 2.3.2 – Опис колекції rates

Таблиця	Назва поля	Тип даних	Опис
rates	EUR	Number	значення курсу євро
	GBP	Number	значення курсу фунта
	PLN	Number	значення курсу злотих
	USD	Number	значення курсу
			американського долара

Таблиця 2.3.3 – Опис колекції requests

Таблиця	Назва поля	Тип даних	Опис
requests	id	Serial	ідентифікаційний номер запиту, іd документа
	userId	Serial	ідентифікаційний номер користувача, якому належить запит на обмін
	timestamp	Object	час створення запиту на обмін
	amount	Number	кількість валюти яку віддає користувач
	convertedAmount	Number	кількість валюти яку отримує користувач
	sourceCurrency	String	валюта з якої відбувається конвертація

Продовження таблиці 2.3.3

targetCurrency	String	валюта	в яку
		відбуваєт	ься
		конвертал	ція
status	String	статус	обробки
		запиту на	обмін

На рисунку 2.3.1 зображено модель NoSQL БД:

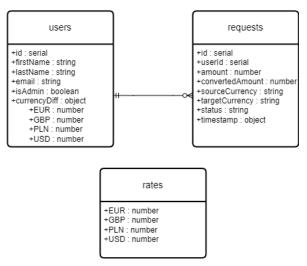


Рисунок 2.3.1 – модель нереляційної БД.

Опис утиліт, бібліотек та іншого стороннього програмного забезпечення, що використовується у розробці наведено в таблиці 2.3.4.

Таблиця 2.3.4 – Опис утиліт

№	Назва утиліти	Опис застосування
п/п		
1	Visual Studio	Головне середовище розробки програмного
	Code	забезпечення.
2	React	JavaScript бібліотека для створення інтерфейсів
		користувача.
3	Firebase	Платформа розробки мобільних та веб застосунків, що
		надає доступ до хмарного сховища.

Продовження таблиці 2.3.4

4	Git	Система контролю версій для зручного управління
		змінами проекту.
5	Redux	JavaScript бібліотека для керування станом застосунку.

2.4 Аналіз безпеки даних

Аналіз безпеки включає в себе оцінку вразливостей ПЗ та застосування основних заходів безпеки ПЗ для запобігання подібних атак:

- Атака брутфорс^[13]: спроба злому пароля шляхом перебору. Завдяки вбудованим заходам безпеки Firebase спроба атаки брутфорс запобігається обмеженням кількості невдалих спроб авторизації.
- Міжсайтовий скриптинг (XSS)^[14] внесення зловмисного коду на вебсторінки користувачів. Firebase використовує власні механізми для захисту від міжсайтового скриптингу. Це включає в себе валідацію та екранування введених даних перед їх збереженням та обробкою.
- SQL-injection^[15] заснований на впроваджені в запит довільного SQLкоду. Для запобігання використовуються параметризовані запити, екранування введених даних, використання ORM, перевірка та обробка введених даних.
- Захист від атак на сесії використовуються різноманітні механізми,
 такі як обмежений термін дії токенів.
- Атака посередника (МІТМ)^[16] втручання третього суб'єкта між зв'язком клієнта з сервером, що імітує їх зв'язок. Запобігається за допомогою SSL/TLS^[17] шифрування між клієнтом та сервером, використовується заголовок HSTS для заборони незашифрованих з'єднань.

Висновки до розділу

У даному розділі було проведено детальний аналіз бізнес-процесів, використано BPMN моделі для моделювання бізнес-процесів. Для двох складних процесів було імплементовано модель.

Було обґрунтовано вибір паттерну архітектури для розробки ПЗ, перераховано переваги обраних фігур та наведено чіткі схеми архітектури. Застосунок має клієнт-серверну мікросервісну архітектуру, оскільки функціонал в системі має розділятись на клієнтську та сервісну частину для делегування функціоналу та покращення оптимізації, також Firebase.

Використання NoSQL Firestore надає гнучкість та масштабованість у зберіганні та отриманні даних. Модель даних ретельно розроблена, враховуючи специфіку додатку та його потреби.

В детальному аналізі безпеки пояснено способи запобігання загроз брутфорс, XSS, SQL injection та МІТМ. Для їх запобігання впроваджені відповідні заходи безпеки, включаючи використання тимчасового блокування облікового запису при багатьох невдалих спробах авторизації, валідацію та екранування даних, використання HTTPS протоколу.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Аналіз якості програмного забезпечення є ключовим етапом у життєвому циклі розробки та експлуатації. Це процес, що визначає, наскільки програмне забезпечення відповідає вимогам, стандартам та очікуванням користувачів. В якості інструменту статичного аналізу коду було під'єднано ESLint для виявлення проблемних шаблонів. Також було проведено аналіз коду за допомогою онлайн аналізатора Codacy^[18] (рисунок 3.2.1).

