

КВАЛІФІКАЦІЙНА РОБОТА

Група ІІЗс-20
Твердохліб М. Я.

2024

ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА

Факультет суспільних та прикладних наук

Кафедра інформаційних технологій

на правах рукопису

Твердохліб Максим Ярославович

УДК 004.738

**Створення онлайн-платформи для обміну валют та відстеження
валютних курсів**

Спеціальність 121 – «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття кваліфікації бакалавра

Нормоконтроль

Студент

_____ Стисло О.В.

(підпис, дата, розшифрування підпису)

_____ Твердохліб М.Я.

(підпис, дата, розшифрування підпису)

Допускається до захисту

Керівник роботи

Завідувач кафедри

_____ к.т.н., доц. Ващишак С.П.

(підпис, дата, розшифрування підпису)

_____ к.ф.-м.н., доц. Бойчук А.М.

(підпис, дата, розшифрування підпису)

КОНСУЛЬТАНТИ РОЗДІЛІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Розділ	Консультант (прізвище, ініціали та посада)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Огляд існуючих аналогів	28.03.2024	Виконано
2.	Проектування та розробка прототипу	17.04.2024	Виконано
3.	Розробка дизайну веб-сайту	25.04.2024	Виконано
4.	Проектування та розробка веб-сайту	01.05.2024	Виконано
5.	Оформлення пояснювальної записки	15.05.2024	Виконано
6.	Оформлення графічного матеріалу та підготовка до захисту роботи	28.05.2024	Виконано

Студент

Твердохліб М.Я.

_____ (підпис)

_____ (прізвище та ініціали)

Керівник роботи

Бойчук А.М.

_____ (підпис)

_____ (прізвище та ініціали)

Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Сторінка	Опис графічного матеріалу	Сторінка	Опис графічного матеріалу
14	Головна сторінка сайту	39	Мокап обміну валют
15	Мобільна версія додатку (емулятор)	40	Мокап сторінки перегляду графіків
16	Головна сторінка сайту	40	Мокап сторінки авторизації
17	Мобільна версія додатку (емулятор)	41	Мокап сторінки реєстрації
33	Загальна структура архітектури	42	Головна сторінка сайту
35	Діаграма архітектури платформи	43	Сторінка обміну валюти
37	Діаграма архітектури бази даних	45	Взаємодія DOM
38	Мокап головної сторінки		

АНОТАЦІЯ

У кваліфікаційній роботі представлено розробку онлайн-платформи для обміну валют та відстеження валютних курсів. Метою проекту було створення зручної та безпечної платформи, яка дозволяє користувачам здійснювати обмін валют, відслідковувати зміни валютних курсів у реальному часі та резервувати валюту в фізичних пунктах обміну.

Робота включає аналіз сучасних фінансових ринків та існуючих технологій обміну валют, огляд існуючих онлайн-платформ для обміну валют, а також роль криптовалют у фінансовій системі. На основі цього аналізу були визначені функціональні та технічні вимоги до системи.

Платформа реалізована з використанням мови програмування Python та веб-фреймворку Flask для серверної частини, а також бази даних SQLite. Для клієнтської частини використані HTML, CSS та JavaScript, що забезпечує адаптивність інтерфейсу під різні пристрої.

Особлива увага приділена безпеці та захисту даних. Реалізовано механізми аутентифікації та авторизації користувачів, шифрування даних у спокої та транзиті, резервне копіювання та відновлення даних, а також моніторинг та виявлення загроз.

Проведене тестування підтвердило відповідність платформи вимогам продуктивності, надійності та безпеки. Результати роботи демонструють успішне вирішення поставлених завдань і показують, що розроблена платформа може бути ефективно використана для обміну валют та відстеження валютних курсів.

КЛЮЧОВІ СЛОВА: ОБМІН ВАЛЮТ, ФІНАНСОВІ РИНКИ, ОНЛАЙН-ПЛАТФОРМА, КРИПТОВАЛЮТА, БЕЗПЕКА ДАНИХ, PYTHON, FLASK, SQLITE.

SUMMARY

This thesis presents the development of an online platform for currency exchange and exchange rate tracking. The aim of the project was to create a convenient and secure platform that allows users to exchange currencies, monitor real-time exchange rate changes, and reserve currency at physical exchange points.

The work includes an analysis of modern financial markets and existing currency exchange technologies, a review of existing online currency exchange platforms, and the role of cryptocurrencies in the financial system. Based on this analysis, the functional and technical requirements for the system were determined.

The platform is implemented using the Python programming language and the Flask web framework for the backend, as well as the SQLite database. For the frontend, HTML, CSS, and JavaScript were used, ensuring the interface is adaptive to various devices.

Special attention was paid to security and data protection. Mechanisms for user authentication and authorization, data encryption at rest and in transit, data backup and recovery, and threat monitoring and detection were implemented.

Testing confirmed the platform's compliance with performance, reliability, and security requirements. The results demonstrate the successful achievement of the objectives and show that the developed platform can be effectively used for currency exchange and exchange rate tracking.

Keywords: CURRENCY EXCHANGE, FINANCIAL MARKETS, ONLINE PLATFORM, CRYPTOCURRENCY, DATA SECURITY, PYTHON, FLASK, SQLITE.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	9
РОЗДІЛ 1. ОГЛЯД ЛІТЕРАТУРИ	11
1.1 Фінансові ринки та технології обміну валют	11
1.2 Використання інформаційних технологій у фінансових операціях.....	12
1.3 Аналіз сучасних онлайн-платформ для обміну валют та відстеження валютних курсів	13
1.4 Проведення аналізу конкурентів	14
1.5 Роль криптовалют у фінансовій системі.....	17
Висновки до розділу 1	19
РОЗДІЛ 2. ВИЗНАЧЕННЯ ВИМОГ.....	20
2.1 Функціональні вимоги.....	20
2.2 Технічні вимоги	24
2.3 Backend частина	26
2.4 Frontend частина.....	28
2.5 Вимоги до безпеки та захисту даних	29
Висновки до розділу 2	32
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РОЗРОБКА. ТЕСТУВАННЯ ПЛАТФОРМИ	33
3.1 Архітектура платформи	33
3.2 Розробка користувацького інтерфейсу	37
3.3 Фронтенд реалізація	41
3.4 Адаптація до різних пристроїв	45
Висновки до розділу 3	47
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТКИ	52
Додаток А	52
Додаток Б.....	56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- CSS – Cascading Style Sheets (каскадні таблиці стилів)
- DBMS – Database Management System (система управління базами даних)
- HTML – HyperText Markup Language (мова розмітки гіпертексту)
- HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту)
- HTTPS – HyperText Transfer Protocol Secure (захищений протокол передачі гіпертексту)
- JSON – JavaScript Object Notation (формат обміну даними)
- SQL – Structured Query Language (структурована мова запитів)
- SSL – Secure Sockets Layer (рівень захищених сокетів)
- TLS – Transport Layer Security (транспортний рівень безпеки)
- UI – User Interface (користувацький інтерфейс)
- UML – Unified Modeling Language (уніфікована мова моделювання)
- UUID – Universally Unique Identifier (унікальний ідентифікатор)
- API – Application Programming Interface (інтерфейс програмування застосунків, додатків)
- XML – Extensible Markup Language (розширювана мова розмітки)
- Аутентифікація – процес перевірки автентичності користувача або системи шляхом підтвердження їхньої ідентичності.
- Авторизація – процес перевірки прав доступу користувача до певних ресурсів або функцій системи.
- База даних (БД) – організований набір структурованої інформації або даних, як правило, зберігається електронно в системі комп'ютера.
- Бекенд – частина системи, яка відповідає за бізнес-логіку, роботу з базами даних та серверні операції.

Клієнт-серверна архітектура – модель взаємодії між двома програмами, де одна з них (клієнт) робить запити до іншої (сервер), яка їх обробляє і повертає результати.

Криптовалюта – цифрова або віртуальна валюта, що використовує криптографію для забезпечення безпеки транзакцій.

Маршрут (route) – шлях, який визначає, як обробляється запит до веб-сервера і які дії виконуються в результаті цього запиту.

Резервне копіювання – процес створення копій даних, які можуть бути використані для відновлення оригінальних даних у разі їх втрати.

Фронтенд – частина системи, з якою безпосередньо взаємодіє користувач, включаючи дизайн та поведінку веб-сторінок.

Шифрування – процес перетворення даних у форму, недоступну для читання неавторизованими особами, з метою забезпечення конфіденційності та захисту інформації.

ВСТУП

Актуальність теми. У сучасному світі, де фінансові ринки стають все більш глобалізованими, а технології розвиваються стрімкими темпами, виникає необхідність створення ефективних та надійних онлайн-платформ для обміну валют та відстеження валютних курсів. Такі платформи дозволяють користувачам оперативно реагувати на зміни курсів валют, здійснювати вигідні фінансові операції та забезпечувати високий рівень захисту своїх коштів. В умовах зростаючої популярності криптовалют та електронних платежів, питання створення подібних платформ стає ще більш актуальним.

Об'єкт роботи. Об'єктом даної кваліфікаційної роботи є процеси обміну валют та відстеження валютних курсів в рамках фінансових онлайн-платформ.

Предмет роботи. Предметом дослідження є розробка інформаційної системи для обміну валют та відстеження валютних курсів, яка забезпечує високий рівень безпеки, ефективності та зручності для користувачів.

Мета роботи. Метою кваліфікаційної роботи є створення онлайн-платформи для обміну валют та відстеження валютних курсів, яка забезпечує користувачам можливість швидко і безпечно здійснювати фінансові операції, відстежувати зміни курсів валют та забезпечувати високу надійність та захищеність даних.

Задачі роботи. Для досягнення поставленої мети в роботі вирішені наступні задачі:

- аналіз існуючих онлайн-платформ для обміну валют та відстеження валютних курсів;
- визначення вимог до функціональних можливостей, технічних аспектів та безпеки нової платформи;
- проектування архітектури платформи та її бази даних;
- розробка користувацького інтерфейсу;
- реалізація механізмів захисту даних та безпеки користувачів;

- тестування платформи та оцінка її ефективності та продуктивності;
- аналіз результатів тестування та внесення необхідних коректив.

Методи роботи. Для виконання поставлених задач використовувалися наступні методи:

- аналіз літературних джерел та існуючих рішень в сфері обміну валют;
- моделювання та проектування інформаційних систем;
- програмування на мовах Python та JavaScript;
- використання технологій HTML, CSS для розробки інтерфейсу;
- методи забезпечення інформаційної безпеки;
- тестування програмного забезпечення.

Практичне значення одержаних результатів. Розроблена онлайн-платформа для обміну валют та відстеження валютних курсів має практичне значення для широкого кола користувачів, які здійснюють фінансові операції. Вона дозволяє забезпечити зручний та безпечний обмін валют, оперативне отримання інформації про зміни курсів та високу надійність захисту персональних даних. Платформа може бути використана як в комерційних цілях, так і для особистого користування, забезпечуючи високий рівень задоволеності користувачів.

Апробація результатів дослідження. Матеріали кваліфікаційної роботи були представлені на XI Міжнародній науковій конференції «Студентські наукові дискусії поза форматом», тема тези «Роль імплементації сучасних онлайн-платформ у оптимізації обміну валют, криптовалют та моніторингу валютних курсів», яка відбулась 11 квітня 2024 року в Університеті Короля Данила.

Структура роботи. Розділи – 3. Загальний обсяг основної частини – 43 сторінки. Список використаних джерел – 20.

РОЗДІЛ 1. ОГЛЯД ЛІТЕРАТУРИ

1.1 Фінансові ринки та технології обміну валют

Фінансові ринки відіграють важливу роль у світовій економіці, забезпечуючи механізми для обміну різноманітних фінансових активів, включаючи валюту. Основними учасниками фінансових ринків є комерційні банки, інвестиційні фонди, корпорації та інші фінансові установи. Ці ринки забезпечують можливість купівлі, продажу та обміну різноманітних фінансових інструментів з метою здійснення різних фінансових операцій.

Різні технології обміну валют використовуються для забезпечення ефективного функціонування фінансових ринків. Традиційні методи обміну валют включають телефонні дзвінки та особисті зустрічі між трейдерами. Однак із розвитком інформаційних технологій з'явилися нові електронні системи обміну валют, які дозволяють учасникам ринку здійснювати операції за допомогою комп'ютерів та мережі Інтернет.

Інтернет-технології відіграють ключову роль у сучасному фінансовому обміні валют. Електронні торгові платформи, такі як Bloomberg Terminal, Reuters Electronic Trading, та інші, надають можливість учасникам ринку здійснювати операції з валютами у реальному часі. Дані платформи забезпечують широкий спектр функціональності, включаючи можливість отримання фінансової інформації, аналізу ринку та здійснення операцій на валютному ринку.

Крім того, розвиток криптовалют та блокчейн технологій відкриває нові можливості для обміну валют. Криптовалюти, такі як Bitcoin та Ethereum, можуть бути використані для здійснення міжнародних фінансових операцій з низькими комісійними та мінімальними затримками.

