

***БАКАЛАВРСКА
РОБОТА***

БР.ІІ – 67.00.00.000 ІІЗ

Група ІІ-22-3

Дзеба Назар

2026

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Дзеба Назар Володимирович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

БАКАЛАВРСЬКА РОБОТА

Розроблення інтелектуального веб-застосунку для аналізу криптовалютного ринку

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

**Робота містить результати власних досліджень, використання ідей, результатів і
текстів інших авторів мають посилання на відповідне джерело:**

Здобувач освітнього ступеня Дзеба Н.В.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Зікратий Сергій Вікторович, доцент, к.т.н.
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІІЗ

доцент.

В.В. Бандура

“ ” 2026 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Дзеба Назару Володимировичу

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) "Розроблення інтелектуального веб-застосунку для аналізу криптовалютного ринку "

керівник проекту (роботи) доц. , к.т.н., Зікратий С.В.

затверджені наказом вищого навчального закладу від “ 21 ” квітня 2026 р. № 179/7

2. Строк подання студентом проекту (роботи) 10 червня 2026 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз вимог до програмного забезпечення

2. Аналіз вимог і постановка задачі

3. Проектування системи

4. Реалізація проекту

5. Тестування та впровадження

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

1 Приклад просторового арбітражу (рис. 1.1, ст. 19)

2 Приклад внутрішньобіржового арбітражу (рис. 1.2, ст. 20)

3 Методологія KDD (рис. 2.1, ст. 31)

4 Збір даних (рис. 2.3, ст. 34)

5 Архітектура системи (рис. 2.11, ст. 46)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

2. Дата видачі завдання 2026 р.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Визначення та обґрунтування теми роботи	17.03.2026	виконано
2	Огляд існуючих концепцій, рішень та сервісів в даній області	21.04.2026	виконано
3	Побудова моделі або алгоритму власного рішення	01.05.2026	виконано
4	Документування реалізації власного оригінального рішення вибраними засобами	21.05.2026	виконано
5	Оформлення пояснювальної записки бакалаврської роботи	09.06.2026	виконано

Студент _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 84 сторінок, 42 рисунка, 2 таблиці, список використаних джерел із 40 найменувань.

Метою роботи є розроблення інтелектуального веб-застосунку для аналізу криптовалютного ринку, який забезпечує збір, обробку та візуалізацію ринкових даних, моніторинг змін вартості криптовалют.

Об'єкт дослідження: аналіз та моніторинг криптовалютного ринку.

Предмет дослідження: методи, моделі та технології розроблення інтелектуальних веб-застосунків для аналізу криптовалютних активів, прогнозування ринкових тенденцій та візуалізації даних.

Результати дослідження: розроблено інтелектуальний веб-застосунок, який забезпечує отримання актуальних даних про криптовалюту, аналіз ринкових показників, візуалізацію динаміки курсів та формування звітів.

У першому розділі проведено аналіз предметної області, розглянуто особливості криптовалютного ринку та принципи функціонування інтелектуальних веб-застосунків.

У другому розділі визначено функціональні та нефункціональні вимоги до системи, обґрунтовано вибір технологій розроблення, а також спроектовано дизайн і архітектуру веб-застосунку.

У третьому розділі описано програмні засоби реалізації системи, розроблено графічний інтерфейс користувача та виконано аналіз результатів використання створеного веб-застосунку.

Висновок: розроблений інтелектуальний веб-застосунок забезпечує ефективний аналіз криптовалютного ринку, підвищує зручність роботи користувачів з фінансовими даними.

КЛЮЧОВІ СЛОВА: КРИПТОВАЛЮТНИЙ РИНОК, ІНТЕЛЕКТУАЛЬНИЙ ВЕБ-ЗАСТОСУНОК, АНАЛІЗ ДАНИХ, ПРОГНОЗУВАННЯ, ВІЗУАЛІЗАЦІЯ ДАНИХ, ВЕБ-ТЕХНОЛОГІЇ, C#, ASP.NET CORE, JAVASCRIPT, API, КРИПТОВАЛЮТА.

ANNOTATION

The bachelor's thesis contains of 84 pages, 42 figures, 2 tables, and a reference list containing 40 sources.

The aim of the work is to develop an intelligent web application for cryptocurrency market analysis that provides data collection, processing, and visualization, monitors cryptocurrency price changes.

Research object: processes of cryptocurrency market analysis and monitoring.

Research subject: methods, models, and technologies for developing intelligent web applications for cryptocurrency asset analysis, market trend forecasting, and data visualization.

Research results: an intelligent web application was developed that provides access to up-to-date cryptocurrency data, analyzes market indicators, visualizes price dynamics, and generates analytical reports for users.

The first section analyzes the subject area, examines the characteristics of the cryptocurrency market, considers the principles of intelligent web applications, and performs a system analysis of the research object.

The second section defines the functional and non-functional requirements of the system, justifies the selected development technologies, and designs the architecture and user interface of the web application.

The third section describes the software tools used for implementation, presents the graphical user interface, and analyzes the results of using the developed web application.

Conclusion: the developed intelligent web application provides effective cryptocurrency market analysis, improves user interaction with financial data, and can be used to support investment decision-making processes.

KEY WORDS: CRYPTOCURRENCY MARKET, INTELLIGENT WEB APPLICATION, DATA ANALYSIS, FORECASTING, DATA VISUALIZATION, WEB TECHNOLOGIES, C#, ASP.NET CORE, JAVASCRIPT, API, CRYPTOCURRENCY.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБЛЕННЯ ІНТЕЛЕКТУАЛЬНИХ ВЕБ-ЗАСТОСУНКІВ ДЛЯ АНАЛІЗУ КРИПТОВАЛЮТНОГО РИНКУ	9
1.1 Основи функціонування інтелектуальних веб-застосунків	9
1.2 Криптовалютний ринок та особливості аналізу криптовалют	11
1.3 Системний аналіз об'єкта дослідження	25
1.4 Висновок до розділу	29
РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОГО ВЕБ-ЗАСТОСУНКУ	30
2.1 Функціональні вимоги до веб-застосунку	30
2.2 Нефункціональні вимоги до системи	41
2.3 Проєктування дизайну та архітектури веб-застосунку	44
2.4 Висновок до розділу	47
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНОГО ВЕБ-ЗАСТОСУНКУ	48
3.1 Програмні засоби реалізації системи аналізу криптовалютного ринку	48
3.2 Розроблення графічного інтерфейсу користувача	57
3.3 Аналіз результатів використання та оцінка ефективності веб-застосунку ..	63
3.4 Висновок до розділу	79
ВИСНОВКИ	80
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА	82

					БР.ІП – 00.00.00.000 ІПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Дзеба Н.В.			Розроблення інтелектуального веб-застосунку для аналізу криптовалютного ринку				Літ.		Арк.	Акрушів
Перевір.		Зікратий С.В.									7	
Реценз.		Процюк В.Р.							ІФНТУНГ ІП-22-3			
Н. Контр.		Піх М.М.										
Затверд.		Бандура В. В.										
					Пояснювальна записка							

ВСТУП

У сучасному світі цифрові фінансові технології стрімко розвиваються та відіграють важливу роль у глобальній економіці. Одним із найбільш динамічних напрямів розвитку фінансових технологій є криптовалютний ринок, який характеризується високою волатильністю, значними обсягами даних та безперервним потоком інформації. Зростання популярності криптовалют, таких як Bitcoin, Ethereum та інших цифрових активів, створює потребу в сучасних програмних засобах, які дозволяють ефективно аналізувати ринкові тенденції, відстежувати зміни курсів та приймати обґрунтовані інвестиційні рішення.

Сучасний криптовалютний ринок функціонує цілодобово та генерує великі обсяги інформації, що потребують оперативної обробки та аналізу. Інвестори, трейдери та аналітики змушені працювати з численними джерелами даних, біржовими платформами, графіками та статистичними показниками.

Актуальність теми зумовлена зростанням ролі криптовалют у світовій фінансовій системі та потребою користувачів у надійних інструментах аналізу ринку. Існуючі програмні рішення часто мають обмежений функціонал, складний інтерфейс або не забезпечують достатнього рівня аналітичної підтримки для прийняття рішень. Крім того, значна кількість платформ орієнтована лише на відображення базових статистичних даних без використання інтелектуальних методів аналізу та прогнозування..

Метою роботи є розроблення інтелектуального веб-застосунку для аналізу криптовалютного ринку, який забезпечує отримання, обробку та візуалізацію актуальних ринкових даних, аналіз динаміки криптовалютних активів, формування аналітичних показників та підтримку прийняття рішень користувачами.

Для досягнення поставленої мети необхідно виконати **такі завдання**: провести аналіз предметної області та особливостей функціонування криптовалютного ринку; дослідити сучасні веб-застосунки та сервіси для аналізу

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

криптовалют; виконати системний аналіз об'єкта дослідження; визначити функціональні та нефункціональні вимоги до програмного продукту; обґрунтувати вибір технологій та інструментів розробки; спроектувати архітектуру та структуру веб-застосунку; реалізувати програмні модулі системи та графічний інтерфейс користувача; провести аналіз результатів використання розробленого програмного забезпечення;

Об'єктом дослідження є процеси аналізу та моніторингу криптовалютного ринку, що включають збір, обробку, зберігання та візуалізацію фінансових даних.

Предметом дослідження є методи, моделі, технології та програмні засоби розроблення інтелектуальних веб-застосунків для аналізу криптовалютного ринку, прогнозування тенденцій та підтримки прийняття рішень на основі аналітичних даних.

Методи дослідження включають аналіз наукової та технічної літератури, системний аналіз предметної області, методи проєктування інформаційних систем, технології веб-розробки, методи обробки та візуалізації даних, а також експериментальне тестування програмного забезпечення для оцінювання його функціональності та ефективності.

У роботі передбачається створення інтелектуального веб-застосунку, який надаватиме користувачам можливість відстежувати актуальні курси криптовалют, аналізувати зміни ринкових показників, переглядати статистичні дані у вигляді інтерактивних графіків та отримувати аналітичну інформацію щодо поточного стану ринку. Особлива увага приділятиметься швидкодії системи, безпеці даних, масштабованості архітектури та зручності користувацького інтерфейсу.

Очікується, що впровадження розробленого веб-застосунку сприятиме підвищенню ефективності аналізу криптовалютного ринку, скороченню часу на пошук та обробку інформації, а також покращенню якості прийняття фінансових та інвестиційних рішень.

Бакалаврська робота містить 84 сторінок, 42 рисунка, 2 таблиці, 3 розділи, список використаних джерел із 40 найменувань.