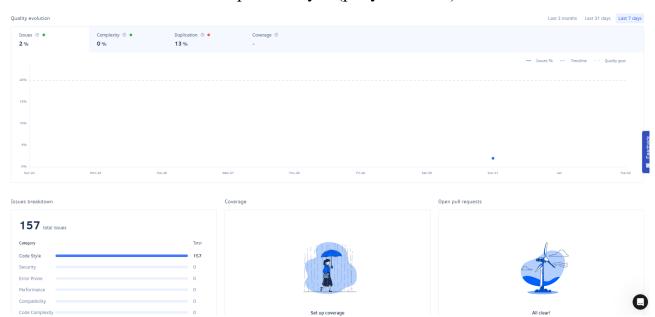


Рисунок 3.2.1 – результати онлайн аналізу коду

З виявлених проблем було піймано тільки мінорні проблеми, пов'язані зі стилем коду, які ніяк не впливають на ефективність та надійність коду.

3.2 Опис процесів тестування

Тестування є важливим аспектом в налагодженні роботи ПЗ, оскільки воно дозволяє оцінити якість готового продукту, перевірити виконання зазначених вимог та знайти можливі вразливості, що дає можливість виявити серйозні проблеми перед випуском застосунку в маси, також дозволяє

ефективно використовувати часові та людські ресурси, маючи повністю налагоджений процес тестування.

Було виконане мануальне тестування програмного забезпечення, опис відповідних тестів наведено у таблицях 3.2.1 - 3.2.10.

Таблиця 3.2.1 – Тест 1.1

Тест	Реєстрація користувача
Модуль	Реєстрація користувача
Номер тесту	1.1
Початковий стан	Користувач знаходиться на сторінці реєстрації
системи	
Вхідні данні	Ім'я, прізвище, електронна пошта, пароль
Опис	У відповідні поля вводяться: електронна пошта в
проведення	правильному форматі, яка до цього не була
тесту	зареєстрована в системі, пароль від 6 до 16 символів,
	який може містити тільки цифри, літери та спеціальні
	символи.
Очікуваний	Реєстрація проходить успішно, користувач додається у
результат	систему, автоматично авторизується і перенаправляється
	на домашню сторінку.
Фактичний	Реєстрація проходить успішно, користувач додається у
результат	систему, автоматично авторизується і перенаправляється
	на домашню сторінку.

Таблиця 3.2.2 – Тест 2.1

Тест	Авторизація користувача
Модуль	Авторизація користувача
Номер тесту	2.1
Початковий стан	Користувач знаходиться на сторінці авторизації
системи	

Вхідні данні	Електронна пошта, пароль
Опис	Користувач, який має активний обліковий запис, вводить
проведення	свою електронну пошту, до якої прив'язаний обліковий
тесту	запис, та пароль від даного облікового запису
Очікуваний	Авторизація проходить успішно, користувач
результат	авторизується в системі і перенаправляється на домашню
	сторінку.
Фактичний	Авторизація проходить успішно, користувач
результат	авторизується в системі і перенаправляється на домашню
	сторінку.

Таблиця 3.2.3 – Тест 3.1

Тест	Розрахунок валюти в калькуляторі
Модуль	Калькулятор валют
Номер тесту	3.1
Початковий стан	Користувач знаходиться на домашній сторінці
системи	
Вхідні данні	Валюти для конвертації, значення валюти
Опис	Користувач обирає валюти для конвертації та вводить
проведення	коректне значення валюти в поле "Enter amount"
тесту	
Очікуваний	Конвертація проходить успішно, користувач отримує
результат	результати конвертації.
Фактичний	Конвертація проходить успішно, користувач отримує
результат	результати конвертації.

Таблиця 3.2.4 – Тест 4.1

Тест	Створення запитну на обмін
Модуль	Запит на обмін

Номер тесту	4.1
Початковий стан	Користувач знаходиться на сторінці створення запиту
системи	
Вхідні данні	Валюти для конвертації, значення валюти
Опис	Користувач обирає валюти для конвертації та вводить
проведення	коректне значення валюти в поле "Enter amount".
тесту	Натискає кнопку підтвердження
Очікуваний	Запит успішно створюється та відображається в історії
результат	запитів.
Фактичний	Запит успішно створюється та відображається в історії
результат	запитів.