Таким чином, різноманітні технології обміну валют відіграють важливу роль у забезпеченні ефективного функціонування фінансових ринків. Розвиток

цих технологій сприяє зростанню ліквідності ринку, підвищенню швидкості операцій, та забезпеченню доступу до ринку для широкого кола учасників [1].

1.2 Використання інформаційних технологій у фінансових операціях

Одним із ключових аспектів функціонування сучасних фінансових ринків є використання інформаційних технологій у фінансових операціях. Розвиток інформаційних технологій значно полегшив та прискорив процеси обміну валют, що сприяло розвитку електронних платформ.

Електронні платформи, такі як Forex (foreign exchange market), надають можливість учасникам ринку здійснювати торгівлю валютними парами у реальному часі. Ці платформи працюють у форматі електронного обміну, де учасники можуть складати та виконувати замовлення за допомогою комп'ютерних систем. Такий підхід значно зменшує час, необхідний для проведення операцій, та дозволяє уникнути зайвих посередників у торгівлі.

Одним із ключових переваг використання інформаційних технологій у фінансових операціях є можливість отримання швидкого та точного аналізу ринку. Завдяки спеціалізованим програмним засобам, трейдери можуть отримувати актуальну інформацію про котирування валют та аналізувати її у реальному часі. Це дозволяє приймати обґрунтовані рішення щодо участі в операціях на валютному ринку та здійснення торгівельних операцій з максимальною ефективністю.

Крім того, використання інформаційних технологій у фінансових операціях дозволяє забезпечити високий рівень безпеки та конфіденційності. Сучасні системи обміну валют використовують передові методи шифрування та захисту даних, що гарантує захист фінансових активів учасників ринку від несанкціонованого доступу [1, 2].

Отже, використання інформаційних технологій у фінансових операціях відкриває широкі можливості для учасників ринку, забезпечуючи їм швидкий, ефективний та безпечний доступ до валютних ринків. Сучасні електронні

платформи та системи обміну валют стають невід'ємною частиною сучасної фінансової інфраструктури, сприяючи розвитку та зростанню фінансових ринків у всьому світі.

1.3 Аналіз сучасних онлайн-платформ для обміну валют та відстеження валютних курсів

В сучасному цифровому світі інновації у фінансових технологіях постійно перетворюють способи, якими люди керують своїми фінансами. Онлайн-платформи для обміну валют та відстеження валютних курсів стають ключовими інструментами у цьому процесі, пропонуючи широкий спектр функцій та можливостей для користувачів. Для кращого розуміння ринкової ситуації та виявлення потенційних можливостей для покращення свого проекту, варто провести аналіз сучасних платформ.

Перед розглядом конкретних платформ важливо зазначити, що ринок фінансових технологій є досить конкурентним, а саме розвиток та популярність онлайн-платформ для обміну валют та відстеження валютних курсів зростає з кожним днем. Зокрема, платформи, такі як Revolut, TransferWise, PayPal, а також багато інших, займають певну нішу на ринку та пропонують свої унікальні функції та послуги.

Перед проведенням аналізу конкурентів слід визначити критерії, за якими буде оцінюватися кожна платформа. Основні аспекти, які можна включити до цього аналізу, включають:

1. Функціонал платформи: різноманіття функцій, що пропонує платформа для обміну валют та відстеження валютних курсів, включаючи можливість переказу грошей, конвертації валют, відстеження валютних курсів у режимі реального часу тощо [2];

2. Зручність та інтерфейс користувача: якість та зручність інтерфейсу впливає на загальне враження від використання платформи. Простота навігації

та доступність основних функцій можуть визначити популярність сервісу серед користувачів;

3. Безпека: важливим аспектом є захист особистих даних та фінансових транзакцій користувачів. Якість системи безпеки та заходи, прийняті для запобігання шахрайству, визначають рівень довіри до платформи;

4. Цінова політика: вартість використання платформи, включаючи комісії за перекази, конвертацію валют та інші послуги, може впливати на рішення користувачів про вибір платформи.

1.4 Проведення аналізу конкурентів

Аналіз сучасних онлайн-платформ для обміну валют та відстеження валютних курсів надають можливість краще зрозуміти ринкові тенденції, виявити переваги та недоліки конкурентів і визначити ключові аспекти, які можна покращити у власному проекті.

Сайт – URL: <https://obmin24.if.ua/> (рис. 1.1). Серед основних плюсів даного проекту можна виділити такі:

- використання SSL-шифрування для забезпечення безпеки даних;
- наявність доступних пунктів обміну в Івано-Франківську;
- впровадження програми лояльності для користувачів.

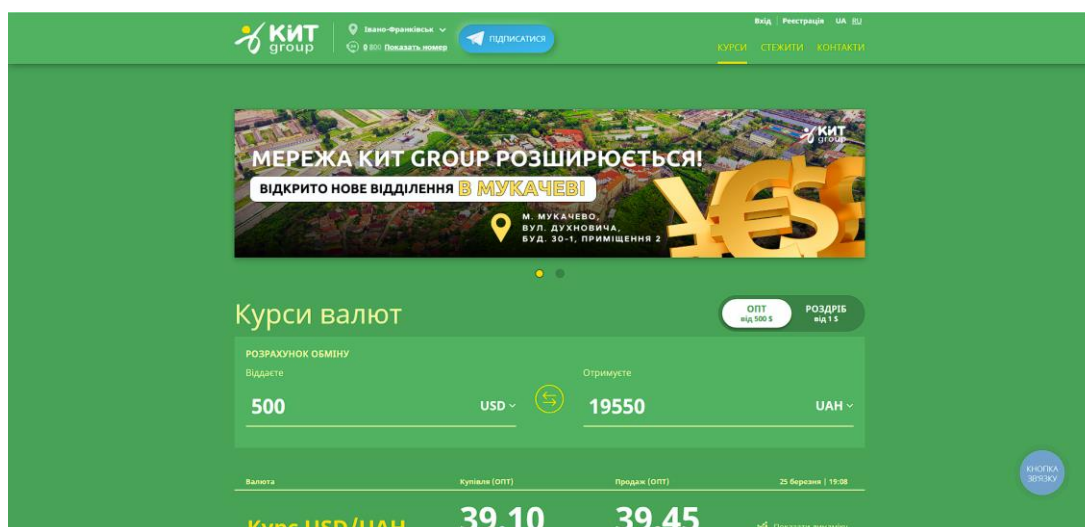


Рисунок 1.1 – Головна сторінка сайту

Недоліки проекту включають:

- не дуже зручний для мобільних пристроїв (рис. 1.2);
- необов'язкова двофакторна аутентифікація;
- не великий вибір валют, відсутність криптовалют (рис. 1.3).

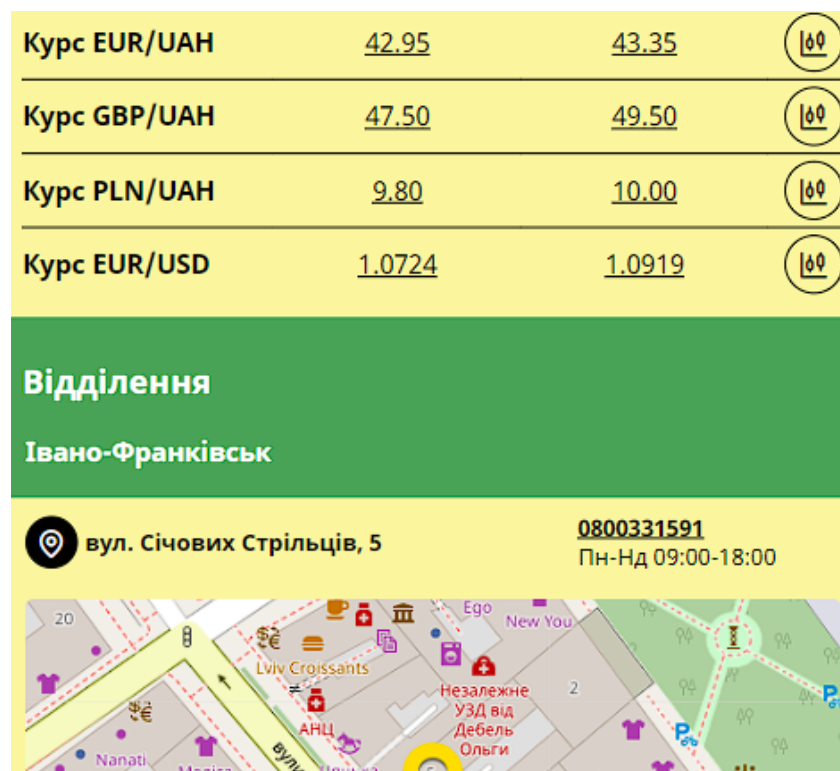


Рисунок 1.2 – Мобільна версія додатку (емулятор)

Рекомендації для покращення проекту:

- оновити інтерфейс сайту;
- зробити сайт більш зручним для мобільних пристроїв;
- зробити двофакторна аутентифікацію обов'язковою.

Висновки: сайт – URL: <https://obmin24.if.ua/> є посереднім сайтом для обміну валют. Він має широкий спектр валют та доступні пункти обміну в Івано-Франківську, але його інтерфейс застарілий, також він не дуже зручний для мобільних пристроїв.

Наступним я розгляну сайт – URL: <https://miniaylo.finance.ua/> (рис. 1.4).

Переваги даного проекту:

- можливість онлайн-обміну валют;

- перекази коштів в Україну та за кордон;
- інформація про кредити, депозити, інвестиції;
- наявність криптовалюти.

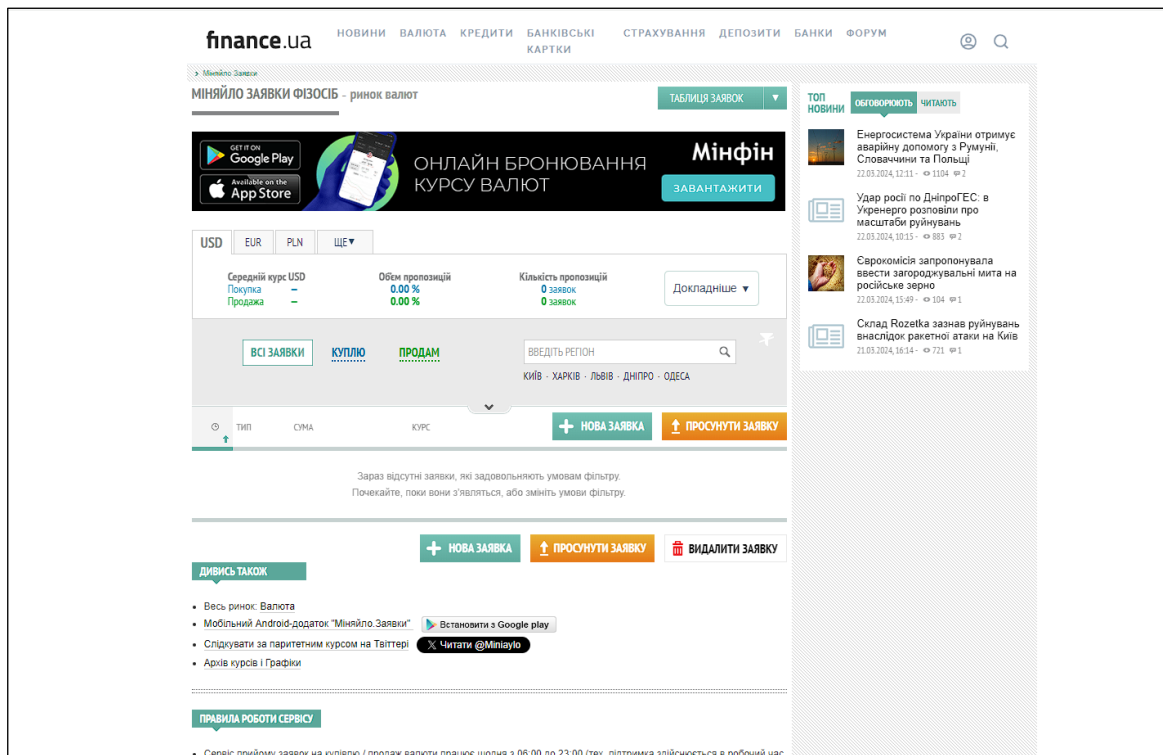


Рисунок 1.4 – Головна сторінка сайту

Недоліки проєкту включають:

- незручний інтерфейс;
- незручний для мобільних пристроїв (рис. 1.5);
- високі комісії за перекази коштів;
- відсутність деяких функцій, таких як форум для обговорення фінансових питань.

Рекомендації:

- зробити інтерфейс сайту більш зручним.
- зробити сайт більш зручним для мобільних пристроїв.
- знизити комісії за перекази коштів.
- додати нові функції, такі як форум для обговорення фінансових питань.