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБЛЕННЯ ІНТЕЛЕКТУАЛЬНИХ ВЕБ-ЗАСТОСУНКІВ ДЛЯ АНАЛІЗУ КРИПТОВАЛЮТНОГО РИНКУ

1.1 Основи функціонування інтелектуальних веб-застосунків

Як руйнівна сила у світовій фінансовій системі, криптовалюти революціонізують способи здійснення транзакцій та перевертають упереджені уявлення про цінність та валюту. Попит на ефективні інструменти для управління, аналізу та моніторингу постійно мінливого ринку криптовалют у режимі реального часу зростає разом із прийняттям та популярністю криптовалют. Щоб задовольнити цей попит, у дослідницькій роботі розроблено трекер криптовалют, використовуючи React JS, потужний інструментарій JavaScript для створення користувацьких інтерфейсів. Щоб робити розумний інвестиційний вибір, ринок криптовалют необхідно постійно контролювати та аналізувати через його волатильність. Інвесторам та трейдерам потрібні надійні та ефективні інструменти, які надають їм доступ до найновіших даних про ціни на криптовалюту, ринкові тенденції та ефективність портфеля. Поточним програмам відстеження біткойнів часто бракує інтерактивних функцій, оновлень даних у режимі реального часу та бажаного користувацького досвіду, необхідних для задоволення мінливих потреб учасників ринку. Отже, існує критична потреба в передовому трекері криптовалют, який поєднує в собі силу даних у режимі реального часу зі зручним та динамічним користувацьким інтерфейсом. React JS з його компонентною архітектурою та декларативним підходом здобув значну популярність серед веб-розробників. Його здатність ефективно обробляти динамічну візуалізацію даних та безперешкодно оновлювати інтерфейси користувача у відповідь на зміни робить його ідеальним вибором для розробки трекера криптовалют, який може забезпечувати аналіз та моніторинг ринку в режимі реального часу. Використовуючи можливості React JS, це дослідження має на меті створити універсальний та зручний трекер криптовалют, який не лише

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

відображає дані в режимі реального часу, але й надає користувачам налаштовувані функції для персоналізованого аналізу та сповіщень. Запропонований трекер криптовалют використовуватиме API, що надаються різними криптовалютними біржами, для отримання ринкових даних у режимі реального часу, включаючи ціни на криптовалюту, обсяги торгів та ринкову капіталізацію [1]. Ці дані будуть оброблятися та відображатися за допомогою компонентів React JS, забезпечуючи безперебійний та адаптивний користувацький досвід. Крім того, трекер інтегруватиме бібліотеки діаграм, щоб користувачі могли візуалізувати історичні цінові тенденції та закономірності, що дозволить їм приймати інвестиційні рішення на основі даних. Крім того, трекер криптовалют включатиме розширені функції, такі як управління портфелем, що дозволить користувачам відстежувати ефективність своїх криптовалютних активів та аналізувати свій загальний інвестиційний портфель. Налаштовувані сповіщення надаватимуть користувачам своєчасні повідомлення про конкретні ринкові умови або цінові коливання, що дозволить їм швидко реагувати на можливості або ризики на ринку крипто валют.

Я вперше почув про біткойн у 2012 році та на той час я дуже мало розумів, як працює базова технологія, що забезпечує роботу біткойна, блокчейн, і, наскільки мені було відомо, біткойн був єдиною криптовалютою. Окрім періодичної перевірки ціни біткойна, я не вивчав криптовалютний простір глибше протягом кількох років.

На початку 2017 року я натрапив на статтю, в якій детально описувалася нова валюта під назвою Ethereum, яка обіцяла повалити біткойн. На відміну від біткойна, Ethereum - це не просто чергова цифрова валюта, Ethereum - це децентралізована платформа, яка запускає додатки поверх блокчейну, відкриваючи простір блокчейну для низки нових варіантів використання. Однією з найважливіших особливостей блокчейну Ethereum є можливість створювати токени. Ці токени обробляються так само, як цифрова валюта, і їх можна використовувати в децентралізованих додатках, з якими вони пов'язані, а також обмінювати на інші цифрові валюти. З відкриттям Ethereum я почав трохи глибше досліджувати простір криптовалют і, на мій приємний подив, там було більше,

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

ніж просто біткойн та Ethereum. Перш ніж я це усвідомив, я став частим відвідувачем різних сабреддів та форумів про криптовалюти.

На відміну від традиційних акцій, криптовалюти не мають відчутної вартості, а ціни коливаються залежно від сприйнятої вартості [2]. Торгівля криптовалютою більше залежить від спекуляцій, ніж від фундаментального аналізу, і в результаті більшість цінових спекуляцій щодо криптовалют відбувається в Інтернеті. Приблизно у травні 2017 року ринок криптовалют пережив вибуховий інтерес. В результаті тисячі нових користувачів зайшли на форуми криптовалют у пошуках наступної найкращої монети для інвестування. Більшість рекомендованих криптовалют або токенів мали великий потенціал для впливу на реальний світ, але на той час не мали продукту, або розробка була за кілька років до завершення. Незважаючи на те, що багато криптовалют не мали продукту або були наполовину завершені, їхня ціна все одно почала зростати. З припливом нових користувачів з'явилося багато недосвідчених трейдерів, які вкладали свої гроші туди, де був консенсус. Ринок криптовалют ще досить молодий, тому існує висока волатильність. Регулювання є рідкісним, а маніпулювати цінами з боку pump and dump або трейдерів з великим банкролом набагато легше, ніж на традиційних фондових ринках. Порівняно з фондовим ринком, для зміни ціни не потрібно багато купівельної чи продажної спроможності, і з появою трейдерів-початківців, які інвестують у криптовалюти виключно тому, що бачать, що їх часто рекомендують, я вважаю, що можна знайти кореляцію між настроями у публікаціях на різних платформах та цінами на згадані криптовалюти.

1.2 Криптовалютний ринок та особливості аналізу криптовалют

Протягом своєї історії гроші дуже змінилися та пройшли багато етапів розвитку: починаючи з самої їхньої появи, коли їхню функцію виконували різного роду предмети: мушлі, дорогоцінні метали чи каміння та банальне проведення бартеру, обміну одного товару на інший, до значного витіснення готівки з обігу та

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

переходу до безконтактних платежів. Сьогодні стрімкий розвиток інформаційних технологій дав поштовх наступному етапу зміни грошей і навіть початку трансформації економічної системи в цілому. Однією з причин таких значних змін є поява електронних грошей та криптовалют [3].

Першою криптовалютою, що з'явилася на світ, став Bitcoin. Поняття криптовалюти зазвичай визначається як віртуальна валюта, яка працює на методах шифрування для генерації грошових одиниць та перевірки їх переказу. Вона не залежить від центрального органу управління, такого як банк [4]. Однією з головних особливостей криптовалюти є її децентралізація завдяки технології блокчейн та відсутність необхідності залучати третіх сторін для проведення транзакцій, оскільки мережа забезпечує рівність усіх учасників (peer-to-peer технологія). Однак криптовалюту не можна класифікувати як класичну категорію грошей через відсутність правового статусу в багатьох країнах [5].

У жовтні 2008 року розробник або група розробників (точно невідомо) під псевдонімом Сатоші Накамото представили так звану Білу книгу, в якій описувалася концепція мережі Bitcoin та аспекти й принципи її функціонування. На початку 2009 року в Інтернеті була опублікована перша версія вихідного коду цієї системи, і згодом була успішно проведена перша транзакція. Переказ коштів у мережі здійснюється за допомогою спеціального програмного забезпечення та принципу асиметричного шифрування. Відкритим ключем є адреса гаманця, згенерований рядок символів і цифр, який не вимагає введення ідентифікаційних даних від користувача, що забезпечує анонімність. Завдяки технології блокчейн також забезпечується прозорість транзакцій: можна відстежувати всі операції з гаманцем, але досить складно встановити зв'язок з власником. Кількість нових генерацій адрес на користувача необмежена. Транзакції в мережі є незворотними [6].

З появою першої криптовалюти незабаром з'явилися й інші проекти. Оскільки вихідний код мережі Bitcoin знаходиться у вільному доступі, деякі з них є майже повною копією так званого форку. Однак деякі криптовалюти кардинально відрізняються за архітектурою. Прикладом такої мережі можна

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

вважати Ethereum, яка займає другий рядок серед криптовалютних компаній за популярністю та капіталізацією. Ethereum - це відкрита програмна платформа на основі технології блокчейн, яка дозволяє створювати децентралізовані додатки, реалізовувати так звані смарт-контракти, що являють собою певний контракт, виконання, модифікація або розірвання якого здійснюється лише за допомогою попередньо визначених програмних засобів. Такі інструкції створюються за допомогою мови програмування Solidity, подібної до JavaScript [7].

Ціноутворення на ринку криптовалют

До появи грошей у тому вигляді, в якому вони існують сьогодні, не виконуючи функції «міри» вартості, вартість товару визначалася в конкретних еквівалентних вартостях іншого товару. Сьогодні купівельна спроможність грошей залежить від специфіки сфери обігу, де вони використовуються та можуть бути обміняні на справжні товари. Принцип попиту та пропозиції визначає ціну конкретної криптовалюти та будь-якого іншого фізичного чи віртуального активу. На ціноутворення впливає потреба людей обмінювати криптовалюту на інші товари. Оскільки віртуальні активи безпосередньо не залежать від економічних процесів чи політики конкретної країни, суттєвим фактором, що впливає на їх ціну, є інтерес людей. Суспільний інтерес до нових, децентралізованих грошей та безконтактної оплати зростає з кожним роком. Недовіра до уряду, політична нестабільність, значна втрата купівельної спроможності фіатних валют на тлі економічних криз – усі ці фактори змушують людей шукати альтернативу та диверсифікувати свої активи, зокрема, інвестуючи в криптовалюту [8]. Привабливими особливостями багатьох криптовалют, зокрема Bitcoin, для пересічних користувачів є проведення транзакцій у мережі без посередників та обмежена кількість одиниць в обігу, що означає майже повну відсутність інфляції. Максимальна кількість монет становить 21 000 000. Перша криптовалюта видобувається шляхом майнінгу – процесу, в якому користувач використовує специфічне обладнання для розв'язання криптографічного рівняння. За це він отримує винагороду у вигляді певної кількості BTC. Кожні чотири роки значення призу падає, а майнінг стає складнішим, що теоретично має стимулювати

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

зростання ціни криптовалют. Безпосереднім фактором ціноутворення є сам майнінг Bitcoin (вартість обладнання та його ефективність, електроенергія, величина винагороди тощо). Отже, за аналогією з дорогоцінними металами та мінералами, перша криптовалюта також характеризується обмеженим ресурсом. Таким чином, Bitcoin є своєрідним «цифровим золотом»[9].

Трохи більше ніж через рік після запуску мережі Bitcoin відбувся перший зафіксований обмін криптовалюти на фізичний товар: це сталося 22 травня 2010 року у Флориді, США. Місцевий програміст заплатив 10 000 BTC за дві піци. Ця подія згодом увійшла в історію під назвою «День піци Bitcoin» [10]. Сьогодні криптовалюти вже приймають тисячі закладів та сервісів по всьому світу. Їх кількість зростає з кожним днем: ресторани, кафе, заправки, магазини електроніки та онлайн-покупки ігор, хостинг-сервіси, поповнення мобільних телефонів тощо. У світі вже існують тисячі пунктів конвертації криптовалюти та терміналів, які можна використовувати для придбання криптовалюти за готівку. Зокрема, багато з них вже розташовані на території України. Також розвиваються системи платіжних карток, які працюють як звичайні банківські картки, конвертуючи з рахунку користувача суму в обраній валюті, еквівалентну вартості платежу, у фіатні гроші, дозволяючи розплачуватися в будь-якому магазині, де приймаються звичайні банківські картки [11]. Початок формування ринку криптовалют розпочався з появи першої криптовалюти – Bitcoin. Хоча спочатку було важко оцінити його вартість через тодішню низьку поширеність, ціна не перевищувала кількох центів. Біткойн зазнав значних змін у ціні через рік після запуску мережі, з появою першої криптовалютної біржі – Mt.Gox. А вже на початку 2011 року вартість першої криптовалюти перетнула позначку в 1 долар США.

З часом біткойн набув широкого поширення, і з 2012 року все більше роздрібних торговців почали впроваджувати платежі в криптовалюті як альтернативний спосіб оплати. Наприкінці 2013 року ціна 1 BTC вперше перетнула позначку в 1000 доларів США завдяки популяризації біткойна в Китаї, відкриттю нових бірж та початку майнінгу криптовалюти в промислових

					БР.ІІІ – 67.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

масштабах.