Таблиця 3.2.5 – **Тест** 4.2

Тест	Відміна запиту на обмін
Модуль	Запит на обмін
Номер тесту	4.2
Початковий стан	Користувач знаходиться на сторінці створення запиту
системи	
Вхідні данні	Новий статус запиту
Опис	Користувач обирає потрібний запит та в полі статусу
проведення	змінює статус з "pending" на "rejected".
тесту	
Очікуваний	Статус запиту успішно змінюється.
результат	
Фактичний	Статус запиту успішно змінюється.
результат	

Таблиця 3.2.6 – Тест 4.3

Тест	Зміна статусу запиту на обмін адміністратором
------	---

Модуль	Запит на обмін
Номер тесту	4.3
Початковий стан	Адміністратор знаходиться на сторінці всіх заявок на
системи	обмін
Вхідні данні	Новий статус запиту
Опис	Адміністратор обирає потрібний запит та в полі статусу
проведення	змінює статус з "take away" на "completed"
тесту	
Очікуваний	Статус запиту успішно змінюється.
результат	
Фактичний	Статус запиту успішно змінюється.
результат	

Таблиця 3.2.7 – Тест 5.1

Тест	Зміна мінімального значення зміни курсу, при зміні на
	яке користувач отримує сповіщення на пошту
Модуль	Зміна інформації профіля
Номер тесту	5.1
Початковий стан	Користувач знаходиться на сторінці профіля
системи	
Вхідні данні	Значення зміни курсу
Опис	Користувач вводить порогові значення зміни курсу, при
проведення	зміні на які приходить лист на пошту, для USD = 0, всі
тесту	інші поля залишає пустими.
Очікуваний	При оновленні курсу на пошту прийшов лист зі
результат	сповіщення про зміну курсу долара.
Фактичний	При оновленні курсу на пошту прийшов лист зі
результат	сповіщення про зміну курсу долара.

Таблиця 3.2.8 – **Тест** 5.2

Тест	Зміна імені та прізвища в інформації профіля
Модуль	Зміна інформації профіля
Номер тесту	5.2
Початковий стан	Користувач знаходиться на сторінці профіля
системи	
Вхідні данні	Ім'я, прізвище
Опис	Користувач вводить нові ім'я та прізвище замість старих.
проведення	Нові дані починаються з великою літери, містять тільки
тесту	літери без пробілів.
Очікуваний	Ім'я та прізвище користувача успішно змінені.
результат	
Фактичний	Ім'я та прізвище користувача успішно змінені.
результат	

Таблиця 3.2.9 – Тест 5.3

Тест	Зміна адреси пошти
Модуль	Зміна інформації профіля
Номер тесту	5.3
Початковий стан	Користувач знаходиться на сторінці профіля
системи	
Вхідні данні	Поштова адреса, пароль
Опис	Користувач вводить нову поштову адресу замість старої
проведення	та актуальний пароль для підтвердження зміни.
тесту	Підтверджує нову поштову адресу по посиланню на
	новій пошті. Авторизується з новою поштою
Очікуваний	Користувача успішно авторизовано з новою поштою
результат	
Фактичний	Користувача успішно авторизовано з новою поштою
результат	

Таблиця 3.2.10 – Тест 6.1

Тест	Блокування доступу до деяких можливостей
Модуль	Верифікація пошти
Номер тесту	6.1
Початковий стан	Користувач знаходиться на домашній сторінці
системи	
Вхідні данні	Пошта користувача не верифікована
Опис проведення	Користувач намагається перейти на сторінку
тесту	створення запитів.
Очікуваний	Система не пускає користувача на сторінку створення
результат	запитів та виводить помилку.
Фактичний	Система не пускає користувача на сторінку створення
результат	запитів та виводить помилку.

3.3 Опис контрольного прикладу

В даному розділі будемо відтворювати взаємодію користувача з вебзастосунком. Тестування було проведено на комп'ютері, який відповідає рекомендованим вимогам в браузері Google Chrome.

Відтворимо сценарій, де користувач бажає зареєструватись або авторизуватись в системі. Введемо коректні дані та подивимось, як програма відпрацює.