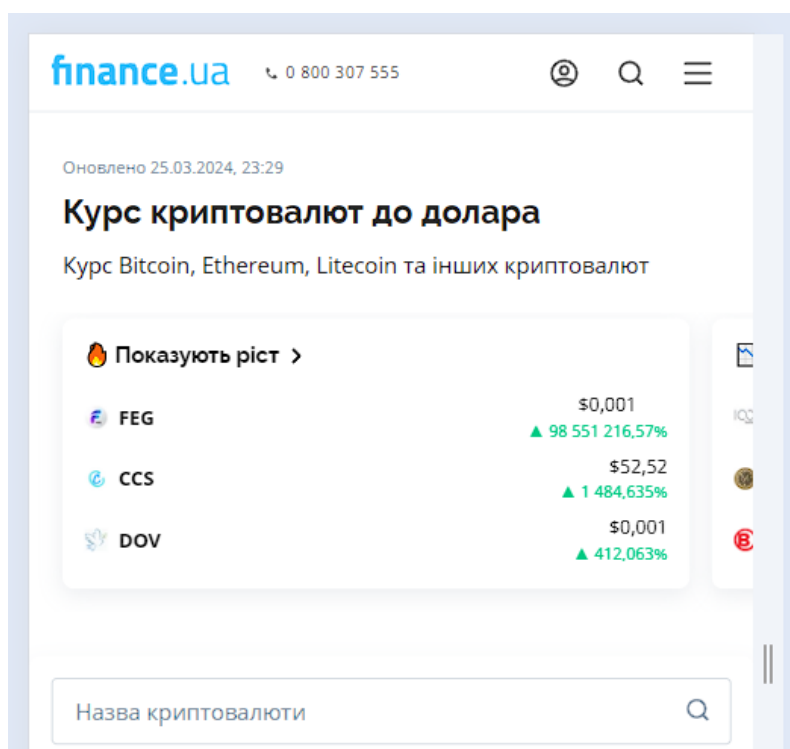


Рисунок 1.5 – Мобільна версія додатку (емулятор)

Висновки: сайт – URL: <https://miniaylo.finance.ua/> пропонує можливість онлайн-обміну валют, переказів коштів, а також інформацію про кредити, депозити та інвестиції. Проте, сайт має незручний інтерфейс, не дуже зручний для мобільних пристроїв, а також високі комісії за перекази коштів.

1.5 Роль криптовалют у фінансовій системі

Криптовалюти, такі як Bitcoin, Ethereum та інші, в останні роки набули значного значення у фінансовій системі, змінюючи традиційні підходи до обміну, інвестування та зберігання активів. У даному розділі детально розглянуто роль криптовалют у сучасній фінансовій системі, їх вплив на ринки, банківську діяльність та інвестиційну стратегію.

1) інноваційність та децентралізація: перш за все, криптовалюти представляють собою інноваційну технологію, яка базується на блокчейні, що дозволяє проводити та записувати транзакції безпечно та ефективно. Ця технологія забезпечує децентралізовану природу, у якій інформація зберігається

на кожному вузлі мережі, а не на централізованому сервері чи базі даних. Це робить криптовалюти незалежними від будь-яких центральних установ і контролюється спільнотою користувачів.

2) фінансова відкритість та прозорість: ще однією важливою характеристикою криптовалют є їхня відкритість та прозорість. Транзакції з криптовалютами здійснюються відкритим та доступним для перегляду блокчейні, що робить їхні дані доступними для перевірки будь-якому користувачеві. Це сприяє відкритості та довірі в систему, оскільки будь-який користувач може перевірити ланцюжок транзакцій.

3) альтернатива традиційним фінансовим послугам: крім того, криптовалюти стають альтернативою традиційним фінансовим послугам, таким як банківські перекази та платіжні системи. Вони дозволяють здійснювати міжнародні перекази коштів швидко та з меншими комісійними в порівнянні з традиційними банківськими послугами. Крім того, криптовалюти можуть бути використані для мікроплатежів та покупок у різних онлайн-магазинах та сервісах.

4) інструмент для інвестування: нарешті, криптовалюти стають популярним інструментом для інвестування та збереження капіталу. Багато людей використовують їх як засіб для диверсифікації інвестиційного портфеля та захисту від інфляції. Крім того, велика частина інвесторів сприймає криптовалюти як засіб для отримання високих прибутків через їхній високий рівень волатильності та можливості швидких зростань цін.

5) виклики та ризики: незважаючи на свою популярність та переваги, криптовалюти також стикаються з рядом викликів та ризиків. Серед них можна виділити високу волатильність цін, можливості для маніпуляцій на ринку, регуляторні ризики, а також проблеми з безпекою та кібербезпекою. Наприклад, зловмисники можуть атакувати криптовалютні біржі та гаманці, що може призвести до втрати активів користувачів.

6) майбутнє криптовалют: незважаючи на виклики та ризики, криптовалюти залишаються важливим елементом фінансової системи та мають

потенціал для подальшого розвитку та інновацій. Розробники працюють над вдосконаленням технологій блокчейну, щоб зробити криптовалюти більш масово доступними та придатними для використання у повсякденному житті. Крім того, уряди та регуляторні органи все більше зацікавлені в регулюванні цього ринку, що може сприяти збільшенню довіри до криптовалют та їхньому подальшому прийняттю в суспільстві [3].

Висновки до розділу 1

У результаті огляду літератури в розділі 1 були розглянуті ключові аспекти фінансових ринків та технологій обміну валют, використання інформаційних технологій у фінансових операціях, аналіз сучасних онлайн-платформ для обміну валют та відстеження валютних курсів, а також роль криптовалют у фінансовій системі.

Під час розгляду фінансових ринків та технологій обміну валют було виявлено, що зростаюча популярність онлайн-платформ для обміну валют є результатом швидкого розвитку технологій та зростаючої потреби у зручних та доступних фінансових інструментах. Використання інформаційних технологій у фінансових операціях підкреслює важливість цифрової трансформації у фінансовій сфері та прагнення до ефективного управління фінансами.

Аналіз сучасних онлайн-платформ для обміну валют та відстеження валютних курсів показав значущі переваги цих платформ у забезпеченні зручності та швидкості обміну, а також підвищення рівня доступності та прозорості фінансових операцій для користувачів.

Нарешті, роль криптовалют у фінансовій системі виявилася досить важливою, зокрема через їхню інноваційність, відкритість, інвестиційний потенціал та можливість використання в якості альтернативи традиційним фінансовим послугам [4].

РОЗДІЛ 2. ВИЗНАЧЕННЯ ВИМОГ

2.1 Функціональні вимоги

Даний підрозділ описує функціональність, яку система повинна надавати користувачам для ефективного взаємодії та досягнення їхніх цілей. Кожна функціональна вимога детально аналізується та формалізується для забезпечення чіткого розуміння її значення та обсягу. Цей розділ включає в себе опис основних функцій системи, взаємозв'язків між ними та їхніх характеристик. Ретельно розроблені функціональні вимоги є основою для подальшої розробки та впровадження системи, дозволяючи забезпечити відповідність потребам користувачів та досягти поставлених цілей проекту.

1. Користувачам надається можливість створення та керування особистими обліковими записами. Процес створення облікового запису є швидким і простим. Новий користувач може зареєструватися на платформі, використовуючи свою електронну адресу та створити пароль для входу. Після успішної реєстрації, користувач має можливість увійти до свого облікового запису, використовуючи введені раніше облікові дані.

У кабінеті користувача він має доступ до різних функцій та налаштувань, що дозволяють персоналізувати його обліковий запис та забезпечують зручний досвід користування платформою. Користувач може переглядати та редагувати свої особисті дані, такі як ім'я, контактні дані та адреса. Він також може змінювати пароль для безпечного входу до системи.

Один обліковий запис дозволяє користувачеві додавати різні види валют до свого портфелю. Для кожної валюти користувач може вказати власний баланс та переглядати історію транзакцій з нею пов'язаних. Це надає користувачеві можливість ефективно відстежувати свої фінансові потоки та керувати своїм портфелем.

Крім того, користувач може налаштувати рівень безпеки свого облікового запису. Платформа надає можливість активувати двофакторну аутентифікацію для додаткового захисту облікового запису. Користувач також може встановити обмеження доступу до свого облікового запису з певних IP-адрес або пристроїв, що підвищує рівень безпеки його даних.

Загалом, управління обліковим записом на платформі забезпечує користувачам зручний та безпечний доступ до всіх функціональних можливостей та сервісів, які надає платформа для обміну валют та відстеження їхніх курсів.

2. Платформа надає користувачам можливість відстежувати валютні курси в режимі реального часу. Під час використання цієї функції, користувач може переглядати актуальні курси обміну для різних валютних пар. Це дозволяє користувачам оперативно реагувати на зміни на фінансових ринках та приймати обґрунтовані рішення з обміну валют.

Платформа також надає користувачам можливість переглядати графіки обміну валют за різні періоди часу. Графіки відображають динаміку змін курсів обміну валют на протязі вибраного періоду, що дозволяє користувачам аналізувати тенденції та прогнозувати майбутні зміни.

Крім того, платформа може надавати користувачам інформацію про ключові фактори, що впливають на курси обміну валют, такі як економічні показники, геополітичні події та інші фактори. Це допомагає користувачам зрозуміти причини змін валютних курсів та приймати обґрунтовані рішення щодо їхніх фінансових операцій.

Функціональність відстеження валютних курсів на платформі дозволяє користувачам отримати доступ до актуальної та об'єктивної інформації про фінансові ринки, що допомагає їм ефективно керувати своїми інвестиціями та операціями з обміну валют.

3. Платформа надає користувачам можливість отримати актуальну інформацію про курс обміну валют у різних обмінних пунктах та фінансових установах. Користувач може здійснити пошук обмінних пунктів за різними критеріями, такими як місцезнаходження, курс обміну, графік роботи тощо [4].

Крім того, платформа надає користувачам можливість переглядати рейтинг та відгуки про обмінні пункти, що дозволяє їм зробити обміркований вибір. Користувач може дізнатися про досвід інших користувачів щодо обмінного пункту, його якість обслуговування та відповідність курсу.

Платформа також надає користувачам можливість здійснювати бронювання валюти в обмінних пунктах. Користувач може обрати валюту, суму, дату та час, коли він планує здійснити обмін, і забронювати необхідну суму валюти. Система автоматично здійснює резервування грошей на обрану дату та час і надає користувачу підтвердження бронювання.

Загалом, функціонал "Інформація про курс обміну валют у живих точках" дозволяє користувачам максимально зручно та ефективно здійснювати обмін валют у найвигідніших умовах.

4. Користувачам надається можливість зарезервувати певну суму валюти для обміну в певний час та місце. Процес резервування дуже простий та зручний для користувача, роблячи планування фінансових операцій максимально ефективним.

Спочатку користувач здійснює вибір валюти, яку він бажає зарезервувати, та вказує потрібну суму. Після цього він обирає зручний для нього час та місце для обміну. Платформа автоматично перевіряє наявність необхідної суми валюти та доступність обмінних пунктів на обраний час.

Після успішної перевірки система резервує зазначену суму валюти до обраного часу. Користувач отримує підтвердження про успішне здійснення резервування разом з усіма необхідними даними щодо дати, часу та місця обміну. У разі зміни планів користувача, він може зробити скасування резервування через платформу.

Ця функція дозволяє користувачам забезпечити собі необхідну суму валюти на майбутній обмін, забезпечуючи при цьому високий рівень зручності та безпеки. Вона допомагає уникнути непередбачуваних ситуацій та забезпечує успішний обмін валюти у відповідності з планами користувача. Додатково, ця функція створює додатковий рівень впевненості для користувача щодо його

фінансових операцій, дозволяючи йому ефективно керувати своїми фінансами.

Загалом, можливість резервування валюти на платформі забезпечує користувачам зручний та безпечний спосіб планування своїх фінансових операцій та максимізації їхньої ефективності [4].

5. На платформі доступний функціонал перегляду графіків обміну валют, що дозволяє користувачам отримати детальну інформацію про динаміку змін курсів обміну валют за різні періоди часу. Цей інструмент стає надійним помічником для аналізу тенденцій на фінансових ринках та прийняття обґрунтованих фінансових рішень.

Користувач має можливість обирати період, за який він бажає переглянути графік обміну валют: це може бути день, тиждень, місяць або будь-який інший визначений період. Після вибору періоду, платформа автоматично побудує графік, що відображає динаміку курсів обміну валют протягом обраного періоду.

Графік обміну валют надає користувачам зрозумілу та вичерпну інформацію про зміни курсів валют в залежності від часу. Він включає в себе різні показники, такі як цінові рівні, обсяги торгів та інші ключові показники, що дозволяють користувачам здійснювати детальний аналіз ринкових умов та прогнозувати майбутні зміни.

Крім того, платформа може надавати користувачам додаткову інформацію та аналітичні матеріали, які допомагають розібратися в причинах змін курсів валют та приймати обґрунтовані інвестиційні рішення.

Загалом, функція перегляду графіків обміну валют на платформі дозволяє користувачам ефективно аналізувати фінансові ринки та приймати обґрунтовані рішення з обміну валют в залежності від ринкових умов [4, 5].

6. На платформі доступний інструмент фільтрації валютних операцій, який дозволяє користувачам зручно та ефективно керувати своїми фінансовими операціями та знайти необхідну інформацію.

Цей інструмент дозволяє користувачам швидко знаходити певні валютні операції за різними параметрами, такими як дата, сума, тип операції

(купівля/продаж), валютна пара тощо. Користувач може використовувати різні комбінації фільтрів для точного визначення потрібних операцій.