Криптовалютний ринок пережив багато злетів і падінь у своїй історії. На початку криптовалютна індустрія зазнавала численних атак, які суттєво вплинули на ціноутворення. Так, у 2014 році, через відсутність належної системи безпеки та належного управління, найпопулярніша на той час біржа Mt.Gox оголосила про банкрутство через крадіжку близько 850 000 BTC. Це найзначніша крадіжка криптовалюти в історії. Користувачі досі намагаються повернути свої кошти. Після цих подій ціна за 1 біткойн впала до 580\$ і продовжила своє тривале падіння майже два роки до 315\$. На початку 2017 року ціна біткойна повернулася до 1000\$. Поява та розвиток нових проєктів, таких як Litecoin, Ripple, Ethereum та інших, бум так званої первинної пропозиції монет (англ. ICO - initial coin offering) - форми залучення початкових інвестицій у проєкт, подібної до первинної публічної пропозиції акцій на фондовому ринку, сприяли черговому стрімкому зростанню капіталізації криптовалютного ринку. Протягом багатьох років інтерес до криптовалют зріс не лише серед пересічних користувачів, а й серед фінансових установ у всьому світі. Значною подією стало введення ф'ючерсних контрактів на Bitcoin на Чиказькій товарній біржі, що дозволило залучити на ринок криптовалют трейдерів з традиційних фондових та Forex ринків та значно збільшити обсяги торгів. У грудні 2017 року капіталізація ринку криптовалют досягла піку, а ціна за 1 Bitcoin досягла майже 20 000 доларів США - свого історичного максимуму [12]. Сьогодні все більше країн намагаються законодавчо врегулювати та легалізувати роботу з криптовалютами та їх правовий статус, а фінансові установи з різних куточків світу розробляють та впроваджують системи, які б дозволили пересічним користувачам інвестувати в криптовалюти та використовувати їх. Поява сотень криптовалютних бірж, маржинальної торгівлі, ф'ючерсів, деривативів та інших торгових інструментів свідчить про зростання суспільного інтересу до криптовалют. В останні роки спостерігається кореляція з фондовими індексами цінних паперів, що означає прихід інвесторів та трейдерів з традиційних ринків. Капіталізація ринку криптовалют станом на кінець квітня 2020 року становить близько 265 мільярдів доларів США [13]. З'являються

					БР.ІІІ – 67.00.00.000 ІІЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

проекти державних цифрових валют, а в Китаї навіть розпочалося тестування національної криптовалюти, що свідчить про неминучу трансформацію світової економічної системи з використанням цифрових грошей.

Арбітраж. Особливості арбітражу на ринку криптовалют

Усунення різниці в ціні з отриманням прибутку шляхом виконання певної послідовності дій, що починаються та закінчуються одним і тим самим активом, що, якщо їх виконати, призводить до його збільшення в кінці цієї послідовності, називається арбітражем. Наприклад, якщо ми купимо акцію А за 10\$ на біржі Х та продамо її на біржі Y за 10,30\$, ми отримаємо 0,30\$ прибутку, без урахування комісій [14].

Концепція арбітражу не нова, і різні типи цінових розбіжностей існують на всіх фінансових ринках, включаючи фондові біржі та валютні ринки. Цей принцип торгівлі вже багато років використовується банками та іншими фінансовими установами по всьому світу. Наприклад, валютний арбітраж передбачає купівлю та продаж валютних пар у різних брокерів, щоб скористатися різницею в цінах для отримання прибутку [15]. Ринок криптовалют не є винятком з арбітражу, де ці розбіжності ще більш значні порівняно з традиційними ринками. Сьогодні існує близько 4000 криптовалют. Побічним ефектом цього стало аналогічне збільшення кількості криптовалютних бірж. Binance, Coinbase, Kraken, Bitfinex, Bittrex – це лише деякі з багатьох популярних бізнесів, що існують. Низький рівень державного регулювання, децентралізація, велика кількість спекулянтів, які прагнуть заробити гроші, висока волатильність – усі ці фактори створюють складні умови для існування єдиної ціни та роблять арбітраж на ринку криптовалют особливо привабливим [16].

Арбітраж на ринку криптовалют майже повністю працює за тими ж принципами, що й на традиційних ринках, але з іншим активом. Значні розбіжності в цінах між біржами, у великих масштабах, часто виникають на тлі політичної нестабільності. Так, у серпні 2019 року в Аргентині на тлі різкого падіння національної валюти ціна за 1 біткойн була на 4% вищою, ніж на міжнародних платформах у перерахунку на долари США. Але навіть якщо

					БР.ІІІ – 67.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

прибрати екстремальні економічні та політичні події, розбіжності в цінах між біржами виникають частіше, ніж здається. Існує багато причин арбітражних ситуацій на ринку криптовалют: від різкого обвалу або зростання цін на одній з бірж, реакції ціни на новини, до так званої «помилки товстого пальця», коли користувач випадково ввів неправильні дані з клавіатури або помилково клацнув мишкою, що призвело до виконання ордера за небажаною ціною або обсягом [17].

Операції, пов'язані з торгівлею криптовалютним шахрайством, проводяться на унікальних платформах - криптовалютних біржах (або криптобіржах). Криптобіржа працює аналогічно валютній або фондовій біржі [18]. В обговореннях вводиться поняття валютної пари - пари, що складається з двох валют, які торгуються одна проти одної. Першу валюту зазвичай називають базовою валютою, а другу - квотою [19]. Наприклад, якщо ви хочете купити BTC за USD, вам потрібно знайти пару BTC/USD. Окрім валютних пар криптовалюта/фіатні гроші, на більшості бірж також популярні пари криптовалюта/криптовалюта, які часто навіть більш затребувані, ніж пари фіатних грошей через певні особливості цього ринку. Всі запити користувачів потрапляють у так званий "біржовий стакан" - список рекомендацій щодо купівлі та продажу певної криптовалюти, організований за рівнем цін. У такому списку зазначено кількість певної криптовалюти, яка продається або пропонується до купівлі за кожною ціною, формуючи глибину ринку [20]. Критичною особливістю криптовалютних бірж є те, що користувачі можуть легко та швидко отримати доступ до переважної більшості з них. З мінімальними витратами перевести придбану криптовалюту на свій гаманець для зберігання, тим самим не залежачи від третіх осіб чи іншої біржі, чим не можуть похвалитися брокери та валютні біржі чи фондові біржі через високі комісії, відсутність децентралізації торгуваних активів, фіксований графік роботи та тривалий час проведення транзакцій. Методи арбітражу, які можна застосовувати на ринку криптовалют, представлені в пунктах нижче.

Просторовий арбітраж

					БР.ІІІ – 67.00.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

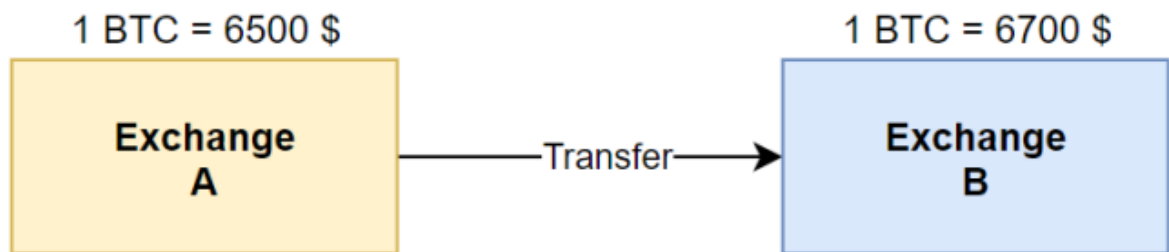


Рисунок 1.1 - Приклад просторового арбітражу

Ця техніка передбачає отримання прибутку з різниці в цінах купівлі та продажу певної криптовалюти, що торгується на двох різних біржах. Наприклад, якщо біржа А може запропонувати BTC за 6500 доларів, а ціна продажу В досягає 6700 доларів, трейдер може скористатися цими 200 доларами, купивши на біржі А та продавши на біржі В – ефективно переміщуючи кошти з однієї платформи на іншу. Враховуючи ступінь децентралізації в криптовалютах, такі розбіжності можуть виникати частіше, ніж ви думаєте. Не забувайте про комісію за проведення торгових операцій та транзакційні витрати на виведення криптовалюти з бірж, також важливий час переказу криптовалюти. Чим він довший, тим ризикованіша можливість отримання прибутку [21]. Приклад виконання просторового арбітражу показано на рис. 1.1.

Внутрішньобіржовий арбітраж

Особливістю цього методу є те, що він відбувається лише в межах однієї біржі. Його принцип полягає у проведенні послідовності торгових операцій, що починаються та закінчуються однією й тією ж валютою. Наприклад, на трьох ринках, що торгуються на біржі, з різними квотами, такі курси: BTC/USD = 6726,94, ETH/USD = 141,58 та ETH/BTC = 0,0214. Продаємо 1 BTC за USD, отримуємо 6726,94 USD, за отримані USD купуємо 47,5133 ETH та продаємо ETH за BTC - отримуємо 1,0167 BTC, прибуток становить 0,0167 BTC. Цей тип арбітражу також називається «Трикутним» (рис. 1.2) [22].

Варто зазначити, що цей метод арбітражу вимагає швидкого збору та аналізу даних і, відповідно, виконання всіх необхідних торгових операцій. Також важливо враховувати торгові комісії та типи виконання ордерів.

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

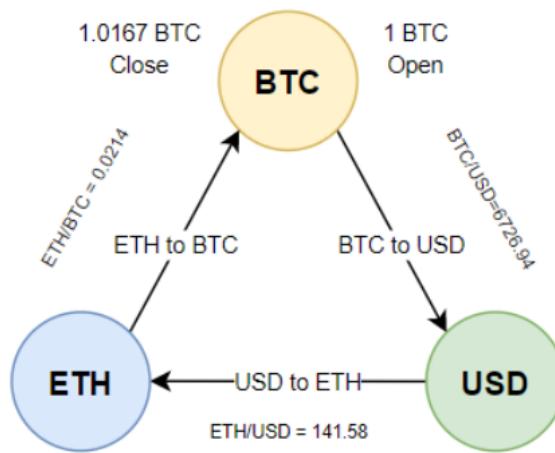


Рисунок 1.2 - Приклад внутрішньобіржового арбітражу

Для пошуку таких арбітражних ситуацій часто використовуються різні алгоритми, що дозволяють знайти найкоротший шлях у зваженому графі від однієї вершини до всіх інших.

Статистичний арбітраж

Цей метод арбітражу вважається високотехнологічною стратегією, зазвичай передбачає математичне моделювання. Цей тип арбітражу базується на підтримці історичного співвідношення між вартістю ідентичних або подібних активів на різних торгових платформах. Цей метод не вимагає переказу криптовалюти з однієї біржі на іншу, тим самим скорочуючи час виконання арбітражних операцій у кілька разів, і працює наступним чином: коли різниця цін між двома біржами вища за їх історичне співвідношення, одночасно будуть виконуватися дві функції: продаж на першій біржі та купівля на другій, і навпаки - коли різниця цін нижча за історичне співвідношення. Коли співвідношення цін наближається до своїх типових значень, позиції закриваються. При використанні цього методу ринкові ризики практично відсутні, оскільки агрегатна функція складається з різноспрямованих частин на одному активі, що захищає від непередбачуваних коливань ринку. Єдиним ризиком такої стратегії є зміна історичного значення співвідношення між інструментами [23].

Слід зазначити, що арбітраж на ринку криптовалют – це досить ризикований вид заробітку, який залежить від фінансової грамотності, навичок

користувачів у роботі з криптовалютою, досвіду, швидкості дій, а часто навіть банальної удачі [24].

Автоматизована торгівля. Огляд існуючих торгових систем

Наприкінці минулого століття, з розвитком інформаційних технологій, сфера торгівлі на фондовому та валютному ринках також зазнала комп'ютеризації. В результаті майже всі торговельні операції здійснюються за допомогою електронних засобів: персональних комп'ютерів, планшетів, смартфонів тощо. Сьогодні користувач може виконувати всі торговельні операції за допомогою веб-інтерфейсу брокера або біржі, використовувати спеціальні торговельні термінали з широким функціоналом або розробляти своє програмне забезпечення, яке обслуговуватиме заданий користувачем алгоритм дій без його участі.