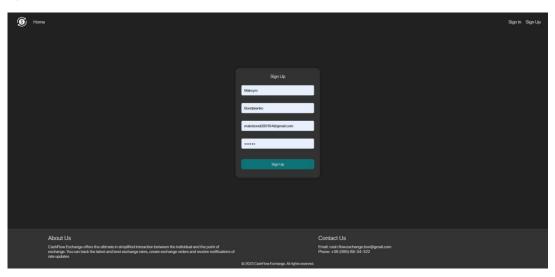


Рисунок 3.3.1 – Відтворення процесу реєстрації

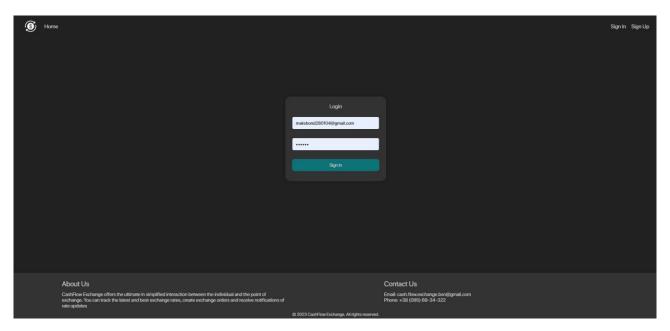


Рисунок 3.3.2 – Відтворення процесу авторизації

В обох випадках користувача буде перекинути на головну сторінку (рисунок 3.3.3) та авторизовано. З головної сторінки можна перевірити функціонал калькулятору валют, відтворимо ситуацію розрахунку в калькуляторі.

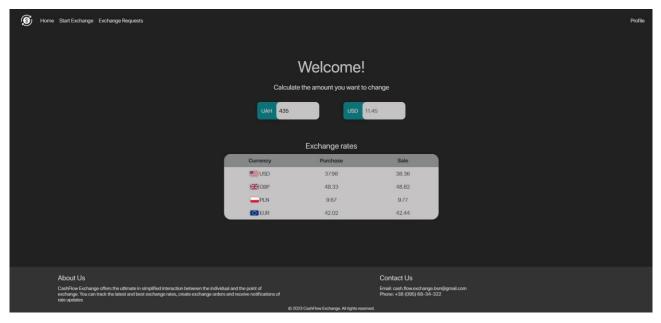


Рисунок 3.3.3 – Головна сторінка (для адміністратора)

Калькулятор відпрацював коректно, спробуємо перейти на сторінку створення запиту на обмін (рисунок 3.3.4) та спробуємо створити запит та подивитись на нього в історії

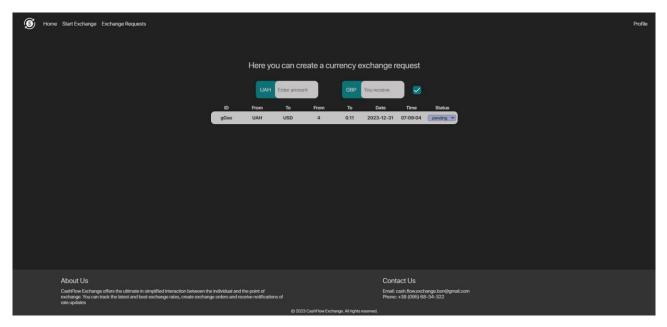


Рисунок 3.3.4 – Сторінка створення запиту на обмін

Спробуємо відмінити запит на створення обравши в пункті статус варіант "rejected"

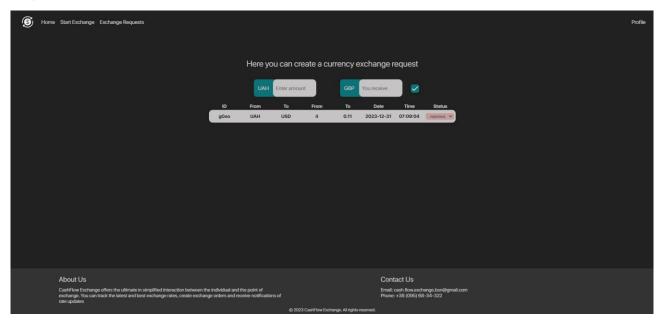


Рисунок 3.3.5 – Запит зі зміненим статусом.

Далі перейдемо на сторінку адміністратора для керування запитами на обмін (рисунок 3.3.6).



Рисунок 3.3.6 – Адміністраторська сторінка для керування створеними запитами.

Тепер спробуємо змінити статус запиту з UD akUa на "completed" (рисунок 3.3.7).

