

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко

підпис

«__» _____ 20__р.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

**«АНАЛІЗ КРИПТОВАЛЮТНИХ БІРЖ ТА ЇХНІХ
СИСТЕМ ТОРГІВ З ВИКОРИСТАННЯМ REACT ТА
REDUX»**

Спеціальність «Інженерія програмного забезпечення»

121 – КРМ – 608м.21810120

Здобувач

_____ І. А. Ревва

підпис

«__» _____ 20__р.

Керівник д-р техн. наук, проф.

_____ І. М. Журавська

підпис

«__» _____ 20__р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри

_____ Є. О. Давиденко

«_____» _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра

Видано студенту групи 608 факультету комп'ютерних наук

_____ Ревві Ігорю Андрійовичу _____

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

_____ Аналіз криптовалютних бірж та їхніх систем торгів з використанням React та Redux _____

Затверджена наказом по ЧНУ від «11» листопада 2023 р. № 234

2. Строк представлення кваліфікаційної роботи «_____» _____ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є ретельний аналіз криптовалютних бірж та їхніх систем торгів, розробка функціональної системи з використанням React та Redux для візуалізації даних і взаємодії з ринками.

4. Перелік питань, що підлягають розробці

- дослідження предметної галузі;
- визначення проблем та пошук рішень;
- моделювання та проектування системи;

- розробка функціональної системи для аналізу сучасних біржових систем;
- впровадження технологій React та Redux;
- тестування системи для аналізу;
- висновки щодо результатів розробки.

5. Перелік графічних матеріалів

Презентація.

Керівник роботи д-р техн. наук, проф. Журавська Ірина Миколаївна

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Ревва Ігор Андрійович

(прізвище, ім'я, по батькові студента)

(підпис)

Дата видачі завдання «___» _____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Аналіз криптовалютних бірж та їхніх систем торгів з використанням React та Redux

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників МКР	08.10.2023	08.10.2023	
2	Отримання завдання на виконання МКР	09.10.2023	12.10.2023	
3	Складання календарного плану роботи на весь період виконання МКР	15.10.2023	16.10.2023	
4	Отримання завдання на передатестаційну практику	21.10.2023	23.10.2023	
5	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	27.11.2023	25.12.2023	
6	Розробка звіту з переддипломної практики	26.12.2023	26.12.2023	
7	Виконання МКР: аналіз криптовалютних бірж, аналіз їх систем торгів, проектування та моделювання системи, розробка системи	28.12.2023	07.02.2024	
8	Попередній захист МКР на засіданні комісії кафедри	08.02.2024	08.02.2024	
9	Доробка та остаточне оформлення МКР	09.02.2024	16.02.2024	
10	Подання МКР рецензенту	17.02.2024	18.02.2024	
11	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	19.02.2024	19.02.2024	
12	Захист МКР перед екзаменаційною комісією (ЕК)	26.02.2024	27.02.2024	

Розробив студент Ревва Ігор Андрійович
(прізвище, ім'я, по батькові) _____ (підпис)
« ____ » _____ 20__ р.

Керівник роботи д-р техн. наук, проф. Журавська Ірина Миколаївна
(посада, прізвище, ім'я, по батькові) _____ (підпис)
« ____ » _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної роботи магістра
«Аналіз криптовалютних бірж та їхніх систем торгів
з використанням React та Redux»

Студент 608м гр.: Ревва Ігор Андрійович

Керівник: д-р техн. наук, проф. Журавська І. М.

Актуальність даної роботи полягає в швидкому розвитку ринку криптовалют та зростанні потреби у вдосконаленні інструментів для аналізу даних криптовалютних бірж. Система призначена для забезпечення користувачів детальною, оновлюваною в реальному часі інформацією, що дозволяє здійснювати обґрунтовані торгові рішення.

Об'єктом дослідження кваліфікаційної роботи є процес розробки системи для аналізу криптовалютних бірж та їхніх торгових механізмів.

Предметом дослідження кваліфікаційної роботи є основні методи та засоби створення системи за допомогою найсучасніших інструментальних засобів.

Мета кваліфікаційної роботи полягає в комплексному аналізі криптовалютних бірж та їхніх систем торгів з використанням сучасних технологій веброзробки, зокрема бібліотеки React та патерну управління станом Redux.

Практичне значення отриманих результатів: використання розробленої системи для аналізу криптовалютних бірж для збору, оброблення та візуалізацію даних у режимі реального часу, що у подальшому дозволить приймати обґрунтовані торгові рішення.

Робота пройшла **апробацію** під час XXVI Всеукраїнської науково-практичної конференції «Могилянські читання» (Миколаїв, 06–10 листопада 2023 р.).

Публікації. Основні положення магістерської роботи опубліковані у збірнику матеріалів XXVI Всеукраїнської науково-практичної конференції «Могилянські читання–2023».

Кваліфікаційна робота магістра складається з вступу, чотирьох розділів, висновків та додатків.

У вступі визначається актуальність теми, формулюється мета та задачі дослідження, об'єкт та предмет дослідження.

Перший розділ присвячений теоретичному аналізу ринку криптовалют, особливостям функціонування криптовалютних бірж та методам аналізу даних на цих біржах.

У другому розділі розглядається поняття системи та їхні архітектурні характеристики, з акцентом на використання технологій React та Redux для розробки клієнтської частини. Також, у цьому розділі обґрунтовуються та обираються технології для розробки системи.

У третьому розділі розглядаються архітектурні підходи до розв'язання проблеми та моделювання об'єкту та предмету роботи. Було представлено декілька UML-діаграм, які можуть представляти систему для аналізу криптовалютних бірж та їх особливостей.

У четвертому фінальному розділі проектується та створюється сервіс для системи. Проводиться детальний огляд та взаємодія користувача з інтерфейсом системи.

В результаті виконання кваліфікаційної роботи магістра було реалізовано систему для аналізу криптовалютних бірж та їхніх систем торгів.

Кваліфікаційна робота магістра викладена на 77 сторінок, вона містить 4 розділи, 31 рис., 27 джерел в переліку посилань, 2 додатка.

Ключові слова: *криптовалютні біржі, система торгів, вебсервіс, JavaScript, TypeScript, SPA, React, Redux, Next.js, Firebase.*

ABSTRACT
of the Master's Thesis
"Analysis of cryptocurrency exchanges and their trading systems
using React and Redux"

Student: Revva Ihor

Supervisor: Dr. Sc. (Engin.), Professor Zhuravska I. M.

The relevance of this work lies in the rapid development of the cryptocurrency market and the growing need to improve tools for analyzing cryptocurrency exchange data. The system is designed to provide users with detailed, real-time information that allows them to make informed trading decisions.

The object of the qualifying work is the process of developing a system for analyzing cryptocurrency exchanges and their trading mechanisms.

The subject of the qualification work is the basic methods and means of creating a system using the most modern tools.

The aim of the qualification work is to comprehensively analyze cryptocurrency exchanges and their trading systems using modern web development technologies, in particular the React library and the Redux state management pattern.

Practical significance of the results: the use of the developed system for analyzing cryptocurrency exchanges to collect, process and visualize data in real time, which will further allow making informed trading decisions.

The work was tested during the XXVI All-Ukrainian Scientific and Practical Conference "Mohyla Readings" (Mykolaiv, November 06–10, 2023).

Publications. The main provisions of the master's thesis were published in the proceedings of the XXVI All-Ukrainian Scientific and Practical Conference "Mohyla Readings-2023".

A master's thesis consists of an introduction, four chapters, conclusions, and appendices.

The introduction defines the relevance of the topic, formulates the purpose and objectives of the study, the object and subject of the study.

The first chapter is devoted to the theoretical analysis of the cryptocurrency market, the peculiarities of cryptocurrency exchanges and methods of data analysis on these exchanges.

The second chapter discusses the concept of systems and their architectural characteristics, with a focus on the use of React and Redux technologies for developing the client side. Also, this section justifies and selects technologies for system development.

The third chapter discusses architectural approaches to solving the problem and modeling the object and subject of work. Several UML diagrams were presented that can represent a system for analyzing cryptocurrency exchanges and their features.

In the fourth and final chapter, a service for the system is designed and created. A detailed review and interaction of the user with the system interface is carried out.

As a result of the master's thesis, a system for analyzing cryptocurrency exchanges and their trading systems was implemented.

The master's thesis is presented on 77 pages, it contains 4 chapters, 31 figures, 27 sources in the list of references, 2 appendices.

Keywords: *cryptocurrency exchanges, trading system, webservice, JavaScript, TypeScript, SPA, React, Redux, Next.js, Firebase.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІЗ ТЕХНІЧНИХ ХАРАКТЕРИСТИК БІРЖОВИХ СИСТЕМ.....	8
1.1 Архітектура біржових систем.....	8
1.2 Системи безпеки в біржових системах.....	10
1.3 Аналіз сучасних трендів у біржових системах.....	11
1.4 Аналіз аналогічних систем для аналізу криптовалютних бірж.....	12
1.5 Формування специфікації вимог до ПЗ інформаційної системи.....	14
Висновки до розділу 1.....	21
2 ТЕХНОЛОГІЇ ТА ПІДХОДИ ДО ПРОЄКТУВАННЯ І СТВОРЕННЯ СИСТЕМИ ДЛЯ АНАЛІЗУ КРИПТОВАЛЮТНИХ БІРЖ ТА ЇХНІХ СИСТЕМ ТОРГІВ.....	22
2.1 Поняття системи та їхні архітектурні характеристики.....	24
2.2 Огляд і аналіз існуючих методів і засобів вирішення завдань КРМ... ..	27
2.3 Обґрунтування та вибір технологій для розробки системи.....	29
2.4 Функціональні компоненти інформаційно-довідкової системи.....	37
Висновки до розділу 2.....	38
3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ.....	40
3.1 Моделювання моделі об'єкту та предмету роботи.....	41
3.2 Класи та методи для вирішення задачі.....	48
3.3 Архітектура системи.....	51
Висновки до розділу 3.....	54
4 РОЗРОБКА ТА ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ.....	55
4.1 Проєктування та створення сервісів для системи.....	55
4.2 Детальний огляд та взаємодія з інтерфейсом системи.....	58
4.3 Тестування розробленої системи.....	66

Висновки до розділу 4	70
ВИСНОВКИ.....	71
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	73
ДОДАТОК А Лістинг коду системи.....	76
ДОДАТОК Б Апробація кваліфікаційної роботи магістра	82

ПЕРЕЛІК СКОРОЧЕНЬ

- БД – база даних
ВЗ – вебзастосунок
КРМ – кваліфікаційна робота магістра
ПЗ – програмне забезпечення
ПК – персональний комп'ютер
ШІ – штучний інтелект
- AES – Advanced Encryption Standard
API – Application Programming Interface
CDN – Content Delivery Network
DDoS – Distributed Denial-of-Service attack
PaaS – Platform as a Service
SPA – Single-Page Application
SQL – Structured Query Language
UML – Unified Modeling Language

ВСТУП

Криптовалютний ринок став ключовим аспектом сучасної фінансової системи, привертаючи увагу як інвесторів, так і науковців. З появою й еволюцією криптовалют та застосуванням технологій блокчейну, створення та обмін цифровими активами отримали новий рівень відкритості й децентралізації. Однак, разом зі зростанням популярності криптовалют виникає потреба в розвинутих та надійних інструментах для торгівлі та аналізу ринку.

Кваліфікаційна робота магістра (КРМ) присвячена аналізу криптовалютних бірж та їхніх систем торгів з використанням сучасних технологій, зокрема бібліотеки React та управління станом застосунку за допомогою Redux. У роботі буде розглянуто архітектури, функціональності та ефективності вебінтерфейсів криптовалютних бірж на основі зазначених технологій.

Сучасний стан ринку криптовалют та його динаміка визначають важливі завдання для інвесторів, трейдерів і розробників. Криптовалютні ринки характеризуються високою волатильністю й швидкими змінами цін. Торгівля на таких ринках вимагає постійного аналізу та моніторингу, щоб вчасно реагувати на зміни й уникнути фінансових втрат. Використання технологій React та Redux дозволяє створити інтерактивні та ефективні інтерфейси для аналізу та моніторингу криптовалютних ринків, а також полегшити розробку та підтримку великих систем торгів.

У цьому контексті, дослідження включає вивчення роботи великих криптовалютних бірж, їхніх технічних характеристик та пропозицій для користувачів. Аналіз існуючих технологій торгів, включаючи алгоритмічну торгівлю та роботизовані стратегії, допоможе визначити найкращі практики та можливості для впровадження технологій React та Redux.

Об'єктом дослідження кваліфікаційної роботи є процес розробки системи для аналізу криптовалютних бірж та їхніх торгових механізмів.

Предметом дослідження кваліфікаційної роботи є основні методи та засоби створення системи за допомогою найсучасніших інструментальних засобів.

Мета кваліфікаційної роботи полягає в комплексному аналізі криптовалютних бірж та їхніх систем торгів з використанням сучасних технологій веброзробки, зокрема бібліотеки React та патерну управління станом Redux.

Для досягнення поставленої мети необхідно виконати наступні **завдання**:

- аналіз технічних характеристик біржових систем: розгляд архітектурних особливостей, принципів безпеки, масштабованості та протоколів обміну на криптовалютних біржах з метою визначення їхнього потенціалу та обмежень;
- аналіз алгоритмів торгів та стратегій: вивчення та оцінка ефективності існуючих алгоритмів торгів та стратегій, а також розгляд можливостей їхнього інтегрування в системи криптовалютних бірж з використанням технологій React та Redux;
- розробка та оптимізація вебінтерфейсів: розробка та реалізація динамічних та ефективних вебінтерфейсів для користувачів криптовалютних бірж з використанням бібліотеки React для забезпечення зручності взаємодії та високого рівня користувацького досвіду;
- використання Redux для управління вебінтерфейсом: впровадження та оптимізація за допомогою Redux для забезпечення консистентності даних та вдосконалення досвіду використання вебінтерфейсів;
- тестування вебінтерфейсу біржової системи є критично важливим етапом в розробці, оскільки саме через цей інтерфейс користувачі взаємодіють із системою, здійснюють торгівельні операції та отримують доступ до фінансової інформації. Функціональне тестування спрямоване на визначення правильності роботи ключових функцій, таких як введення та виведення даних, виконання торгових операцій та взаємодія з графічним інтерфейсом.

Практичне значення отриманих результатів: використання розробленої системи для аналізу криптовалютних бірж для збору, оброблення та візуалізацію даних у режимі реального часу, що у подальшому дозволить приймати

обґрунтовані торгові рішення. Завдяки цій системі, трейдери зможуть вибирати оптимальні стратегії на основі об'єктивної інформації про ринкову динаміку. Система надає можливість визначати точки входу та виходу, що сприяє ефективному управлінню позиціями та мінімізації ризиків.

Моніторинг ринкових тенденцій в режимі реального часу допомагає учасникам ринку швидко реагувати на зміни та адаптувати свої стратегії відповідно до нових умов.

Система також може використовуватися для підвищення ефективності інвестування. Інвестори отримують інструменти для аналізу та оцінки ризиків, пов'язаних з різними фінансовими інструментами, що дозволяє їм краще розуміти можливі ризики та вигоди і вчасно реагувати на зміни в ринкових умовах.

Робота пройшла **апробацію** під час XXVI Всеукраїнської науково-практичної конференції «Могилянські читання» (Миколаїв, 06–10 листопада 2023 р.).

Публікації. Основні положення магістерської роботи опубліковані у збірнику матеріалів XXVI Всеукраїнської науково-практичної конференції «Могилянські читання–2023» [10].

1 АНАЛІЗ ТЕХНІЧНИХ ХАРАКТЕРИСТИК БІРЖОВИХ СИСТЕМ

Криптовалютні ринки та цифрові активи стають все більш актуальними та значущими в сучасному світі фінансів та інвестицій. Швидкий та часто непередбачуваний розвиток криптовалют і блокчейн-технологій породжує необхідність аналізу ринкових даних для успішної торгівлі та прийняття обґрунтованих інвестиційних рішень.

Криптовалютні ринки характеризуються високою волатильністю й швидкими змінами цін. Торгівля на таких ринках вимагає постійного аналізу та моніторингу, щоб вчасно реагувати на зміни й уникнути фінансових втрат. Точність та оперативність аналізу ринкових даних є критичними для трейдерів та інвесторів. Інформація про ціни, торгові обсяги й інші показники грають важливу роль у прийнятті торговельних рішень.

Трейдерам і аналітикам криптовалютних ринків потрібні інструменти, які дозволяють ефективно й своєчасно аналізувати та прогнозувати рух цін криптовалют для досягнення успішних торгових стратегій.

Тому в цьому розділі проаналізуємо технічні аспекти криптовалютних біржових систем та розглянемо архітектурні рішення, протоколи обміну, системи безпеки та масштабованості для виявлення їх впливу на ефективність та надійність бірж.

1.1 Архітектура біржових систем

Архітектуру біржових систем можна поділити на декілька видів, а саме:

а) **Централізована архітектура** – це архітектура біржових систем базується на ідеї концентрації всіх торговельних операцій та обробки даних у єдиному центрі. В такій системі, біржовий сервер відіграє ключову роль, обробляючи заявки та управляючи транзакціями. Цей підхід спрощує контроль над обсягами даних та забезпечує єдність управління, але може призводити до невдачі та обмежує масштабованість.

б) **Децентралізована архітектура** – це архітектура у якої використовується розподілене управління та рішення. Система розміщена на різних серверах, що розташовані у різних частинах світу. Кожен сервер відповідає за обробку торгових партій або групи користувачів. Децентралізована архітектура забезпечує більшу стійкість до відмов та підвищує масштабованість, але може призводити до ускладнень у контролі та координації.

в) **Ланцюжкова архітектура** – це архітектура орієнтована на поєднання переваг централізованої та децентралізованої моделей. У цій системі окремі блокчейн-ланцюги використовуються для обробки транзакцій та підтвердження їхньої правомірності, але підтверджені дані потрапляють у централізовану систему для фінальної узгодженості та виконання.

У контексті архітектури обговорюється питання масштабованості. Використання горизонтального та вертикального масштабування для забезпечення ефективної роботи під великими обсягами торгів та збільшенням кількості користувачів. Розглядаються техніки, такі як шарування, кешування та використання технологій контейнеризації для оптимальної масштабованості.

Обговорюється використання технологій, що дозволяють ефективно керувати контентом на біржі. Системи кешування, розподілений зберігання та використання Content Delivery Network (CDN) для забезпечення швидкості та доступності даних для користувачів.

Оцінка якості мережі та її впливу на біржову систему. Розгляд питань пропускної здатності, латентності та маршрутизації, щоб забезпечити швидку та надійну передачу даних.

Обговорюється впровадження систем забезпечення резервування для попередження відмов та техніки відновлення для відновлення нормального функціонування системи після непередбачених ситуацій.

У розділі враховуються сучасні тренди в області архітектури біржових систем, такі як використання технологій блокчейн, розумних контрактів та інших інновацій для поліпшення безпеки та ефективності.

1.2 Системи безпеки в біржових системах

Одним із ключових елементів систем безпеки є процес автентифікації, що забезпечує перевірку ідентичності користувача перед входом в систему. Для забезпечення високого рівня безпеки використовуються двофакторна аутентифікація та біометричні технології. Авторизація контролює рівень доступу користувачів до конкретних ресурсів та функціоналу системи.

Шифрування використовується для захисту конфіденційної інформації, щоб уникнути несанкціонованого доступу під час передачі даних між серверами та клієнтами, а також при зберіганні даних. Застосовуються алгоритми шифрування з високою ступенем надійності, такі як Advanced Encryption Standard (AES).

Системи виявлення та запобігання інцидентам безпеки використовують аналіз поведінки та алгоритми виявлення аномалій для реагування на потенційно небезпечні події. Інтегровані системи виявлення вразливостей слідкують за можливими ділянками атак та надають засоби їхнього нейтралізації.

Для забезпечення стабільності роботи біржових систем використовуються засоби захисту від розподіленого атак, таких як DDoS. Вони виявляють та фільтрують трафік, який може викликати перевантаження серверів, забезпечуючи при цьому доступ користувачам.

Системи аудиту відслідковують всі акції користувачів та події в системі. Це допомагає виявляти та розслідувати потенційно небезпечні дії, а також служить засобом відновлення системи після інциденту.

Враховується захист від внутрішніх загроз, що можуть виникнути внаслідок недбалого використання або намагань зловмисників отримати несанкціонований доступ. Обмеження привілеїв та внутрішні системи виявлення зловживань є частиною стратегії захисту [6].

У випадку використання блокчейн та смарт-контрактів, системи безпеки включають заходи для перевірки та валідації коду смарт-контрактів, щоб уникнути вразливостей та використання невірної коду.

1.3 Аналіз сучасних трендів у біржових системах

У розділі враховуються сучасні тренди в області архітектури біржових систем, такі як використання технологій блокчейн, розумних контрактів та інших інновацій для поліпшення безпеки та ефективності.

Технологія блокчейн стає важливим елементом біржових систем, надаючи безпеку та прозорість у торгівельних операціях. Системи, що використовують блокчейн, дозволяють миттєві та недорогі транзакції, а також гарантують невідмінність та непереборність даних.

Використання смарт-контрактів на блокчейні дозволяє автоматизувати та узгоджувати угоди без посередництва. Це підвищує ефективність та надійність торгівельних операцій, забезпечуючи автоматичне виконання умов угод.

Зростання кількості альтернативних криптовалют: крім біткоіна, на ринку з'являється все більше альтернативних криптовалют, таких як Ethereum, Ripple, та інші. Це створює нові можливості для торгівлі та розширює спектр інструментів для інвесторів.

Також можна зазначити інтеграцію штучного інтелекту (ШІ) та машинного навчання. ШІ та машинне навчання використовуються для прогнозування та аналізу ринкових тенденцій. Автоматизовані системи на основі ШІ можуть аналізувати величезний обсяг даних та приймати швидкі та точні рішення.

Однією з характеристик сучасних бірж є зростання популярності мобільних платформ призвело до розширення можливостей мобільної торгівлі. Користувачі можуть здійснювати операції та відстежувати ринкові події зі своїх смартфонів, що робить торгівлю більш доступною та зручною.

Розумні аналітичні інструменти, засновані на алгоритмах та штучному інтелекті, допомагають трейдерам зрозуміти та прогнозувати ринкові тенденції. Вони підтримують прийняття рішень.

Токенізація активів за допомогою блокчейну дозволяє представляти реальні активи, такі як нерухомість чи мистецтво, у вигляді токенів. Це сприяє ліквідності та доступності для нових класів інвесторів.

1.4 Аналіз аналогічних систем для аналізу криптовалютних бірж

Сучасні сайти для аналізу криптовалютних бірж і їхніх систем торгів включають різноманітні платформи, кожна з яких має свої унікальні функції та переваги.

CoinMarketCap – один з найвідоміших ресурсів у галузі криптовалют, який пропонує широкий спектр інформації про різні криптобіржі, включаючи обсяги торгівлі, рейтинги бірж та інші важливі показники.

Переваги:

- велика база даних криптовалют та бірж;
- надає інформацію про обсяги торгів, ринкову капіталізацію та інші важливі показники.

Недоліки:

- основний фокус на статистичних даних, менше уваги до аналітичних звітів;
- інформація може бути занадто загальною для досвідчених трейдерів.

Binance Academy – це платформа, яка забезпечує детальний аналіз та навчальні матеріали з криптовалют, включаючи фундаментальний і технічний аналіз. Це корисний ресурс для тих, хто хоче зрозуміти як аналізувати ринок криптовалют та блокчейн-дані.

Переваги:

- великий вибір навчальних матеріалів та статей про криптовалюту;
- покриває теми фундаментального та технічного аналізу.

Недоліки:

- може відчуватися як промоційний інструмент для Binance;
- недостатньо інформації про інші криптобіржі.

Traders Union – вони пропонують рейтинги криптобірж з оцінками, які базуються на аналізі їхніх основних параметрів роботи, що дозволяє визначити загальний рівень якості умов для криптотрейдингу, які надаються цими біржами.