На початку ХХІ століття автоматизована торгівля, або так звана «роботизована торгівля», почала стрімко розвиватися. Трейдеру більше не потрібно стежити за змінами ціни та сидіти на графіках цілими днями, оскільки реалізувати свою стратегію можна за допомогою унікальних торгових систем. Такі автоматизовані системи збирають дані з бірж. Потім, згідно із заздалегідь заданими інструкціями, вони виконують певний алгоритм, який обслуговує всю послідовність дій замість користувача, згідно із заздалегідь заданими інструкціями [25].

Перевагами такого підходу до торгівлі є можливість отримувати та аналізувати величезну кількість даних і перевіряти багато ринкових умов за лічені секунди, що не під силу людині. Водночас також зменшується ризик ручних помилок під час виконання торгових операцій. Найголовніше, що це виключає емоційний вплив на процес торгівлі, який часто призводить до збитків. Таким чином, алгоритмічна торгівля забезпечує систематичний та незмінний підхід до активної торгівлі порівняно з людиною. Сьогодні багато автоматизованих торгових систем дозволяють користувачам створювати власні алгоритми торгівлі з метою отримання прибутку. Такі процедури охоплюють виконання певної послідовності дій залежно від результатів розрахунків обсягів за допомогою складних математичних формул та початкове швидке отримання та обробку

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

необхідної інформації. Зокрема, використання такої системи буде ефективним у пошуку та виконанні арбітражних можливостей на ринку криптовалют. Доступ до необхідних даних та можливість автоматизації торгових операцій забезпечуються криптовалютами біржами за допомогою API (Application Programming Interface) - інтерфейсу прикладного програмування, який являє собою набір методів роботи з бізнесом [26]. Кілька популярних торгових систем для ринку криптовалют представлені в наступних параграфах.

Сервіс пошуку арбітражних ситуацій на ринку криптовалют. Допомагає початківцям та досвідченим користувачам знаходити найвигідніші та найнадійніші арбітражні можливості. Система збирає актуальні та необхідні дані пошуку онлайн (ціни, обсяги торгів, технічні характеристики валют тощо) з найпопулярніших криптобірж, аналізує їх та пропонує користувачам найвигідніші варіанти арбітражних операцій. Результат роботи системи відображається у вигляді таблиці, в якій користувач може встановити свої параметри пошуку - прибуток, обсяг транзакцій, час транзакцій, ступінь надійності операції, а також скласти список бірж, де потрібно шукати транзакції. Сервіс забезпечує пошук просторових та внутрішньобіржових арбітражних ситуацій. Система реалізована як веб-сервіс, що пропонує кросплатформну продуктивність. Є можливість надсилати сповіщення про виявлення арбітражних можливостей. Автоматичне проведення арбітражних ситуацій та можливість впровадження власних торгових алгоритмів відсутня [27]. Головна сторінка сервісу показана на рис. 1.3.

The screenshot shows the Arby.Trade web interface. At the top, there are search filters: 'Profit not less than' (10%), 'Volume not less' (10), 'Price taken into account' (BTC, Buy at Ask, sell at Bid), 'Transaction time' (30 min), and 'Reliability' (5 stars). Below these are checkboxes for 'Take into account the Commission' (checked for the transaction and for transfer). A green 'Apply' button is on the right. The main table displays search results with columns: Pair, Profit, Time, Market, Bid, Ask, Volume, BTC, 20M, 24h, and Reliability. The table lists five rows of arbitrage opportunities, alternating between Bitrex/Binance and Hitbtc/Livecoin pairs.

Pair	Profit	Time	Market	Bid	Ask	Volume	BTC	20M	24h	Reliability
Bitcoin	5.01%	~5 min	Bitrex Binance	0.08040030 0.08457830	0.08457830 0.08040030	15000	-2.87%	+3.09%	+2.88%	★★★★○
Bitcoin	4.50%	~20 min	Hitbtc Livecoin	0.08040030 0.08457830	0.08158946 0.08574736	0.50	-20.11%	+30.75%	+20.11%	★★★★○
Bitcoin	5.01%	~5 min	Bitrex Binance	0.08040030 0.08457830	0.08457830 0.08040030	15000	-2.87%	+3.09%	+2.88%	★★★★○
Bitcoin	4.50%	~20 min	Hitbtc Livecoin	0.08040030 0.08457830	0.08158946 0.08574736	0.50	-20.11%	+30.75%	+20.11%	★★★★○
Bitcoin	5.01%	~5 min	Bitrex Binance	0.08040030 0.08457830	0.08457830 0.08040030	15000	-2.87%	+3.09%	+2.88%	★★★★○

Рисунок 1.3 - Інтерфейс Arby.Trade

Blackbird

Торгова система написана на C++, яка виявляє арбітражні ситуації на більш ніж п'яти криптовалютних біржах та автоматично їх виконує. Основне завдання системи - знаходити та усувати цінові різниці на ринку шляхом продажу криптовалют на одній біржі та купівлі на іншій. Використовується стратегія низького ринкового ризику, подібна до статистичного арбітражу, яка передбачає торговельну діяльність, що включає паралельні купівлі та продажі на двох різних біржах без необхідності переказу криптовалюти з бізнесу на біржу. Є мінімальний графічний інтерфейс. Основна кількість інструкцій від користувача надходить через командний рядок [28].

Gekko

Безкоштовна торгова система з відкритим вихідним кодом, яка підтримує кілька популярних криптовалютних бірж. Вона написана на JavaScript та працює на Node.js. Головним завданням плану є автоматизація ваших торгових стратегій. Gekko надає можливість створювати торгові стратегії, використовуючи популярні індикатори технічного аналізу або використовуючи готові. Для перевірки та визначення оптимальної стратегії можливо моделювати торгівлю, використовуючи історичні ринкові дані. Також існує система плагінів, яка дозволяє додавати свій функціонал [29]. Вигляд інтерфейсу системи показано на рис. 1.4.



Рисунок 1.4 - Інтерфейс Gekko. Перевірка вашої стратегії на основі історичних даних

					БР.ІП – 67.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

Zenbot

Торговий робот написаний на JavaScript та працює на платформі Node.js з використанням бази даних MongoDB. Реалізована підтримка широкого спектру функціональності призначена для використання серед більш ніж 12 криптовалютних бірж. Система надає можливість створювати власні торгові стратегії на основі технічного аналізу. Архітектура підтримки плагінів пропонує можливість самостійно додавати нові обговорення та розробляти новий функціонал і процеси. Zenbot надає можливість створювати ордери на купівлю або продаж з багатьма конфігураціями, що дозволяє системі працювати як торговий термінал. Є можливість проводити торговельні симуляції для тестування стратегій на основі історичних даних. Система працює відповідно до інструкцій користувача через термінал [30].

Більшість існуючих систем є платними за використання, з необхідністю щомісячної абонентської плати. Деякі роботи дозволяють розробляти, тестувати та впроваджувати власні торгові стратегії. Однак, мало які системи можуть виявляти або виконувати арбітражні можливості на ринку та навіть збирати всі необхідні дані для успішного здійснення таких операцій. Також відсутня технічна реалізація автоматичного переказу криптовалют з біржі на біржу в жодній із систем, що виключає можливість проведення класичного просторового арбітражу. Недоліками розробок також є невелика кількість підтримуваних бірж та фіксований список пар криптовалют, що зменшує кількість знайдених арбітражних можливостей. Критичним для систем пошуку арбітражу є надсилання сповіщень про такі ситуації та дистанційне керування системою.

На основі проведеного порівняльного аналізу торгових систем для ринку криптовалют можна зробити висновок, що поточна система пошуку та виконання арбітражних транзакцій буде кросплатформною системою з підтримкою широкого кола бірж та всіх можливих пар криптовалют, з можливістю віддаленого управління, повідомленнями користувачів про виявлені ситуації, з можливістю їх налаштування, а головне – з компонентом для автоматизації виконання послідовності операцій, що дозволить повністю зібрати інформацію

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

для успішного арбітражу. Крім того, реалізація власних торгових стратегій не є обов'язковою, оскільки для проектованої системи обрано метод просторового арбітражу на ринку криптовалют.

Таблиця 1.1

Порівняння можливостей системи

Функціонал / Назва системи	Arby.Trade	Gekko	Blackbird	Zenbot
Збір даних про ціни та обсяги (Collecting data on prices and volumes)	Так	Так	Так	Так
Збір технічних даних про криптовалюту (Collecting technical data on cryptocurrencies)	Так	Ні	Ні	Ні
Пошук арбітражних ситуацій (Search for arbitration situations)	Так	Ні	Так	Так
Можливість здійснення арбітражних угод (Ability to perform arbitrage transactions)	Ні	Ні	Так	Так
Можливість реалізації власних алгоритмів торгівлі (Ability to implement your trading algorithms)	Ні	Так	Ні	Так
Кроссплатформеність (Cross-platform)	Так	Так	Так	Так
Push-сповіщення (Push notifications)	Так	Через плагіни	Ні	Через плагіни

На основі проведеного порівняльного аналізу торгових систем для ринку криптовалют можна зробити висновок, що поточна система пошуку та виконання арбітражних транзакцій буде кроссплатформною системою з підтримкою широкого кола бірж та всіх можливих пар криптовалют, з можливістю віддаленого управління, повідомленнями користувачів про виявлені ситуації, з можливістю їх налаштування, а головне – з компонентом для автоматизації виконання послідовності операцій, що дозволить повністю зібрати інформацію для успішного арбітражу. Крім того, реалізація власних торгових стратегій не є обов'язковою, оскільки для проектованої системи обрано метод просторового арбітражу на ринку криптовалют.

У підрозділі було розглянуто та проаналізовано інформацію про

проектовану систему, що є першим кроком для її майбутнього розвитку. Розкрито поняття криптовалюти, принципи та аспекти її функціонування. Розглянуто фактори, що впливають на ціноутворення, та роль криптовалют у повсякденному житті. Відображено особливості проведення арбітражу на ринку криптовалют, описано популярні методи. Проаналізовано існуючі торговельні системи, розглянуто їх можливості та функціональність, а також виділено сильні та слабкі сторони. На основі аналізу визначено необхідний функціонал та проблеми, які повинна вирішувати розроблена інформаційна система.

1.3 Системний аналіз об'єкта дослідження

Для початку розробки інформаційної системи необхідно проаналізувати об'єкт дослідження, виявити та класифікувати кілька актуальних проблем, а також знайти рішення. Одним із методів вирішення таких проблем є побудова дерева цілей. Дерево цілей – це ієрархічна діаграма, що складається з взаємозв'язків та підпорядкованих цілей системи, які відображають розподіл цілей системи на підцілі та завдання або окремі дії. Цей метод базується на декомпозиції та є універсальним, і є одним з основних у системному аналізі. Декомпозиція – це спосіб поділу системи на її компоненти. Метод дерева цілей використовується для встановлення взаємозв'язку між головною метою та способами її досягнення, які представлені в конкретних діях або завданнях. Основний принцип побудови дерева цілей полягає у спрощенні складного процесу до його елементарних компонентів. Для дотримання цього правила дотримуйтесь наступного підходу:

- мета, розташована на найвищому рівні, є основою та своєрідною точкою відліку для розділення цілей, розташованих на найнижчому рівні;

Цілі, що знаходяться на найнижчому рівні, є засобами реалізації завдання на найвищому рівні та представлені таким чином, що їх об'єднання забезпечує виконання головної мети [31].