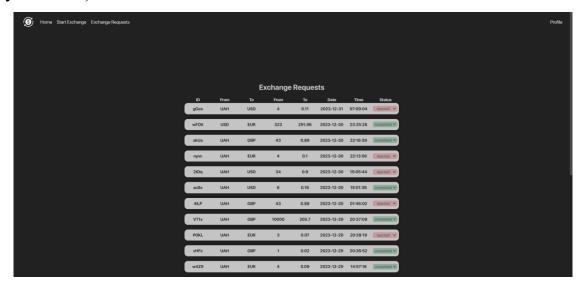


Рисунок 3.3.7 — Статус потрібного запиту змінено Тепер перейдемо на сторінку профіля (рисунок 3.3.8)

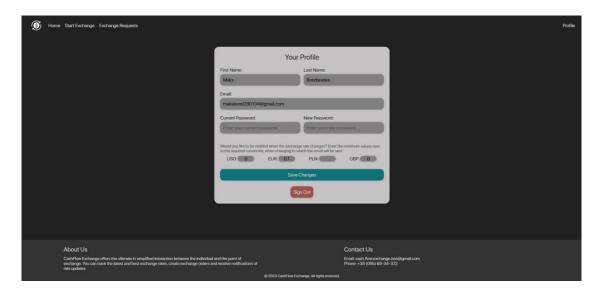


Рисунок 3.3.8 – Сторінка профіля.

Тепер спробуємо змінити ім'я, прізвище та пароль (рисунок 3.3.9)



Рисунок 3.3.9 – Ім'я, прізвище та пароль змінено.

Ну і вийдемо з акаунту шляхом натискання кнопки "Sign out" на сторінці профіля, після чого користувача переадресовує на головну сторінку.

Висновки до розділу

У даному розділі було проведено детальний аналіз якості ПЗ за допомогою онлайн аналізаторів, який не виявив серйозних проблем в коді, тільки мінорні проблеми в стилі коду, також на якість коду при написанні позитивну вплинув інструмент статичного аналізу ESLint^[19]. Було описано та виконано тестування застосунку, в результаті якого помилок в роботі ПЗ не було помічено. Було повністю описано та проведено використання більшості

функціоналу ПЗ на комп'ютері, що повністю відповідає вимогам, тестування дало позитивні результати. Отже, застосунок був повністю протестований та готовий до використання.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Даний проект буде розгорнуто на платформі Vercel^[20]. Потрібно буде розгорнути тільки клієнтську частину, оскільки серверна частина за допомогою Firebase вже розгорнута у хмарі та функціонує без додаткових для цього зусиль.

Покрокова інструкція:

- 1. створити обліковий запис Vercel;
- 2. розмістити свій проект у гілці deploy в GitHub репозиторії;
- 3. у вкладці "Import Git Repository" потрібно під'єднати репозиторій з проектом;
- 4. в конфігурації проєкту потрібно задати ім'я проєкту, значення "Vite" у виборі "Framework Preset", в "Root Directory" вказати './frontend';
- 5. у вкладці "Environment Variables" треба задати всі змінні середовища з .env.local файлу проекту;
- 6. натиснути кнопку "Deploy";
- 7. якщо з'явились якісь проблеми, то в створеному розгортанні треба вказати гілку "deploy" в налаштуваннях та перевірити пункт "Root Directory".
- 8. перейти по посиланню на розгорнутий проект (рисунок 4.1.1)

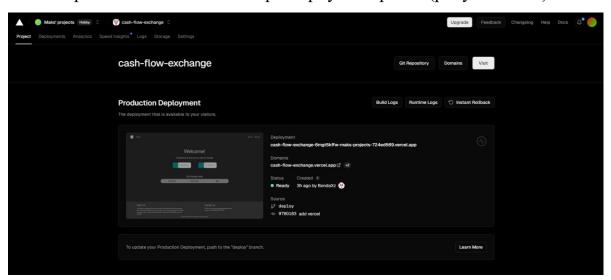


Рисунок 4.1.1 – Вигляд проекту після розгортання в інтерфейсі Vercel

Для подальшої розробки зручніше використовувати локальне розгортання, тож в термінали потрібно перейти до папки з проектом і прописати почергово: "npm install" та "npm run dev". Після чого з'явиться посилання на локально розгорнутий проект.

4.2 Підтримка програмного забезпечення

Остання версія застосунку зберігається на GitHub репозиторії застосунку, там можна отримувати всі оновлення програми, для цього достатньо клонувати застосунок собі на комп'ютер.