Переваги:

- надає рейтинги та оцінки криптобірж;
- може бути корисним для порівняння та вибору бірж для трейдингу.

Недоліки:

- зосередженість на рейтингах та оцінках може не завжди відображати повну картину;
- менше уваги до навчальних ресурсів та аналітики ринку.

Кожна з цих платформ має свої особливості, і вибір залежить від конкретних потреб користувача. Наприклад, якщо вам потрібна загальна інформація про ринок і рейтинги бірж, CoinMarketCap може бути найкращим вибором. Для глибшого аналізу та навчальних матеріалів, Binance Academy може бути більш відповідним. Traders Union надає конкретні оцінки та детальні огляди різних криптобірж, що може бути корисним для прийняття обґрунтованих торгових рішень.

Порівнюючи CoinMarketCap, Binance Academy та Traders Union, можна побачити, що кожна з цих платформ служить різним цілям і має свої унікальні особливості.

CoinMarketCap вирізняється своєю здатністю надавати широкий спектр даних про різні криптовалюти та біржі, включаючи обсяги торгів та ринкову капіталізацію. Цей ресурс ідеально підходить для отримання загального огляду ринку криптовалют та стеження за основними тенденціями. Його інтуїтивно зрозумілий інтерфейс та оновлення в реальному часі роблять його дуже зручним для користувачів. Однак, він може не задовольнити потреби більш досвідчених трейдерів, які шукають більш глибокий аналіз та індивідуальні рекомендації.

Binance Academy, з іншого боку, є більш освітньо орієнтованим ресурсом. Ця платформа надає детальну інформацію та навчальні матеріали, які стосуються всіх аспектів криптовалют, включаючи фундаментальний та технічний аналіз. Це

робить її ідеальним місцем для новачків, які хочуть навчитися основам криптовалютного трейдингу, а також для досвідчених користувачів, які прагнуть поглибити свої знання. Однак, оскільки це пов'язано з платформою Binance, інформація може бути схильна до суб'єктивного представлення цієї конкретної біржі.

Traders Union займає інший підхід, зосереджуючись на наданні детальних рейтингів та оцінок різних криптобірж. Цей ресурс є незамінним для тих, хто шукає порівняльний аналіз бірж, з урахуванням таких факторів, як зручність інтерфейсу, комісії та інші важливі параметри торгівлі. Однак, як і в інших випадках, він може не повністю враховувати всі нюанси кожної біржі та зосереджується більше на кількісних оцінках, ніж на якісному аналізі.

У підсумку, вибір між цими трьома платформами залежить від конкретних потреб користувача. Порівняючи CoinMarketCap, Binance Academy та Traders Union, можна побачити, що кожна з цих платформ служить різним цілям і має свої унікальні особливості.

1.5 Формування специфікації вимог до ПЗ інформаційної системи

Під час створення вебсайтів, застосунків та інших програмних продуктів існує багато різних методик розробки. Вибір конкретної методології залежить від ряду факторів: особливостей проєкту, бюджету, встановлених термінів та особистих уподобань розробника.

Система була створена за допомогою фреймворку Next.js. Next.js – це фронтенд фреймворк з відкритим кодом, побудований на основі React, що спрямований на покращення процесу розробки вебзастосунків [2]. Він забезпечує зручність у створенні як статичних, так і динамічних сайтів на базі React. Основні характеристики Next.js включають рендеринг на стороні сервера, автоматизоване розбиття коду, підвищену швидкість завантаження сторінок та маршрутизацію, засновану на файловій системі. Крім того, Next.js пропонує переваги для SEO,

забезпечуючи попереднє рендерування сторінок, що поліпшує їх індексацію пошуковими системами.

Фреймворк гармонійно поєднується з численними розробницькими інструментами та сервісами, включаючи Webpack, Babel та Node.js, надаючи розширені можливості для створення застосунків. Next.js також має функцію гарячого перезавантаження, що дозволяє розробникам відразу бачити зміни без необхідності перезавантаження сторінки, що робить розробку швидшою та більш продуктивною. Особливістю Next.js є його універсальність, оскільки він підходить як для малих, так і для великих корпоративних проєктів, забезпечуючи масштабованість і високу продуктивність.

1.5.1 Опис фреймворку Next.js

Головна відмінність фреймворку Next полягає у використанні мови програмування TypeScript, яка є перевіреною та надійною.

Конфігурація проєкту в Next.js виконується легко та інтуїтивно (рис. 1.1).

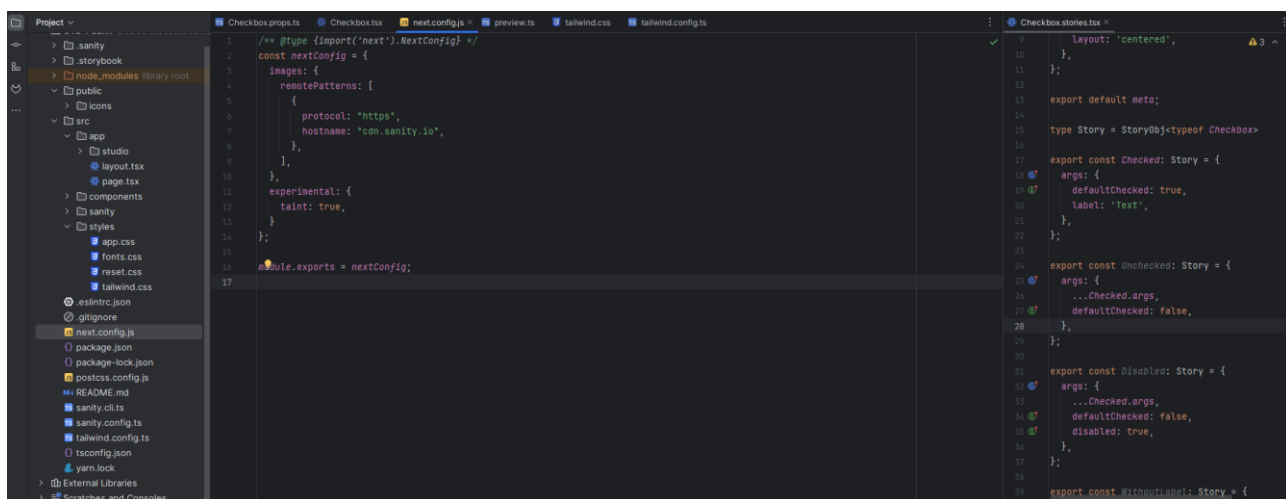


Рисунок 1.1 – Налаштування конфігурації Next.js для проєкту

Недоліки:

- складність освоєння через систему впровадження залежностей (Dependency Injection) та використання Observable;
- для забезпечення належної роботи та високої швидкості потрібно інвестувати час у тестування та оптимізацію;

- наявність детальної документації з великою кількістю дрібних аспектів, які потрібно вивчити;
- реалізація динамічних компонентів виявляється складним і довгим процесом.

Переваги:

- підтримка від Microsoft та Google;
- інструменти для розробників;
- реактивне програмування за допомогою RxJS;
- вбудована система впровадження залежностей;
- шаблони, базовані на розширенні HTML;
- вбудована кросбраузерна підтримка Shadow DOM;
- підтримка HTTP, WebSockets, Service Workers для кросбраузерності;
- мінімальна необхідність у додаткових налаштуваннях.

1.5.2 Опис основного API

Для створення системи буде використана RapidAPI. RapidAPI є одним з найбільших ринків Application Programming Interface (API), що забезпечує розробникам зручний доступ до тисяч різноманітних API. Цей сервіс дозволяє знаходити, підключати та інтегрувати API різних постачальників у свої проєкти.

Ось деякі ключові особливості та переваги RapidAPI:

- доступні тисячі API, що охоплюють різноманітні області, від фінансових та спортивних даних до соціальних медіа та погоди;
- RapidAPI надає єдиний інтерфейс для взаємодії з різними API, що спрощує процес розробки та інтеграції;
- платформа надає інструменти для моніторингу використання API, що дозволяє розробникам стежити за викликами API та виявляти потенційні проблеми;
- RapidAPI спрощує управління ключами доступу до API, дозволяючи централізовано керувати автентифікацією та дозволами;

- сервіс надає кодові фрагменти для інтеграції API у різні мови програмування, що робить його доступним для широкого кола розробників;
- RapidAPI дозволяє вибрати різні тарифні плани, від безкоштовних з обмеженим доступом до преміальних з розширеними можливостями.

RapidAPI є популярним вибором серед розробників, оскільки він забезпечує швидкий доступ до широкого спектру API, спрощує процес інтеграції та надає корисні інструменти для моніторингу та управління.

Coinranking API, доступний через платформу RapidAPI, є потужним інструментом для отримання даних про криптовалюту. Цей API надає вичерпну інформацію про ринкові тенденції, ціни, капіталізацію та інші ключові метрики, пов'язані з криптовалютами (рис. 1.2).

The screenshot displays the RapidAPI interface for the 'GET Get coins' endpoint. On the left, there is a description: 'Get a list of coins. Coins are by default ordered by their rank, which - somewhat simplified - means that they are ordered on marketcap. The response not only returns a list of coins, but also statistics regarding the requested list, such as the volume in the last 24 hours.' Below this is a 'Full documentation' link and a 'Personal Account' dropdown menu for 'Igor Revva'. The 'RapidAPI App' field is set to 'default-application_8691823' and is marked as 'REQUIRED'. The 'Request URL' field is set to 'rapidapi.com' and is also marked as 'REQUIRED'. On the right, the 'Code Snippets' section shows a Node.js Axios example with the following code:

```
const axios = require('axios');

const options = {
  method: 'GET',
  url: 'https://coinranking1.p.rapidapi.com/coins',
  params: {
    referenceCurrencyUuid: 'yhjMzLPhuID1',
    timePeriod: '24h',
    'tiers[0]': '1',
    orderBy: 'marketCap',
    orderDirection: 'desc',
    limit: '50',
    offset: '0'
  },
  headers: {
    'X-RapidAPI-Key': '94968496c8msh5ab3ad37155593ap15934ejsn829ba0ab870c',
    'X-RapidAPI-Host': 'coinranking1.p.rapidapi.com'
  }
};
```

Рисунок 1.2 – Приклад отримання даних про криптовалюту

API включає деталізовані дані про окремі криптовалюти, включаючи їх ранжування, доступність на ринках, та зміни ціни за різні періоди та надає інформацію про загальну ринкову капіталізацію криптовалют та об'єми торгів на різних біржах.

Цей API є цінним ресурсом для розробників фінтех-застосунків, інвестиційних платформ, а також для аналітиків і ентузіастів криптовалют, оскільки він забезпечує швидкий доступ до важливих даних про крипторинок.

1.5.3 Опис сучасних баз даних та бази даних для системи аналізу криптовалютних бірж

Основною вимогою до баз даних є збереження інформації без втрат. Дивно, але деякі бази даних, навіть ті, що були перевірені часом, не завжди відповідають цьому критерію, демонструючи проблеми навіть у простих сценаріях. Розробники прагнуть до систем, де дані дублюються мінімум на трьох серверах для забезпечення надлишковості під час технічного обслуговування одного з них.

Іншою вимогою до баз даних є їхня здатність працювати з сучасним обладнанням. З часом технології розвиваються, і бази даних повинні адаптуватися до змін, таких як збільшення кількості ядер в процесорах, інтеграція оперативної пам'яті в чіпи та зниження вартості флеш-пам'яті. Однак, швидкість світла залишається незмінною, що впливає на швидкість мережевих передач. Тому сучасні бази даних повинні підтримувати синхронну реплікацію в межах одного дата-центру та асинхронну між різними дата-центрами.

Аналіз баз даних ґрунтувався на даних від самих постачальників, незалежних тестах та реальних випадках використання. Були виключені вбудовані бази даних через відсутність автоматичної реплікації по мережі.

Комерційні SQL БД, такі як Microsoft SQL Server та Oracle Database, є відомими представниками цього класу. Вони перевірені часом і ефективно використовують можливості сучасного обладнання завдяки інноваціям, таким як in-memory tables і column stores. Обидві бази підтримують кластеризацію та мають розширені можливості мови SQL.

Обидві бази можуть ліцензуватися за моделлю «ціна за ядро процесора», що робить ціну незалежною від кількості користувачів..

Відкриті SQL БД, такі як MySQL і PostgreSQL, є популярним вибором для великої кількості завдань. Вони підтримують кластеризацію і використовуються у великих проєктах, дозволяючи міграцію між різними системами. Основним недоліком є необхідність ручного шардінгу і відсутність автоматичного ребалансування кластера.

NoSQL БД, хоча вважаються застарілими, продовжують бути важливими гравцями на ринку. Вони були створені для зберігання та обробки великої кількості неструктурованої інформації і залишаються критично важливими для збереження даних клієнтів.

Хмарні бази даних становлять окрему категорію, кожен з основних PaaS-провайдерів пропонує варіанти для зберігання структурованих даних.

NewSQL – це категорія баз даних, що використовують відомий SQL-сервер. Цей напрямок у розробці програмного забезпечення створив нову галузь розподілених баз даних, які використовують SQL для обробки запитів. Одним із видатних прикладів є Google Spanner, який гарантує linearizability, забезпечуючи таким чином глобальну послідовність записів. Це важливо для синхронізації часу на серверах по всьому світу, для чого Google використовує атомні годинники та GPS-приймачі.

Однак, атомні годинники залишаються платними для загального використання, тому розробники Spanner створили аналогічну базу даних із меншими гарантіями щодо транзакцій, але достатньо надійну для більшості застосунків. CockroachDB відрізняється надійністю при відмовах ПЗ або зв'язку між основними центрами. Вона пропонує сучасні транзакції та автоматичне оновлення кластера при втраті пакетів, використовуючи популярну SQL-мову запитів. Її недоліками є відсутність текстових індексів і складність у вирішенні деяких проблем.

Move code to data – це підхід, де серверні застосунки отримують запити від клієнтів та обробляють їх, звертаючись за даними до сервера бази даних. При великій кількості даних, їх передача через мережу може зайняти багато часу. Тому

виникає потреба перемістити обробку даних безпосередньо у базу даних, з використанням технологій, таких як Apache Hadoop.

Існує також зростаючий інтерес до інтеграції серверів застосунків і баз даних для OLTP-навантажень в реальному часі, з відповідними технологіями, такими як Tarantool. Її архітектура «cooperative multitasking» без блокувань є привабливою, хоча розробка на ній вимагає більше зусиль. Однак, мова програмування Lua, яка є популярною в ігровій індустрії, все ще має обмежену аудиторію серед розробників.

БД – це спеціалізована структура, створена для зберігання, зміни та обробки взаємопов'язаної інформації, особливо великих обсягів даних. Бази даних є ключовими для створення динамічних вебсайтів, які обробляють значні масиви інформації. Вони проєктуються та розробляються за допомогою серверних мов програмування або на платформах CMS, відрізняючись від сайтів, створених статично в HTML, без готових шаблонів даних.

Недоліки MySQL:

- сфера застосування обмежена порівняно з NoSQL-рішеннями;
- Firebase ставить певні обмеження на вибірку даних і запис в декілька місць одночасно;
- робота з деякими структурами даних у Firebase може бути незручною.

Переваги MySQL:

- це база даних, а не бекенд, тому розробка застосунків на Firebase без серверної частини поки що неможлива;
- це NoSQL-база з усіма її перевагами і недоліками.

Динамічні вебсайти можуть бути реалізовані дуже швидко, оскільки результати взаємодії між скриптами та БД після запиту від клієнта надсилаються на вебсервер.

Нарешті, БД повинна бути гнучкою та масштабованою, щоб витримувати зростаючі обсяги даних та запитів від користувачів системи. Це може вимагати

використання розподілених баз даних або технологій хмарного зберігання, залежно від специфічних потреб і обмежень проєкту.

Висновки до розділу 1

У цьому розділі проаналізовано найважливіші аспекти біржових систем, враховуючи їх архітектуру та систему безпеки, а також останні тренди. Враховуючи швидкий розвиток технологій, важливо враховувати необхідність постійного вдосконалення та адаптації біржових систем для продажу криптовалют.

Було проведено детальний огляд та аналіз наявних методів та інструментів, які використовуються для вирішення завдань у сфері криптовалютного ринкового менеджменту. Це включає оцінку різних підходів до аналізу даних ринку, торгових стратегій, моделей прогнозування, а також технологічних платформ, таких як блокчейн та інші інструменти для роботи з криптовалютами. Важливою частиною є аналіз ефективності та надійності цих методів у контексті швидкозмінного та непередбачуваного ринку криптовалют.

Coinranking API на RapidAPI є важливим інструментом для отримання детальної та актуальної інформації про криптовалютний ринок. Цей API вирізняється своєю здатністю надавати реальні дані про ціни криптовалют, історичні дані, інформацію про ринкову капіталізацію та об'єми торгів. Висока надійність, швидкість відгуку та легкість інтеграції роблять його ідеальним рішенням для розробників, аналітиків та інвесторів, які шукають глибоке розуміння криптовалютного ринку. Гнучкість тарифних планів також дозволяє використовувати Coinranking API в різних проєктах, від невеликих особистих досліджень до великих комерційних застосунків.

Специфікація вимог охоплює технічні характеристики, функціональні можливості, інтерфейс користувача, міру безпеки, а також вимоги до інтеграції з іншими системами або платформами. Важливим аспектом є врахування масштабованості та гнучкості системи, а також її здатності адаптуватися до змін умов ринку криптовалют.

2 ТЕХНОЛОГІЇ ТА ПІДХОДИ ДО ПРОЄКТУВАННЯ І СТВОРЕННЯ СИСТЕМИ ДЛЯ АНАЛІЗУ КРИПТОВАЛЮТНИХ БІРЖ ТА ЇХНІХ СИСТЕМ ТОРГІВ

На сьогоднішній день вебсайти та вебзастосунки часто стикаються з викликом надмірної кількості інформації, яка або подається користувачам некоректно, або, в деяких випадках, не подається взагалі. Для цього необхідно використати сучасні методи для розробки інтелектуальних та інформаційних систем. Програмування у веброзробці в основному зосереджене на інтеграції бази даних (БД) з вебзастосунками, які пропонують користувачам інтуїтивно зрозумілий інтерфейс, різноманітні складні форми, можливості для онлайн-розрахунків та інші функціональні особливості.

Якісно розроблені вебсайти зазвичай відповідають наступним критеріям:

- наявність набору сукупностей (підсистем), які знаходяться в тісній взаємодії, та в кінцевому результаті вибудовуються в цілісний проєкт;
- складність опису, а саме це велика кількість функціоналу, різність процесів, складової даних та взаємозв'язків між ними, що вимагає значного моделювання.

Вебсайт – це ефективний та простий спосіб комунікації в сучасну цифрову епоху. Як і інші інструменти, розроблені для забезпечення безпосередньої відповіді. Основна мета вебсайту полягає в тому, щоб спочатку привернути увагу відвідувача, а потім заохочувати його до використання послуг, покупки товарів та інших дій. На жаль, під час створення сайту, не всі дотримуються цього правила, що, в свою чергу, зазвичай призводить до того, що користувачеві не цікавий сайт з першої ж сторінки, отже він не затримується на сайті надовго і залишає його, ледь зайшовши.

Кожен вебсайт, не важливо якої він буде тематики, має містити на головній сторінці тільки важливу для кінцевого користувача інформацію, та давати можливість легко орієнтуватися по потрібним користувачу матеріалам і розділам,

тому ще це потенційний клієнт. Інакше сайт не буде ефективним і не матиме сенсу існування. Адже на даний час специфіка сайтів постійно удосконалюється для максимально зручного і простого користування.

Якщо приділити невелику кількість часу, то будь-який сайт зможе перетворитися з маленького та не надійного на більш продуктивний інструмент для бізнесу. Також, важливим фактором залишається те, що на потенційних клієнтів щодня зрушується велика кількість зайвої та набридливої інформації в поєднанні з різними рекламними повідомленнями, що в свою чергу призводять до великої конкуренції. Вебсайт, який здатний на те, щоб викликати увагу і зацікавленість підштовхує користувачів не лише відвідувати сторінки і залишити коментарі, але й надає довіру в очах користувача, на підставі якої йому в майбутньому захочеться знову відвідати вебсайт, а саме це підштовхує таких людей на те, щоб повідати про надзвичайний застосунок своєму другові або подрузі та іншим своїм знайомим.

Також, беручи до уваги специфіку даного сайту, варто розуміти, що сайт на таку тематику повинен бути не тільки привабливим візуально, але й максимально бути корисним для споживача. Адже мова йде про сферу надання сервісу з обслуговування автомобілів де надзвичайно велика конкуренція.

Отже, є декілька основних компонентів, які має містити подібний сайт.

Для початку, опис сайту – для чого він і на кого орієнтований. Головна сторінка повинна містити розділи, де є якомога більше інформації про діяльність компанії, розділ з контактами, в якому легко можна знайти потрібні контакти для зв'язку і подальшого діалогу між користувачем і установою. Кожен з розділів має бути максимально зручним для використання. Не буде зайвою сторінка користувача, де має міститись особистий кабінет, з якого споживач зможе безпосередньо робити різні маніпуляції із замовленнями.

Для успішної реалізації проєкту та подальшого його просування, сайт має бути детально описаний, побудовані функціональні та інформаційні моделі. Досвід, отриманий при виконанні КРМ вказує на те, що розробка подібного

проєкту дуже важка і тривала робота, яка вимагає великий спектр умінь та навичок від розробників, які займаються розробкою проєкту.

2.1 Поняття системи та їхні архітектурні характеристики

Система – це поєднання в собі сукупності, яка властива до апаратних засобів, які мають включають в собі комп'ютери та носії по типу смартфона чи планшету, що містять великий обсяг інформації із вже заздалегідь завантаженою на них інформацією. Також вони виконані безпосередньо з можливістю для запису та зчитування інформації у вигляді комп'ютерних застосунків, БД та інше. Інформаційна система, яка була виконана з можливістю обробки значної кількості інформації та команд для користувача системи та надання йому онлайн-сервісів, які побудовані за допомогою результатів обробки масової інформації та функцій [3].

Поняття системи в архітектурі та технологіях описує складний набір взаємопов'язаних компонентів, що працюють разом для досягнення певних цілей. Ключовими елементами будь-якої системи є її компоненти, їх взаємодія, інтерфейси, а також зовнішнє середовище, в якому система функціонує. Архітектурні характеристики системи визначають структуру, організацію та стратегію взаємодії її компонентів. Ці характеристики включають:

- модульність: система поділяється на незалежні модулі або компоненти, кожен з яких виконує певну функцію. Це полегшує розробку, тестування та управління системою;
- інтерфейси: вони визначають, як компоненти взаємодіють один з одним та з зовнішнім світом. Інтерфейси забезпечують стандартизований спосіб комунікації між компонентами, забезпечуючи їхню взаємну сумісність і координацію;
- масштабованість: це здатність системи ефективно працювати при зміні обсягу використання або вимог. Масштабовані системи можуть адаптуватися до зростаючих або зменшення навантажень без втрати продуктивності;

– гнучкість і адаптивність: система повинна бути спроектована так, щоб вона могла адаптуватися до змін в технологіях, вимогах користувачів чи зовнішньому середовищі;

– продуктивність: визначає, наскільки ефективно система використовує свої ресурси (такі як час, пам'ять, процесор) для виконання задач. Висока продуктивність важлива для задоволення вимог користувачів та забезпечення бажаного рівня обслуговування.

Ці архітектурні характеристики визначають здатність системи задовольняти потреби користувачів і ефективно функціонувати в динамічному технологічному ландшафті.

2.1.1 Класифікація з різновидів систем

Класифікація систем залежить від різноманітності критеріїв та перспектив. Одним із способів класифікації є розгляд за характером їхнього функціонування, яке може включати фізичні системи, такі як механічні чи електронні пристрої, і біологічні системи, до яких належать різноманітні живі організми. Інший підхід полягає у розгляді соціальних систем, які охоплюють людські взаємодії та організаційні структури, а також інформаційних систем, що займаються обробкою та зберіганням даних.