Головною метою, що визначається методом побудови дерева цілей, є створення інформаційної системи для пошуку арбітражних ситуацій на ринку

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

криптовалют. Ця проблема може бути вирішена лише після виконання всіх її підцілей. Тому побудоване дерево має чотири рівні: основна мета, підцілі, завдання, які необхідно виконати, та критерії оцінки якості виконання. В результаті створення дерева цілей були сформовані такі підцілі:

- Збір та обробка інформації;
- Застосування методів системного аналізу;
- Розробка програмного рішення для системи.

Розглянемо першу підціль – «Збір та обробка інформації». На цьому етапі проводиться аналіз вимог та взаємозв'язків між компонентами системи, здійснюється аналітичний огляд літератури та інших джерел, що стосуються предметної області, та прогнозований план напрямку. На основі аналізу було визначено кілька таких завдань:

- Дослідження методів арбітражу. Для створення інформаційної системи пошуку арбітражних ситуацій, коректності, повноти та точності розрахунків у такій системі необхідно ознайомитися з самим поняттям арбітражу, можливими методами його проведення та його особливостями на ринку криптовалют;

- Збір даних щодо роботи з біржами. Це завдання передбачає ознайомлення з принципом роботи на криптовалютних біржах, аналіз технічної документації API методів обміну для успішного створення відповідних модулів, якісної типізації та впровадження в систему;

- Аналіз існуючих рішень. Для створення якісного та багатофункціонального продукту необхідно проаналізувати існуючі програмні рішення, які вирішують подібні проблеми або близькі до проектованої системи. Виділити їхні сильні та слабкі сторони, функціональність тощо.

Створіть правильну функціональну модель проектованої системи, використовуйте виявлені переваги аналізованих систем, вирішіть виявлені проблеми та представте їх рішення у вашій системі.

Для першої підцілі ми обрали такі параметри оцінки якості, як релевантність та корисність, які є основними властивостями для збору та аналізу

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

інформації. Друга підціль – «Застосування методів системного аналізу». Перед створенням інформаційної системи необхідно вибрати альтернативні системи, визначити серед них оптимальну інформаційну систему та спроектувати її центральні процеси, функції та завдання для вирішення. Результатом цієї підцільі є визначення типу інформаційної системи, що проектується, та серії діаграм для пояснення процесів, що відбуваються. Ця підціль передбачає вирішення кількох таких завдань:

- Це визначення типу інформаційної системи. Це завдання передбачає застосування методу аналізу ієрархій. Шляхом парних порівнянь значень оцінок якості серед альтернативних систем вибирається найоптимальніший варіант системи;
- Побудова контекстної діаграми. Для відображення основного процесу або функції системи та її деталей необхідно створити контекстну діаграму DFD або IDEF0;
- Побудова діаграм декомпозиції процесів. На основі діаграми контексту потрібно декомпонувати її для створення наступних діаграм, що визначають процеси нижчого рівня та деталі їх реалізації.

Як критерії для оцінки цієї підцільі було обрано оптимальність та адаптивність.

Остання підціль для досягнення головної мети – «Розробка програмного рішення для системи», яка включає виконання наступних дій та завдань:

- Вибір інструментів реалізації програмного забезпечення для системи. На основі прототипу системи необхідно вибрати середовище розробки та інструменти реалізації програмного забезпечення, які найкраще підходять для її розробки, щоб задовольнити вимоги та завдання, які повинна виконувати ця система;
- Створення програмного компонента системи. Це завдання полягає в написанні коду необхідних модулів для повноцінного функціонування системи з використанням заздалегідь вибраних інструментів реалізації програмного забезпечення;

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

- Виконання тестів та усунення несправностей. Це завдання включає перевірку та налагодження виявлених проблем у процесі розробки програмного забезпечення;

- Підтримка оновлень системи. Необхідно забезпечити інформаційну систему підтримкою програмного продукту та його оновленням, що вкрай необхідно у випадках змін у роботі бірж або появи нових методів арбітражу, ідей щодо розширення функціональності для забезпечення актуальності системи тощо. Також не варто забувати про можливість виникнення аномалій під час роботи системи, які необхідно виявляти та усувати.

Обраними критеріями для оцінки якості виконання цієї підцілі є масштабованість та

продуктивність, оскільки ці властивості відіграють вирішальну роль під час функціонування системи та для її подальшого розвитку в майбутньому. Для визначення типу інформаційної системи обрано метод аналізу ієрархій. Цей метод використовується для вирішення багатьох практичних задач на різних етапах проектування системи. Суть процесу полягає у використанні парних порівнянь критеріїв оцінки якості виконання підцілей для вибору пріоритетних рішень.

Як альтернативи було обрано три типи інформаційних систем:

- Інформація та управління (A1);
- Інформаційно-пошукова система (A2);
- Інформаційно-довідкова система (A3).

Визначення найоптимальнішого типу інформаційної системи, яку потрібно проектувати, здійснюється за допомогою попередньо вибраних шести критеріїв якості:

- Релевантність (C1);
- Корисність (C2);
- Оптимальність (C3);
- Адаптивність (C4);
- Масштабованість (C5);
- Продуктивність (C6).

Оскільки дерево цілей системи вже побудовано, тепер необхідно створити матрицю парних порівнянь критеріїв та визначити значення власних значень (HF) та власних векторів (HV), що використовуються в наступному аналізі. Розрахунок важливості пріоритетів буде визначатися шляхом знаходження взаємозв'язків критеріїв з використанням балів для їх оцінки в діапазоні від 1 до 9, де 1 – це критерій рівноваги, а 9 – абсолютна перевага одного стандарту над іншим.

1.4 Висновок по розділу

У першому розділі було розглянуто теоретичні основи розроблення інтелектуальних веб-застосунків для аналізу криптовалютного ринку. Проаналізовано принципи функціонування інтелектуальних веб-систем, особливості криптовалютного ринку та ключові фактори, що впливають на зміну вартості цифрових активів. Проведено системний аналіз об'єкта дослідження, визначено основні функціональні компоненти та вимоги до майбутнього програмного продукту. Отримані результати створюють теоретичне підґрунтя для проєктування та реалізації веб-застосунку, здатного забезпечувати аналіз і візуалізацію ринкових даних.

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОГО ВЕБ-ЗАСТОСУНКУ

2.1 Функціональні вимоги до веб-застосунку

Наступні вимоги будуть відповідати методології KDD (рисунок 2.1). KDD розшифровується як «виявлення знань у базах даних» і стосується процесу пошуку знань у даних, який є важливим для проекту аналізу даних. Життєвий цикл KDD пояснюється нижче:

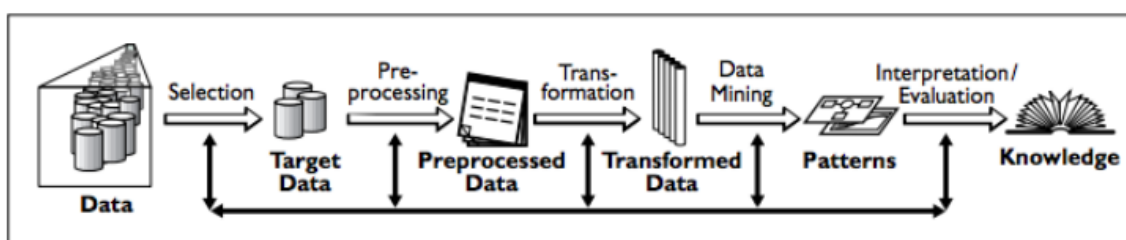


Рисунок 2.1 - Методологія KDD

Вибір

Цей крок включає визначення відповідного набору даних для проєкту. Цей проєкт буде зосереджений на коментарях та публікаціях з reddit.com та twitter.com. API Twitter та reddit пропонують багато кінцевих точок, більшість з яких не стосуються обсягу цього проєкту, тому цей крок також включатиме вибір відповідної інформації.

Попередня обробка

Крок попередньої обробки очищує та стирає набір даних, видаляючи будь-які небажані знаки пунктуації, пробіли та спеціальні символи, які можуть заважати аналізу [32]. Цей крок важливий, оскільки набір даних, який використовується в цьому проєкті, – це коментарі та публікації з онлайн-платформ, вони, безумовно, міститимуть емодзі та інші спеціальні символи, які можуть вплинути на результат. Оскільки цей проєкт збирає дані з трьох різних

джерел, деякі поля, такі як дата, можуть бути представлені по-різному, крок попередньої обробки також стандартизує набори даних у єдиний формат.

Трансформація

Цей крок у KDD передбачає генерування кращих даних з нашого очищеного набору даних. Використання методів перетворення або зменшення розмірності зменшить кількість змінних, що розглядаються, або знайде інваріантні представлення для даних. Цей крок також класифікує дані на позитивні нейтральні або негативні настрої за допомогою методів аналізу настроїв.

Інтелектуальний аналіз даних

Аналіз даних є одним з найважливіших етапів проекту, оскільки саме тут ми застосовуватимемо алгоритми до набору даних для пошуку тенденцій та відповідної інформації. На цьому етапі будуть проаналізовані як історичні ціни, так і коментарі до криптовалют, щоб знайти подібності або тенденції між настроями в коментарях та рухом цін на згадані криптовалюти. На цьому етапі будуть використані машинне навчання та статистичні алгоритми.

Інтерпретація

Останній крок методології KDD включає інтерпретацію результатів отриманих даних та документування висновків. Результати також будуть відображені за допомогою візуалізацій, щоб користувачі могли краще зрозуміти отримані дані.

Функціональні вимоги

Функціональні вимоги до системи, які будуть потрібні для завершення проекту:

1. Система збиратиме дані та зберігатиме їх у форматі CSV.
2. Система очищуватиме та стандартизуватиме дані.
3. Система класифікуватиме дані за допомогою текстового аналізу.
4. Система проаналізує дані.
5. Користувач зможе увійти за допомогою облікового запису Google.
6. Користувач зможе шукати криптовалюту

7. Користувач зможе переглядати дані для певної криптовалюти.

Користувач зможе переглянути N найкращих криптовалют.

Діаграма варіантів використання (рисунок 2.2)

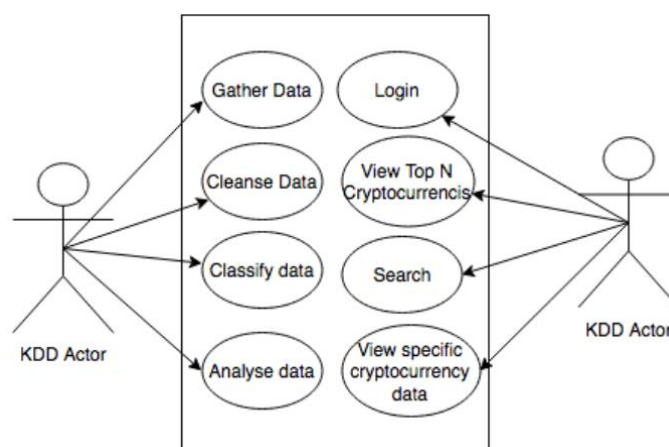


Рисунок 2.2 - Діаграма варіантів використання

Вимога 1: Збір даних (рисунок 2.3)

Опис і пріоритет

Ця вимога є найважливішою для системи. Це перший крок і один з найважливіших, оскільки дані необхідні для проведення аналізу.

Область застосування варіантів використання

Визначте, яка інформація з платформ є релевантною для цілей цього аналізу, та зберіть відповідні дані.

Опис

Цей варіант використання описує збір даних з різних платформ для обговорення криптовалют шляхом парсингу або вилучення з API.

Діаграма варіантів використання

Опис потоку Передумова

Система не має даних для активації

Цей варіант використання починається, коли <Actor> запускає скрипти Python, які або витягують дані з API, або збирають дані з сайтів без API.