Висновки до розділу

В даному розділі було детально описано повний процес розгортання додатку, щоб кожен бажаючий міг розгорнути даний застосунок на платформі Vercel або локально на своєму комп'ютері, щоб можна було вносити зміни в код та дивитись ці зміни моментально, що дозволяє робити Vite.

ВИСНОВКИ

В ході виконання курсової роботи була проведена інтенсивна робота над проектуванням та розробкою веб-застосунку для підтримки роботи пункту обміну валют. Застосовані сучасні технології та архітектурні підходи дозволили досягти високого рівня ефективності та гнучкості додатку.

Проект доступний для впровадження будь-яким фізичним пунктом обміну валют, який бажає оптимізувати свою роботу завдяки сучасним технологіям. Даний додаток був би корисним для будь-якого обмінника, який веде активну роботу на ринку та має велику кількість клієнтів.

Отримані результати курсової роботи вважаються задовільними та відповідають сучасному рівню наукових і технічних знань в області веброзробки та інформаційних технологій. Використання сучасних технологій, таких як React для клієнтської частини та Firebase для серверної, свідчить про актуальність обраного стеку та відповідність його сучасним тенденціям у розробці веб-додатків.

Використання BPMN для моделювання бізнес-процесів дозволило стандартизувати термінологію та концепції, спрощуючи спільне розуміння команди розробників та бізнес-аналітиків.

Досліджувати дану тематику більш широко вважаю важливим, оскільки обмін валют ще стосується цифрових валют (криптовалют), що досі ϵ новим напрямком у сфері фінансів.

В якості середовища розробки обрано Visual Studio Code, який давно зарекомендував себе, як якісний продукт для розробки подібних застосунків.

Огляд безпекових питань, таких як запобігання брутфорс, XSS, SQL injection, MITM, демонструє уважний підхід до аспектів безпеки та відповідність сучасним стандартам забезпечення інформаційної безпеки.

Використання NoSQL бази даних Firestore вказує на усвідомлення сучасних підходів до зберігання та обробки даних у розподілених системах. Використання зовнішньої АРІ для отримання актуальних курсів валют

відображає вміння інтегрувати зовнішні ресурси для забезпечення актуальності та коректності інформації.

Усі поставлені задачі було успішно виконано та реалізовано у вебзастосунку. Після реалізації застосунок було протестовано на різних пристроях з різними технічними та програмними характеристиками, щоб переконатись в коректній роботі застосунку незалежно від різних умов.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Currency Exchange: Definition, How It Works, and Where to Find It [Електронний ресурс] https://www.investopedia.com/terms/c/currency-exchange.asp.
- 2) Гроші як сміття: 6 світових прикладів гіперінфляції [Електронний ресурс]

 https://loyer.com.ua/uk/groshi-yak-smittya-6-svitovyh-prykladiv-giperinflyacziyi/
- 3) Firebase Documentation [Електронний ресурс] https://firebase.google.com/docs.
- 4) React [Електронний ресурс] https://react.dev/
- 5) JavaScript [Електронний ресурс] https://developer.mozilla.org/ru/docs/Web/JavaScript
- 6) Vite [Електронний ресурс] https://vitejs.dev/guide/why
- 7) React Redux [Електронний ресурс] https://react-redux.js.org/
- 8) What is Business Process Modeling Notation [Електронний ресурс] https://www.lucidchart.com/pages/bpmn
- 9) What is Client Server Architecture? [Електронний ресурс] https://intellipaat.com/blog/what-is-client-server-architecture/
- 10) Microservice architecture style [Електронний ресурс] https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices
- 11) Node.js Guides [Електронний ресурс] https://nodejs.org/en/guides
- 12) What is a NoSQL database? [Електронний ресурс] https://www.ibm.com/topics/nosql-databases
- 13) What Is A Brute Force Attack? [Електронний ресурс] https://www.fortinet.com/resources/cyberglossary/brute-force-attack
- 14) Cross Site Scripting (XSS) [Електронний ресурс] https://owasp.org/www-community/attacks/xss/
- 15) SQL injection [Електронний ресурс] https://portswigger.net/web-security/sql-injection

- 16) Man in the middle (MITM) attack [Електронний ресурс] https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/
- 17) What Is An SSL/TLS Certificate? [Електронний ресурс] https://aws.amazon.com/what-is/ssl-certificate/
- 18) Codacy [Електронний ресурс] https://www.codacy.com/
- 19) ESLint [Електронний ресурс] https://eslint.org/
- 20) Vercel [Електронний ресурс] https://vercel.com/docs/deployments/overview