Також існують закриті та відкриті системи: закриті системи мають обмежену взаємодію з навколишнім середовищем, тоді як відкриті системи активно взаємодіють зі своїм середовищем. Ще один важливий критерій – це ступінь складності, за яким системи можуть бути простими, складаючись з небагатьох елементів та зв'язків, або складними, включаючи багато взаємопов'язаних компонентів та процесів.

Адаптивні системи відрізняються здатністю до змін та вдосконалення у відповідь на зовнішні впливи, на відміну від статичних систем, що залишаються незмінними протягом часу. Крім того, системи можуть класифікуватися за

призначенням, наприклад, продуктивні, управлінські, інформаційні, освітні та інші.

Важливість класифікації систем полягає у забезпеченні більш глибокого розуміння їхніх властивостей, поведінки та способів взаємодії, що є ключовим для ефективного проєктування, аналізу, управління та оптимізації цих систем у різних областях застосування.

2.1.2 Класифікація різновидів систем для користувачів

Динамічні та статичні системи – динамічні системи змінюють свій стан або поведінку з часом, відповідно до внутрішніх або зовнішніх впливів. Натомість статичні системи не змінюють своєї поведінки та структури з плином часу.

Автономні та контрольовані системи – автономні системи здатні функціонувати незалежно без зовнішнього управління або втручання. Контрольовані системи, навпаки, потребують зовнішнього керування чи впливу для своєї роботи.

Детерміновані та стохастичні системи – детерміновані системи мають чітко визначену поведінку, яка може бути передбачена на основі вхідних даних і внутрішніх правил. Стохастичні системи мають елементи випадковості або непередбачуваності у своїй поведінці, що залежить від різних випадкових факторів.

Розподілені та централізовані системи – у розподілених системах компоненти розташовані в різних місцях та взаємодіють через мережу. Централізовані системи мають один або декілька центрів керування, які координують і контролюють всі інші елементи системи.

Ієрархічні та мережеві системи – в ієрархічних системах елементи організовані у вигляді ієрархії, де кожен рівень має свою роль та обов'язки. Мережеві системи мають менш структурований, більш гнучкий взаємозв'язок між елементами, що дозволяє більш розподілену взаємодію та координацію.

Лінійні та нелінійні системи – лінійних системах вихідні сигнали прямо пропорційні вхідним. В нелінійних системах ця залежність більш складна, і вона може включати експоненціальні, логарифмічні або інші типи відносин.

Знання та розуміння різних типів систем і їх класифікації дозволяє більш ефективно проектувати, аналізувати та управляти системами в різних областях, включаючи інженерію, ІТ, управління, біологію та соціальні науки. Кожен тип системи має свої унікальні властивості та вимоги, що вимагають спеціалізованого підходу та розуміння для їх ефективного використання та управління.

Крім того, розуміння специфіки різних систем може допомогти у виявленні потенційних викликів та можливостей для інновацій та покращення. Особливо це актуально в сучасному світі, де технологічний прогрес та зростаюча складність соціальних та економічних систем вимагають глибокого розуміння та гнучкого підходу до системного аналізу та управління.

2.2 Огляд і аналіз існуючих методів і засобів вирішення завдань КРМ

PhpStorm представляє собою комерційний, крос-платформний інструмент для розробки PHP, включаючи інтегровану підтримку для цієї мови. PHP, як скриптова мова, переважно використовується для створення динамічних вебзастосунків та серверної розробки, що пояснює назву цієї середи. Крім того, PhpStorm також підтримує інші мови програмування. У рамках цього проєкту будуть використовуватися такі технології, як JSX, Tailwind, React, Redux і Git (рис. 2.1).

Недоліки PhpStorm:

- цей інструмент є платним, на відміну від безкоштовних аналогів, наприклад, Visual Studio Code або Eclipse;
- PhpStorm може споживати значну кількість системних ресурсів, що особливо помітно на застарілих або менш потужних комп'ютерах;
- через багатofункціональність середовища, новачкам може бути складно адаптуватися та швидко навчитися користуватися ним;

- інтерфейс програми може сприйматися як перенасичений через велику кількість інструментів і опцій, що може ускладнювати знаходження необхідних функцій;
- низький поріг входу в сферу розробки.

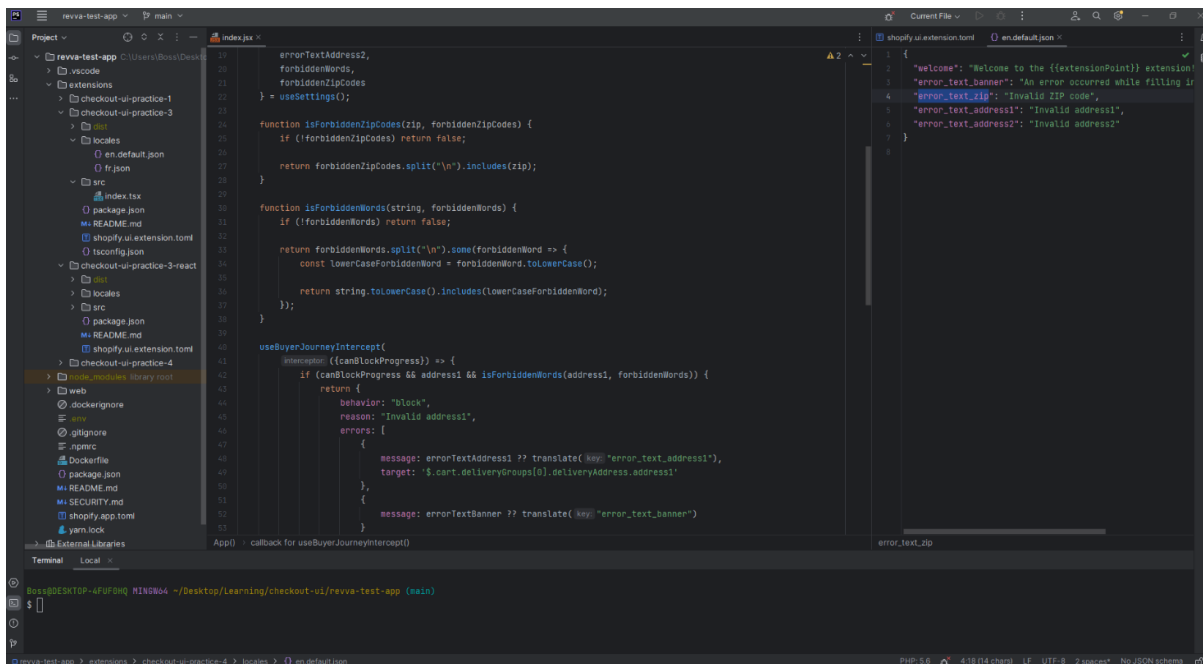


Рисунок 2.1 – Внутрішній інтерфейс середи розробки PhpStorm

Переваги PhpStorm:

- підтримка широкого спектру технологій: PhpStorm працює з PHP, HTML, CSS, JavaScript, TypeScript та іншими мовами, що робить його гнучким інструментом для веброзробки;
- розширені функції IDE для аналізу, рефакторингу, контекстних підказок, автоматичного завершення коду та інші специфічні для PHP можливості;
- вбудований дебагер PHP в PhpStorm допомагає легко знаходити і виправляти помилки в кодї;
- PhpStorm ефективно співпрацює з такими популярними PHP-фреймворками, як Laravel, Symfony, Zend Framework, Yii та інші;
- легка інтеграція з системами контролю версій, такими як Git, SVN, Mercurial;

– вбудовані засоби для роботи з SQL та базами даних, дозволяючи розробникам виконувати базоданні операції прямо з середовища розробки.

PhpStorm продовжує бути одним із лідерів серед інструментів для PHP-розробки через свою багатofункціональність і адаптивність. Проте, при виборі інструментарію для розробки, важливо зважати на як його сильні сторони, так і потенційні обмеження.

2.3 Обґрунтування та вибір технологій для розробки системи

У сучасному світі існує чимало нових та вже відомих фреймворків, які можуть бути використані для створення систем аналізу криптовалютних бірж. Фреймворк – це набір комерційних інструментів, що відкриває перед розробниками широкі можливості для створення програмного забезпечення. Він інтегрує в себе різноманітні наперед визначені класи та методи. До фреймворку можна додати специфічний для проєкту код, розширюючи його функціональність.

Серед переваг фреймворків можна виділити:

– економічна вигода полягає в тому, що багато фреймворків є безкоштовними та надають відкритий код, що спрощує розробку застосунків і знижує вартість кінцевого вебзастосунку;

– підвищена продуктивність є ключовою особливістю фреймворків. Вони істотно покращують ефективність розробки, зокрема, оптимізаційні процеси. Тому краще використовувати фреймворки з високою оптимізацією та добре розробленими шаблонами, ніж розробляти ці елементи самостійно;

– безпека є важливою перевагою популярних JS фреймворків. Вони зарекомендували себе не тільки завдяки значній підтримці спільноти розробників на GitHub, але й завдяки більш продуманій системі безпеки.

Давайте розглянемо найпопулярніші фреймворки та бібліотеки для фронтенду, які мають високий рейтинг сьогодні:

1) **React.js**. Цей відомий JS фреймворк [1] був запущений у 2013 році командою розробників, які також працювали над створенням відомих онлайн-

платформ, таких як Instagram, Facebook і WhatsApp. Він дозволяє легко та швидко створювати складні та масштабовані односторінкові застосунки(SPA). React особливо популярний для створення користувацьких інтерфейсів (рис. 2.2).

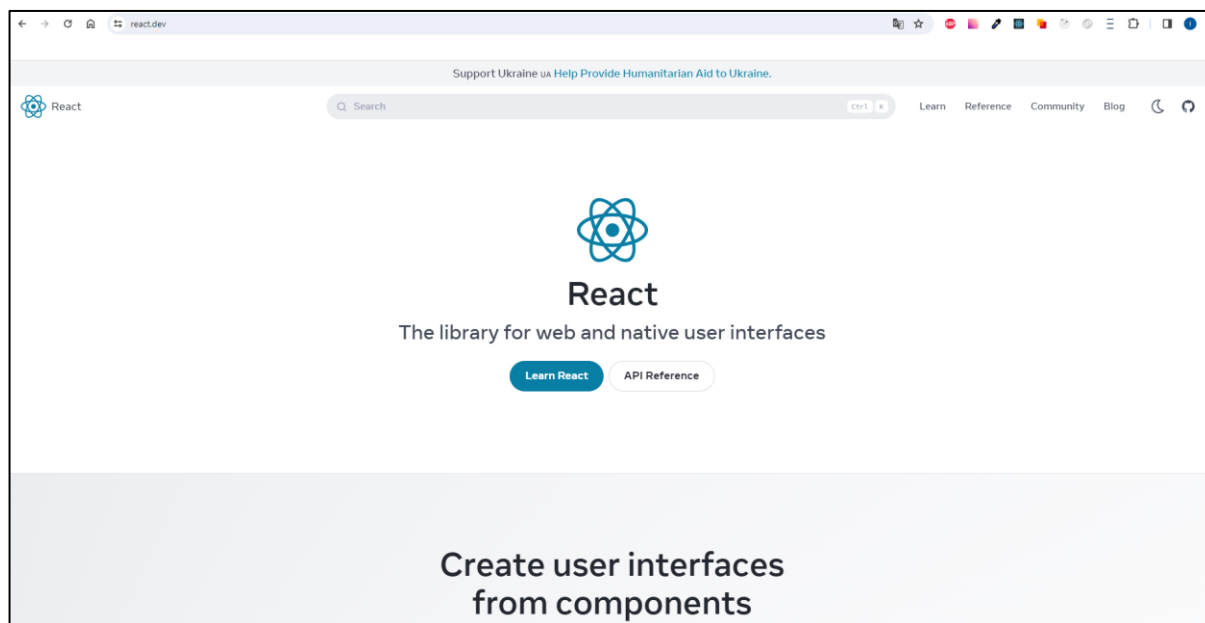


Рисунок 2.2 – Офіційне джерело React.js

Недоліки:

- неточний та досить повільний алгоритм віртуальною моделі документу.

Переваги:

- одноразове вивчення React дозволяє застосовувати його у різних проєктах;
- безкоштовний доступ та відкритий код;
- можливість повторного використання написаного коду;
- підтримка гнучкої віртуальної DOM для ефективного управління інтерфейсом.

Ця технологія обирається через її універсальність та ефективність у різних проєктах, забезпечену одноразовим вивченням React, можливість повторного використання коду, відкритий код та підтримку гнучкої віртуальної DOM. Незважаючи на недоліки, такі як неточний та повільний алгоритм віртуальної моделі документу, переваги значно перевищують ці обмеження.

2) **Angular.js.** Цей фреймворк [13] для розробки вебсайтів, створений відомою компанією Google, вже вважається дещо застарілим. Він найкраще підходить для створення динамічних вебпорталів і використовує HTML для структурування звичайних вебсторінок. Angular.js є незамінним інструментом не тільки для дизайнерів програмного забезпечення та веброзробників, але й для інших фахівців. Всі його версії швидко завоювали популярність серед розробників (рис. 2.3).

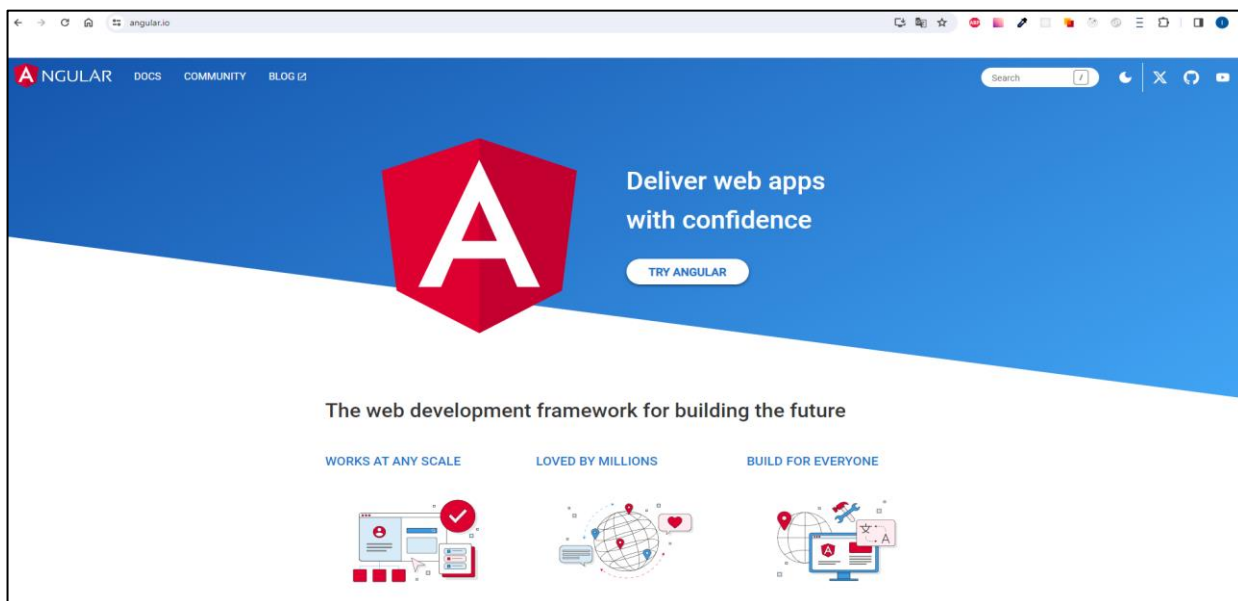


Рисунок 2.3 – Офіційне джерело AngularJS

Недоліки:

- вивчення може бути складним для тих, хто має обмежений досвід у розробці;
- Vue.js видається значно простішим для розробки вебзастосунків і їх архітектури;
- об'ємна документація фреймворку вимагає знайомства з численними концепціями;
- потребує розуміння складного асинхронного програмування при роботі з серверною частиною.

Переваги:

- велика кількість відкритого вихідного коду;
- можливість зберігання та повторного використання шаблонів коду;
- різноманітна підтримка елементів MVC;
- ефективне використання у середовищі Agile;
- багатий набір інструментів для тестування.

Незважаючи на переваги, ця технологія не обирається через складність вивчення для розробників з обмеженим досвідом, більш простий альтернативний вибір у вигляді Vue.js для розробки вебзастосунків, об'ємну документацію, що вимагає глибокого знайомства з багатьма концепціями, та складності асинхронного програмування.

3) Vue.js. Цей JS фреймворк користується популярністю в середовищі розробників, особливо для створення користувацьких односторінкових застосунків [4]. Розроблений з нуля для більш широкого використання в майбутньому, Vue.js інтегрує легкодоступні основні бібліотеки, зосереджені на рівні розробки, та відомі бібліотеки, що сприяють ефективній розробці масштабних односторінкових застосунків.

Недоліки:

- у шаблонізаторах runtime-середовища спостерігається чимало помилок;
- підхід, який використовується в React, виявляється більш адаптивним для розробки.

Переваги:

- стандартні компілятори, що використовуються за замовчуванням, є зручними у використанні;
- гнучкий перехід від бібліотеки до фреймворку під час розвитку проекту;
- зручність розробки односторінкових застосунків;
- поєднання легкості у читанні та написанні програмного коду.

Ця технологія не обирається через проблеми з помилками у шаблонізаторах runtime-середовища та тому, що підхід, використовуваний в React, виявляється

більш адаптивним для розробки. Ці недоліки переважають навіть такі переваги, як стандартні зручні компілятори, гнучкість переходу від бібліотеки до фреймворку, зручність розробки SPA, та легкість у читанні та написанні програмного коду.

4) Ember.js. Цей JS фреймворк, що з'явився у 2011 році, відомий своїм відкритим кодом та популярністю серед розробників. Він дозволяє ефективно створювати масштабовані односторінкові вебзастосунки із забезпеченням двостороннього зв'язку даних. Ember.js виділяється своїми функціональними можливостями у роботі з віртуальною моделлю документу на серверній частині та має значну підтримку від спільноти (рис. 2.4).

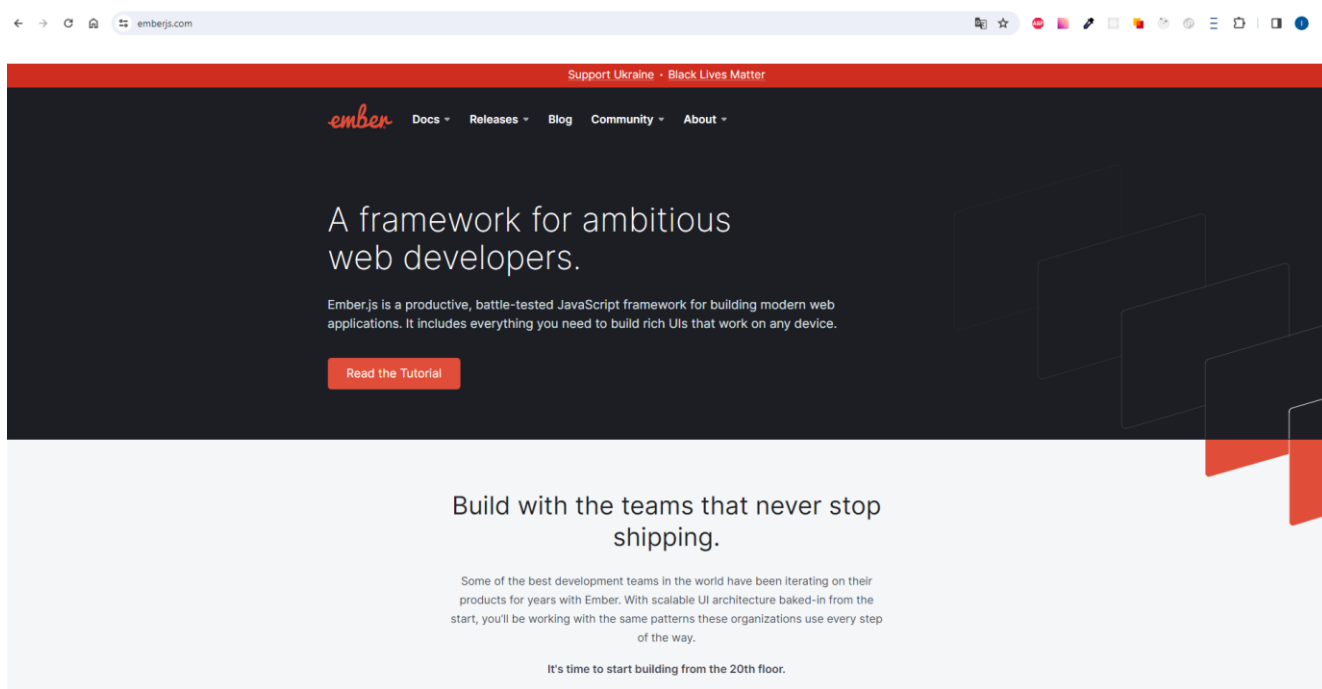


Рисунок 2.4 – Офіційне джерело Ember.js

Недоліки:

- суворо типізована структура в проєктах;
- брак стандартного комплексу елементів інтерфейсу користувача (UI/UX).

Переваги:

- простота та зручність у налаштуванні;
- охоплення широкого спектру інтерфейсів;
- наявність двостороннього зв'язку між компонентами.

Ця технологія не обирається через її сувору типізовану структуру, яка може ускладнити розвиток проєктів, та відсутність стандартного набору елементів інтерфейсу користувача, що обмежує гнучкість у дизайні.

5) jQuery. Ця бібліотека не є фреймворком, але зате є однією з найстаріших і найбільш популярних у світі. jQuery давно та міцно увійшла в екосистему мови програмування JavaScript. Ця бібліотека, хоч і вважається застарілою, надає розробникам альтернативні способи написання коду, що є коротшими за JavaScript, тим самим зменшуючи навантаження на вебсторінку. Використовуючи функції для роботи з віртуальною моделлю документу та інтеграцію з CSS, jQuery стає зручним інструментом для вирішення різноманітних завдань (рис. 2.5).

Недоліки:

- багато функцій jQuery, спрямованих на спрощення роботи з віртуальною моделлю документу, вже ефективно реалізовані в сучасному чистому JavaScript.

Переваги:

- jQuery використовується на більшості вебсайтів, забезпечуючи швидке виконання скриптів;
- сумісна з багатьма новітніми версіями браузерів;
- ідеально підходить для новачків, які розробляють прості застосунки.

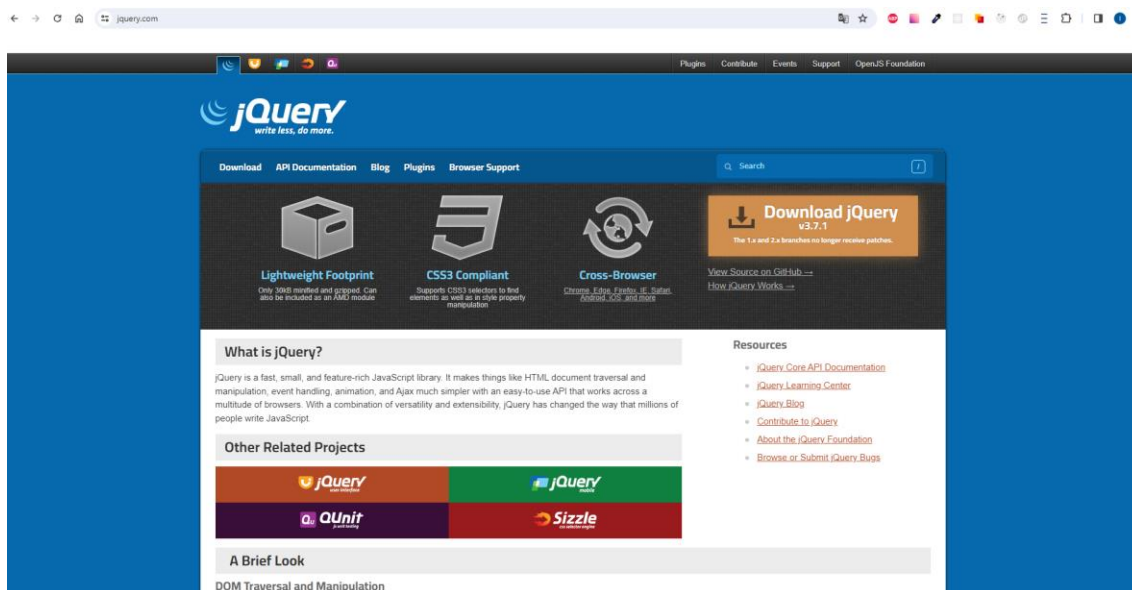


Рисунок 2.5 – Офіційне джерело jQuery

6) **Node.js**. Цей набір інструментів дозволяє створювати як стандартні, так і інноваційні застосунки. Він вирізняється своєю сучасністю, простотою використання та високою швидкістю. Компанії, такі як Google і Trello, є лише деякими з відомих прихильників цього інструментарію для розробки своїх онлайн-сервісів. Набір особливо ефективний у проєктах, пов'язаних з вводом/виводом даних та застосунками що потребують постійного обміну даними (рис. 2.6).