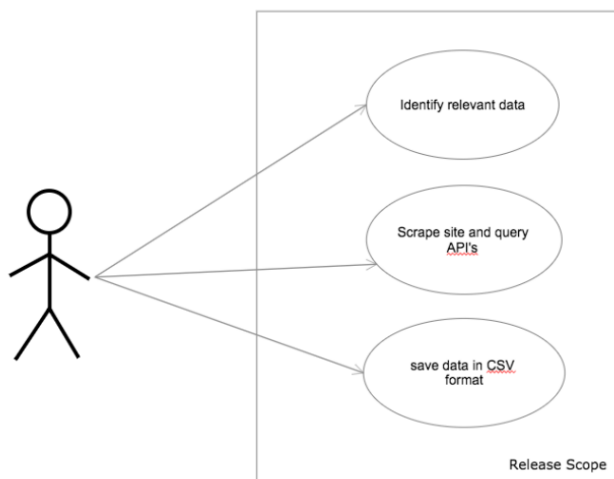


Рисунок 2.3 - Збір даних

Основний потік

1. <Actor> визначає відповідні дані
2. <Actor> запускає скрипт
3. Скрипт витягує дані
4. Дані зберігаються у файлі CSV

Умова публікації

Система імпортує файл до RStudio

Вимога 2: Очищення та стандартизація даних (рисунок 2.4).

Опис і пріоритет

Дані будуть очищені та стандартизовані до узгодженого формату шляхом видалення розділових знаків та пробілів. Цей крок необхідний для забезпечення максимальної точності нашого аналізу шляхом видалення будь-яких небажаних даних.

Область застосування варіантів використання

Метою цього випадку використання є очищення коментарів шляхом видалення будь-яких небажаних символів та стандартизація даних у єдиний формат.

Опис

Цей варіант використання описує очищення та стандартизацію коментарів.

Діаграма варіантів використання

					БР.ІП – 67.00.00.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

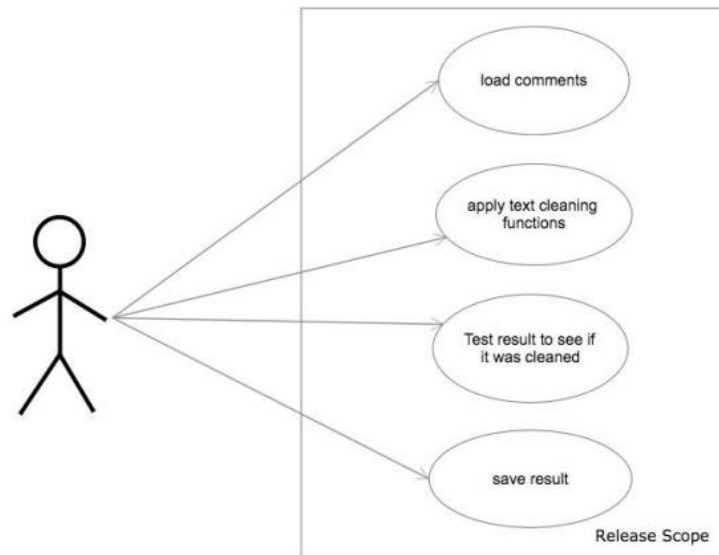


Рисунок 2.4 - Очищення та стандартизація

Опис потоку Передумова

Коментарі містять зайві пробіли та пунктуацію. Активація

Цей варіант використання починається, коли <Actor> імпортує набір даних до основного потоку R.

1. <Actor> завантажує набір даних у RStudio.
2. <Actor> застосовує функції очищення даних та стандартизації.
3. <Actor> перевіряє результат, щоб побачити, чи він був оброблений правильно.
4. Коментарі очищені та стандартизовані.
5. Результат зберігається у форматі CSV.

Вимога 3: Аналіз настроїв (рисунок 2.5).

Опис і пріоритет

Коментарі будуть класифіковані за допомогою аналізу настроїв. Ця вимога є необхідною, оскільки проєкт зосереджений на аналізі настроїв даних.

Область застосування варіантів використання

Мета цього випадку використання полягає в тому, щоб класифікувати коментарі на позитивні, негативні або нейтральні.

Опис

Цей варіант використання описує процес виконання аналізу настроїв у

коментарях.

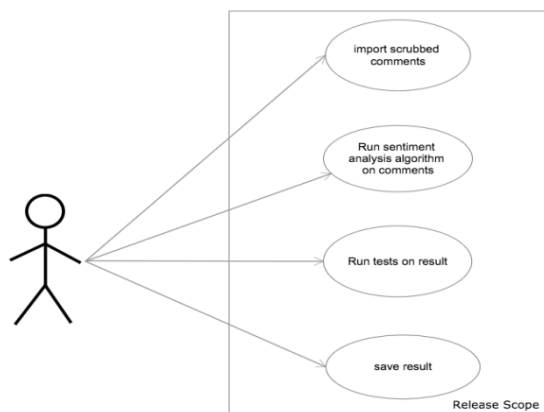


Рисунок 2.5 - Аналіз настроїв

Опис потоку Передумова

Коментарі було видалено з небажаних даних, які могли б заважати процесу аналізу.

Активація

Цей варіант використання починається, коли <Actor> імпортує оброблені дані до основного потоку RStudio.

<Actor> завантажує оброблені дані в R.

<Actor> запускає алгоритм аналізу настроїв для коментарів.

<Актор> переглядає результати.

<Actor> зберігає результати.

Система готова аналізувати коментарі та ціни.

Вимога 4: Аналіз даних (рисунок 2.6).

Опис і пріоритет

Ця вимога використовуватиме статистичні алгоритми та машинне навчання, щоб знайти кореляцію між настроями коментарів користувачів та цінами на криптовалюту.

Область застосування варіантів використання

Метою цього випадку використання є аналіз коментарів користувачів за допомогою статистичних алгоритмів для створення моделей, які дозволять нам

визначити, чи впливають вони на ціну обговорюваної криптовалюти.

Опис

Цей варіант використання описує процес аналізу коментарів та цін.

Опис потоку Передумова

Коментарі було класифіковано за допомогою аналізу настроїв. Активація

Цей варіант використання починається, коли <Astor> завантажує дані.

Основний потік

1. <Astor> завантажує дані в R.
2. <Astor> запускає алгоритми статистичного аналізу коментарів.
3. <Astor> запускає алгоритми машинного навчання на коментарях.

<Astor> інтерпретує результати.

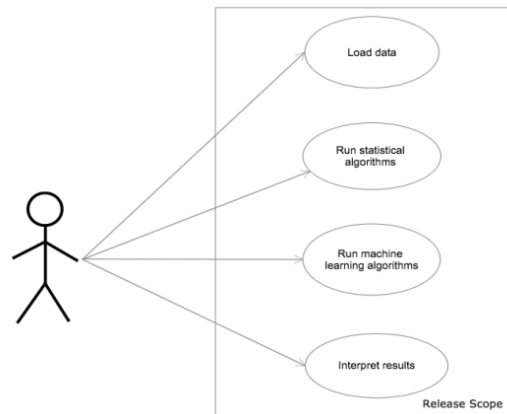


Рисунок 2.6 - Аналіз даних

Вимога 5: Вхід (рисунок 2.7)

Опис і пріоритет

Користувач зможе увійти в додаток, використовуючи свій обліковий запис Google

Область застосування варіантів використання

Метою цього випадку використання є вхід користувача в систему.

Опис

Цей варіант використання описує, як користувач увійде у веб-додаток, використовуючи свою електронну адресу та пароль.

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Опис потоку Передумова

Актор не увійшов у систему. Активация.

Цей варіант використання починається, коли Актор натискає кнопку входу. Основний потік.

1. Актор натискає кнопку входу.
2. Актор вводить свою електронну пошту та пароль.

Актор увійшов у систему.

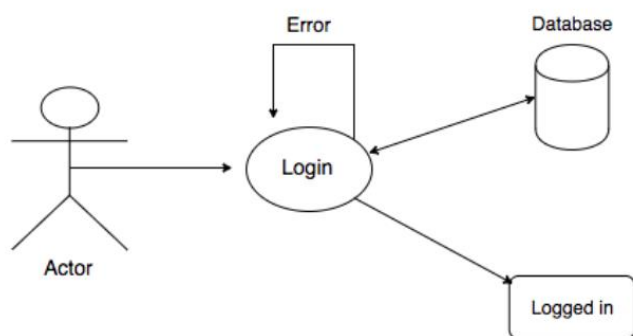


Рисунок 2.7 - Вхід

Винятковий потік

E1: Недійсні облікові дані

1. Актор вводить свої дані.
2. Система повертає попередження про недійсну електронну адресу або пароль.

3. Варіант використання продовжується на кроці 2 основного потоку.

E2: Користувач не зареєстрований

1. Актор вводить свої дані
2. Система повертає помилку через нерозпізнану електронну адресу

Припинення

Система реєструє користувача.

Умова публікації

Система переходить у стан очікування

Вимога 6: Пошук (рисунок 2.8)

Опис і пріоритет

Користувач зможе шукати певну крипто валюту

Сфера застосування

Мета цього випадку використання полягає в тому, щоб користувач шукав певну криптовалюту.

Опис: Цей варіант використання описує, як користувач шукатиме криптовалюту та буде перенаправлено на потрібну сторінку.

Діаграма варіантів використання

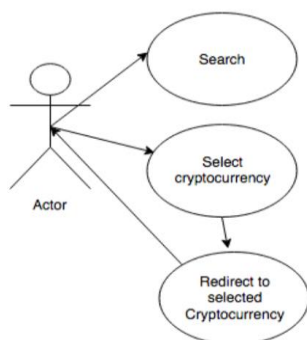


Рисунок 2.8 - Пошук

Активация веб-додатку «Актор»

Цей варіант використання починається, коли Актор починає вводити текст у рядок пошуку. Основний потік

1. Актор починає друкувати
2. Система повертає запропоновані результати
3. Актор вибирає криптовалюту.
4. Система перенаправляє на сторінку вибраної криптовалюти

Винятковий потік

E1: Немає результатів

1. Актор вводить свої дані про криптовалюту, якої не існує.
2. Система не повертає жодних результатів.
3. Варіант використання продовжується на кроці 1 основного потоку.

Припинення

Система перенаправляє користувача.

Умова публікації

Система переходить у стан очікування

Вимога 7: Переглянути N найкращих криптовалют (рисунок 2.9).

Опис і пріоритет

Користувач зможе переглянути N найкращих криптовалют за ринковою капіталізацією.

Область застосування варіантів використання

Мета цього випадку використання полягає в тому, щоб користувач переглядав 50 найкращих криптовалют за ринковою капіталізацією.

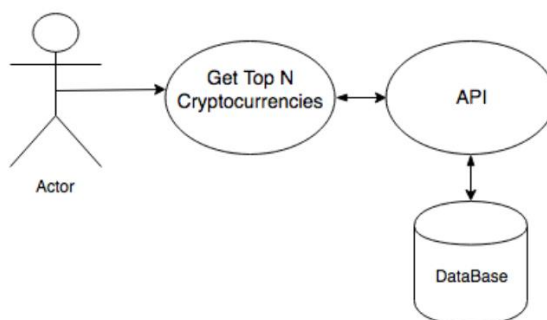


Рисунок 2.9 - Перегляд найпопулярніших криптовалют

Опис

Цей варіант використання описує, як буде переглядатися N найкращих криптовалют.

Діаграма варіантів використання

Опис потоку Передумова

Актор увійшов у систему Активація

Цей варіант використання починається, коли Актор переходить на головну сторінку

Основний потік

1. Актор завантажує головну сторінку.

2. Система запитує дані.
3. Система повертає дані.

Винятковий потік

E1: Помилка отримання

1. Актор завантажує головну сторінку.
2. Набір даних повертає помилку.
3. Випадок використання починається з кроку 1 основного потоку.

Припинення

Актор виходить із системи.