Наприклад, користувач може шукати всі свої купівельні операції доларів у проміжку з 1 січня по 15 лютого цього року. Він також може фільтрувати операції за валютним парами, вибираючи лише ті операції, що стосуються певної валютної пари, наприклад, USD/EUR або USD/GBP.

Після виконання фільтрації, користувач отримує список відповідних операцій, який може бути відсортований за різними критеріями, такими як дата, сума або тип операції. Це дозволяє користувачам швидко знайти необхідну інформацію та аналізувати свої фінансові операції.

Загалом, функція фільтрації валютних операцій на платформі забезпечує користувачам зручний та ефективний спосіб управління своїми фінансами та швидкий доступ до необхідної інформації про їхні операції.

2.2 Технічні вимоги

У даному підрозділі описуються основні технічні вимоги до розроблення системи обміну валют та відстеження валютних курсів. Цей підрозділ визначає функціональні та технічні характеристики системи, необхідні для досягнення поставлених цілей та задоволення потреб користувачів. Вимоги до архітектури, інтеграції, безпеки та ефективності системи ретельно аналізуються з метою забезпечення найвищої якості та надійності продукту. Крім того, у цьому розділі розглядається вибір технологічних засобів та платформ, що найбільш відповідають вимогам проекту. Ретельно розроблені технічні вимоги визначають основу для успішної реалізації системи та забезпечують високу якість та ефективність роботи програмного забезпечення.

1. При розробці даної онлайн-платформи для обміну валют та відстеження валютних курсів було завчасно обрано базу даних. Для даного проекту було використано SQLite – яка вбудована в Python. Дана БД легка, швидка та ефективна реляційна база даних. Python, як мова програмування, відома своєю

простотою та потужністю, що робить її ідеальним інструментом для швидкого розроблення та реалізації проєктів будь-якої складності. З використанням SQLite разом з Python вдалося добре організувати роботу з базою даних та швидко реалізувати основну функціональність проєкту без необхідності встановлення та налаштування складних серверів баз даних:

1) тип бази даних: використовується SQLite, що забезпечує легку інтеграцію та простоту у використанні.

2) продуктивність: з врахуванням обмежень проєкту, продуктивність SQLite вже вистачає для ефективної роботи платформи.

3) масштабованість: SQLite може бути легко перенесена на більш потужну базу даних у майбутньому, якщо зросте обсяг даних або потрібна більша продуктивність.

2. Безпека: враховуючи важливість забезпечення безпеки даних для проєкту, було вдосконалено заходи безпеки для бази даних SQLite. Для цього було використано різноманітні підходи та механізми захисту, включаючи:

1) аутентифікація і авторизація: застосовано механізми аутентифікації користувачів та контролю доступу до бази даних. Кожен користувач має свій унікальний ідентифікатор і обмеження доступу до відповідних даних відповідно до їх ролі в системі.

2) шифрування даних: для захисту конфіденційності даних використовується шифрування. Всі дані, що зберігаються в базі даних, шифруються за допомогою надійних алгоритмів шифрування, що забезпечує їхню конфіденційність у випадку несанкціонованого доступу.

3) моніторинг та журналювання: реалізовано систему моніторингу та журналювання, що дозволяє вчасно виявляти та реагувати на будь-які потенційні загрози безпеці даних. Логи та події системи ретельно аналізуються для виявлення незвичних або підозрілих активностей.

3. Резервне копіювання та відновлення: хоча SQLite не має автоматичних інструментів резервного копіювання, я розробив процедури регулярного збереження даних. Однією з ключових складових забезпечення безпеки даних є

система резервного копіювання та відновлення. Це надійний механізм, який гарантує, що навіть у випадку втрати даних або системних збоїв, інформація може бути легко відновлена:

1) регулярні резервні копії: проводяться регулярні резервні копії бази даних, що забезпечує збереження актуальних копій даних на випадок виникнення непередбачуваних подій.

2) захищене зберігання копій: усі резервні копії даних зберігаються в захищеному і недоступному для злоумисників місці. Використовуються надійні засоби зберігання даних з відповідними механізмами контролю.

3) тестування відновлення: періодично проводяться тести відновлення для перевірки ефективності та надійності системи резервного копіювання. Це дозволяє вчасно виявляти та усувати будь-які проблеми з відновленням даних.

4) сумісність: SQLite ідеально сумісна з Python, що полегшує розробку та інтеграцію з іншими частинами проекту [5].

З використанням SQLite було добре враховано основні вимоги щодо бази даних для проекту та готовність до подальшого розвитку та вдосконалення системи.

2.3 Backend частина

У своєму проекті я використав Flask, який є мікро фреймворком для веб-розробки на Python. Flask надає всі необхідні інструменти для створення надійного і масштабованого серверного додатку, а його простота та гнучкість дозволяють легко адаптуватися під специфічні вимоги проекту.

Flask є популярним вибором для розробки веб-додатків завдяки своїм численним перевагам:

1. Легкість і простота використання: Flask має невеликий обсяг коду та простий у налаштуванні, що дозволяє швидко розпочати роботу над проектом.

2. Гнучкість і розширюваність: Flask дозволяє легко інтегрувати сторонні бібліотеки та розширення, що робить його ідеальним для реалізації проекту.

3. Підтримка модульності: Flask підтримує створення модульних додатків, що спрощує структуру проекту та його підтримку.

4. Наявність великої кількості розширень: серед розширень, що широко використовуються, є Flask-SQLAlchemy для роботи з базами даних, Flask-Login для аутентифікації користувачів, Flask-Migrate для управління міграціями бази даних та багато інших.

Архітектура бекенд-частини включає:

1. Маршрутизація та обробка запитів:
2. Використання RESTful архітектури для організації API, що забезпечує зручний доступ до функціоналу платформи.

3. Реалізація маршрутизації запитів для обробки основних операцій, таких як створення транзакцій, отримання курсів валют.

Робота з базою даних:

1. Використання SQLAlchemy як ORM (Object-Relational Mapping) для управління базою даних SQLite.

2. Створення моделей для основних таблиць: користувачів, транзакцій та курсів валют.

3. Реалізація функцій для виконання CRUD-операцій (створення, читання, оновлення, видалення записів).

Безпека проекту включає:

1. Впровадження аутентифікації та авторизації користувачів за допомогою Flask-Login.

2. Використання хешування паролів з бібліотекою bcrypt для забезпечення безпечного зберігання паролів.

3. Використання токенів доступу для авторизації запитів до захищених ресурсів платформи.

Обробка помилок та логування:

1. Реалізація обробки помилок для забезпечення стабільності роботи платформи та інформування користувачів про можливі проблеми.

2. Логування важливих подій та помилок для подальшого аналізу та налагодження системи.

Резервне копіювання та відновлення даних:

1. Реалізація механізмів автоматичного резервного копіювання бази даних для захисту даних від втрати або пошкодження.

2. Загалом, бекенд-частина проєкту побудована на основі Flask, що забезпечує високу надійність, гнучкість та масштабованість системи. Використання сучасних технологій та підходів до розробки веб-додатків дозволяє забезпечити стабільну роботу платформи та відповідну безпеку [5, 6].

2.4 Frontend частина

Frontend частина веб-додатку відповідає за візуальне відображення даних та інтерактивність інтерфейсу, який користувач взаємодіє з ним. У цьому пункті зосередження відбувається саме на розробці Frontend частини за допомогою HTML, CSS та JavaScript, з можливим використанням фреймворків та бібліотек, таких як React або Vue.js.

Основна мета цього етапу – забезпечити правильне відображення контенту на різних пристроях та браузерах, а також забезпечити зручну та естетичну взаємодію з користувачем. Для цього використовуються техніки респонсивного дизайну, які дозволяють адаптувати вигляд сторінки до різних розмірів екранів.

При розробці веб-сторінок використовується HTML (HyperText Markup Language) для визначення структури та контенту сторінок. HTML включає в себе набір тегів, які визначають різні елементи сторінки, такі як заголовки, абзаци, списки, таблиці, зображення тощо. Теги HTML надають структурну основу для вмісту сторінки та дозволяють браузерам правильно інтерпретувати та відображати цей вміст для користувачів.

CSS (Cascading Style Sheets) використовується для стилізації та оформлення вмісту сторінок. CSS дозволяє визначати такі аспекти як кольори, шрифти, розміри, відступи, межі, анімації та багато іншого. Він дозволяє надати

вмісту сторінки відповідний вигляд та відображення, створюючи зручне та привабливе середовище для користувачів.

HTML і CSS спільно використовуються для створення зручного та естетичного інтерфейсу веб-додатків. HTML визначає структуру та контент сторінки, тоді як CSS відповідає за її оформлення та візуальне представлення. Вони спільно доповнюють один одного, дозволяючи розробникам створювати різноманітні та привабливі веб-інтерфейси, які забезпечують зручну та приємну взаємодію з користувачем [5].

2.5 Вимоги до безпеки та захисту даних

Забезпечення безпеки та захисту даних є однією з найважливіших складових будь-якого інформаційного проекту. У цьому підрозділі кваліфікаційної роботи розглядаються вимоги до безпеки та методи захисту даних проекту. Відповідна увага приділяється аспектам аутентифікації, авторизації, шифруванню, контролю доступу та іншим механізмам, що забезпечують конфіденційність, цілісність та доступність даних. Ретельне вивчення та впровадження вимог до безпеки дозволяє запобігти можливим загрозам та забезпечити високий рівень захищеності системи. В цьому розділі будуть розглянуті ключові аспекти забезпечення безпеки, а також вимоги до захисту даних, які допоможуть забезпечити безпеку та відповідність стандартам.

1. Механізми аутентифікації та авторизації. У проекті реалізовано надійні механізми аутентифікації та авторизації, які забезпечують безпеку та конфіденційність користувачів:

1) аутентифікація користувачів: для входу в систему користувачі повинні надати свої логін та пароль. Після цього дані перевіряються у базі даних, де пароль зберігається у вигляді хешу. Такий підхід забезпечує безпеку паролів навіть у разі несанкціонованого доступу до бази даних.

2) авторизація користувачів: після успішної аутентифікації користувача система перевіряє його роль та дозволи. Кожній ролі призначаються

визначені функціональність та ресурси, до яких вона має доступ. Це дозволяє контролювати рівень доступу до різних частин системи та забезпечити приватність даних.

3) хешування паролів: паролі користувачів зберігаються у базі даних у вигляді хешів, саме це забезпечує захист паролів навіть у випадку, якщо база даних стане доступною для зловмисників.

4) сесії та токени авторизації: після успішної аутентифікації користувача генерується токен авторизації, який використовується для подальшої ідентифікації. Цей токен зберігається в сесії користувача та передається з кожним запитом до сервера для підтвердження ідентифікації.

5) двофакторна аутентифікація: для підвищення безпеки системи може бути включена двофакторна аутентифікація, яка вимагає введення додаткового коду, отриманого через SMS, мобільного додатку або іншого засобу [2, 6].

Усі ці механізми дозволяють забезпечити найвищий рівень безпеки та захисту даних у проекті.

2. Резервне копіювання та відновлення даних. У проекті реалізована надійна система резервного копіювання та відновлення даних, що забезпечує безпеку та надійність інформаційних ресурсів:

1) система резервного копіювання: система автоматично створює резервні копії бази даних та інших важливих даних згідно з налаштованою частотою. Ці копії зберігаються на надійних серверах, забезпечуючи захист від випадкових втрат даних. Частота та механізми резервного копіювання налаштовані таким чином, щоб забезпечити мінімальні можливості втрати даних у випадку будь-яких непередбачуваних ситуацій.

2) відновлення даних: якщо сталася ситуація, коли потрібно відновити дані, система має процедури відновлення, які дозволяють швидко та ефективно відновити важливу інформацію з резервних копій. Це забезпечує мінімальний час недоступності та максимальну надійність відновлення даних.

3) тестування та перевірка: регулярні перевірки та тестування системи резервного копіювання забезпечують його ефективність та надійність. Це

включає в себе перевірку цілісності та доступності резервних копій, а також перевірку готовності та ефективності процедур відновлення. Будь-які виявлені недоліки виправляються негайно, щоб забезпечити найвищий рівень захищеності та безпеки даних.

Цей підхід до резервного копіювання та відновлення даних гарантує, що дані залишаються безпечними та доступними у будь-який час, забезпечуючи стабільну та надійну роботу мого проекту.

3. Шифрування даних у спокої та транзиті. У проекті велика увага приділяється безпеці даних у спокої та транзиті. За допомогою шифрування забезпечується конфіденційність та цілісність інформації під час її зберігання.

1) шифрування даних у спокої: всі дані, що зберігаються у базі даних, підлягають шифруванню. Це забезпечує, що навіть у випадку несанкціонованого доступу до бази даних, інформація залишається зашифрованою і недоступною для злоумисників. Використовуються сучасні алгоритми шифрування, такі як AES (Advanced Encryption Standard), щоб забезпечити максимальний рівень захисту.