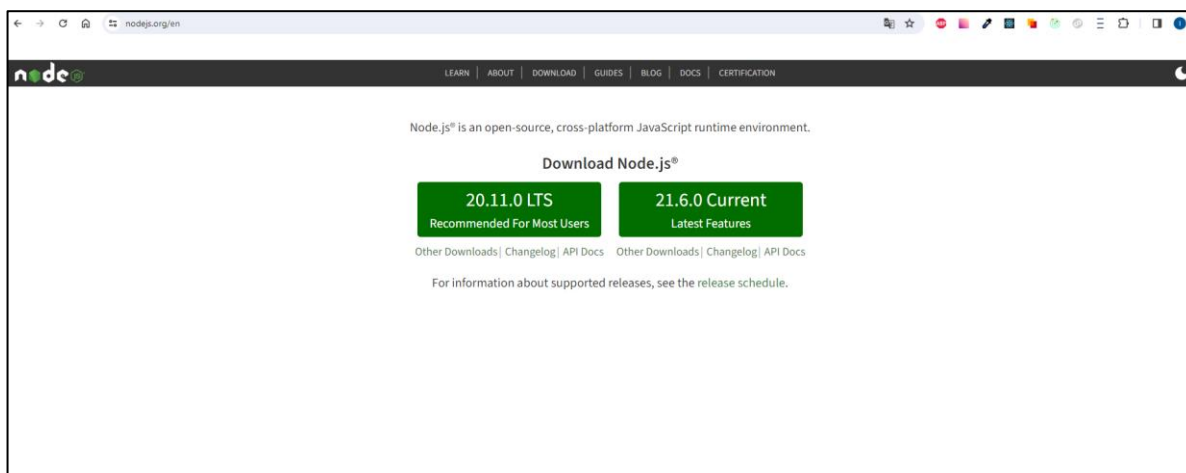


Рисунок 2.6 – Офіційне джерело Node.js

Недоліки:

- не найкращий вибір для складних обчислювальних процесів;
- розробка без попереднього тестування модульного пакета недоцільна.

Переваги:

- відрізняється високою швидкістю та легкістю у використанні;
- ефективно функціонує на різних онлайн-хостингах;
- підтримується на швидкодіючих серверах.

Ця технологія не обирається, оскільки вона не є найкращим вибором для складних обчислювальних процесів, а також через необхідність попереднього тестування модульного пакета перед розробкою. Ці недоліки переважають її переваги, такі як висока швидкість, легкість використання, ефективна робота на різних онлайн-хостингах та підтримка на швидкодіючих серверах.

7) **Meteor.js**. Ця система є оптимальним рішенням для розробки full-stack проєктів, оскільки вона включає багатий набір функцій, необхідних для реалізації різноманітних завдань. Вона забезпечує розробникам усе необхідне для створення фронтенду, розробки бекенд-застосунків та управління базами даних (рис. 2.7).

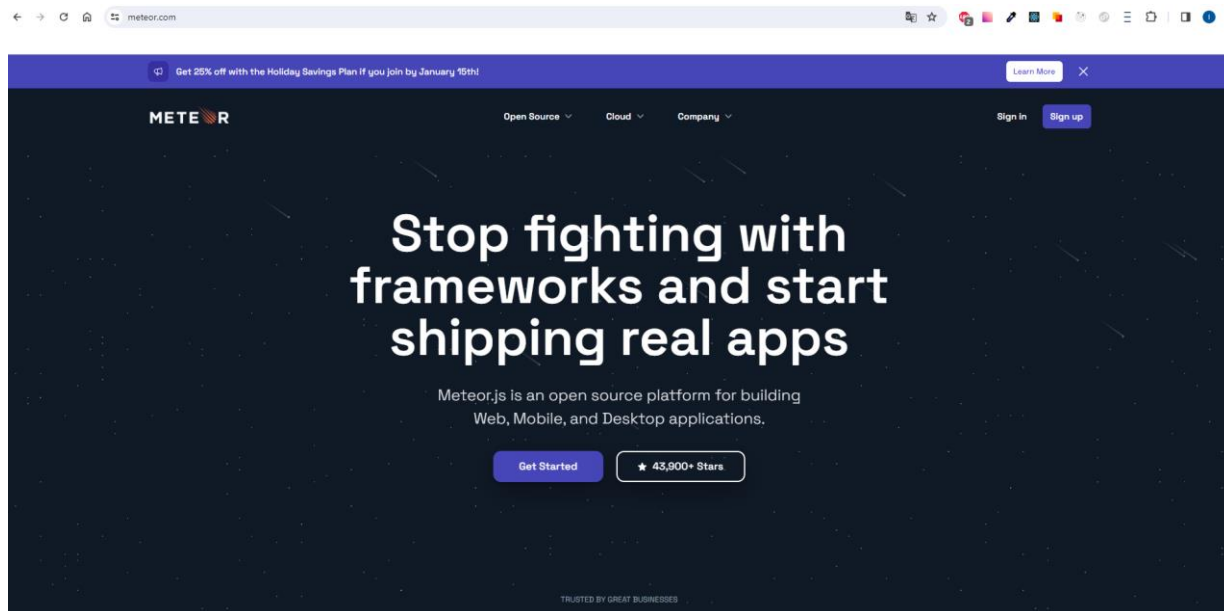


Рисунок 2.7 – Офіційне джерело Meteor.js

Недоліки:

- складність інтерфейсу розробки для новачків;
- обмежена підтримка деяких функцій, включаючи стандартні.

Переваги:

- платформа, що підтримує як фронтенд, так і бекенд;
- можливість розробки та дизайну повноцінних застосунків;
- розробка з елементами реактивності;
- ефективна обробка даних;
- низький поріг входу для початківців у розробці.

Ця технологія не обирається через складність інтерфейсу розробки для новачків та обмежену підтримку деяких стандартних функцій. Незважаючи на переваги, такі як підтримка як фронтенду, так і бекенду, можливість розробки повноцінних застосунків, реактивність, ефективну обробку даних та низький поріг

входу для початківців, ці недоліки роблять її менш вибірковою для певних проєктів.

Фреймворки пропонують готові рішення у вигляді структур та компонентів, що полегшують процес розробки інтерфейсів і сприяють швидшій реалізації функціоналу. Вони також сприяють розробці модульного коду, що спрощує управління та розвиток застосунків. Крім того, фреймворки вдосконалюють рендеринг, управління станами системи та надають інші інструменти для збільшення ефективності застосунків.

2.4 Функціональні компоненти інформаційно-довідкової системи

Всі сайти зазвичай створюють за допомогою поділу функціоналу сайту та його компонентів, тобто створюють багаторівневе відображення.

Розробка інтернет-сайтів являє собою створення компонентів та модулів, які потрібні для того, щоб створити сайт відповідно до бажання замовника чи потреб цільової аудиторії.

Необхідні компоненти – це великий та значимий етап у процесі створення платформи. Перед тим, як вибрати необхідні компоненти потрібно знати які саме компоненти можуть знадобитися користувачу [5]. Це, в першу чергу, сторінки та форми без яких не може обійтись жоден сайт:

- головна сторінка, з описом функцій та послуг сайту;
- сторінка входу та реєстрації користувача;
- сторінка зворотного зв'язку чи контактів з компанією;
- особистий кабінет;
- сторінка «фішка», на якій буде унікальний чи надзвичайно корисний контент.

Реалізація функціоналу створюється за допомогою відповідної програмної системи. Така система має багаторівневе відображення компонентів (рис. 2.8).



Рисунок 2.8 – Програмна система та її компоненти

Компонент подання існує для взаємодії користувачів та програми, що була створена, тобто обробляє події, роботу кнопок, рух різного типу контролерів, тобто відповідає за інтерфейс користувача.

Прикладний компонент існує для того, щоб описати набір правил та алгоритмів, які використовуються для реалізації системи. Описує відгук системи на дії користувачів або інші внутрішні події. Також працює для обробки даних.

За зберігання, вдосконалення, вибір та видалення різного типу даних відповідає компонент керування ресурсами. Також він пов'язаний з розв'язуванням прикладних завдань.

Не останнім за важливістю етапом проектування є розподіл функціональних компонентів у відповідності до обраної архітектури.

Висновки до розділу 2

У цьому розділі було детально продемонстровано послідовність проектування інформаційної системи. Описані технології та підходи до проектування і створення шаблону інформаційної системи, поняття інформаційної системи та їхня архітектура та структура. Детально описано роботу з БД Firebase у даній системі. Показаний зміст деяких таблиць БД. Було розглянуто функціональні компоненти інформаційно-довідкової систем.

Обґрунтування вибору конкретних підходів і технологій включає вибір алгоритмів аналізу даних, розробку або адаптацію торгових стратегій, а також використання специфічних інструментів і платформ. Цей вибір базується на попередньому аналізі та порівнянні різних методів, з урахуванням їх ефективності, масштабованості та безпеки в контексті конкретних цілей та вимог кваліфікаційної роботи магістра.

Розробка інформаційно-довідкових систем здійснюється шляхом створення і інтеграції різноманітних компонентів та модулів, що відповідають потребам замовника та цільової аудиторії. Основні компоненти, такі як головна сторінка, сторінка входу, сторінка контактів, особистий кабінет та унікальні контентні сторінки, є критично важливими для ефективної взаємодії з користувачами. Використання багаторівневого відображення компонентів у програмній системі дозволяє досягти гнучкості та ефективності в процесі розробки та експлуатації інформаційно-довідкових систем.

3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

У цьому розділі представлено моделювання та програмну реалізацію системи для аналізу криптовалютних бірж. Метою системи є забезпечення користувачів інструментами для ефективного аналізу ринкових даних з різних криптовалютних бірж. Система виконує збір, обробку та візуалізацію даних, дозволяючи користувачам отримувати актуальну інформацію про ціни, торгові об'єми, та інші ключові показники ринку.

Особлива увага приділяється використанню React для створення компонентного інтерфейсу та Redux для управління станом застосунку що є критично важливим для забезпечення високої продуктивності та зручності використання системи. Крім того, розглядаємо питання безпеки та приватності, які є особливо актуальними в контексті обробки фінансових даних. В цьому розділі забезпечується глибоке розуміння того, як архітектурні рішення та дизайн інтерфейсу впливають на загальну ефективність та зручність використання системи.

Розробка програмного забезпечення для аналізу криптовалютних бірж та їх систем торгівлі є надзвичайно важливою та актуальною задачею в сучасному світі цифрових фінансів. Враховуючи динамічну природу криптовалютного ринку, ефективність та надійність програмного забезпечення грають ключову роль. Цей розділ детально описує основи архітектури та проєктування такого типу програмного забезпечення, з особливим фокусом на використанні сучасних технологій, таких як React і Redux, для створення гнучкого, масштабованого та інтуїтивно зрозумілого застосунку.

У цілому, ця система є комплексним інструментом, який забезпечує широкі можливості для аналізу криптовалютного ринку, відзначаючись високою продуктивністю, надійністю та користувацькою зручністю.

3.1 Моделювання моделі об'єкту та предмету роботи

Для моделювання об'єкту та предмету роботи є багато шляхів для вирішення поставленої задачі, а саме Unified Modeling Language (UML) діаграми, які можна розділити на окремі.

Діаграма прецедентів, або діаграма варіантів використання, (**Use Case Diagram**) в UML є ключовим інструментом для візуалізації функціональних вимог системи та її взаємодій з зовнішніми елементами або користувачами. Ця діаграма демонструє, як різні користувачі (відомі як актори) взаємодіють з системою для досягнення конкретних цілей (рис. 3.1). До основних аспектів діаграми варіантів використання належать:

- актори (Actors): актори представляють ролі людей, інших систем або зовнішніх процесів, які взаємодіють з системою. Наприклад, в системі криптовалютних бірж акторами можуть бути трейдери, аналітики ринку або платіжні системи;
- варіанти використання (Use Cases): кожен варіант використання описує послідовність дій, яку система виконує, реагуючи на певний запит актора. Варіанти використання відображають функціональність системи, не вдаючись у деталі реалізації. Наприклад, для криптовалютної біржі типовими варіантами використання можуть бути «Здійснення торгових операцій», «Аналіз ринкових тенденцій» чи «Поповнення рахунку»;
- взаємодії між елементами: діаграма показує, як актори взаємодіють з варіантами використання. Це допомагає зрозуміти, які функціональні можливості системи важливі для різних користувачів;
- контекст системи: діаграма варіантів використання також демонструє межі системи, виділяючи, що входить до системи, а що знаходиться поза її межами.

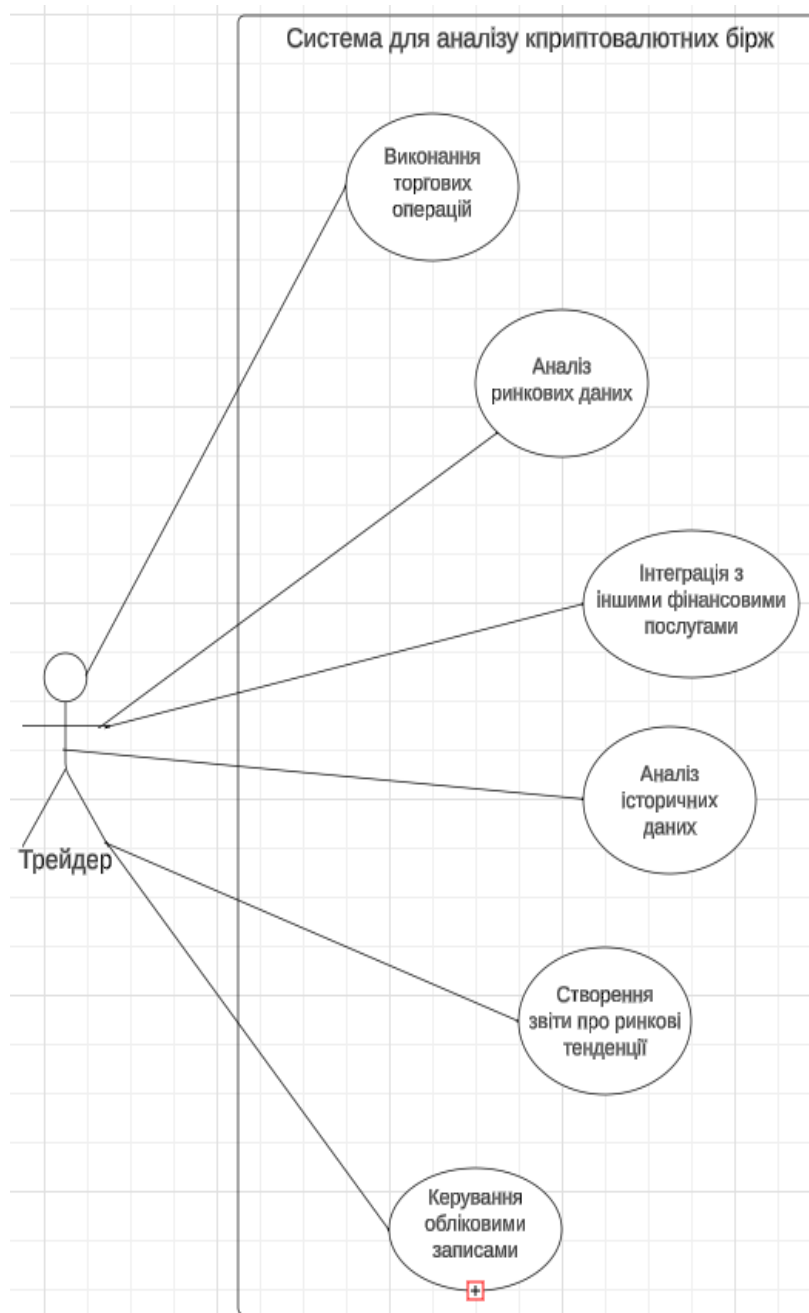


Рисунок 3.1 – UML-діаграма варіантів використання моделі об'єкту та предмету роботи

Діаграма варіантів використання є важливою частиною процесу аналізу та проектування, оскільки вона допомагає розробникам та зацікавленим сторонам розуміти функціональні вимоги та взаємодії в системі. Це сприяє кращому плануванню проекту та зменшує ризики непорозумінь або неврахування важливих аспектів системи на ранніх етапах розробки.

Діаграма послідовності (Sequence Diagram) в UML – це динамічна модель, яка ілюструє, як об'єкти взаємодіють у певних сценаріях. Вона зосереджена на представленні порядку та часу взаємодій між об'єктами або компонентами системи. Основними елементами діаграми послідовності є об'єкти, повідомлення між ними та часові рамки цих взаємодій.

На діаграмі послідовності об'єкти представлені вертикальними лініями (так званими «лініями життя»), а повідомлення між об'єктами – горизонтальними стрілками. Ці стрілки показують, як повідомлення або виклики функцій передаються між об'єктами в конкретному порядку. Вертикальна розташування стрілок відображає часову послідовність подій, зверху вниз на діаграмі.

Діаграма послідовності часто використовується для моделювання сценаріїв використання або конкретних процесів у системі. Вона допомагає розробникам розуміти, як компоненти системи взаємодіють для виконання певної задачі або досягнення певного результату. Це особливо корисно для виявлення та аналізу складних взаємодій, визначення необхідних інтерфейсів та виявлення потенційних проблем у взаємодії між компонентами (рис. 3.2).

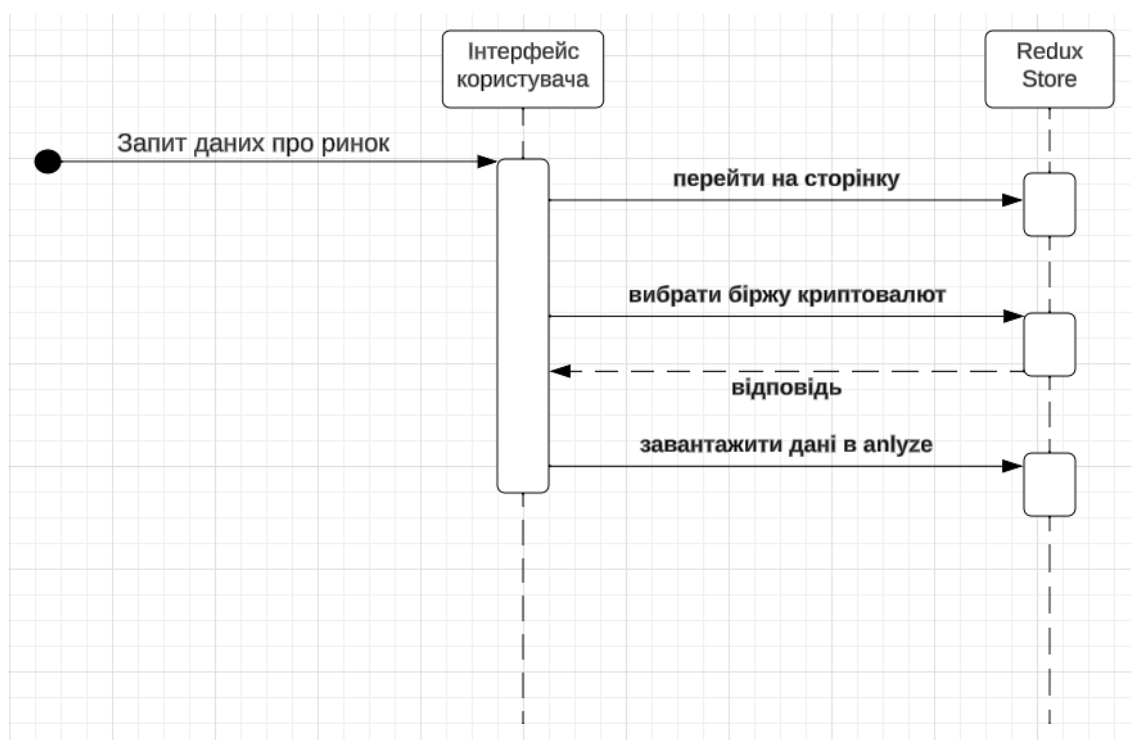


Рисунок 3.2 – UML-діаграма послідовності

На діаграмі послідовності можна детально відобразити, як ці компоненти взаємодіють під час конкретних сценаріїв, наприклад, при аналізі ринкових даних або під час виконання торгових операцій. Це допомагає розробникам зрозуміти динаміку системи та ідентифікувати ключові моменти, де можуть виникати затримки або помилки, а також допомагає у визначенні оптимальних місць для оптимізації продуктивності чи вдосконалення логіки системи.

Загалом, діаграма послідовності є важливим інструментом у процесі проектування та аналізу програмного забезпечення, дозволяючи розробникам глибше зрозуміти динаміку та взаємодії всередині системи. Це сприяє створенню більш ефективних, надійних та зрозумілих архітектур програмних рішень.

Діаграма розгортання (Deployment Diagram) в UML – це вид структурної діаграми, яка використовується для візуалізації фізичного розміщення компонентів системи на апаратному обладнанні. Ця діаграма показує, як програмне забезпечення розподілене по різних вузлах системи, і є важливою для розуміння того, як різні частини системи взаємодіють на фізичному рівні (рис. 3.3).

Основні елементи діаграми розгортання:

- вузли (nodes): вузли представляють фізичні ресурси в системі, такі як комп'ютери, сервери або пристрої. Вони можуть бути або апаратними (як сервери або мережеве обладнання), або віртуальними (як віртуальні машини).
- артефакти (artifacts): артефакти представляють конкретні фізичні файли програмного забезпечення, які розгорнуті на вузлах, наприклад, виконувані файли, бібліотеки, конфігураційні файли тощо.
- зв'язки (associations): зв'язки між вузлами і артефактами вказують на те, де і як артефакти розміщені на фізичних ресурсах.
- комунікаційні шляхи (communication paths): ці елементи вказують на можливі шляхи передачі даних між вузлами, наприклад, мережеві з'єднання.