Умова публікації

Система переходить у стан очікування

Вимога 8: Конкретні дані про криптовалюту (рисунок 2.10)

Опис і пріоритет

Користувач зможе переглядати всі дані, що стосуються певної криптовалюти, такі як ціна, обсяг згадок за 24 години, настрої та будь-які інші відповідні результати аналізу.

Область застосування варіантів використання

Мета цього випадку використання полягає в тому, щоб користувач переглядав та взаємодівав з усіма даними, що стосуються певної криптовалюти.

Опис

Цей варіант використання описує, як користувач переглядатиме дані, що стосуються певної криптовалюти.

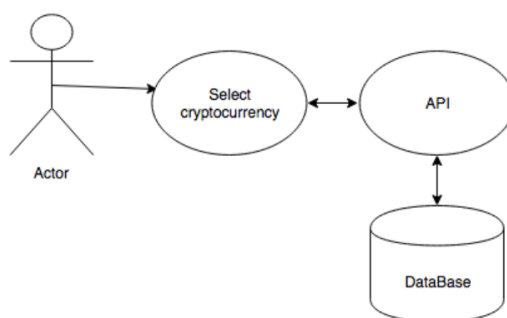


Рисунок 2.10 - Дані щодо конкретних крипто валют

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

Опис потоку Передумова

Актор увійшов у систему Активація

Цей варіант використання починається, коли Актор знаходиться на головній сторінці. Основний потік.

1. Актор переходить на головну сторінку.
2. Актор вибирає криптовалюту.
3. Система перенаправляє на сторінку огляду для цієї валюти
4. Актор переглядає дані.

Винятковий потік

E1: Помилка отримання

1. Актор вибирає криптовалюту.
2. База даних повертає помилку.
3. Варіант використання продовжується на кроці 2 основного потоку.

Припинення

Актор виходить із системи або залишає сайт.

Умова публікації

Система переходить у стан очікування

2.2 Нефункціональні вимоги до системи

Вимога доступності

Система та всі її компоненти будуть загальнодоступні через GitHub.

Користувачі зможуть отримати доступ до програми через веббраузер на мобільному пристрої, планшеті та ПК.

Вимога відновлення

Весь код і набори даних будуть резервно копіюватися локально, а також за допомогою хмарних сховищ, таких як Google Диск і GitHub.

Вимога надійності

Система відповідатиме всім функціональним вимогам. Скрипти в системі

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

працюватимуть, поки на комп'ютері встановлено необхідні залежності. Веб-додаток працюватиме постійно, і у разі збою сервера користувачеві доведеться знову увійти в систему.

Вимоги щодо довкілля

Система працюватиме на Mac, Windows та Linux. На комп'ютері потрібно буде встановити Python 3 та RStudio. Скрипти Python та R матимуть деякі залежності, які також потрібно встановити. Веб-додаток доступний через веббраузер і може бути перегляданий на будь-якому стандартному пристрої з підключенням до Інтернету.

Вимога розширюваності

Система буде розширюваною як за рахунок збільшення розміру даних, так і застосування складніших алгоритмів для пошуку закономірностей та індикаторів у коментарях. Завдяки кластеризації система зможе виявляти приховану семантичну структуру в коментарях.

Вимоги до даних

Для проведення аналізу потрібен достатній набір даних. Коментарі будуть зібрані з найпопулярніших платформ для обговорення криптовалют, а ціни будуть взяті з агрегатора цін на криптовалюту. Я обрав 2 джерела для коментарів та публікацій, оскільки різні платформи обслуговують різних користувачів з різними думками та рівнем експертизи.

Reddit.com

Reddit.com – один з найбільших сайтів для новачків та досвідчених криптоентузіастів. Сайт побудований на основі сабреддитів або спільнот, присвячених певним темам, де користувачі можуть вільно ділитися посиланнями та публікувати запитання чи теми для обговорення. Унікальною особливістю reddit є можливість створювати сабреддит на будь-яку тему, яка спадає на думку. Це дозволило криптовалютній спільноті розширитися до спільнот та підспільнот, присвячених нішевим областям [33]. Декількома прикладами цього може бути сабреддит про біткойн. /r/btc , який в основному призначений для новин та обговорення торгівлі біткойнами, /r/BitcoinBeginners, який призначений для

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

новачків у світі біткойна, які мають запитання або потребують поради, а також /r/CryptoCurrency, призначений для відкритого обговорення всіх криптовалют, а не лише біткойна. Різноманітність бази користувачів на Reddit дозволить цьому проєкту збирати публікації та коментарі від користувачів, починаючи від тих, хто купує вперше, і закінчуючи ентузіастами блокчейну. Будуть обрані найпопулярніші спільноти криптовалют на Reddit, а дані будуть зібрані за допомогою API Reddit [34].

Twitter.com

Хоча Twitter не призначений виключно для обговорення криптовалют, він пропонує чудовий засіб обміну, завдяки якому новини швидко поширюються. База користувачів набагато більша, ніж на Reddit, і там свої думки публікують користувачі будь-якого рівня, від тих, хто купує щось по-новому, до розробників блокчейну. Twitter пропонує API для розробників, і цей проєкт буде зосереджений переважно на твітах, що обговорюють криптовалюти зі списку 100 найкращих криптовалют coinmarketcap.com.

Coinmarketcap.com

Coinmarketcap.com є одним із найпопулярніших джерел цін та статистики криптовалют, яке відстежує такі дані про криптовалюту, як ринки, 24-годинний обсяг, ринкову капіталізацію та історичні дані. Coinmarketcap.com пропонує публічний API, і цей проєкт використовуватиме його для отримання поточних та історичних даних про ціни на 100 найкращих криптовалют.

Вимоги користувача. З точки зору клієнта, система повинна відображати результати аналізу лаконічно та легко для читання. Усі візуалізації повинні бути чітко позначені разом з усією відповідною інформацією щодо методів та алгоритмів, що використовуються для обчислення результатів [35]. Веб-додаток вимагає від користувачів доступу до веббраузера на пристрої, підключеному до Інтернету. Користувачі також повинні зареєструватися.

Вимоги до зручності використання

Веб-додаток повинен мати простий у навігації інтерфейс користувача, і для використання будь-яких його функцій не потрібен певний рівень знань. Веб-

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

додаток також повинен чітко та лаконічно відображати помилки для користувача.

2.3 Проєктування дизайну та архітектури веб-застосунку

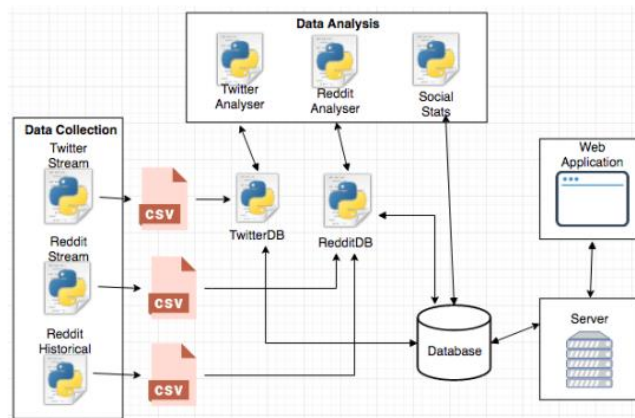


Рисунок 2.11 - Архітектура системи

Наведена вище архітектурна діаграма показує систему з вищого рівня. Система збирає дані як з Reddit, так і з Twitter, використовуючи відповідні API. Як Reddit, так і Twitter пропонують потокові API для збору твітів і постів, щойно вони надходять на кожен сайт. Система також збирає дані про історичні публікації та коментарі лише з Reddit, оскільки ця функція недоступна через API Twitter. Усі зібрані дані зберігаються у файлах CSV [36]. Потім система аналізує зібрані дані з Reddit і Twitter, витягуючи старі дані з бази даних для порівняння та оновлення за необхідності. Заключним етапом аналізу даних є створення соціальної статистики з проаналізованих даних для веб-застосунку.

Потім веб-сервер може витягувати ці дані з бази даних, надаючи їх веб-застосунку. Система використовує базу даних MongoDB для зберігання результатів аналізу, історичних цін на криптовалюту та різних інших даних, необхідних для веб-застосунку. На відміну від реляційних баз даних, які базуються на таблицях, MongoDB - це документоорієнтована база даних без схеми, де кожен документ має атрибути key/value. Загалом база даних містить 10 різних колекцій, і кожна колекція містить масив documents.

На рисунку 2.12 наведено огляд даних, зібраних та збережених з різних

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

subreddits, пов'язаних з криптовалютою. Кожен документ, що належить до Сабреддіт міститиме дані про найпопулярніших користувачів, найменш популярних користувачів, найактивніших користувачів, найчастіші слова, найчастіші біграми та найпопулярніші криптовалюти в цьому сабреддіті.

Проектування баз даних

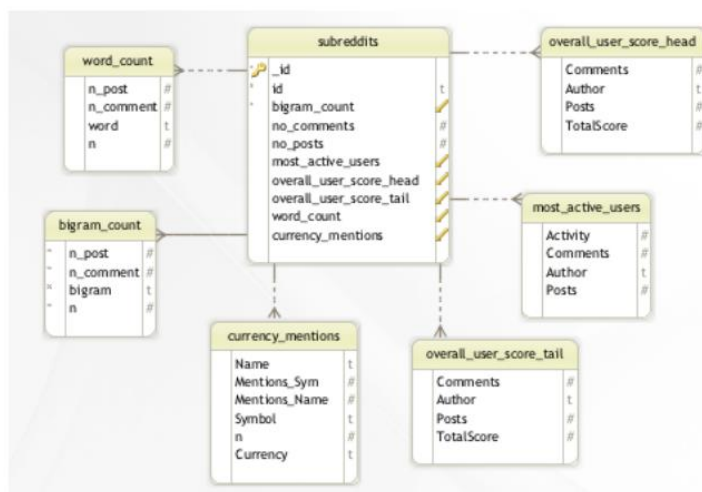


Рисунок 2.12 - Колекція Subreddit

5 колекцій, показаних на рисунку 2.13, мають спільну тему зберігання даних, зібраних щодня, з криптовалютних сабреддітів та твітів, пов'язаних із криптовалютами.

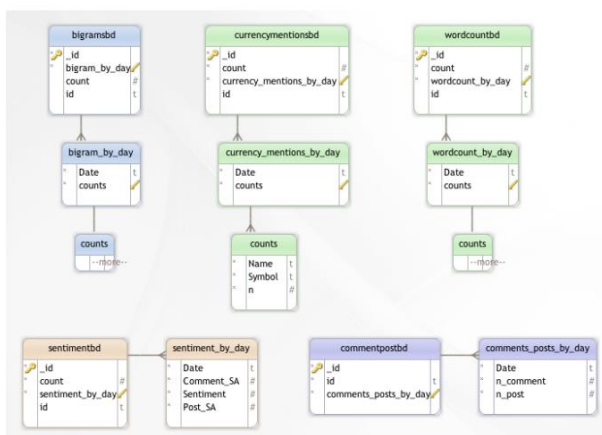


Рисунок 2.13 - Різні колекції

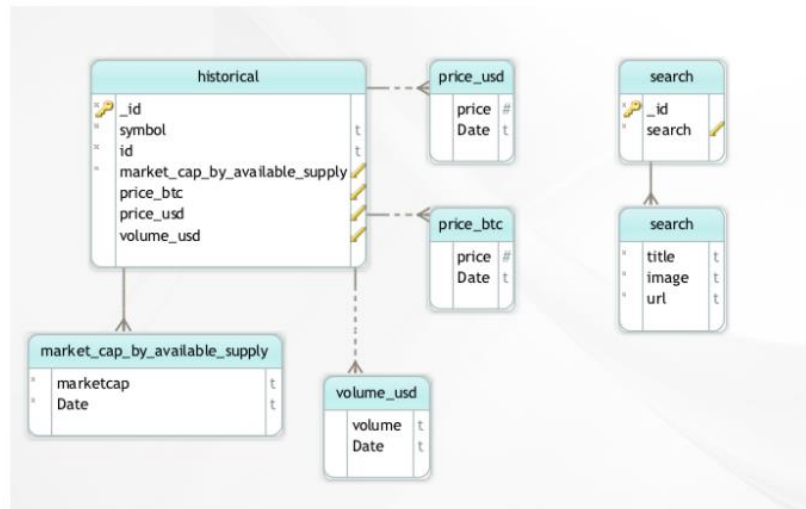


Рисунок 2.14 - Історичні ціни та пошукові колекції

Історична колекція, показана на рисунку 2.14, містить ціни в доларах США та BTC, а також ринкову капіталізацію та обсяг для різних криптовалют. Пошукова колекція є специфічною для веб-застосунку та містить масив криптовалют, доступних для пошуку.