2) шифрування даних у транзиті: під час передачі даних між клієнтом і сервером також використовується шифрування. Всі дані, що передаються через мережу, зашифровані з використанням протоколів забезпечення безпеки, таких як SSL (Secure Sockets Layer) або TLS (Transport Layer Security). Це забезпечує конфіденційність даних під час їх передачі та запобігає можливим атакам на дані під час транзиту.

3) керування ключами: усі ключі шифрування та розшифрування даних тримаються у безпеці та керуються відповідно до найвищих стандартів безпеки. Вони генеруються випадковим чином і зберігаються в безпечному сховищі. Доступ до ключів мають лише авторизовані користувачі з необхідними правами доступу.

4) аудит безпеки: проводяться регулярні аудити безпеки для перевірки ефективності застосованих заходів захисту даних. Вони включають в себе перевірку конфігурації системи, аналіз потенційних вразливостей та моніторинг

відомих загроз. Будь-які виявлені вразливості виправляються негайно для забезпечення максимального рівня безпеки даних.

Шифрування даних у спокої та транзиті гарантує, що інформація залишається захищеною під час зберігання та передачі, забезпечуючи конфіденційність та цілісність даних [5, 6].

Висновки до розділу 2

Розділ 2 кваліфікаційної роботи присвячений визначенню вимог до проекту зі створення онлайн-платформи для обміну валют та відстеження валютних курсів. У цьому розділі були ретельно проаналізовані функціональні, технічні вимоги до безпеки та захисту даних, що вимагається для успішної реалізації проекту.

В результаті досліджень та аналізу літератури було визначено ключові функції та технічні аспекти системи, а також вимоги до безпеки, які потрібно врахувати при її розробці. Функціональні вимоги включають створення механізмів обміну валют, перегляду валютних курсів та графіків, резервування валют, а також фільтрацію валютних операцій. Технічні вимоги передбачають використання вбудованих механізмів безпеки та бази даних SQLite для забезпечення ефективності та безпеки системи.

У розділі також було розглянуто вимоги до безпеки та захисту даних, що включають в себе механізми аутентифікації та авторизації, резервне копіювання та відновлення даних, шифрування даних у спокої та транзиті, моніторинг та виявлення загроз. Всі ці вимоги спрямовані на забезпечення надійності, конфіденційності та цілісності даних у системі.

Розділ 2 відіграє ключову роль у формулюванні вимог до проекту та визначенні його технічних параметрів. Його вивчення та аналіз дозволяє точно спланувати розробку системи та забезпечити її високу якість та безпеку.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РОЗРОБКА. ТЕСТУВАННЯ ПЛАТФОРМИ

У даному розділі детально описується процес проектування та розробки онлайн-платформи для обміну валют та відстеження валютних курсів, а також результати її тестування. Основна мета цього розділу – продемонструвати, як були реалізовані технічні та функціональні вимоги, визначені у попередньому розділі, та оцінити ефективність та продуктивність розробленої системи.

3.1 Архітектура платформи

У цьому підрозділі буде детально розглянуто архітектуру платформи для обміну валют та відстеження валютних курсів. Архітектура системи вже спроектована та реалізована таким чином, щоб забезпечити ефективність, безпеку та масштабованість платформи. Архітектура складається з декількох основних компонентів: клієнтська частина, серверна частина, база даних та інтеграція з зовнішніми сервісами (рис. 3.1) [4].

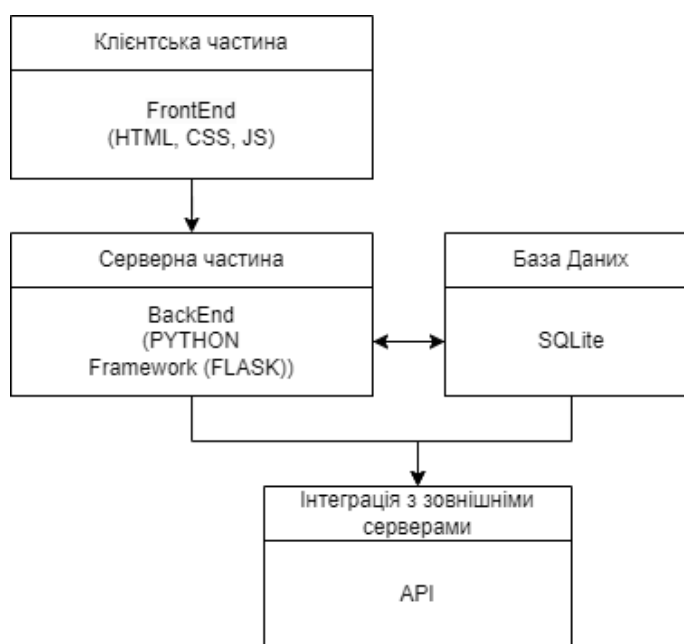


Рисунок 3.1 – Загальна структура архітектури

Як можна побачити з (рис. 3.1) архітектура платформи базується на клієнт-серверній моделі, де клієнтська частина (frontend) взаємодіє з серверною частиною (backend) через API (інтерфейс програмування додатків). Серверна частина взаємодіє з базою даних та зовнішніми сервісами для отримання та обробки необхідної інформації:

1. Клієнтська частина (frontend) платформи відповідає за взаємодію з користувачем. Вона розроблена з використанням сучасних веб-технологій, таких як HTML, CSS та JavaScript. Користувачі можуть переглядати курси валют, здійснювати обмін валют, резервувати валюту та переглядати графіки валютних курсів. Інтерфейс користувача створено таким чином, щоб забезпечити зручність та інтуїтивність взаємодії.

2. Серверна частина (backend) платформи відповідає за обробку запитів від клієнтської частини, виконання бізнес-логіки та взаємодію з базою даних. Для розробки серверної частини я використав Python. Серверна частина забезпечує аутентифікацію та авторизацію користувачів, обробку фінансових операцій, зберігання та обробку даних про валютні курси.

3. База даних використовується для зберігання інформації про користувачів, фінансові операції, курси валют та інші дані, необхідні для функціонування платформи. У проекті використано SQLite, яка є вбудованою базою даних для Python. SQLite забезпечує простоту налаштування та використання, що є важливим для прототипування та розробки.

4. Інтеграція з зовнішніми сервісами платформа інтегрується з кількома зовнішніми сервісами для отримання актуальних даних про курси валют. Це можуть бути API фінансових установ, банків та інших організацій, що надають дані про валютні курси в реальному часі. Інтеграція з цими сервісами забезпечує актуальність та точність інформації, яку отримують користувачі [5].

Діаграма архітектури (рис. 3.2) представляє високорівневий огляд системи та її компонентів. Основні компоненти включають:

1. Користувачів, які взаємодіють з платформою через веб-браузер або мобільний додаток.


```

        username = db.Column(db.String(80), unique=True,
nullable=False)
        password_hash = db.Column(db.String(120), nullable=False)

```

– таблиця транзакцій (`transactions`), (рис. 3.3): зберігає дані про фінансові операції. Код для реалізації бази даних:

```

class Transaction(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'),
nullable=False)
    amount = db.Column(db.Float, nullable=False)
    currency = db.Column(db.String(10), nullable=False)
    timestamp = db.Column(db.DateTime, nullable=False)

```

– таблиця курсів валют (`exchange_rates`), (рис. 3.3): зберігає поточні та історичні курси валют. Код для реалізації бази даних:

```

class ExchangeRate(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    currency = db.Column(db.String(10), nullable=False)
    rate = db.Column(db.Float, nullable=False)
    timestamp = db.Column(db.DateTime, nullable=False)

```

Архітектура серверної частини: серверна частина (бекенд) розроблена з використанням Flask, легкого веб-фреймворку для Python. Вона забезпечує обробку запитів від клієнта, взаємодію з базою даних та виконання бізнес-логіки. Основні компоненти бекенду включають:

1. Маршрути для управління користувачами: реєстрація, аутентифікація та управління профілем.
2. Маршрути для управління транзакціями: створення, перегляд та аналіз транзакцій, і їхнє управління.

3. Маршрути для управління курсами валют: додавання, оновлення та перегляд курсів валют.

Код реалізації серверної частини детально описаний в (Додаток А). В цьому додатку наведені всі необхідні SQL-запити для створення таблиць, а також реалізація маршрутів API.

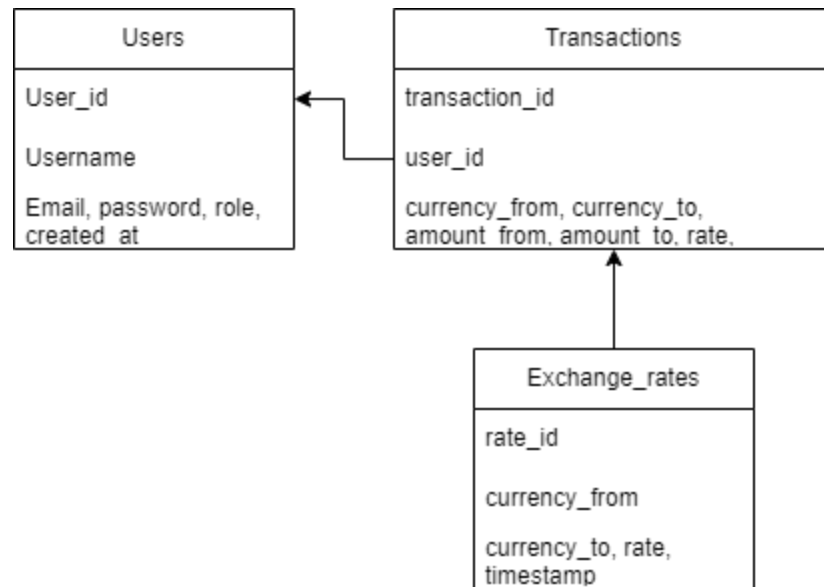


Рисунок 3.3 – Діаграма архітектури бази даних

3.2 Розробка користувацького інтерфейсу

Користувацький інтерфейс включає в себе веб-сторінки та елементи взаємодії, які дозволяють користувачам взаємодіяти з платформою зручним та ефективним способом.

Веб-сторінки та макети: для початку розробки користувацького інтерфейсу були розроблені макети веб-сторінок з урахуванням функціональних вимог та дизайну. Макети включають в себе головну сторінку (рис. 3.4), сторінки для обміну валют (Рис. 3.5), перегляду графіків курсів валют (Рис. 3.6), а також сторінки авторизації (Рис. 3.7), та реєстрації (Рис. 3.8).

Мокап головної сторінки (рис. 3.4) включає в себе кілька ключових елементів, які забезпечують зручність та ефективність користування платформою. Серед них:

1. Кнопки "Реєстрація" та "Вхід", розташовані у верхньому правому куті сторінки, що дозволяють користувачам швидко створити новий обліковий запис або увійти до існуючого.

2. Центральна кнопка із переходом до реєстрації, яка розташована на головному банері і привертає увагу нових відвідувачів.

3. Додаток із вибором валюти, розташований на головній панелі, який дозволяє користувачам швидко обрати потрібну валюту для обміну чи відстеження курсів.

4. Блок із перевагами сервісу, що містить перелік ключових переваг платформи, таких як "Безпечний обмін", "Найкращі курси" та "Миттєві транзакції", щоб переконати користувачів у надійності та зручності використання нашого сервісу.

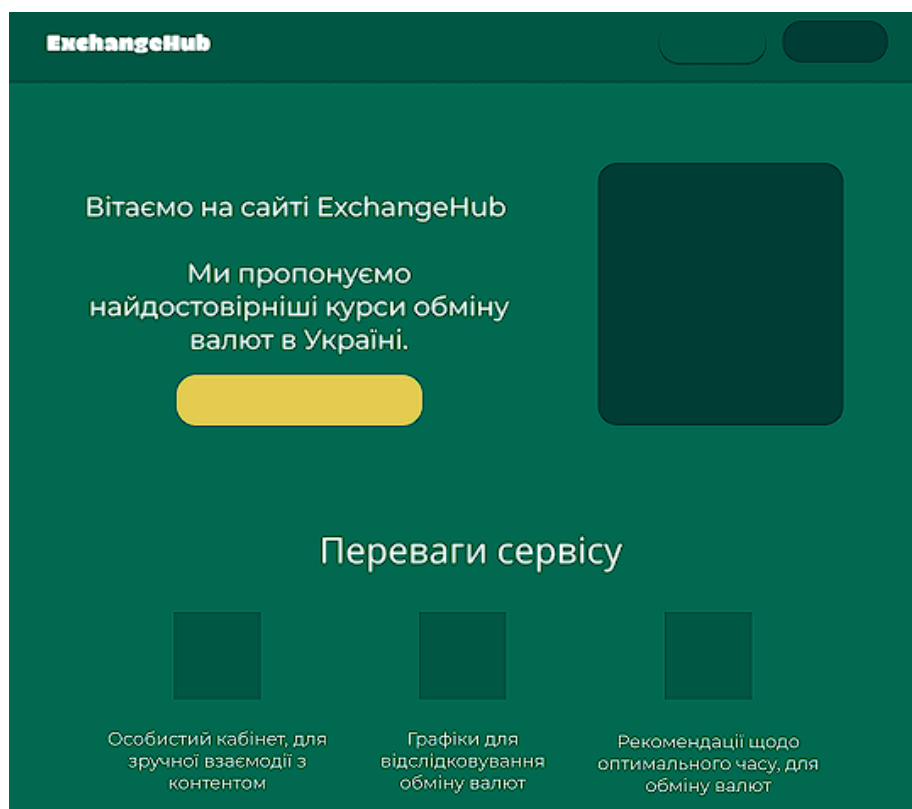


Рисунок 3.4 – Мокап головної сторінки

Мокап сторінки обміну валют (рис. 3.4), включає в себе декілька основних елементів, які забезпечують зручність та ефективність обміну валют. Серед них:

1. Блок, який містить конвертор для валюти, розташований у верхній частині сторінки, що дозволяє користувачам швидко та легко здійснювати обмін валют.