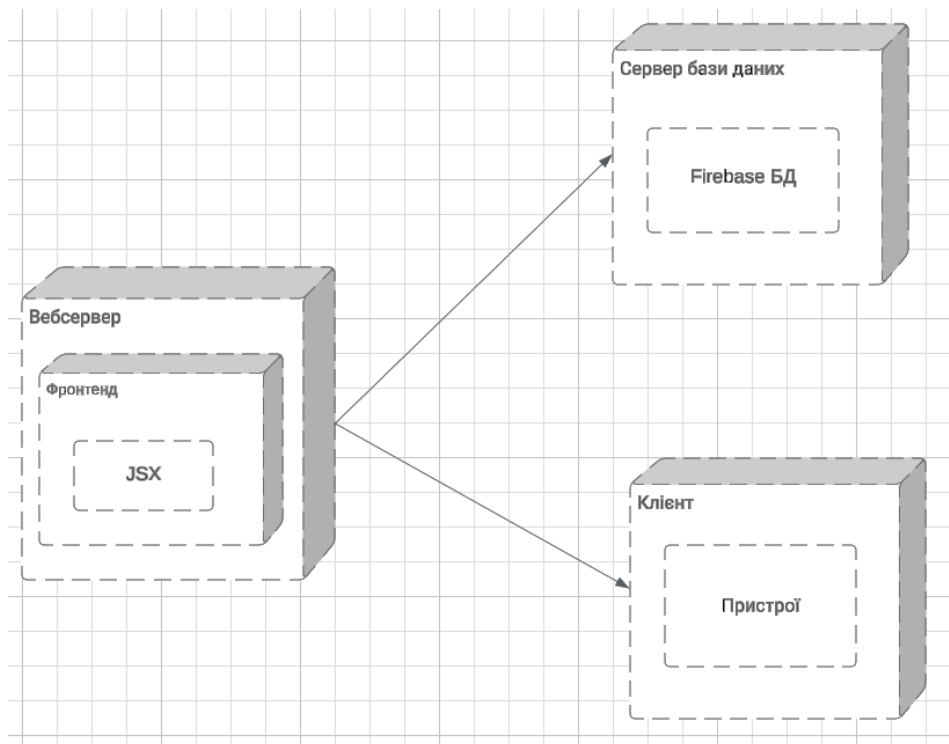


Рисунок 3.3 – UML-діаграма розгортання

На діаграмі представлено розподіл компонентів, таких як клієнтські пристрої, вебсервери, аплікаційні сервери, бази даних та зовнішні API, а також показано мережеві з'єднання між ними. Це дає змогу ідентифікувати, як ресурси системи взаємодіють та залежать один від одного, а також виявити потенційні «вузькі місця» та вимоги до продуктивності. Діаграма також важлива для розуміння розподілу навантаження, масштабованості, безпеки системи. Вона допомагає розробникам та архітекторам чітко зрозуміти, як програмне забезпечення розподіляється на апаратному обладнанні, та визначити оптимальні стратегії розгортання для забезпечення ефективності та надійності системи.

У підсумку, діаграма розгортання відіграє ключову роль у процесі проектування та розробки системи, дозволяючи команді заздалегідь планувати інфраструктурні вимоги та рішення, що сприяє створенню ефективного та надійного програмного продукту.

Діаграма діяльності (Activity Diagram) є важливим інструментом для візуалізації поведінки процесів та робочих потоків у системах. Вона допомагає

моделювати бізнес-процеси, робочі процедури, а також детальні послідовності операцій у програмних системах.

Діаграма діяльності використовується для моделювання таких процесів, як аналіз ринкових даних, процес торгівлі, обробка замовлень, управління користувачькими запитами та інших ключових процедур. Це дозволяє розробникам та аналітикам краще зрозуміти та оптимізувати робочі потоки, а також виявити потенційні проблеми або вдосконалити рішення (рис. 3.4).

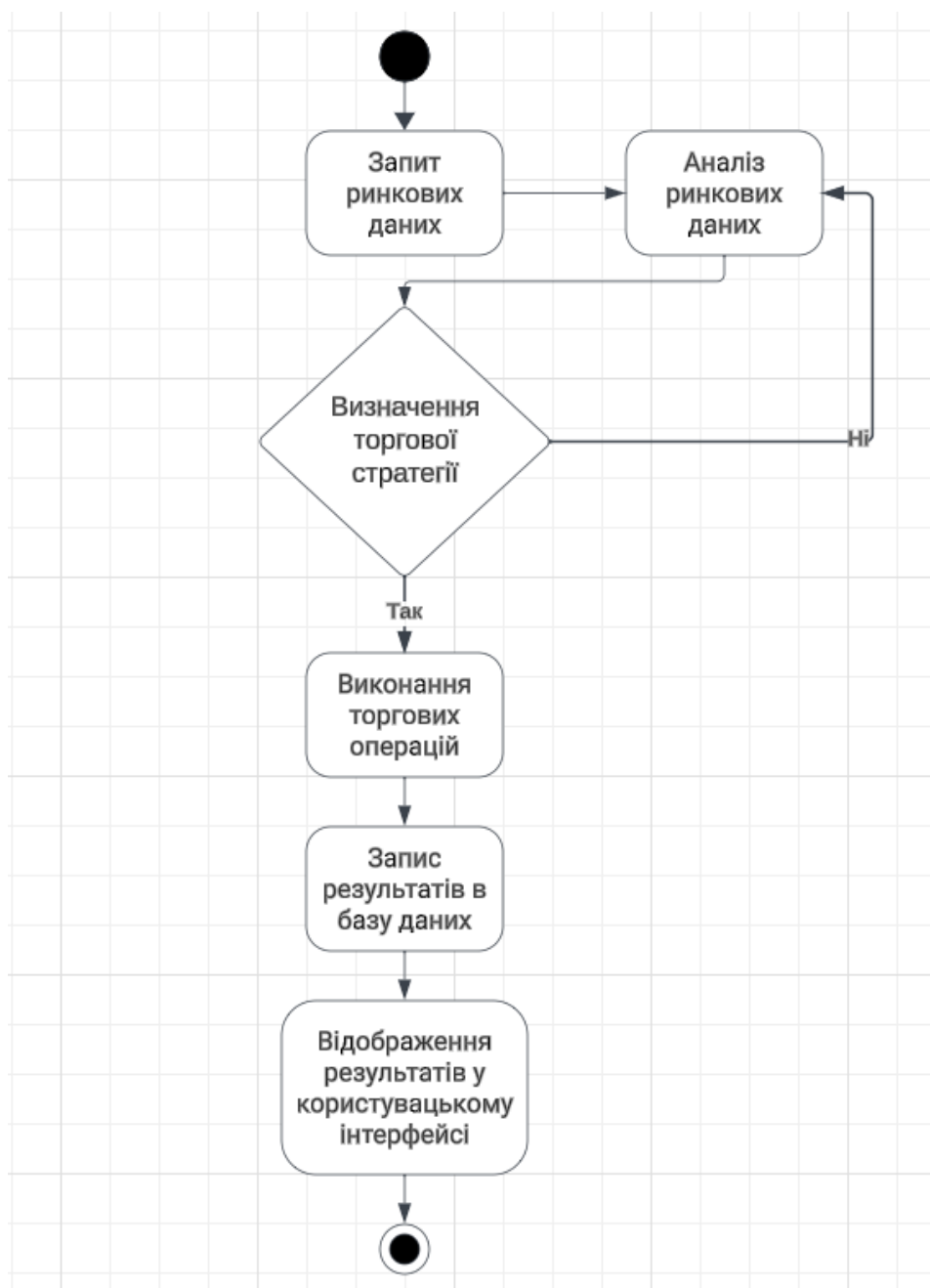


Рисунок 3.4 – Діаграма діяльності розроблюваною системи

Наведена діаграма діяльності відображає логічний потік процесу аналізу та торгівлі на криптовалютних біржах, починаючи зі збору даних і закінчуючи відображенням результатів торгівлі у користувацькому інтерфейсі.

Основні характеристики діаграми діяльності:

- вузли дій: це фундаментальні елементи діаграми, які представляють виконання операцій або дій;
- потоки контролю: показують напрямок виконання від одного вузла дії до іншого. Це допомагає відстежити, як управління переміщається в процесі;
- вузли рішення та злиття: використовуються для моделювання рішень, де потік контролю розділяється на кілька шляхів, або злиття декількох шляхів у один;
- початкові та кінцеві точки: визначають початок та завершення робочого потоку або процесу;
- синхронізація та паралелізм: показують, де процеси або дії виконуються одночасно або потребують синхронізації;
- плаваючі вузли: використовуються для організації дій по ролях або відділах, показуючи, хто або що відповідальне за певні дії в процесі.

Використання цих елементів у діаграмі діяльності дозволяє візуально представити потік робочого процесу в системі, включаючи ключові дії, рішення, що приймаються на основі даних, та послідовність кроків, що ведуть до кінцевих результатів.

Отже, створення різних типів UML діаграм для моделювання системи для аналізу криптовалютних бірж з використанням React та Redux надає цінний інструментарій для візуалізації, аналізу та планування системи. Кожна діаграма вносить вклад у загальне розуміння та розробку проєкту.

Діаграма розгортання відіграє важливу роль у визначенні фізичної структури системи, показуючи розподіл компонентів програмного забезпечення на апаратному обладнанні та їх взаємозв'язки. Це критично важливо для планування інфраструктури та забезпечення надійності та ефективності системи. Нарешті,

діаграма активностей візуалізує робочі процеси та бізнес-процедури, виявляючи послідовності дій і рішень, які формують потоки роботи в системі.

Кожна з цих діаграм сприяє глибокому розумінню різних аспектів системи, її структури, поведінки, фізичного розміщення та робочих процесів. Разом вони формують комплексне бачення проєкту, що є вирішальним для ефективного проєктування, реалізації та управління системою аналізу криптовалютних бірж.

3.2 Класи та методи для вирішення задачі

Діаграма класів (Class Diagram) є однією з найважливіших діаграм, яка використовується для моделювання структури системи на рівні класів, їх атрибутів, методів та відносин між ними. Вона є ключовим інструментом у об'єктно-орієнтованому проєктуванні, допомагаючи розробникам візуалізувати архітектуру програмного забезпечення та сприяти кращому розумінню та організації коду. Клас на такій діаграмі зазвичай містить ім'я класу, атрибути (або поля) та методи (або функції). Атрибути представляють інформацію, яку об'єкт буде зберігати, тоді як методи визначають поведінку або дії, які клас може виконувати.

Використання діаграми класів дозволяє аналізувати і виявляти потенційні проблеми у структурі програмного забезпечення на ранніх етапах розробки, забезпечуючи більш ефективний і організований підхід до проєктування. Вона також сприяє кращій комунікації та розумінню між членами команди, надаючи чітке уявлення про внутрішню структуру системи і способи її реалізації (рис. 3.5).

Елементи, які можна включити в діаграму класів для вашої теми:

- класи для представлення даних;
- класи для користувачів та аутентифікації;
- класи для торгівлі;
- класи для аналізу даних;
- класи для інтеграції з API бірж.

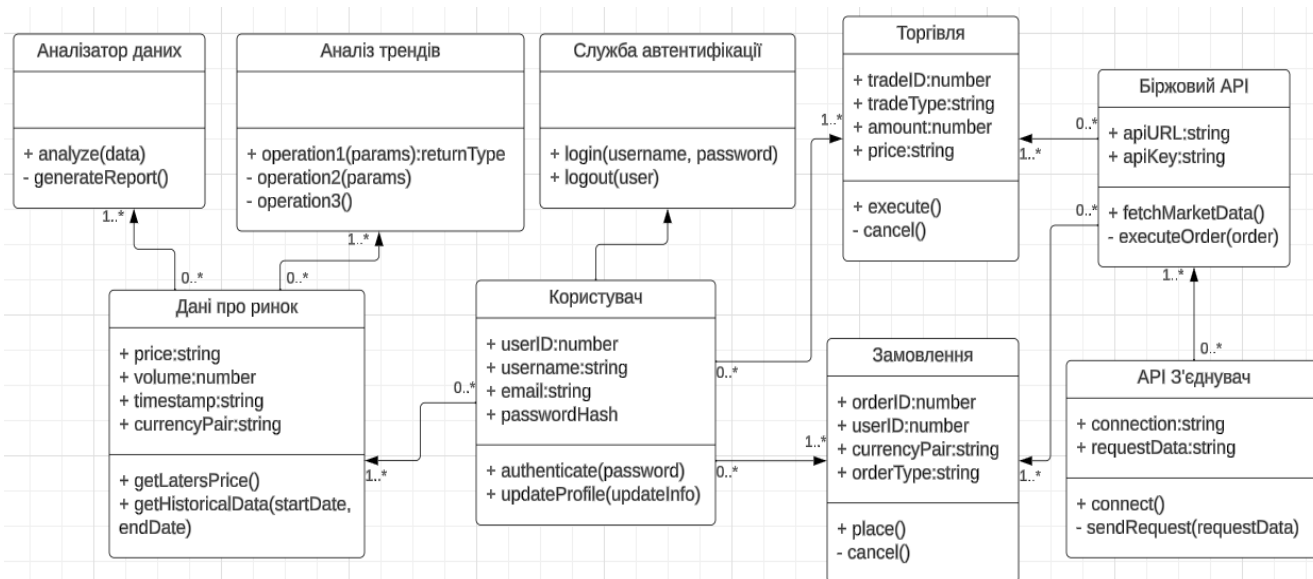


Рисунок 3.5 – UML-діаграма класів моделі об'єкту та предмету роботи

Ця діаграма відображає не тільки структуру та взаємозв'язки між класами, але й забезпечує основу для реалізації чіткого коду з добре визначеними відповідальностями для кожного компоненту. Вона сприяє кращому розумінню загальної архітектури системи та є ключовим інструментом для ефективного планування розробки.

Центральну роль у цій структурі відіграють React та Redux, які забезпечують ефективне управління інтерфейсом користувача та станом застосунку відповідно.

React використовується для створення інтуїтивно зрозумілого та відгукового інтерфейсу. Це дозволяє легко інтегрувати різноманітні UI компоненти, такі як графіки, таблиці та інші візуальні елементи, які є ключовими для аналізу даних криптовалютних бірж. Компонентний підхід React також сприяє більш чистому та організованому коду, що спрощує розширення та підтримку застосунку.

Redux, з іншого боку, використовується для управління станом застосунку. Він забезпечує централізоване сховище, де зберігаються всі дані стану, що полегшує управління інформацією та спільний доступ до неї між різними компонентами. Такий підхід допомагає в реалізації складних функцій, таких як

асинхронні запити до API криптовалютних бірж та ефективно оновлення інтерфейсу користувача відповідно до змін у даних.

Опис архітектури застосунку, в якому використовуються React та Redux, включаючи загальну структуру та компоненти. Це може включати:

- React Components: опис компонентів інтерфейсу, таких як графіки криптовалют, таблиці ринкових даних тощо;
- Redux Store: структура Redux store для управління станом застосунку;
- Middleware: використовуються, опис ролі middleware, таких як Redux Thunk чи Saga, для асинхронних операцій;
- Data Parsing and Normalization: методи, що використовуються для обробки та нормалізації отриманих даних;
- Error Handling: механізми обробки помилок при взаємодії з зовнішніми API;
- Caching and Optimization: підходи до кешування та оптимізації запитів для підвищення продуктивності.

Значну увагу в розділі приділено інтеграції з API криптовалютних бірж. Це включає методи аутентифікації, виконання запитів та обробки відповідей. Особлива увага приділяється ефективному парсингу та нормалізації отриманих даних, що є важливим для забезпечення точності та надійності аналізу.

Опис інтеграції з API різних криптовалютних бірж включає:

- API Integration: методи інтеграції з API, включаючи аутентифікацію, запити та обробку відповідей;
- Data Representation: застосунок представляє та відображає ринкові дані від різних бірж.

Також важливою є обробка помилок та забезпечення безпеки при взаємодії з зовнішніми API. Механізми кешування та оптимізації запитів відіграють ключову роль у підвищенні продуктивності та швидкості роботи застосунку.

Модульне тестування також є важливою частиною процесу розробки. Воно допомагає забезпечити надійність та стабільність різних компонентів застосунку, виявляючи та виправляючи помилки на ранніх стадіях розробки.

Діаграма класів глибоко занурює нас у структуру системи, представляючи класи, їх властивості та методи, а також взаємозв'язки між ними. Це ключово для визначення архітектури програмного забезпечення та його компонентів. Діаграма послідовностей розкриває взаємодію між об'єктами у різних сценаріях використання, допомагаючи зрозуміти, яким чином відбувається обмін повідомленнями та даними у системі. Вона є незамінною для аналізу динамічної поведінки компонентів.

3.3 Архітектура системи

Архітектура системи для аналізу криптовалютних бірж розроблена з метою забезпечити ефективність, масштабованість та високу доступність. Вона складається з декількох ключових компонентів, кожен з яких відіграє свою роль у загальній структурі системи:

- клієнтська частина (Front-End) реалізована з використанням React, клієнтська частина відповідає за представлення даних користувачам у зручному та інтуїтивно зрозумілому вигляді. Використання React дозволяє створювати динамічні та відгукові інтерфейси, які забезпечують відмінне користувацьке досвід. Адаптивний дизайн забезпечує оптимальне відображення на різних пристроях;
- серверна частина (Back-End) розроблена на Node.js із використанням Express як основного фреймворку, що забезпечує гнучке та ефективне управління запитами. Відповідає за збір, обробку та надання даних клієнтській частині. Цей процес включає збір даних з зовнішніх API криптовалютних бірж, їх обробку та агрегацію;

- Firebase використовується як основна система управління базами даних. Вона забезпечує високу продуктивність при роботі з великими обсягами даних та їхньому швидкому доступі;
- система інтегрована з різними зовнішніми API криптовалютних бірж для збору реальних ринкових даних. Це дозволяє системі надавати користувачам актуальну інформацію в режимі реального часу;
- застосовані різноманітні заходи безпеки, включаючи шифрування даних, захист від SQL-ін'єкцій, XSS атак та інших загроз. Аутентифікація та авторизація користувачів для забезпечення безпечного доступу до системи;
- для реалізації функціональності реального часу, такої як оновлення цін та об'ємів торгів, використовуються WebSockets. Це дозволяє системі миттєво передавати оновлення без необхідності перезавантаження сторінки.
- Масштабованість та висока доступність: архітектура системи розроблена з урахуванням потреб масштабування та забезпечення високої доступності. Це означає, що систему можна легко розширювати відповідно до зростання навантаження та користувацької бази.

Загалом, архітектура системи для аналізу криптовалютних бірж забезпечує високий рівень продуктивності, надійності та зручності для кінцевих користувачів, враховуючи сучасні тенденції та вимоги в області розробки програмного забезпечення.

Описана архітектура системи для аналізу криптовалютних бірж є прикладом мікросервісної архітектури. Мікросервісна архітектура – це підхід до розробки програмного забезпечення, який передбачає розбиття застосунку на набір менших, незалежних сервісів, кожен з яких виконує певну функцію і спілкується з іншими сервісами через легкі протоколи, часто використовуючи API (рис. 3.6).

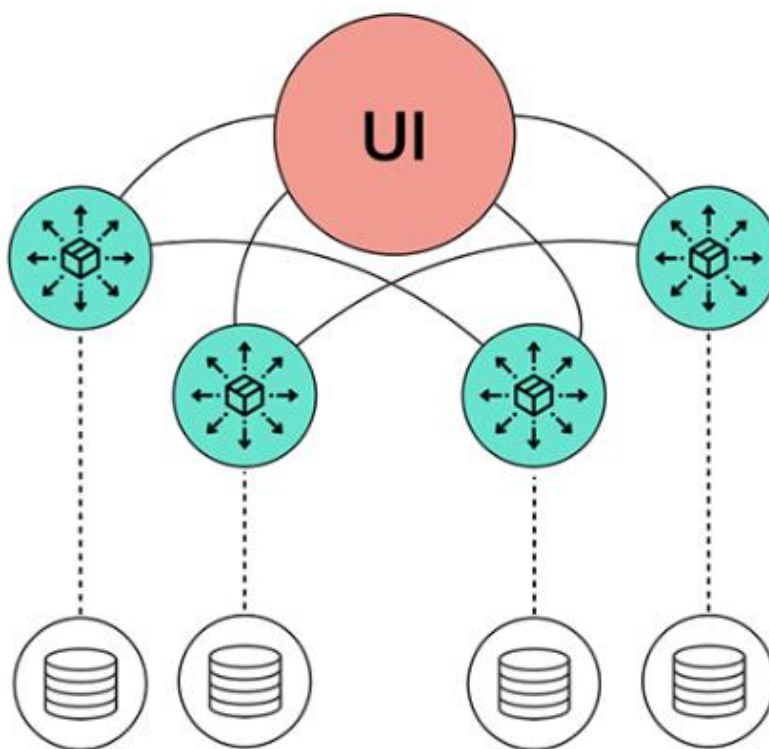


Рисунок 3.6 – Концептуальна модель системи

Основні характеристики мікросервісної архітектури, які відповідають опису вашої системи:

- система розділена на незалежні модулі (сервіси), кожен з яких відповідає за окрему функціональність;
- кожен сервіс можна масштабувати незалежно, що забезпечує ефективне розподілення ресурсів;
- різні сервіси можуть бути розроблені, тестовані та розгорнуті незалежно один від одного.
- Використання API для комунікації: сервіси спілкуються між собою за допомогою легких API, що дозволяє встановлювати гнучкі зв'язки між різними частинами системи.

Цей підхід дозволяє створювати гнучкі та легко масштабовані системи, які можуть ефективно адаптуватися до змінних вимог та обсягів роботи, що є ідеальним для системи аналізу криптовалютних бірж.

Висновки до розділу 3

Побудова UML діаграми класів для системи аналізу криптовалютних бірж із використанням React та Redux є важливим кроком у структуруванні та плануванні кваліфікаційної роботи магістра. Діаграма демонструє складну архітектуру системи, інтегруючи різноманітні компоненти та їх взаємодії. Вона включає класи, що представляють користувачів, ринкові дані, торгові операції та аналітичні інструменти, а також інтерфейси для зовнішніх криптовалютних бірж. Важливою особливістю є глибока інтеграція між бізнес-логікою і користувацьким інтерфейсом, де React компоненти спілкуються з бекендом, використовуючи Redux для управління станом.

Використання UML діаграми як візуального інструменту допомагає уникнути помилок у проєктуванні, спрощує комунікацію між членами команди та забезпечує єдине розуміння структури системи. В кінцевому підсумку, ця діаграма служить фундаментом для розробки надійного та ефективного програмного забезпечення, здатного впоратися зі складними вимогами аналізу криптовалютних бірж.

4 РОЗРОБКА ТА ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

У цьому розділі представлено детальний опис процесу розробки, тестування та аналізу системи, призначеної для аналізу криптовалютних бірж та їхніх торгових систем, реалізованої з використанням технологій React та Redux. Мета даного розділу полягає у демонстрації практичного застосування теоретичних знань, отриманих в ході підготовки кваліфікаційної роботи магістра, та навичок програмування для створення функціональної та ефективної системи

Розробка системи включає в себе декілька ключових етапів. Спочатку спроектуємо та створим сервіс для системи, на основі якого сформульовано основні задачі та цілі проєкту. Далі буде наведено детальний опис інтерфейсу користувача та логіки взаємодії з системою.

Тестування системи є не менш важливим етапом, що дозволяє забезпечити її надійність та ефективність. У розділі представлено методіку тестування, що була використана, включаючи як автоматизоване, так і ручне тестування. Окрім цього, розглянуто підходи до виявлення та усунення помилок, а також до оптимізації продуктивності системи.

На завершення розділу проведено аналіз отриманих результатів. Демонструється, як система відповідає поставленим вимогам, а також оцінюється її ефективність та потенціал для подальшого розвитку та удосконалення. Також обговорюються можливі напрямки для майбутнього розширення функціональності системи, зокрема, інтеграція з додатковими криптовалютними біржами та вдосконалення алгоритмів аналізу даних.

4.1 Проєктування та створення сервісів для системи

Процес проєктування та створення сервісів для системи базувався на використанні сучасних інструментів та технологій для забезпечення ефективної роботи та взаємодії з API криптовалют та новинних ресурсів.

Інтеграція з Coinranking через RapidAPI дозволяє нашій системі отримувати різноманітну інформацію про криптовалютні ринки. За допомогою сервісу `cryptoApi.js`, можна отримувати дані про криптовалютні ринки, такі як ціни, обсяги торгів, ринкову капіталізацію та інші ключові показники, за допомогою відповідних запитів API. Для цього створено зручний інтерфейс, який дозволяє виконувати запити та обробляти отримані дані відповідно до потреб нашої системи.