2. Інтерактивні поля для вибору різних валют, де користувачі можуть вибрати валюту, яку хочуть обміняти, та валюту, яку хочуть отримати.

3. Поле для введення суми обміну, де користувачі можуть вказати необхідну суму для обміну.

4. Автоматичний розрахунок курсу, який показує поточний курс обміну та суму, яку користувач отримає після конвертації.

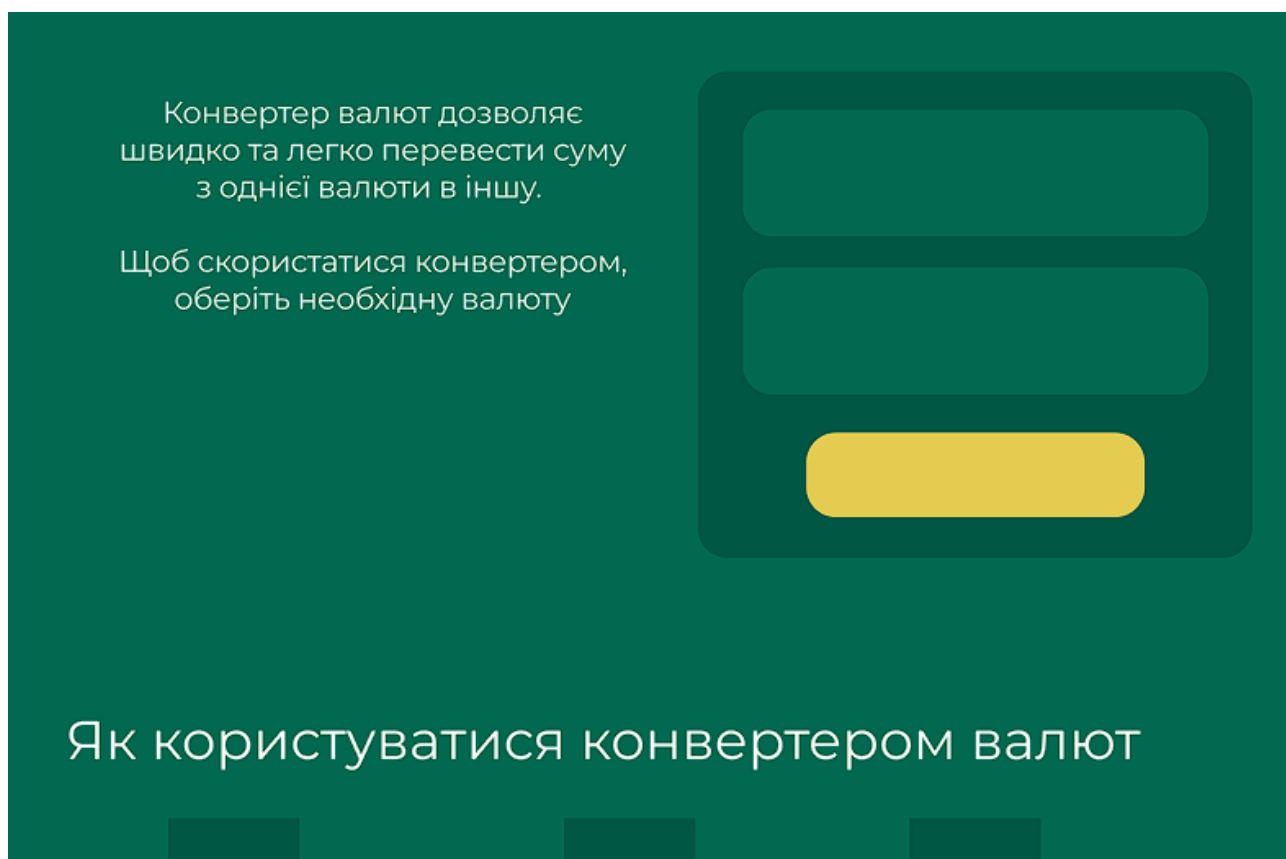


Рисунок 3.5 – Мокап обміну валют

Наступний мокап перегляду графіків (рис. 3.6), користувач може вибрати валюту з відповідного блоку вибору валют. Після вибору валюти в центральному блоці автоматично відобразиться графік зміни курсу цієї валюти за обраний

період. Нижній блок зі списком популярних валют забезпечує швидкий доступ до перегляду графіків інших популярних валют на сайті.

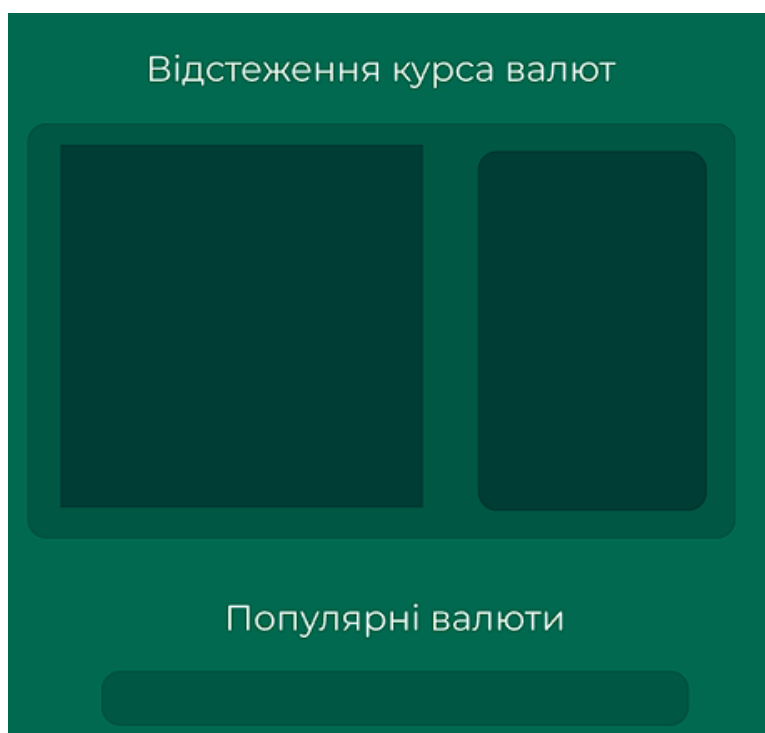


Рисунок 3.6 – Мокап сторінки перегляду графіків

Наступні сторінки це реєстрація, та вхід для користувача: на сторінці входу користувачі можуть увійти в свій акаунт, ввівши електронну пошту або логін та пароль. Після введення цих даних і натискання кнопки "Увійти", користувач отримає повідомлення про успішний вхід або про помилки у введених даних.

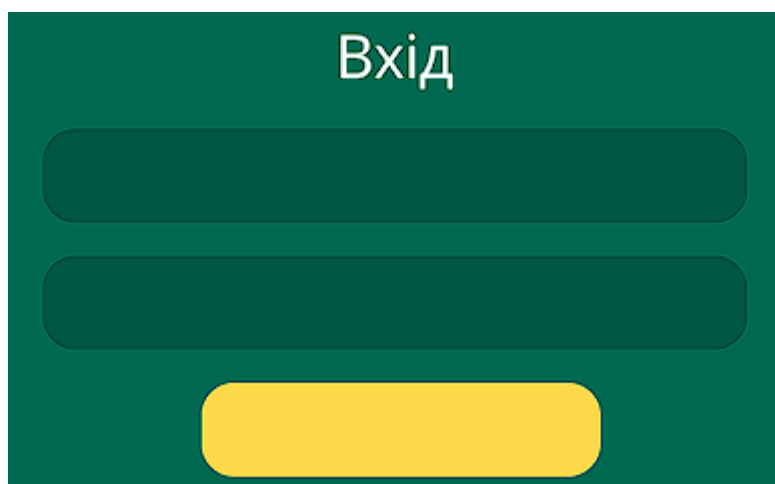


Рисунок 3.7 – Мокап сторінки авторизації

На сторінці реєстрації користувачі можуть створити новий акаунт, заповнивши форму реєстрації з необхідною інформацією. Після заповнення всіх полів і натискання кнопки "Зареєструватися", користувач отримає повідомлення про успішну реєстрацію або про помилки, які потрібно виправити, для успішної реєстрації

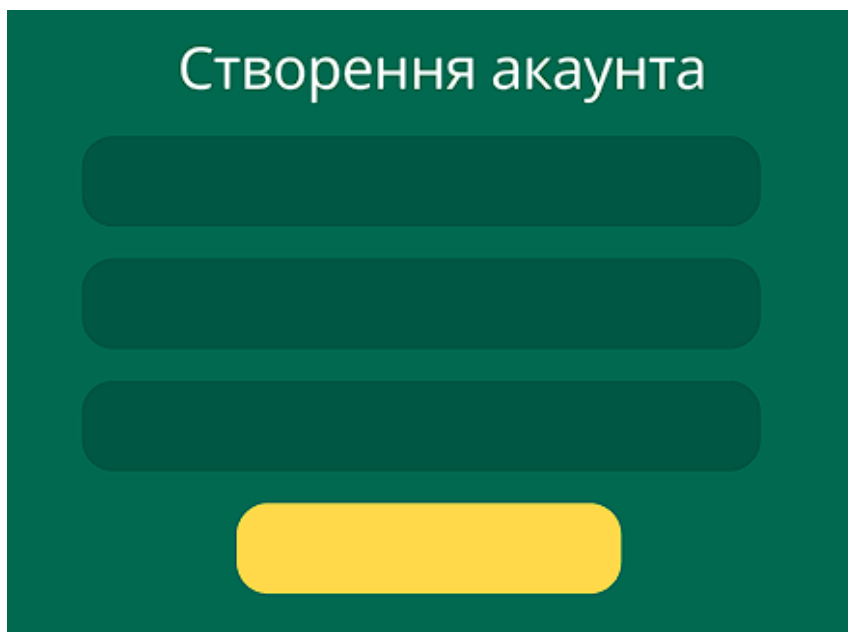


Рисунок 3.8 – Мокап сторінки реєстрації

3.3 Фронтенд реалізація

Для розробки користувацького інтерфейсу (UI) онлайн-платформи для обміну валют, я застосував технології HTML, CSS і JavaScript. Кожна з цих технологій грає ключову роль у забезпеченні зручності та інтерактивності платформи. У цьому розділі детально описується процес реалізації фронтенду, включаючи створення HTML-структури, стилізацію за допомогою CSS, а також додавання інтерактивності через JavaScript.

HTML (HyperText Markup Language) є основою будь-якого веб-документа. Він використовується для створення структури веб-сторінок, визначення елементів, таких як заголовки, параграфи, таблиці, форми та інші блоки контенту.

Для платформи було розроблено кілька ключових сторінок, включаючи головну сторінку (рис 3.9), та сторінку обміну валют (3.10).

Після розробки макетів веб-сторінок розпочалася фаза реалізації фронтенду. Для цього було обрано технології HTML, CSS та JavaScript. HTML використовувався для створення структури сторінок, CSS – для стилізації та оформлення, а JavaScript – для реалізації інтерактивних елементів [7].