Сервіс `cryptoApi.js` був розроблений з використанням `@reduxjs/toolkit/query/react`, що надає можливість легкої інтеграції з Redux для управління станом застосунку. Цей сервіс дозволяє отримувати дані про криптовалютні ринки, деталі конкретних криптовалют, історію торгів та список доступних обмінних бірж. Він використовує базовий запит `fetchBaseQuery` для виконання HTTP-запитів, а також створює заголовки з використанням ключів, які надаються через змінні середовища (рис. 4.1).

```
import { createApi, fetchBaseQuery } from '@reduxjs/toolkit/query/react';

const cryptoApiHeaders = {
  'X-RapidAPI-Host': process.env.REACT_APP_CRYPTO_RAPIDAPI_HOST,
  'X-RapidAPI-Key': process.env.REACT_APP_RAPIDAPI_KEY,
};
const createRequest = (url) => ({ url, headers: cryptoApiHeaders });

export const cryptoApi = createApi({
  reducerPath: 'cryptoApi',
  baseQuery: fetchBaseQuery({ baseUrl: process.env.REACT_APP_CRYPTO_API_URL }),
  endpoints: (builder) => ({
    getCryptos: builder.query({
      query: (count) => createRequest(`/coins?limit=${count}`),
    }),
    getCryptoDetails: builder.query({
      query: (coinId) => createRequest(`/coin/${coinId}`),
    }),
    getCryptoHistory: builder.query({
      query: ({ coinId, timeperiod }) => createRequest(`/coin/${coinId}/history?timeperiod=${timeperiod}`),
    }),
    getExchanges: builder.query({
      query: () => createRequest(`/exchanges`),
    }),
  }),
});
```

a)

```

export const {
  useGetCryptosQuery,
  useGetCryptoDetailsQuery,
  useGetExchangesQuery,
  useGetCryptoHistoryQuery,
} = cryptoApi;

```

б)

Рисунок 4.1 – Фрагменти коду для сервісу *cryptoApi.js* (а, б)

Наприклад, за допомогою методу *getCryptos*, можна отримати список криптовалют з обмеженням кількості (рис. 4.1). Метод *getCryptoDetails* дозволяє отримати детальну інформацію про певну криптовалюту за її ідентифікатором. Також є методи для отримання історії торгів та списку доступних обмінних бірж. Отримані дані можуть бути оброблені та використані у системі для подальшого аналізу, візуалізації або інших цілей відповідно до потреб користувачів.

Аналогічно, сервіс *cryptoNewsApi.js* був розроблений для отримання новин про криптовалюту з використанням відповідних API. Обидва сервіси використовують базовий запит *fetchBaseQuery* для виконання HTTP-запитів та створюють необхідні заголовки для виконання запитів (рис. 4.2).

```

import { createApi, fetchBaseQuery } from '@reduxjs/toolkit/query/react';

const cryptoNewsHeaders = {
  'x-bingapis-sdk': 'true',
  'x-rapidapi-key': process.env.REACT_APP_RAPIDAPI_KEY,
  'x-rapidapi-host': process.env.REACT_APP_NEWS_RAPIDAPI_HOST,
};

const createRequest = (url) => ({ url, headers: cryptoNewsHeaders });

export const cryptoNewsApi = createApi({
  reducerPath: 'cryptoNewsApi',
  baseQuery: fetchBaseQuery({ baseUrl: process.env.REACT_APP_NEWS_API_URL }),
  endpoints: (builder) => ({
    getCryptoNews: builder.query({
      query: ({ newsCategory, count }) => createRequest(`/news/search?q=${newsCategory}&safeSearch=Off&textFormat=Raw&freshness=Day&count=${count}`),
    }),
  }),
});

export const { useGetCryptoNewsQuery } = cryptoNewsApi;

```

Рисунок 4.2 – Фрагмент коду для сервісу *cryptoNewsApi.js*

Ці сервіси надають зручний інтерфейс для взаємодії з зовнішніми ресурсами, а їхні функції можуть бути легко використані в компонентах React для відображення даних та взаємодії з користувачем. Завдяки цим сервісам система отримує доступ до актуальних даних про криптовалютні ринки та новини, що дозволяє користувачам отримувати актуальну інформацію та приймати обґрунтовані торговельні рішення.

4.2 Детальний огляд та взаємодія з інтерфейсом системи

У цьому розділі наведено опис інтерфейсу та функціоналу розробленої системи аналізу криптовалютних бірж. Також було зосереджено увагу на важливості інтуїтивного та функціонального дизайну для ефективного використання користувачами.

Дизайн інтерфейсу користувача розроблено з метою забезпечити зручність та простоту використання, не жертвуючи при цьому функціональністю. Це включає в наступне:

- чітка та проста навігація, яка дозволяє користувачам легко переходити між різними секціями системи;
- ефективне використання кольорів та шрифтів для підвищення читабельності та зручності інтерфейсу;
- графічне представлення даних, включаючи діаграми та графіки, які надають користувачам швидкий огляд ринкових тенденцій.

У сукупності, ці аспекти ілюстрації та взаємодії з інтерфейсом створюють користувацький досвід, який є одночасно приємним, інтуїтивним і безпечним, сприяючи ефективній та продуктивній роботі користувачів з системою.

Розглянемо секцію «Top 10 Cryptos In The World», яка відображає список криптовалют із їх цінами, ринковими капіталізаціями та щоденними змінами вартості. Кожна криптовалюта в списку має номер, назву, логотип, та три характеристики: ціну, ринкову капіталізацію та денну зміну ціни у відсотках (рис. 4.3).

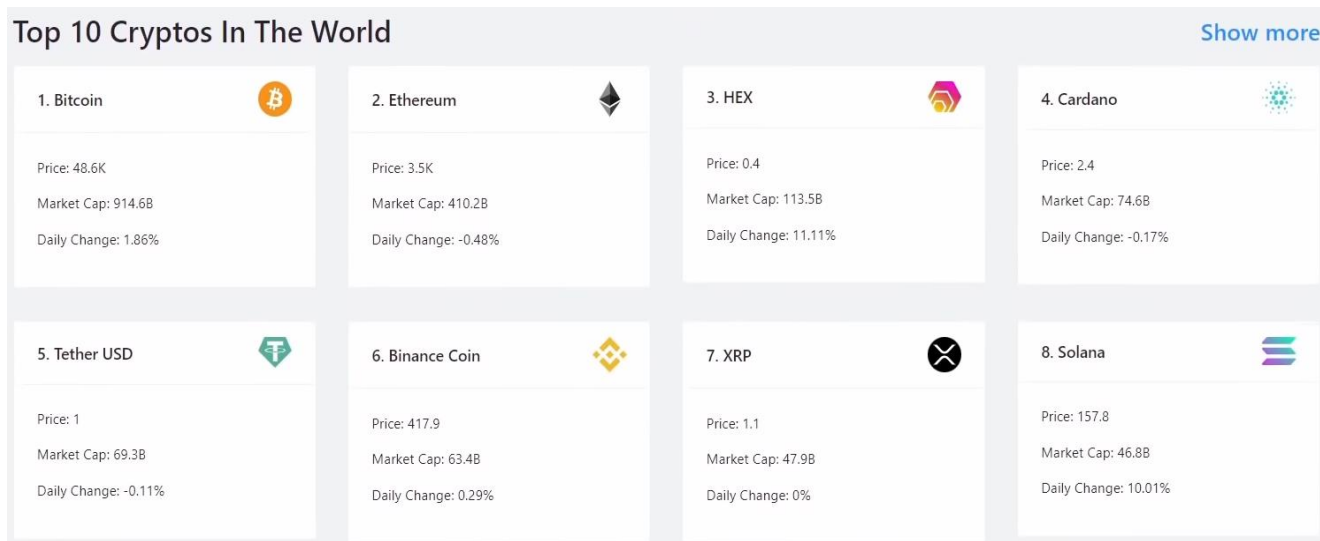


Рисунок 4.3 – Секція з рейтингом криптовалют

Користувач може взаємодіяти з цією секцією наступним чином:

- користувачі можуть переглянути основні дані про кожну криптовалюту, такі як ціну, ринкову капіталізацію та денну зміну вартості;
- натиск на назву криптовалют або логотипи, дозволяє користувачам перейти на іншу сторінку або розділ з більш детальною інформацією про обрану криптовалюту;
- у верхньому правому куті є кнопка «Show more», яка, дозволяє користувачам переглядати додаткові криптовалюти, що входять до рейтингу, але не відображені в цьому візуальному виразі.

Далі, перейдемо до секції «Global Crypto Stats», яка містить статистику світового ринку криптовалют. Вказано загальну кількість криптовалют (12,176), кількість бірж (373), загальну ринкову капіталізацію (\$2.4Т, тобто трильйони доларів), загальний обсяг торгів за 24 години (\$92.3В, тобто мільярди доларів), а також загальну кількість ринків (80К, тобто тисяч). Дизайн секції мінімалістичний, з інформацією, організованою у дві колонки для легкого сприйняття. Ця інформація допомагає користувачам отримати швидке уявлення про загальний стан ринку криптовалют (рис. 4.4).



Рисунок 4.4 – Секція зі статистикою криптовалют

Наступною переглянемо секцію «Global Crypto Stats», яка відображає найновіші новини у світі криптовалют. Вона складається з ряду карток новин, кожна з яких містить заголовок статті, зображення, джерело і час публікації. Вгорі секції знаходиться заголовок «Latest Crypto News», а в правому верхньому куті кнопка «Show more», що дозволяє користувачам переглядати додаткові новинні статті (рис. 4.5).

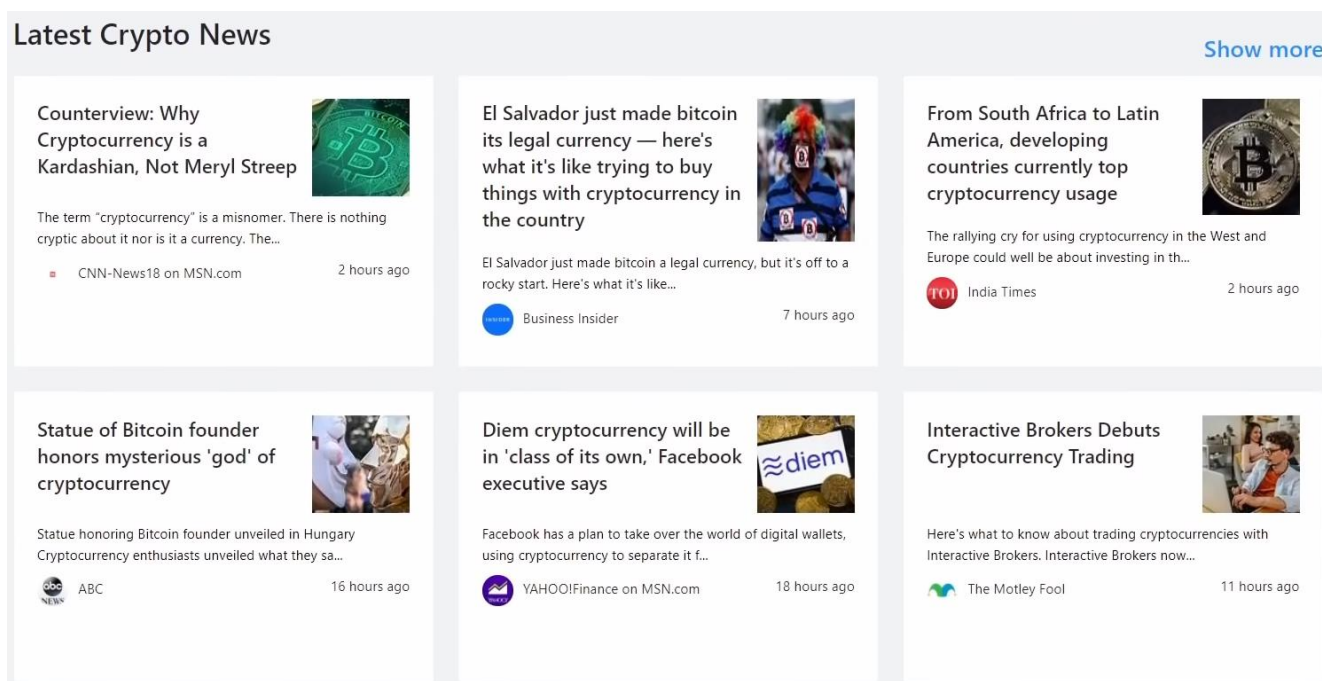


Рисунок 4.5 – Секція із новинами про криптовалюти та біржі

Розглянемо навігаційну панель, яка відображає найновіші має темно-синій фон і білий текст, що створює контраст для легкого читання (рис. 4.6). Навігаційна

панель містить логотип та назву сервісу вгорі, під якими розташовані чотири посилання для навігації: «Home», «Cryptocurrencies», «Exchanges» та «News».

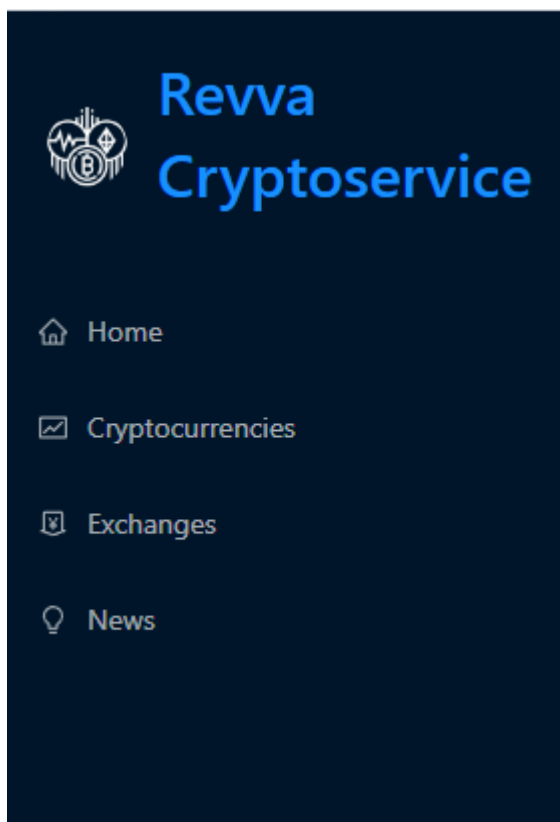


Рисунок 4.6 – Навігаційна панель

Користувач може взаємодіяти з цією панеллю наступним чином:

- натиснувши на будь-яке з посилань, користувач буде направлений на відповідну сторінку системи;
- логотип та назва сервісу також є клікабельними і перенаправляють користувача на головну сторінку;
- кнопка з іконкою меню у верхньому правому куті, за логікою коду, дозволяє користувачам увімкнути або вимкнути відображення навігаційного меню, залежно від розміру екрану або користувацьких переваг.

Код компоненту `Navbar`, написаний на JavaScript з використанням бібліотеки React, вказує, що стан активності меню змінюється при кліку на кнопку меню, а також автоматично адаптується до розміру екрану – якщо ширина екрану менше або дорівнює 800 пікселів, меню стає неактивним, тобто складається (Додаток А).

Перейдемо до нижнього колонтитула системи, який включає авторське право та посилання для навігації. Текст включає повідомлення про авторське право і нижче розташовані три посилання (рис. 4.7).

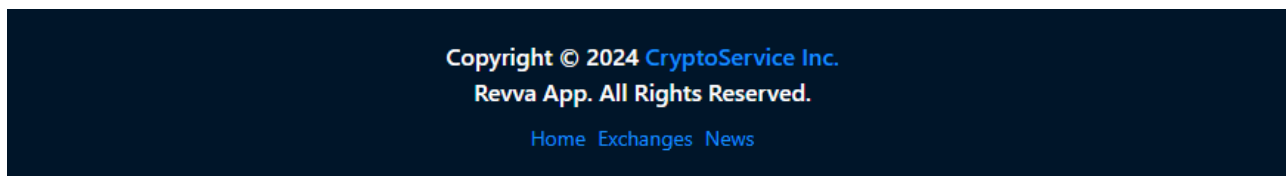


Рисунок 4.7 – Нижній колонтитул

Ця частина вебсайту часто використовується для швидкого доступу до ключових розділів вебсайту та може містити додаткові юридичні посилання, контактну інформацію або посилання на політику конфіденційності, яких не видно на цьому зображенні.

Розглянемо сторінку з криптовалютою, на прикладі Bitcoin (BTC). Компонент *CryptoDetails.tsx* (див. Додаток А) відображає графік цін на BTC. Вгорі сторінки розташований заголовок з текстом нижче, який пояснює, що показана жива ціна BTC у доларах США, а також пропонується ознайомитися зі статистикою вартості, ринковою капіталізацією та пропозицією. У правому верхньому куті є інформація про зміну ціни у відсотках та поточну ціну BTC (рис. 4.8).

Користувач може взаємодіяти з цією сторінкою таким чином:

- натиснувши на вибір часового періоду для перегляду історії цін, користувач може вибрати з розкривного меню часовий період (наприклад, «7d» для семи днів), який змінить дані на графіку і покаже зміни цін за обраний період;
- користувач може подивитися статистику, таку як ринкова капіталізація, обсяг торгів за 24 години, ранг у рейтингу криптовалют, і інші значущі показники (див. рис. 4.8);

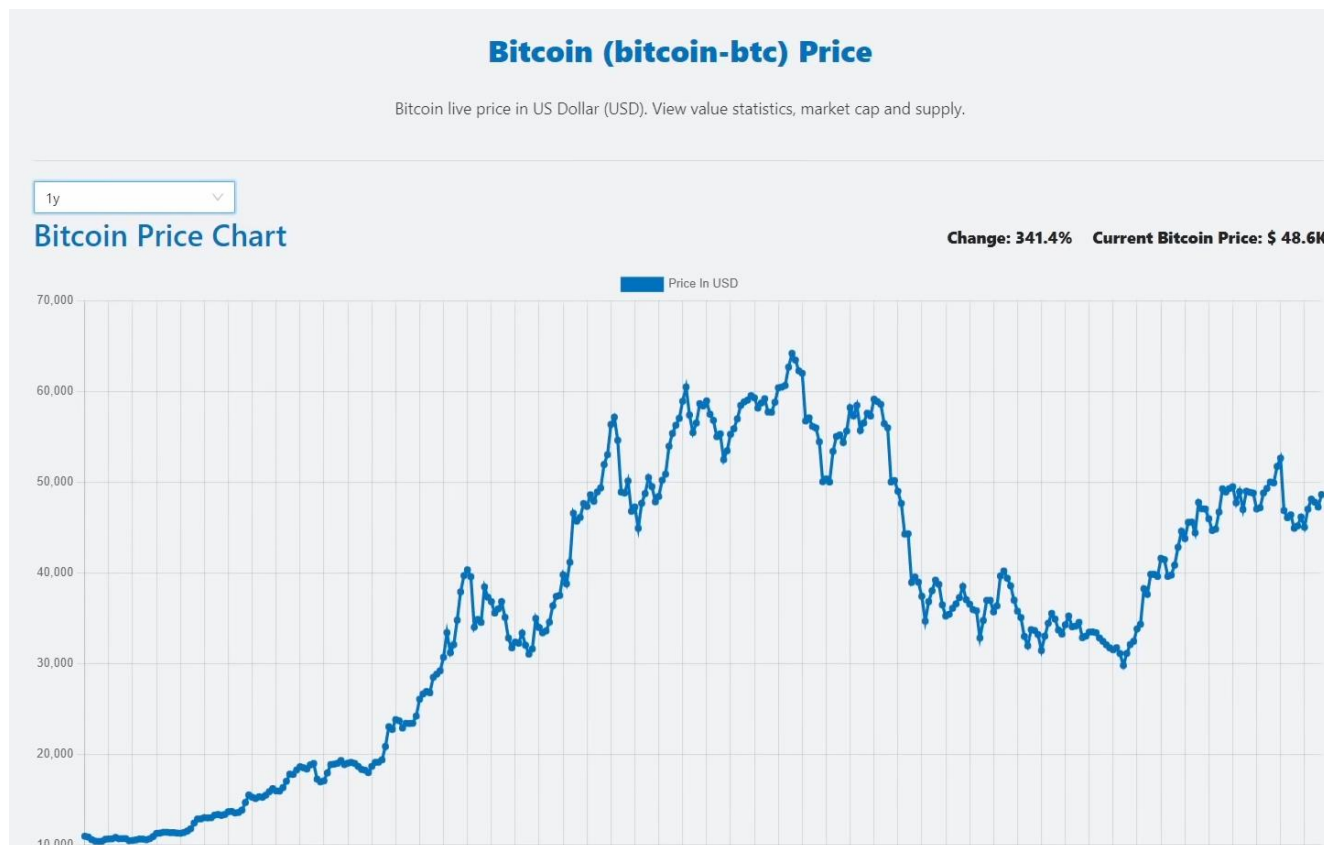


Рисунок 4.8 – Сторінка криптовалюти BTC (інфографіка)

- у розділі «What is Bitcoin?» користувачі можуть прочитати опис та історію криптовалюти;
- у розділі «Bitcoin Links» користувачі можуть знайти та перейти за посиланнями на зовнішні ресурси для отримання більш повної інформації (рис. 4.9).

<p>What is Bitcoin?</p> <p>Bitcoin is the first digital currency that allows users to send and receive money, without the interference of a central bank or government. Instead, a network of thousands of peers is controlling the transactions; a decentralized system.</p> <p>Why does bitcoin have value?</p> <p>Bitcoin's useful qualities (decentralized, borderless, secure) aren't the only reason the coin is worth so much. Due to its scarcity (and it's hard to produce), Bitcoin is often nicknamed 'Digital Gold', in reference to 'classic' physical gold. Like gold, Bitcoin also has a finite supply of coins available; there will only ever be 21 million bitcoin. And now you know why the creation of new bitcoins is also called mining.</p>	<p>Bitcoin Links</p> <table border="0"> <tr> <td>Website</td> <td>bitcoin.org</td> </tr> <tr> <td>Website</td> <td>Bitcoin Whitepaper</td> </tr> <tr> <td>Website</td> <td>bitcoinmagazine.com</td> </tr> <tr> <td>Bitcointalk</td> <td>BitcoinTalk</td> </tr> </table>	Website	bitcoin.org	Website	Bitcoin Whitepaper	Website	bitcoinmagazine.com	Bitcointalk	BitcoinTalk
Website	bitcoin.org								
Website	Bitcoin Whitepaper								
Website	bitcoinmagazine.com								
Bitcointalk	BitcoinTalk								

Рисунок 4.9 – Сторінка криптовалюти (детально інформація)

Код сторінки, написаний на React.js, включає використання стейтів для управління станом вибору часового періоду і викликає API для отримання даних про деталі криптовалюти та її історії цін. Графік цін є компонентом *LineChart.tsx* (див. Додаток А), який відображає історію цін згідно вибраного часового періоду.

Далі перейдемо до сторінки з новинами про криптовалюти. Вона організована у вигляді сітки з картками новин. Кожна картка містить заголовок статті, короткий опис, зображення, назву видання та час, коли новина була опублікована (рис. 4.10).

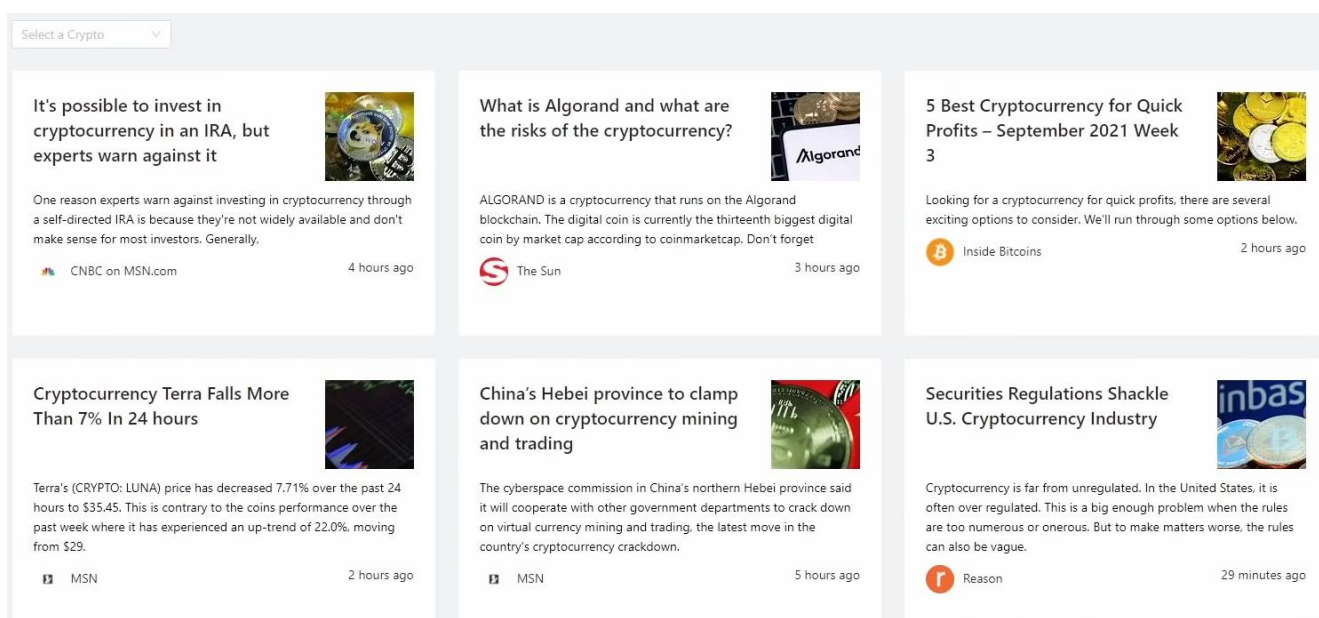


Рисунок 4.10 – Сторінка з новинами про криптовалюти

Користувачі можуть взаємодіяти з цією сторінкою наступним чином:

- клікнувши на заголовок новини або зображення, вони можуть перейти до повного тексту статті для детального читання;
- новини можуть бути сортовані або фільтровані за допомогою інтерактивного елемента в верхньому лівому куті, де написано «Select a Crypto», дозволяючи користувачам вибрати новини, що стосуються конкретної криптовалюти;
- користувачі можуть також подивитися в якому часі новина була опублікована, щоб оцінити актуальність інформації.