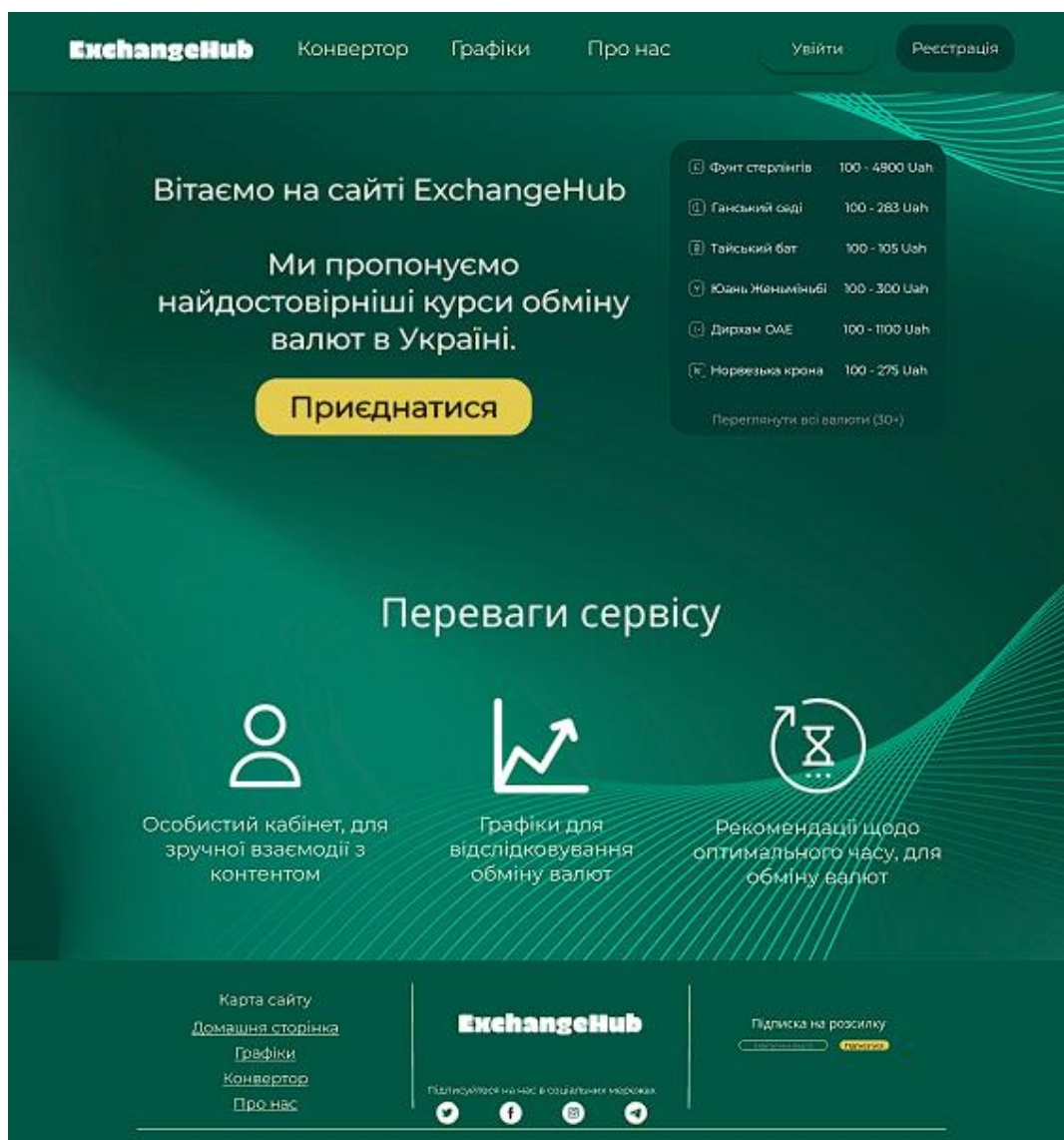


Рисунок 3.9 – Головна сторінка сайту

Базова структура HTML коду, яку я використовував для всіх сторінок на онлайн-платформі. Ця структура містить основні елементи, які будуть спільними для всіх сторінок, такі як заголовок, навігація, основний контент.

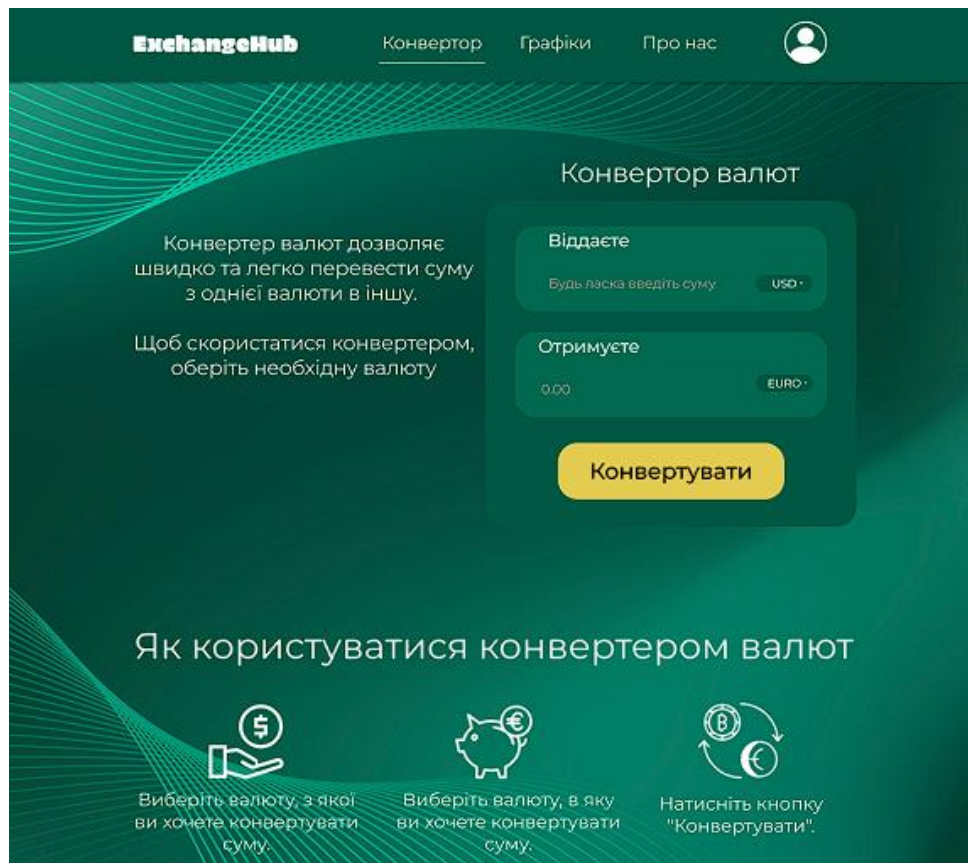


Рисунок 3.10 – Сторінка обміну валюти

Детальний опис структури HTML-документа для платформи включає кілька ключових елементів. У заголовку документа `<head>` встановлюється кодування документа як UTF-8 за допомогою тега `<meta charset="UTF-8">`. Налаштування сторінки для правильного відображення на мобільних пристроях виконується тегом:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Заголовок сторінки, який відображається у вкладці браузера, визначається тегом `<title>`. Підключення файлу зі стилями CSS здійснюється через тег:

```
<link rel="stylesheet" href="styles.css">
```

Для відкладеного виконання скрипту після завантаження HTML використовується тег:

```
<script src="script.js" defer></script>
```

У тілі документа `<body>` розміщуються основні елементи сайту. Тег `<header>` містить заголовок сайту та навігаційне меню, з основним заголовком сайту, представленим тегом `<h1>`. Навігаційний блок `<nav>` включає нумерований список `` посилань, кожне з яких є елементом списку ``, що веде на відповідні сторінки [8].

Основний контент сторінки розміщується в блоці `<main>`, де секція `<section>` змінюється залежно від конкретної сторінки. Нижній колонтитул `<footer>` містить інформацію про авторські права.

Також для платформи я створив окремий файл `styles.css`, який містить усі стилі, необхідні для різних елементів сторінок (Додаток Б).

JavaScript (JS) використовується для додавання динамічної поведінки та інтерактивності на веб-сторінки. JS дозволяє створювати багатий користувацький досвід шляхом реагування на події, такі як кліки миші, натискання клавіш та зміни форми. За допомогою JavaScript можна маніпулювати DOM (Document Object Model), (рис 3.10) змінюючи структуру, вміст та стилі HTML-документів на льоту, що дозволяє оновлювати частини веб-сторінки без необхідності її перезавантаження.

Окрім цього, JS надає можливість взаємодіяти з сервером через API-запити, використовуючи такі технології, як XMLHttpRequest або Fetch API. Це дозволяє завантажувати дані з сервера та відправляти їх назад без перезавантаження сторінки, що значно покращує продуктивність та користувацький досвід.

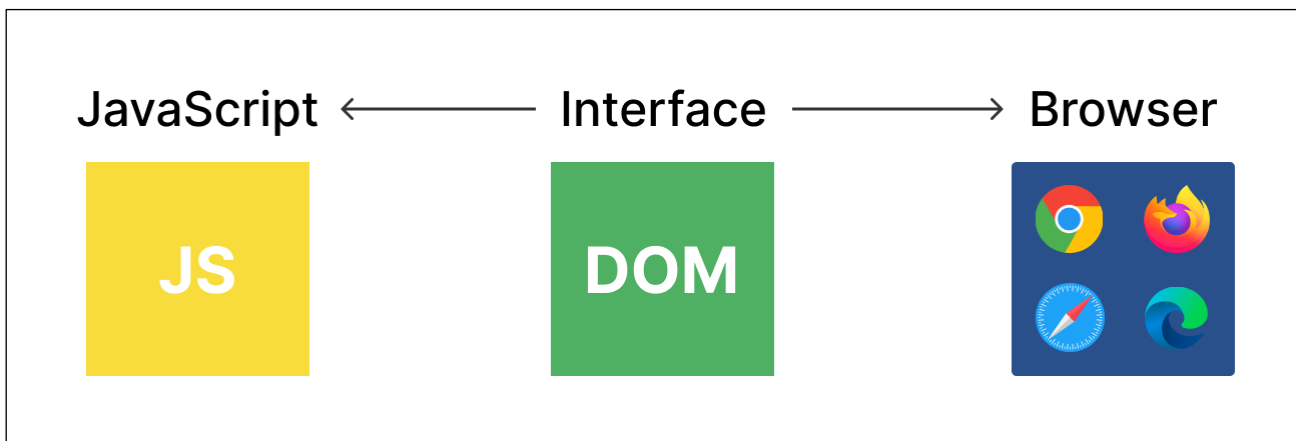


Рисунок 3.10 – Взаємодія DOM

Для платформи було створено файл `script.js` (Додаток Б), який містить функціонал для оновлення контенту, валідації форм та інтеграції з бекендом.

Код реалізації фронтенд частини детально описаний у (Додаток Б). В цьому додатку наведені всі необхідні файли, та загальні структури коду які я використовував у своєму проєкті [8].

3.4 Адаптація до різних пристроїв

Адаптація веб-платформи до різних пристроїв є важливим аспектом, що забезпечує зручний доступ користувачів незалежно від того, який пристрій вони використовують. Цей підрозділ розглядає підходи та технології, які я використав для досягнення оптимальної роботи платформи на різних екранах, включаючи десктопи, планшети та смартфони [9].

Використання адаптивного дизайну. У проєкті реалізовано адаптивний дизайн, який дозволяє автоматично змінювати макет сторінок відповідно до розміру екрану користувача. Основними принципами адаптивного дизайну є:

1. Медіа-запити: використання CSS медіа-запитів для застосування різних стилів до різних розмірів екранів. Наприклад, можна змінювати розташування елементів, розміри шрифтів та відступи.

2. Гнучкі зображення: використання зображень, які автоматично змінюють свої розміри відповідно до розмірів контейнера, щоб уникнути переповнення або деформації.

Приклад медіа-запитів у CSS. Для забезпечення адаптивності платформи я використовував медіа-запити. Ось приклад CSS-коду, який демонструє, як змінюються стилі залежно від розміру екрану:

```
/* Стили для десктопів (більше 900px) */
@media (min-width: 900px) {
  nav ul {
    display: flex;
    flex-direction: row;
    justify-content: space-between;
  }
  nav ul li {
    margin: 0 2em;
  }
}
```

Використання фреймворків для адаптивного дизайну. Для спрощення процесу створення адаптивних інтерфейсів я використовував CSS фреймворки, такі як Bootstrap. Цей фреймворк забезпечує готові компоненти та класи для створення адаптивних макетів, що значно прискорює процес розробки, підключення bootstrap:

```
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
```

Тестування адаптивності. Для забезпечення якісної адаптації до різних пристроїв я провів тестування на різних емуляторах та реальних пристроях. Зокрема, тестування охоплювало такі пристрої:

1. Смартфони: iPhone, Android-пристрої різних розмірів.
2. Планшети: iPad, Android-планшети.

3. Десктопи та ноутбуки: різні роздільні здатності екранів, включаючи Full HD, 2K та 4K.

Це тестування дозволило переконатися, що всі елементи інтерфейсу відображаються коректно та зручно на різних пристроях.

Висновки до розділу 3

У розділі 3 було детально розглянуто процес проектування, розробки та тестування онлайн-платформи для обміну валют. Архітектура платформи була спроектована з урахуванням вимог надійності та масштабованості, а також інтеграції з базою даних, яка забезпечує ефективне зберігання та управління даними. Користувацький інтерфейс розроблений з акцентом на зручність та інтуїтивність, що робить платформу доступною для широкого кола користувачів на різних пристроях. Після проведення тестування платформи були отримані позитивні результати щодо її ефективності та продуктивності, що підтвердило здатність системи витримувати значні навантаження та забезпечувати високу швидкість обробки запитів. Всі ці аспекти забезпечують стабільну, безпечну та зручну роботу платформи для кінцевих користувачів.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було успішно реалізовано онлайн-платформу для обміну валют та відстеження валютних курсів. Проект охоплював аналіз сучасних фінансових ринків, існуючих технологій обміну валют та розробку системи з урахуванням функціональних, технічних та безпекових вимог. Платформа забезпечує користувачам зручний та інтуїтивно зрозумілий інтерфейс для обміну валют та відстеження курсів у реальному часі, а також резервування валют у фізичних пунктах обміну.