Цей інтерфейс дозволяє користувачам швидко сканувати останні події в індустрії криптовалют, вибираючи статті, які їх найбільше цікавлять, та отримуючи доступ до різноманітних джерел інформації.

Останньою оглянемо сторінку з інформацією про криптовалютні біржі та їхні системи торгів (рис. 4.11). Вибрана біржа – Binance, яка займає перше місце у списку, і до неї наведено детальний опис. Нижче Binance перелічені інші біржі за рейтингом: OKEx на 2-му місці, Huobi Global на 3-му та Coinbase Pro на 5-му.





Exchanges	24h Trade Volume	Markets	Change
1.  Binance	\$22.3B	1.2K	24.2%
<p>Binance is a blockchain ecosystem comprising a cryptocurrency exchange, startup incubator, startup investor, and coin information provider. Binance is a highly popular exchange and allows its users to trade hundreds of cryptocurrencies, including Dogecoin, Bitcoin, and Ethereum.</p> <p>The exchange was founded in July 2017, by Changpeng Zhao (CZ), and was initially based in China, but was later moved out of China due to China's increasing regulation of cryptocurrency. The name 'Binance' comes from the two words 'binary' and 'finance'. Binance exchange requires KYC (know-your-customer), which is the process that banks gather identifying data and contact information from current and potential customers, to prevent fraud. Binance is capable of processing 1,400,000 orders per second and supports many cryptocurrencies. Binance serves more than 15,000,000 users worldwide, operates in over 40 countries, and is based in over 180 countries & regions.</p> <p>Binance coin (BNB)</p> <p>Binance coin (BNB) is the native asset on Binance Chain and powers the Binance ecosystem. BNB has several use cases, like fueling transactions on the chain, paying for transaction fees on Binance Exchange, and making in-store payments.</p>			
2.  OKEx	\$5.3B	462	5.7%
3.  Huobi Global	\$4.9B	827	5.3%
5.  Coinbase Pro	\$4.3B	268	4.7%

Рисунок 4.11 – Сторінка з інформацією про криптовалютні біржі

Користувачі можуть взаємодіяти з цією сторінкою наступним чином:

- перегляд інформації про кожну біржу, включаючи назву, обсяг торгів, кількість ринків та відсоток змін;
- вивчення детального опису вибраної біржі, який містить історію заснування, основні характеристики та сервіси, які вона пропонує, а також інформацію про власну криптовалюту біржі, якщо така є (наприклад, Binance Coin для Binance);
- можливість клікнути на назву біржі або інші інтерактивні елементи, для переходу на офіційний сайт біржі або на сторінку з більш глибоким аналізом її діяльності.

Ця сторінка допомагає інвесторам та трейдерам криптовалют оцінити та порівняти різні біржі на основі наданої інформації, щоб зробити інформований вибір щодо місця торгівлі. Для кожної біржі вказані наступні параметри: цілодобовий обсяг торгів, кількість ринків та відсоток змін.

4.3 Тестування розробленої системи

Тестування розробленої системи є важливим етапом в процесі розробки програмного забезпечення, що дозволяє перевірити коректність роботи системи та виявити можливі помилки перед її впровадженням у виробництво. Для забезпечення високої якості продукту, тестування повинно бути всебічним та систематизованим.

Перейдемо до фінальної стадії розробки – тестування. Для цього проведемо декілька тестів за допомогою різних джерел. Перший тест виконано за допомогою сервісу від компанії Google – LightHouse (рис. 4.12).

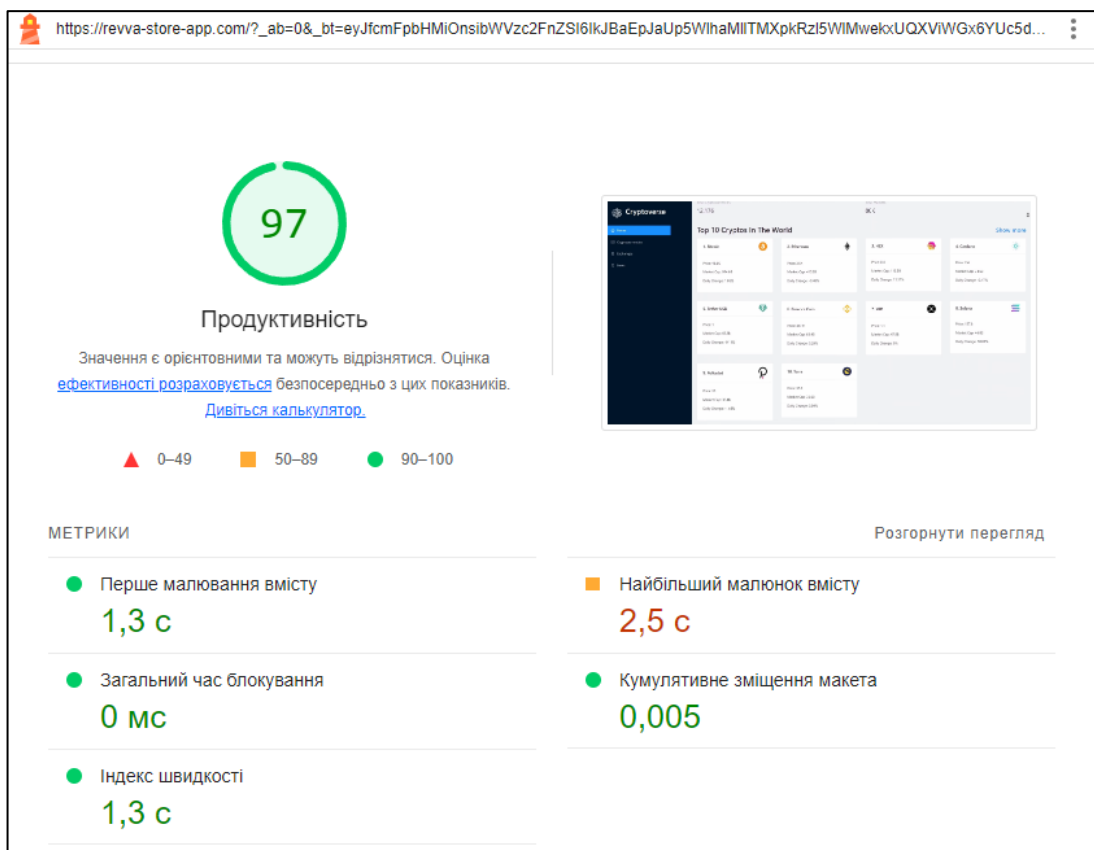


Рисунок 4.12 – Результати тесту за допомогою LightHouse

Результат тестування системи за допомогою інструменту LightHouse показує високий рівень продуктивності з оцінкою 97 із 100, що свідчить про оптимізоване завантаження та відмінну ефективність роботи веб-сайту. Час першого малювання вмісту складає 1,3 секунди, а індекс швидкості також становить 1,3 секунди, що вказує на швидке відображення вмісту сторінки. Кумулятивне зміщення макета дуже низьке (0,005), що демонструє стабільність візуального макета під час завантаження сторінки.

Наступним онлайн-сервісом обрано PageSpeed Insights. Це система інструментів, що була також розроблена компанією Google для оптимізації роботи вебсайтів (рис. 4.13).

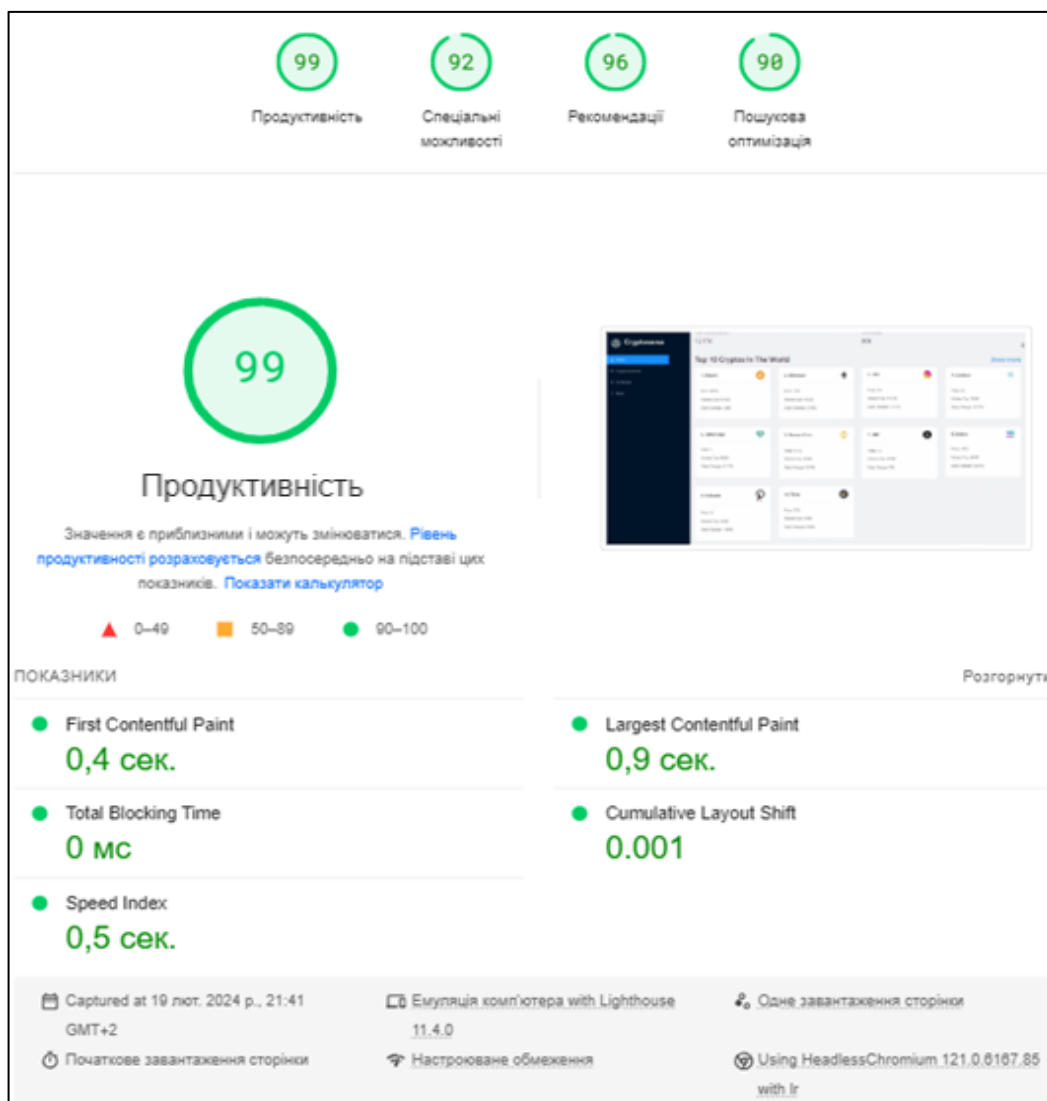


Рисунок 4.13 – Результат тесту за допомогою PageSpeed Insights

Результати тестування за допомогою PageSpeed Insights свідчать про відмінну продуктивність вебсайту з загальним рейтингом 99 зі 100. Вебсайт демонструє швидке завантаження, з часом First Contentful Paint всього 0,4 секунди та Speed Index 0,5 секунди, що підкреслює миттєве відображення вмісту користувачам. Значення Cumulative Layout Shift вкрай низьке (0,001), що вказує на мінімальні візуальні зміни під час завантаження, забезпечуючи стабільний користувацький досвід.

Для ще одного тесту візьмемо онлайн-джерело – GTmetrix. GTmetrix – це сервіс, який дозволяє зробити аналіз швидкості завантаження сайту та отримати рекомендації та подальші дії щодо оптимізації. Завдяки йому QA-інженер зможе проаналізувати багато різних показників та вказати на недоліки, після чого вже розробник зможе приступити до відповідно їх усунення. Онлайн-сервіс є значно швидшим за такі аналоги, як Pingdom Tools, WebPagetest та інші (рис. 4.14).

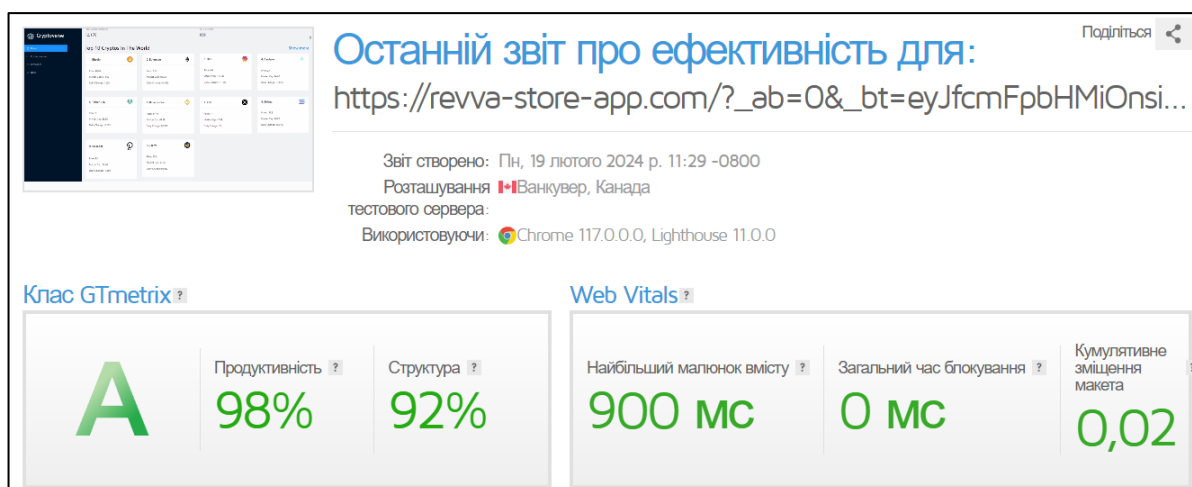


Рисунок 4.14 – Результат тесту за допомогою GTmetrix

Результати тестування веб-сайту за допомогою GTmetrix показують високу продуктивність, з оцінкою 98% за продуктивність та 92 % за структуру сторінки, що свідчить про оптимальну оптимізацію та відповідність кращим практикам. Час найбільшого малюнку вмісту (Largest Contentful Paint) становить лише 900 мілісекунд, що вказує на швидке завантаження найбільших елементів сторінки,

а кумулятивне зміщення макета (Cumulative Layout Shift) дуже низьке (0,02), гарантуючи стабільність візуального відображення під час завантаження.

Тестування розробленої системи не тільки виявляє потенційні помилки, але й дозволяє оцінити її продуктивність, сумісність та використання ресурсів, що є важливими аспектами для успішного функціонування системи. Окрім перевірки функціональності, тестування допомагає визначити, як система буде поводитися під час пікових навантажень і чи легко її можна буде масштабувати згідно з вимогами користувачів.

Використання таких інструментів як Lighthouse, PageSpeed Insights та GTmetrix забезпечує глибоке занурення в аналіз продуктивності вебзастосунку. Ці інструменти дозволяють не тільки виявити та усунути проблеми, що сповільнюють швидкість завантаження сторінок, але й оптимізувати ресурси для покращення загального користувацького досвіду.

За допомогою Lighthouse можна оцінити не тільки продуктивність, але й доступність, застосованість вебзастосунків, а також відповідність SEO-стандартам. PageSpeed Insights надає інформацію про те, як можна поліпшити час завантаження сторінки на мобільних пристроях та десктопах. GTmetrix же пропонує зручний спосіб візуалізації продуктивності завантаження сторінок та детальний звіт про необхідні оптимізації.

Крім того, для високої якості тестування важливо враховувати різноманітність користувацьких сценаріїв, включаючи різні типи браузерів, операційних систем і мережевих умов. Це допомагає забезпечити стабільність і надійність системи в різних умовах експлуатації.

Останнім етапом тестування є ретельний аналіз зібраних даних та внесення відповідних коректив у застосунок. Після усунення виявлених недоліків, важливо провести повторне тестування, щоб переконатися, що усі проблеми було виправлено і що система готова до впровадження. У якості підтвердження у розділі використовуються рисунки з урахуванням повторних тестів.

Враховуючи вищевказане, можна стверджувати, що ефективне тестування вебзастосунку Crypto Service App є інтегральною частиною розробки, що гарантує якість продукту та задоволеність користувачів. Використання передових інструментів тестування, таких як Lighthouse, PageSpeed Insights, та GTmetrix, надає розробникам детальне розуміння аспектів продуктивності, доступності та оптимізації вебсайту.

Висновки до розділу 4

Отже, було спроектовано та створено сервіс для системи, який з'єднує її з даними що приходять API. Ці дані відображають статистику криптовалют, новини та інформацію про біржі.

Кожна частина інтерфейсу була ретельно аналізована для забезпечення зручності та інформативності для користувачів. Зокрема, було встановлено, що інтерфейс надає користувачам зручні інструменти для доступу до актуальних даних про ціни криптовалют, їх ринкову капіталізацію та денні зміни вартості.

Новинний розділ відображає останні події у світі криптовалют, дозволяючи користувачам залишатися в курсі найсвіжіших трендів та розвитків.

Розділ бірж надає детальну інформацію про ключові торгові платформи, включаючи їх торговий обсяг та кількість ринків.

Також, було проведено декілька етапів тестування для системи, які показали чудовий результат. Для тестування було використано наступні сервіси: Lighthouse, PageSpeed Insights, та GTmetrix. Ці інструменти не лише допомагають виявити та вирішити проблеми, які можуть негативно вплинути на користувацький досвід, але й сприяють підвищенню швидкості завантаження, покращенню SEO, та забезпеченню кращої сумісності з різноманітними пристроями та браузерами. Таке тестування є ключовим для забезпечення, що застосунок буде не лише функціональним і безпечним, але й конкурентоспроможним на ринку.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи магістра було досягнуто поставлену мету – розроблено повнофункціональну систему аналізу криптовалютних бірж, що включає збір, обробку та візуалізацію даних з різних бірж за допомогою технологій React та Redux.

Для досягнення поставленої мети КРМ було виконано наступні завдання:

- було проаналізовано технічні характеристики біржових систем: досліджено архітектурні особливості, принципи безпеки та протоколи обміну криптовалютних бірж;

- було проаналізовано алгоритми торгів та стратегій, а також було розглянуто можливості їх інтегрування в системи криптовалютних бірж з використанням технологій React та Redux;

- було розроблено повнофункціональну систему, яка дозволяє користувачам здійснювати глибокий аналіз різноманітних криптовалютних бірж.;

- було використано сучасні технології, таких як React, Redux, Next.js, TypeScript, що дозволило створити ефективний, гнучкий та інтуїтивно зрозумілий інтерфейс. Ці технології сприяли підвищенню продуктивності розробки та забезпечили високий рівень взаємодії з користувачем;

- було протестовано вебінтерфейс біржової системи для визначення коректності роботи ключових функцій, таких як введення та виведення даних, виконання торгових операцій та взаємодія з графічним інтерфейсом.

Розроблена система забезпечує збір, обробку та візуалізацію даних у режимі реального часу, що є ключовим для прийняття обґрунтованих торгових рішень. Використання сучасних технологій, а саме React та Redux, забезпечило створення інтерактивного та зручного користувацького інтерфейсу, що є важливим для забезпечення ефективної взаємодії з системою. Впровадження Next.js та TypeScript підняло якість та надійність коду, забезпечивши стабільність роботи та гнучкість системи.

Інноваційність роботи полягає в розширенні інструментарію для аналізу ринку криптовалют. Розроблена система забезпечує користувачам доступ до широкого спектру даних, сприяючи кращому розумінню ринкових тенденцій та допомагаючи в інвестиційному аналізі.

Після розробки системи було проведено тестування за допомогою трьох сервісів: LightHouse, GTmetrix та PageSpeed Insights. Результат показав, що система для аналізу криптовалют та їх систем торгів швидка та оптимізована.

У перспективі можна буде розвивати систему за допомогою впровадження додаткових інноваційних технологій, таких як машинне навчання та штучний інтелект, для підвищення точності аналізу та прогнозування ринкових тенденцій. Це дозволить системі адаптуватися до постійно змінних умов ринку, забезпечуючи її актуальність та цінність для користувачів.

Результати кваліфікаційної роботи магістра пройшли апробацію під час XXVI Всеукраїнської науково-практичної конференції «Могилянські читання» (Миколаїв, 06–10 листопада 2023 р.).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Choi D. Full-Stack React, TypeScript, and Node Build Cloud-ready Web Applications Using React 17 with Hooks and GraphQL. United Kingdom : Packt Publishing Ltd. 2023. 648 p. ISBN 978-1-83921-993-1.
2. Kumar R., Mishra A., Mollick M., Chaurasiya H., Roy K. Cryptocurrency Tracker for Real- Time Market Analysis and Monitoring Using React JS. *International Journal of Engineering Applied Sciences and Technology*. 2022. Vol. 8. №. 1. P. 177–184. DOI: 10.33564/IJEAST.2023.v08i01.028.
3. Prasetyo D., Ariati N., Lubis H., Akbar A. Cryptocurrency Exchange Selection Decision Support System Using Preference Selection Index. *SaNa: Journal of Blockchain, NFTs and Metaverse Technology*. 2023. Vol. 1. №. 1: 23–29. DOI: 10.58905/sana.v1i1.148.
4. Roldan C. React Cookbook Create Dynamic Web Apps with React Using Redux, Webpack, Node.js, and GraphQL. United Kingdom : Packt Publishing Ltd. August 2018. 580 p. ISBN 978-1-78398-072-7.
5. Sathish Kumar R., Indhumathi V. Crypto-Currency App Using React. *International Journal of Research Publication and Reviews*. 2023. Vol. 4. № 6. P. 3294–3299. ISSN 2582-7421.
6. Tarasov V. A. Application of Technical Analysis for Cryptocurrency Market Research. *Visnik Sums'kogo deržavnogo unìversitetu*. 2022. P. 274–282. DOI: 10.21272/1817-9215.2022.4-29.
7. Tsapro I. V. Adaptation of the mechanistic approach to the analysis of the cryptocurrency market. *Connectivity*. 2023. №. 1 (161). 10.31673/2412-9070.2023.014647.
8. Zhu T., Liao Ch., Zhou Z., Li X., Zhang Q. Fast Popularity Value Calculation of Virtual Cryptocurrency Trading Stage Based on Machine Learning. *Frontiers in Physics*. 2022. Vol. 9. DOI: 10.3389/fphy.2021.788508.

9. Безверхий О., Куценко О. Ефективність застосування бібліотеки React. *Інформаційні технології та суспільство*. 2022. Вип. 2 (4). С. 31–34. DOI: 10.32689/maup.it.2022.2.2.
10. Ревва І. А., Обухова К. О. Аналіз криптовалютних бірж та їхніх систем торгів з використанням React та Redux. *Могилянські читання – 2023 : тези доп. XXVI Всеукр. наук.-метод. конф. Миколаїв, 6–10 листоп. 2023 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2023. С. 385–387.*
11. Рейтинг криптобірж – 2023: як інвесторам знайти баланс між безпекою та якістю послуг. URL: <https://mind.ua/publications/20259453-rejting-kriptobirzh-2023-yak-investoram-znajti-balans-mizh-bezpekoju-ta-yakistyu-poslug> (дата звернення: 09.10.2023).
12. Фундаментальний і технічний аналіз криптовалют: зв'язок, подібності та відмінності. URL: <https://www.cryptohackers.club/2019/07/fundamentalniy-i-tehnicniy-analiz-kryptovalyut.html> (дата звернення: 10.10.2023).
13. Biswas N. *Beginning React and Firebase: Create Four Beginner-Friendly Projects Using React and Firebase*. USA : Apress Berkeley. November 2021. DOI: 10.1007/978-1-4842-7812-3.
14. Biswas N. *Using React Redux with TypeScript*. In book: *TypeScript Basics*. USA : Apress Berkeley. October 2023. P. 153–160. DOI: 10.1007/978-1-4842-9523-6_11.
15. Arslanian H. *Crypto Exchanges*. In: *The Book of Crypto*. Palgrave Macmillan, Cham. May 2022. P. 335–350. DOI: 10.1007/978-3-030-97951-5_18.
16. Appel H. *Regulatory Classification of Crypto Assets*. In: *Quick Guide Crypto Assets*. Springer Gabler, Wiesbaden. March 2023. P. 27–58. DOI: 10.1007/978-3-658-40462-8_2.
17. Khan S. M. *API Development Using Asp.Net Core Web API: A practical approach for developing the APIS in asp.net core*. 2023. 221 p.

18. Callaghan M. D. Understand Your Service API. *In: Angular for Business*. Apress, Berkeley, CA. December 2023. P. 133–136. DOI: 10.1007/978-1-4842-9609-7_10.
19. Himschoot P. Single-Page Applications and Routing. *In: Microsoft Blazor*. Apress, Berkeley, CA. December 2021. P. 351–387. DOI: 10.1007/978-1-4842-7845-1_9.
20. Mukhiya S., Wei T., Lee J. Redux Quick Start Guide. Packt. 2019. 204 p. ISBN: 978-1-78961-008-6
21. Thakkar M. Next.js. *In: Building React Apps with Server-Side Rendering*. Apress, Berkeley, CA. April 2020. P. 93–137. DOI: 10.1007/978-1-4842-5869-9_3.
22. Thakkar M. Adding Server-Side Rendering to Your React Application. *In: Building React Apps with Server-Side Rendering*. Apress, Berkeley, CA. April 2020. P. 139–152. DOI: 10.1007/978-1-4842-5869-9_4.
23. Duldulao D. B., Cabagnet R. J. L. Navigating React Router. Navigating React Router. *In: Practical Enterprise React*. Apress, Berkeley, CA. August 2021. P. 55–90. DOI: 10.1007/978-1-4842-6975-6_5.
24. Subramanian V. React Router. *In: Pro MERN Stack*. Apress, Berkeley, CA. May 2019. P. 233–265. DOI: 10.1007/978-1-4842-4391-6_9.
25. Porfirenko V., Melnyk, M., Ivanov V., Hobta M., Diachenko T., Vysochylo O. Organization and Features of Investing in the Cryptocurrency Market in Ukraine. *In: Khoury, R.E., Nasrallah, N. (eds) Intelligent Systems, Business, and Innovation Research. Studies in Systems, Decision and Control*. Vol. 489. Springer, Cham. January 2024. P. 213–225. DOI: 10.1007/978-3-031-36895-0_18.
26. Tripathi A., Choudhary A., Arora S. K., Arora G., Shakya G., Rajwanshi B. Crypto Bank: Cryptocurrency Wallet Based on Blockchain. *Recent Trends in Image Processing and Pattern Recognition (RTIP2R 2023): Communications in Computer and Information Science*. Vol. 2027. Springer, Cham. January 2024. P. 223–236. DOI: 10.1007/978-3-031-53085-2_19.

ДОДАТОК А

Лістинг коду системи

Код компоненту *Navbar.tsx*:

```
import React, { useState, useEffect } from 'react';
import { Button, Menu, Typography, Avatar } from 'antd';
import { Link } from 'react-router-dom';
import { HomeOutlined, MoneyCollectOutlined, BulbOutlined, FundOutlined,
MenuOutlined } from '@ant-design/icons';

import icon from '../images/cryptocurrency.png';

const Navbar = () => {
  const [activeMenu, setActiveMenu] = useState(true);
  const [screenSize, setScreenSize] = useState(undefined);

  useEffect(() => {
    const handleResize = () => setScreenSize(window.innerWidth);

    window.addEventListener('resize', handleResize);

    handleResize();

    return () => window.removeEventListener('resize', handleResize);
  }, []);

  useEffect(() => {
    if (screenSize <= 800) {
      setActiveMenu(false);
    } else {
      setActiveMenu(true);
    }
  }, [screenSize]);

  return (
    <div className="nav-container">
      <div className="logo-container">
        <Avatar src={icon} size="large" />
        <Typography.Title level={2} className="logo"><Link to="/">Revva
<br/>Cryptoservice</Link></Typography.Title>
        <Button className="menu-control-container" onClick={() =>
setActiveMenu(!activeMenu)}><MenuOutlined /></Button>

```

```

</div>
{activeMenu && (
  <Menu theme="dark">
    <Menu.Item icon={<HomeOutlined />}>
      <Link to="/">Home</Link>
    </Menu.Item>
    <Menu.Item icon={<FundOutlined />}>
      <Link to="/cryptocurrencies">Cryptocurrencies</Link>
    </Menu.Item>
    <Menu.Item icon={<MoneyCollectOutlined />}>
      <Link to="/exchanges">Exchanges</Link>
    </Menu.Item>
    <Menu.Item icon={<BulbOutlined />}>
      <Link to="/news">News</Link>
    </Menu.Item>
  </Menu>
)}
</div>
);
};

```

export default Navbar;

Код компоненту *CryptoDetails.tsx*:

```

import React, { useState } from 'react';
import HTMLReactParser from 'html-react-parser';
import { useParams } from 'react-router-dom';
import millify from 'millify';
import { Col, Row, Typography, Select } from 'antd';
import { MoneyCollectOutlined, DollarCircleOutlined, FundOutlined,
  ExclamationCircleOutlined, StopOutlined, TrophyOutlined, CheckOutlined,
  NumberOutlined, ThunderboltOutlined } from '@ant-design/icons';

import { useGetCryptoDetailsQuery, useGetCryptoHistoryQuery } from
'../services/cryptoApi';
import Loader from './Loader';
import LineChart from './LineChart';

const { Title, Text } = Typography;
const { Option } = Select;

const CryptoDetails = () => {
  const { coinId } = useParams();

```

```

const [timeperiod, setTimeperiod] = useState('7d');
const { data, isFetching } = useGetCryptoDetailsQuery(coinId);
const { data: coinHistory } = useGetCryptoHistoryQuery({ coinId, timeperiod });
const cryptoDetails = data?.data?.coin;

if (isFetching) return <Loader />;

const time = ['3h', '24h', '7d', '30d', '1y', '3m', '3y', '5y'];

const stats = [
  { title: 'Price to USD', value: `\$ ${cryptoDetails?.price &&
millify(cryptoDetails?.price)}`, icon: <DollarCircleOutlined /> },
  { title: 'Rank', value: cryptoDetails?.rank, icon: <NumberOutlined /> },
  { title: '24h Volume', value: `\$ ${cryptoDetails?.volume &&
millify(cryptoDetails?.volume)}`, icon: <ThunderboltOutlined /> },
  { title: 'Market Cap', value: `\$ ${cryptoDetails?.marketCap &&
millify(cryptoDetails?.marketCap)}`, icon: <DollarCircleOutlined /> },
  { title: 'All-time-high(daily avg.)', value: `\$
${cryptoDetails?.allTimeHigh?.price &&
millify(cryptoDetails?.allTimeHigh?.price)}`, icon: <TrophyOutlined /> },
];

const genericStats = [
  { title: 'Number Of Markets', value: cryptoDetails?.numberOfMarkets, icon:
<FundOutlined /> },
  { title: 'Number Of Exchanges', value: cryptoDetails?.numberOfExchanges, icon:
<MoneyCollectOutlined /> },
  { title: 'Approved Supply', value: cryptoDetails?.supply?.confirmed ?
<CheckOutlined /> : <StopOutlined />, icon: <ExclamationCircleOutlined /> },
  { title: 'Total Supply', value: `\$ ${cryptoDetails?.supply?.total &&
millify(cryptoDetails?.supply?.total)}`, icon: <ExclamationCircleOutlined /> },
  { title: 'Circulating Supply', value: `\$ ${cryptoDetails?.supply?.circulating
&& millify(cryptoDetails?.supply?.circulating)}`, icon: <ExclamationCircleOutlined
/> },
];

return (
  <Col className="coin-detail-container">
    <Col className="coin-heading-container">
      <Title level={2} className="coin-name">
        {data?.data?.coin.name} ({data?.data?.coin.symbol}) Price
      </Title>
      <p>{cryptoDetails.name} live price in US Dollar (USD). View value

```

```

statistics, market cap and supply.</p>
  </Col>
  <Select defaultValue="7d" className="select-timeperiod" placeholder="Select
Timeperiod" onChange={(value) => setTimeperiod(value)}>
    {time.map((date) => <Option key={date}>{date}</Option>)}
  </Select>
  <LineChart coinHistory={coinHistory}
currentPrice={millify(cryptoDetails?.price)} coinName={cryptoDetails?.name} />
  <Col className="stats-container">
    <Col className="coin-value-statistics">
      <Col className="coin-value-statistics-heading">
        <Title level={3} className="coin-details-heading">{cryptoDetails.name}
Value Statistics</Title>
        <p>An overview showing the statistics of {cryptoDetails.name}, such as
the base and quote currency, the rank, and trading volume.</p>
      </Col>
      {stats.map(({ icon, title, value }) => (
        <Col className="coin-stats">
          <Col className="coin-stats-name">
            <Text>{icon}</Text>
            <Text>{title}</Text>
          </Col>
          <Text className="stats">{value}</Text>
        </Col>
      ))}
    </Col>
    <Col className="other-stats-info">
      <Col className="coin-value-statistics-heading">
        <Title level={3} className="coin-details-heading">Other Stats
Info</Title>
        <p>An overview showing the statistics of {cryptoDetails.name}, such as
the base and quote currency, the rank, and trading volume.</p>
      </Col>
      {genericStats.map(({ icon, title, value }) => (
        <Col className="coin-stats">
          <Col className="coin-stats-name">
            <Text>{icon}</Text>
            <Text>{title}</Text>
          </Col>
          <Text className="stats">{value}</Text>
        </Col>
      ))}
    </Col>
  </Col>

```

```

</Col>
<Col className="coin-desc-link">
  <Row className="coin-desc">
    <Title level={3} className="coin-details-heading">What is
{cryptoDetails.name}</Title>
    {HTMLReactParser(cryptoDetails.description)}
  </Row>
  <Col className="coin-links">
    <Title level={3} className="coin-details-heading">{cryptoDetails.name}
Links</Title>
    {cryptoDetails.links?.map((link) => (
      <Row className="coin-link" key={link.name}>
        <Title level={5} className="link-name">{link.type}</Title>
        <a href={link.url} target="_blank" rel="noreferrer">{link.name}</a>
      </Row>
    ))}
  </Col>
</Col>
</Col>
);
};

```

```
export default CryptoDetails;
```

Код компоненту *LineChart.tsx*:

```

import React from 'react';
import { Line } from 'react-chartjs-2';
import { Col, Row, Typography } from 'antd';

const { Title } = Typography;

const LineChart = ({ coinHistory, currentPrice, coinName }) => {
  const coinPrice = [];
  const coinTimestamp = [];

  for (let i = 0; i < coinHistory?.data?.history?.length; i += 1) {
    coinPrice.push(coinHistory?.data?.history[i].price);
  }

  for (let i = 0; i < coinHistory?.data?.history?.length; i += 1) {
    coinTimestamp.push(new
Date(coinHistory?.data?.history[i].timestamp).toLocaleDateString());
  }

```

```
const data = {
  labels: coinTimestamp,
  datasets: [
    {
      label: 'Price In USD',
      data: coinPrice,
      fill: false,
      backgroundColor: '#0071bd',
      borderColor: '#0071bd',
    },
  ],
};

const options = {
  scales: {
    yAxes: [
      {
        ticks: {
          beginAtZero: true,
        },
      },
    ],
  },
};

return (
  <>
    <Row className="chart-header">
      <Title level={2} className="chart-title">{coinName} Price Chart </Title>
      <Col className="price-container">
        <Title level={5} className="price-change">Change:
{coinHistory?.data?.change}%</Title>
        <Title level={5} className="current-price">Current {coinName} Price: $
{currentPrice}</Title>
      </Col>
    </Row>
    <Line data={data} options={options} />
  </>
);
};

export default LineChart;
```


ДОДАТОК Б

Апробація кваліфікаційної роботи магістра

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
ДНУ «Інститут модернізації змісту освіти»
Південний науковий центр НАН та МОН
Інститут української археографії та джерелознавства
імені М. С. Грушевського НАН України



«МОГИЛЯНСЬКІ ЧИТАННЯ – 2023:

**досвід та тенденції розвитку суспільства в Україні: глобальний,
національний та регіональний аспекти»**

XXVI Всеукраїнська науково-практична конференція

ТЕЗИ ДОПОВІДЕЙ

Миколаїв, 6–10 листопада 2023 року

Миколаїв – 2023

Могиллянські читання–2023: досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти.

Гекова Т. В., Кумпан І. Д. Розробка інтегрованої системи автоматизованого управління логістичними процесами в промисловому середовищі365

Гекова Т. В., Купчик В. М. Розробка та впровадження системи автоматизації в складах з використанням автономних транспортних засобів366

Підсекція:

МОДЕЛІ, МЕТОДИ ТА ЗАСОБИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Антіпова К. О., Андрєєва Ю. А. Аналіз відомостей про ноутбуки та їх характеристики367

Гончарова Н. В., Гончаров Д. С. Ймовірно-графічні моделі для оцінки сценаріїв формування контингенту студентів ЗВО371

Горбань Г. В. Застосування мобільної об'єктної СКБД Realm у застосунках React Native374

Давиденко Є. О., Бондаренко С. В. Інтелектуальне управління проектами377

Кірей К. О. Підвищення відмовостійкості автоматизованої системи розрахунків оператора електронних комунікацій380

Лункіна А. В., Обухова К. О. Програмне забезпечення автоматизації бізнес-процесів онлайн-школи маркетингу382

Раленко В. С., Стоєв Є. Д. Унікалізація ігрових застосунків для ОС Android383

Рєва І. А., Обухова К. О. Аналіз криптовалютних бірж та їхніх систем торгів з використанням React та Redux385

Стоєв Є. Д., Раленко В. С. Використання вебсокетів в інформаційних системах реального часу387

Федас Ю. М., Боровльова С. Ю. Використання geCAPTCHA в React388

Фісун М. Т., Кандиба І. О. Використання інструментарію Python для реалізації застосунків семантичного вебу390

Підсекція:

КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Dvornik O. V. The effect of filtering and smoothing on the correlation of blood pressure and heart rate data sets391

Yaremchuk O. M. Data mining of ambulatory blood pressure monitoring393

Аскеров В. В., Павлова О. О. Метод покращення перевірок AML шляхом зміни парадигми ставлення до кожної окремої транзакції395

Басє В. О., Пузирьов С. В. Оркестрація розподіленої сенсорної IoT-мережі на базі BPNM396

Біліська А. Є., Говоруценко Т. О. Система догляду за домашніми тваринами на основі технології Інтернет речей (IoT)398

Булавський Д. В., Бойко А. П. Моніторинг якості води на основі модуля ESP32401

Галицький О. С., Павлова О. О. Система керування квадрокоптером з використанням ретранслятора403

Горішана О. М., Бурлаченко І. С. Інноваційні периферійні пристрої в комп'ютерних системах406

Гюльмамедов Н. М., Невідомий Д. О., Чередищенко Д. О., Волочай П. О., Бурлаченко І. С. Автомобільна платформа для перевезення медикаментів або поранених людей408

Мотиви читання – 2023: досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти.

потенційних вразливостей безпеки в додатку. Зміни, методи та класи з некорумбованими назвами ускладнюють пошук та виправлення конкретних компонентів додатка зовнішніми.

5. Виконання вимог відповідності: В деяких випадках, зокрема при розробці фінансових або медичних додатків, вимоги до безпеки можуть вимагати додаткової обробки коду.

Для додаткової унікальності Android застосунку та його коду доцільно використовувати програмне забезпечення, яке розроблене на мові програмування Java та використовує бібліотеку JavaParser API.

```
private void refreshStatistics() {
    for (Map.Entry<String, Pair> sourceItem : listSourceItems.entrySet()) {
        for (ParseItem.ComparisonItem item : module.getItemValue(0).b) {
            try {
                methodDeclaration = class.stream().filter((MethodDeclaration method) -> {
                    methodDeclaration.isProtected(0);
                    methodDeclaration.isPrivate(0);
                    methodDeclaration.isPublic(true);
                });
            } catch (Exception e) {
                //
            }
        }
    }
}

for (Map.Entry<String, Pair> sourceItem : listSourceItems.entrySet()) {
    for (ParseItem.ComparisonItem item : module.getItemValue(0).b) {
        try {
            methodDeclaration = class.stream().filter((MethodDeclaration method) -> {
                methodDeclaration.isProtected(0);
                methodDeclaration.isPrivate(0);
                methodDeclaration.isPublic(true);
            });
        } catch (Exception e) {
            //
        }
    }
}
```

Рисунок 1 – Приклад використання JavaParser API

Завершуючи, унікалізація гравих застосунків на платформі Android – це важливий крок для досягнення успіху в конкурентному світі мобільних ігор. Розробники повинні постійно працювати над тим, щоб надати своїм грі унікальну стиль і якість, які відрізняють їх серед інших. Незважаючи на жорстку конкуренцію, можливість для творчого вираження та комерційного успіху на платформі Android залишаються великими для тих, хто готовий працювати над створенням унікальних ігор.

УДК 004.35

*Рева І. А.,
магістранти спеціальності 121 «Інженерія програмного забезпечення»,
Обухова К. О.,
ст. викладач кафедри комп'ютерної інженерії,
ЦНУ імені Петра Могили, м. Миколаїв, Україна*

**АНАЛІЗ КРИПТОВАЛЮТНИХ БІРЖ ТА ЇХНІХ СИСТЕМ ТОРГІВ
З ВИКОРИСТАННЯМ REACT ТА REDUX**

Криптовалютні ринки та цифрові активи стають все більш актуальними та значущими в сучасному світі фінансів та інвестицій. Швидкий та часто непередбачуваний розвиток криптовалют і блокчейн-технологій породжує необхідність аналізу ринкових даних для успішної торгівлі та прийняття обґрунтованих інвестиційних рішень.

Криптовалютні ринки характеризуються високою волатильністю й швидкими змінами ціни. Торгівля на таких ринках вимагає постійного аналізу та моніторингу, щоб вчасно реагувати на зміни й уникнути фінансових втрат.

Тем доповіді

Точність та оперативність аналізу ринкових даних є критичними для трейдерів та інвесторів. Інформація про ціни, торгові обсяги й інші показники грають важливу роль у прийнятті торговельних рішень. Тому інвестори шукають шляхи для ефективного аналізу ринку та максимізації своїх прибутків у цьому конкурентному середовищі.

Використання сучасних технологій, таких як React та Redux, дозволяє розробникам створювати швидкі, ефективні й інтуїтивно зрозумілі інструменти для аналізу та моніторингу криптовалютних ринків.

Трейдерів і аналітиків криптовалютних ринків потрібні інструменти, які дозволяють ефективно й оперативно аналізувати та прогнозувати рух цін криптовалют для досягнення успішних торговельних стратегій.

Розподіл капіталу на криптовалютному ринку за типами активів наведено на рис. 1.

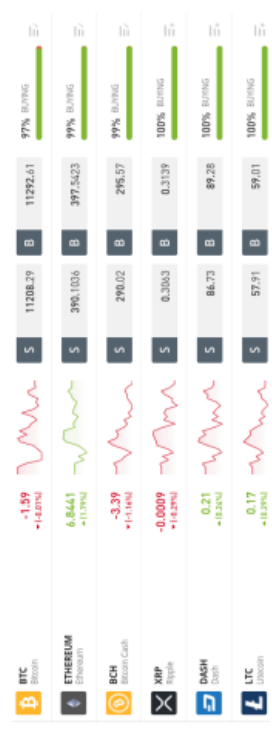


Рисунок 1 – Капітал на криптовалютному ринку

Наведена діаграма відображає структуру розподілу капіталу на криптовалютному ринку та вказує на важливість і різноманітність типів активів у цьому сегменті фінансового ринку. Bitcoin (BTC) та Ethereum (ETH) виступають як ключові гравці, аналітики представляють широкий спектр можливостей, стабільності та забезпечують стабільність, а токеновані активи відкривають нові можливості для інвесторів.

Bitcoin є першою та найбільш важливою криптовалютою на ринку. Його вага свідчить про домінування та вплив BTC на всеосяжному Ethereum є другою за важливістю криптовалютою та важливою блокчейн-платформою, яка дозволяє створювати смарт-контракти та децентралізовані застосунки.

Окремі застосунки є розробкою застосунку для аналізу криптовалютних бірж та їх систем торгів з використанням технологій React та Redux. Даний застосунок має спростити та полегшити аналіз ринку криптовалют, надаючи користувачам інтуїтивно зрозумілий інтерфейс і потужні аналітичні можливості.

Для розробки застосунку будуть використані сучасні технології, такі як React та Redux. React – це бібліотека для створення інтерфейсів користувача, яка дозволяє швидко й ефективно розробляти зручні та інтерактивні вебзастосунки. Redux – це бібліотека для управління станом застосунку, що спрощує обробку й оновлення даних у великих та складних застосунках.

Розробка застосунку для аналізу криптовалютних бірж та їх систем торгів є важливим кроком у полегшенні роботи трейдерів й інвесторів на сьогоднішньому динамічному ринку. Такий застосунок може бути покращений шляхом інтеграції розширених аналітичних засобів, оптимізації продуктивності та додаванням нових функцій, що полегшать прийняття розумних торговельних рішень.

Подальші напрями розробки можуть включати:

- розширення функціональності для аналізу ринкових даних та паттернів;
 - інтеграцію з різними криптовалютними біржами та їх API для більш широкого покриття ринку;
 - реалізацію можливості автоматизованих торговельних стратегій;
 - удосконалення інтерфейсу та додавання персоналізованих налаштувань для користувачів.
- Таким чином, розробка застосунку для аналізу криптовалютних бірж та систем торгів на основі сучасних технологій є вкрай актуальною задачею, яка сприятиме покращенню та оптимізації аналітичних процесів у сфері криптовалютних інвестицій то торгівлі.

Список використаних джерел

1. Бевзверкий О., Кученко О. Ефективність застосування бібліотеки React. *Інформаційні технології та суспільство*, 2022. Вип. 2 (4). С. 31-34. DOI: 10.32689/inf.it.2022.2.2.
2. Рейтинг криптобірж – 2023: як інвесторам знайти баланс між безпекою та якістю послуг. URL: <https://mind.ua/publications/20259453-rejting-kriptobirzh-2023-yak-investoram-znaji-balans-mizh-bezpekoju-i-yakistyu-poslug> (дата звернення: 09.10.2023).