Забезпечення безпеки даних було одним із ключових аспектів роботи. Реалізовані механізми аутентифікації та авторизації, шифрування даних у спокої та транзиті, а також системи резервного копіювання та моніторингу забезпечують високий рівень захисту платформи. Проведене тестування підтвердило ефективність та надійність системи.

Отримані результати демонструють, що платформа відповідає поставленим вимогам і стандартам, забезпечуючи безпеку, продуктивність та зручність користувачів. Подальший розвиток проекту передбачає розширення функціоналу та інтеграцію з додатковими фінансовими сервісами. Дана кваліфікаційна робота підкреслює важливість використання сучасних інформаційних технологій у фінансовій сфері та відкриває нові можливості для професійного зростання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Базилевич В.Д., Базилевич К.С., Вишнеvsька О.С. Фінансовий ринок: Підручник. — К.: Знання, 2008. URL: <https://repository.ldufk.edu.ua/handle/34606048/21146> (дата звернення: 28.03.2024)
2. Павловська Н.В., Лихачова О.І. Міжнародні валютні відносини: Навч. посібник. — Київ: Знання, 2011. URL: <https://dspace.onu.edu.ua/server/api/core/bitstreams/30f5ee6f-7f5d-48f0-ab42-8f087df4b81c/content> (дата звернення: 28.03.2024)
3. Дюран М., Гребельний М. Інтернет-технології в банківській сфері. — Київ: Либідь, 2010. URL: <https://ir.kneu.edu.ua/server/api/core/bitstreams/90576c62-9d7a-46b8-a69f-09f1bc3963b4/content> (дата звернення: 28.03.2024)
4. Сорокін І.А., Ковальчук В.В. Інформаційні системи і технології в економіці: Підручник. — К.: Знання, 2011. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/2677115a-ac4e-49a5-bcda-297bface0c05/content> (дата звернення: 30.03.2024)
5. Сосна І.В. Інформаційна безпека в комп'ютерних системах та мережах. — Київ: Либідь, 2012. URL: <https://repository.kpi.kharkov.ua/server/api/core/bitstreams/26a79adc-66e5-4f2e-b6e3-8396ee15f499/content> (дата звернення: 30.03.2024)
6. Плєскач В.Л., Савченко І.В. Інформаційні системи і технології на підприємствах. — К.: Либідь, 2010. URL: <https://www.westudents.com.ua/knigi/210-nformatsyn-sistemi-tehnolog-na-rdprimstvah-pleskach-vl.html> (дата звернення: 02.04.2024)
7. Kurose J.F., Ross K.W. Computer Networking: A Top-Down Approach. — Pearson, 2017. URL: <https://github.com/ece-cohort/Computer-Networking-A-Top-Down-Approach-7th-Edition> (дата звернення: 02.04.2024)
8. Shostack A. Threat Modeling: Designing for Security. — Wiley, 2014. URL: <https://archive.org/details/threatmodelingde0000shos> (дата звернення: : 24.04.2024)

10. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. — Wiley, 2015. URL: <https://onlinelibrary.wiley.com/doi/epdf/10.1002/9781119183471.fmatter> (дата звернення: 26.04.2024)

11. Managing Risk and Information Security – Malcolm W. Harkins, 2016 URL: <https://link.springer.com/book/10.1007/978-1-4842-1455-8> (дата звернення: 28.04.2024)

12. McDowell M. Information Security: Protecting the Global Enterprise. — Apress, 2012. URL: <https://archive.org/details/informationsecur0000ripk> (дата звернення: 01.05.2024)

13. Python Software Foundation. Python 3.9 Documentation. URL: <https://docs.python.org/3.9/index.html> (дата звернення: 28.04.2024)

14. Grinberg M. Flask Web Development: Developing Web Applications with Python. — O'Reilly Media, 2018. URL: https://books.google.com.ua/books/about/Flask_Web_Development.html?id=VKRWAwAAQBAJ&redir_esc=y (дата звернення: 14.05.2024)

15. O'Hara R., Heinz G. The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise. — Addison-Wesley Professional, 2015. URL: <https://www.oreilly.com/library/view/flask-web-development/9781491991725/> (дата звернення: 14.05.2024)

16. Ferguson N., Schneier B., Kohno T. Cryptography Engineering: Design Principles and Practical Applications. — Wiley, 2010. URL: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118722367.fmatter> (дата звернення: 14.05.2024)

17. Anderson R. Security Engineering: A Guide to Building Dependable Distributed Systems. — Wiley, 2020. URL: <https://onlinelibrary.wiley.com/doi/book/10.1002/9781119644682> (дата звернення: 22.05.2024)

18. Armstrong T. JavaScript: The Definitive Guide. — O'Reilly Media, 2020. URL: https://archive.org/details/javascriptdefini0000flan_6edi (дата звернення: 23.05.2024)

19. Resig J., Bibeault B., and Maras R. Secrets of the JavaScript Ninja. — Manning Publications, 2016. URL: <https://www.manning.com/books/secrets-of-the-javascript-ninja-second-edition?query=secrets> (дата звернення: 24.05.2024)

20. Welling L., Thomson L. PHP and MySQL Web Development. — Addison-Wesley Professional, 2016. URL: <https://archive.org/details/phpmysqlwebdevel0000well> (дата звернення: 24.05.2024)

ДОДАТКИ

Додаток А

App.py: Основний файл додатку Flask

```

from flask import Flask
from routes import user_routes, transaction_routes, exchange_rate_routes
from database import init_db

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///currency_exchange.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

init_db(app)

# Register blueprints
app.register_blueprint(user_routes)
app.register_blueprint(transaction_routes)
app.register_blueprint(exchange_rate_routes)

if __name__ == '__main__':
    app.run(debug=True)

```

Database.py: Скрипт для створення та управління базою даних

```

from flask_sqlalchemy import SQLAlchemy

db = SQLAlchemy()

class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    password_hash = db.Column(db.String(120), nullable=False)

class Transaction(db.Model):

```

```

    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'),
nullable=False)
    amount = db.Column(db.Float, nullable=False)
    currency = db.Column(db.String(10), nullable=False)
    timestamp = db.Column(db.DateTime, nullable=False)

class ExchangeRate(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    currency = db.Column(db.String(10), nullable=False)
    rate = db.Column(db.Float, nullable=False)
    timestamp = db.Column(db.DateTime, nullable=False)

def init_db(app):
    db.init_app(app)
    with app.app_context():
        db.create_all()

```

Routes.py: Реалізація маршрутів для API

```

from flask import Blueprint, request, jsonify
from database import db, User, Transaction, ExchangeRate
from werkzeug.security import generate_password_hash, check_password_hash
from datetime import datetime

user_routes = Blueprint('user_routes', __name__)
transaction_routes = Blueprint('transaction_routes', __name__)
exchange_rate_routes = Blueprint('exchange_rate_routes', __name__)

# User routes
@user_routes.route('/register', methods=['POST'])
def register():
    data = request.get_json()
    username = data['username']
    password = generate_password_hash(data['password'])

    new_user = User(username=username, password_hash=password)
    db.session.add(new_user)

```

```

db.session.commit()

return jsonify({'message': 'User registered successfully'})

@user_routes.route('/login', methods=['POST'])
def login():
    data = request.get_json()
    user = User.query.filter_by(username=data['username']).first()
    if user and check_password_hash(user.password_hash,
data['password']):
        return jsonify({'message': 'Login successful'})
    else:
        return jsonify({'message': 'Invalid credentials'}), 401

# Transaction routes
@transaction_routes.route('/transaction', methods=['POST'])
def create_transaction():
    data = request.get_json()
    new_transaction = Transaction(
        user_id=data['user_id'],
        amount=data['amount'],
        currency=data['currency'],
        timestamp=datetime.utcnow()
    )
    db.session.add(new_transaction)
    db.session.commit()

    return jsonify({'message': 'Transaction created successfully'})

@transaction_routes.route('/transactions/<int:user_id>', methods=['GET'])
def get_transactions(user_id):
    transactions = Transaction.query.filter_by(user_id=user_id).all()
    return jsonify([
        {
            'id': t.id,
            'amount': t.amount,
            'currency': t.currency,
            'timestamp': t.timestamp
        } for t in transactions])

```

```
# Exchange rate routes
@exchange_rate_routes.route('/exchange_rate', methods=['POST'])
def create_exchange_rate():
    data = request.get_json()
    new_rate = ExchangeRate(
        currency=data['currency'],
        rate=data['rate'],
        timestamp=datetime.utcnow()
    )
    db.session.add(new_rate)
    db.session.commit()

    return jsonify({'message': 'Exchange rate added successfully'})

@exchange_rate_routes.route('/exchange_rates', methods=['GET'])
def get_exchange_rates():
    rates = ExchangeRate.query.all()
    return jsonify([
        {
            'currency': r.currency,
            'rate': r.rate,
            'timestamp': r.timestamp
        } for r in rates])
```


Додаток Б

Index.html – основний файл для html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Онлайн-платформа для обміну валют</title>
  <link rel="stylesheet" href="styles.css">
  <script src="script.js" defer></script>
</head>
<body>
  <header>
    <h1>Онлайн-платформа для обміну валют</h1>
    <nav>
      <ul>
        <li><a href="/">Головна</a></li>
        <li><a href="/exchange">Обмін валют</a></li>
        <li><a href="/charts">Графіки курсів</a></li>
        <li><a href="/login">Увійти</a></li>
        <li><a href="/register">Зареєструватися</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <!-- Основний контент сторінки -->
    <section>
      <!-- Вміст цієї секції буде змінюватися залежно від
конкретної сторінки -->
    </section>
  </main>
  <footer>
    <p>&copy; 2024 Онлайн-платформа для обміну валют</p>
  </footer>
</body>
</html>
```

Style.css – основний файл для css

```
*,
*::before,
*::after {
    padding: 0;
    margin: 0;
    border: 0;
    box-sizing: border-box;
}
a {
    text-decoration: none;
    color: inherit;
}
li {
    list-style: none;
}
img {
    vertical-align: top;
}
h1,
h2,
h3,
h4,
h5,
h6 {
    font-weight: inherit;
    font-size: inherit;
}
html,
body {
    line-height: 1;
    height: 100%;
}
/* Загальні стилі */
body {
    font-family: Arial, sans-serif;
```

```
margin: 0;
padding: 0;
background-color: #f5f5f5;
}

header {
background-color: #007bff;
color: white;
padding: 10px 0;
}

header h1 {
text-align: center;
margin: 0;
}

nav ul {
display: flex;
justify-content: center;
list-style: none;
padding: 0;
}

nav ul li {
margin: 0 15px;
}

nav ul li a {
color: white;
text-decoration: none;
}

footer {
text-align: center;
padding: 10px 0;
background-color: #343a40;
color: white;
position: fixed;
bottom: 0;
width: 100%;
```

```
}
```

Script.js – основний файл для js

```
document.addEventListener('DOMContentLoaded', function() {
  const form = document.querySelector('#exchange-form');
  form.addEventListener('submit', function(event) {
    event.preventDefault();
    const amount = document.querySelector('#amount').value;
    const fromCurrency = document.querySelector('#from-
currency').value;
    const toCurrency = document.querySelector('#to-currency').value;

    fetch(`/api/exchange?amount=${amount}&from=${fromCurrency}&to=${toCurrenc
y}`)
      .then(response => response.json())
      .then(data => {
        document.querySelector('#result').textContent =
`Результат обміну: ${data.result}`;
      })
      .catch(error => {
        console.error('Error:', error);
      });
  });
});
```

Інтеграція коду:

```
fetch('/api/currency-rates')
  .then(response => response.json())
  .then(data => {
    const ratesTable = document.querySelector('#rates-table');
    data.rates.forEach(rate => {
      const row = document.createElement('tr');
      row.innerHTML = `
        <td>${rate.currency}</td>
        <td>${rate.rate}</td>
```

```
        `;  
        ratesTable.appendChild(row);  
    });  
})  
.catch(error => {  
    console.error('Error:', error);  
});
```



метадані

Заголовок

Створення онлайн-платформи для обміну валют, та відстеження валютних курсів

Автор

Твердохліб М. Я. Науковий керівник / Експерт

підрозділ

King Danylo University

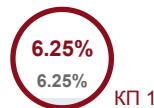
Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		34

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

**25**

Довжина фрази для коефіцієнта подібності 2

7773

Кількість слів

61294

Кількість символів

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	Асистент Вчителя Інформатики 12/8/2023 Borys Grinchenko Kyiv Metropolitan University (Кафедра комп'ютерних наук)	22	0.28 %
2	https://krs.chmnu.edu.ua/jspui/bitstream/123456789/2969/1/%D0%9C%D1%96%D0%BB%D1%94%D0%B2%20%D0%92%D0%BB%D0%B0%D0%B4%D0%B8%D1%81%D0%BB%D0%B0%D0%B2%20%D0%91%D0%9A%D0%A0.pdf	20	0.26 %
3	Асистент Вчителя Інформатики 12/8/2023 Borys Grinchenko Kyiv Metropolitan University (Кафедра комп'ютерних наук)	20	0.26 %
4	https://blog.learnhub.africa/2023/02/19/building-apis-with-python-a-comprehensive-guide/	19	0.24 %