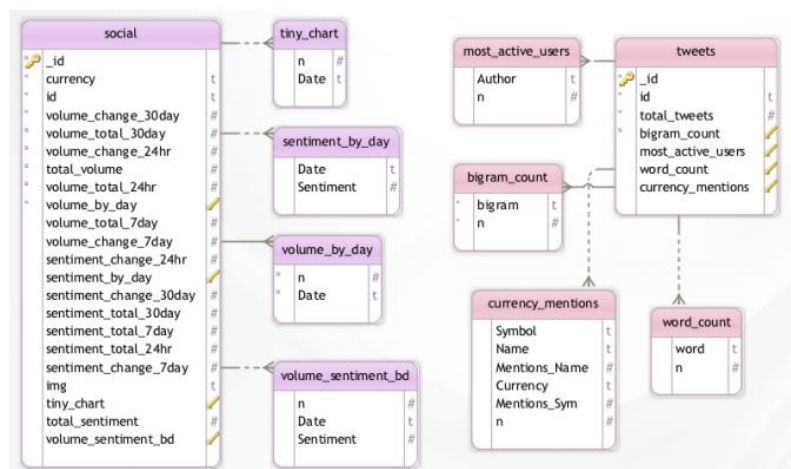


Рисунок 2.15 - Колекції соціальних мереж та твітів.

Подібно до колекції subreddits, на рисунку 2.15 показано колекцію твітів, яка містить такі дані, як частота слів та біграм, найактивніші користувачі та найчастіше згадувані криптовалюти. Колекція соціальних мереж, також показана на рисунку 2.15, містить соціальну статистику для різних криптовалют, таку як загальна кількість настроїв та їх зміна протягом 24 годин, 7 днів та 30 днів, а також

загальна кількість згадок у соціальних мережах та їх зміна, також протягом 24 годин, 7 днів та 20 днів. Колекція соціальних мереж також містить щоденні дані про згадки та настрої у соціальних мережах для певної криптовалюти.

2.4 Висновок по розділу

У другому розділі визначено функціональні та нефункціональні вимоги до інтелектуального веб-застосунку для аналізу криптовалютного ринку. Сформовано перелік основних можливостей системи, серед яких збір, обробка, аналіз та відображення ринкової інформації в режимі реального часу. Розглянуто вимоги до продуктивності, масштабованості, безпеки та зручності використання. Також спроектовано архітектуру та дизайн системи, що забезпечують ефективну взаємодію між клієнтською та серверною частинами застосунку. Результати розділу стали основою для подальшої програмної реалізації системи.

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНОГО ВЕБ-ЗАСТОСУНКУ

3.1 Програмні засоби реалізації системи аналізу криптовалютного ринку

Одним із найважливіших рішень перед початком розробки програмного забезпечення, яке слід прийняти якомога швидше, є вибір програмних засобів, які будуть використані для створення системи. До програмних засобів належать сама мова програмування (або мови, якщо їх кілька), бібліотеки та фреймворки, операційна система, середовище розробки, програмне забезпечення для тестування тощо.

Найважливішим фактором, що впливає на вибір мови програмування, є безпосередньо сама проектована система, для кого вона створюється та які завдання повинна вирішувати та над чим працювати. Наступним фактором є тип системи, в якій вона має бути реалізована: мобільний додаток, консольний додаток, розширення для браузера, веб-додаток, веб-сайт тощо. Не менш важливим фактором, який слід враховувати при виборі мови програмування, є популярність обраної мови. Вона тісно пов'язана з кількістю доступних фреймворків та бібліотек, підтримкою мови та майбутнім розвитком. Вибір популярної мови програмування полегшить оновлення та розвиток проектованої системи. Окрім вибору мови програмування, слід вибрати набір необхідних інструментів для побудови програмного забезпечення: середовище розробки та інше програмне забезпечення, що використовується в процесі створення проекту.

Порівняння Node.js та Python

Перш ніж говорити про Node.js, необхідно згадати безпосередньо саму мову програмування JavaScript. JavaScript – це мова програмування, створена Netscape Communications у 1995 році. У ці роки виникла потреба зробити веб-сторінки динамічними для реалізації анімації, взаємодії з користувачем тощо. Отже, виникла потреба у створенні скриптової мови програмування для

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

вирішення таких задач. Подібний синтаксис та швидке зростання популярності Java дали мові програмування відповідну назву. Але JavaScript був дуже повільним порівняно з традиційними компільованими мовами програмування. Роки досліджень та розробок вивели мову програмування на новий рівень і зробили можливим створення масштабних додатків, таких як Google Maps.

Довгий час JavaScript виконувався лише в браузері. Однак у 2009 році з'явився фреймворк під назвою Node.js, який дозволяв відтворювати код JavaScript на сервері. Node.js - це платформа, написана на C++, яка дозволяє виконувати файли JavaScript. Це дозволило JavaScript використовувати цю платформу для виконання кількох дій, таких як виконання дій із системними файлами, які недоступні у версії для браузера. Node.js базується на движку V8, розробленому Google. Це швидка та прогресивна віртуальна машина з вбудованим інтерпретатором, який компілює JavaScript у нативний код. На відміну від таких мов програмування, як C++, C#, Java, Python тощо, JavaScript на платформі Node.js - це подієво-керована мова програмування, яка забезпечує асинхронні можливості введення та виведення. Кожен процес викликається, коли відбувається певна подія, а це означає, що жоден з методів не блокує потік. Оскільки код JavaScript виконується поза браузером на платформі Node.js, продуктивність цієї мови програмування значно вища, оскільки движок V8 дозволяє ефективніше використовувати ресурси. Крім того, архітектура платформи, керована подіями та без потоків, дозволить вам обробляти кілька запитів одночасно, що позитивно впливає на ефективність та пришвидшує виконання коду.

Синтаксис у Node.js майже нічим не відрізняється від браузерного JavaScript. Також Node.js надає можливість використовувати TypeScript з великою кількістю типів [37]. У Node.js під час розробки застосунку бажано створювати набір модулів та мікросервісів, які взаємодіють один з одним за допомогою певного механізму та виконують свої процеси окремо. Це робить процес розробки платформи Node.js досить гнучким, оскільки немає потреби створювати монолітну систему, і завжди можна легко додати новий модуль. Активна та чисельна спільнота створює значну перевагу для Node.js – наявність десятків

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

тисяч модулів та бібліотек для полегшення роботи та вирішення різних загальнодоступних завдань. У Node.js модулі керуються за допомогою менеджера пакетів npm, де інструкції виконуються шляхом введення певних команд у терміналі.

Сфера застосування Node.js полягає, по суті, у розробці серверної частини веб-додатків. Однак платформа надає всі можливості для створення мобільних додатків, настільних додатків, хмарних сервісів, рішень для сегмента Інтернету речей тощо. Також, значною перевагою Node.js є кросплатформність, тобто створений додаток працюватиме на таких популярних операційних системах, як Windows, Linux, Mac тощо [38].

Розробка мови програмування Python розпочалася наприкінці 1989 року, коли над проектом почав працювати Гвідо ван Россум, голландський програміст з Центру математики та інформатики (CWI) в Амстердамі. Спочатку розробка вважалася хобі, а саме програмне рішення мало на меті вирішити існуючі проблеми та стати наступником мови ABC. Перша версія мови програмування Python, 0.9.0, була опублікована в 1991 році. Вона суттєво відрізняється від таких мов, як Java або C++. Основним завданням під час її розробки було створення мови, код якої легко читати. У 2000 році вийшов Python 2.0. З виходом цієї версії функціональність мови програмування була значно розширена: з'явився збирач сміття, який автоматично виконував операції над Ram, підтримка Unicode тощо.

Сьогодні Python - це мова програмування високого рівня з відкритим вихідним кодом. Остання версія мови на момент написання - Python 3.8. Python є досить популярною мовою та має активну аудиторію. Через це також існує багато бібліотек з відкритим доступом для вирішення різних завдань. Встановлення та керування додатковими пакетами та бібліотеками в Python здійснюється системою, яка працює в командному рядку та називається pip. Програми, створені на Python, працюватимуть на основних популярних операційних системах, оскільки мова є кросплатформною. Її можна використовувати як для серверних, так і для клієнтських завдань.

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Однією з найважливіших переваг та особливостей мови програмування Python є її синтаксис. Мова є відносно лаконічною та простою і не вимагає квадратних дужок, натомість правильних табуляцій або пробілів. Завдяки цим особливостям синтаксису мови, фахівцям різного рівня його легко читати та розуміти таблиця 3.1 .

Таблиця 3.1

Порівняльна характеристика Node.js та Python

Критерій (Criteria)	Node.js	Python
Архітектура (Architecture)	Подієво-орієнтована, не блокує потік виконання (<i>Event-driven doesn't block the flow</i>)	Синхронне виконання коду, блокує потік виконання (<i>Synchronous code execution blocks the flow</i>)
Продуктивність (Performance)	Підтримка багатопоточності (<i>Multithreading support</i>)	Однопоточковість (Single-threading)
Синтаксис (Syntax)	Схожий на браузерний JavaScript (<i>Similar to browser-based JavaScript</i>)	Лаконічний, на основі відступів (табуляції), легко читається (<i>Concise, tab-based, and easy to read</i>)
Масштабованість (Scalability)	Легко масштабується, мікросервісна модель (<i>Easily scalable, microservices model</i>)	Важко масштабується, монолітна модель (<i>Hard-to-scale, monolithic model</i>)
Бібліотеки (Libraries)	Велика кількість (Large number)	Велика кількість (Large number)
Універсальність (Versatility)	Можливість розробки вебзастосунків, десктопних та мобільних додатків (<i>Ability to develop web, desktop and mobile applications</i>)	Можливість створення вебзастосунків, десктопних та мобільних додатків (<i>Ability to build web, desktop and mobile applications</i>)
Популярність та спільнота (Popularity and community)	Висока (High)	Висока (High)
Сфера застосування (Application)	Складні односторінкові вебзастосунки (SPA), інтернет речей (IoT), робота в режимі реального часу (<i>Complex single-page web applications, Internet of things, real-time operation</i>)	Великі дані (Big data), 3D-моделювання, штучний інтелект (<i>Big data, 3D modelling, artificial intelligence</i>)

Дизайн мови Python передбачає використання монолітної моделі в розробці. Звичайно, використовуючи спеціальні бібліотеки та пакети, можна створювати додатки, які дозволяють писати асинхронний код та керувати подіями

в програмі. Але це не є нативним рішенням для мови, і налагодження в цьому випадку займає досить багато часу. Мова програмування Python є інтерпретованою мовою програмування та повільніша за традиційні мови, код яких компілюється. Ця мова не підтримує багатопоточність, а працює лише з одним потоком, що пов'язано з технічними особливостями мови. Однак, знову ж таки, використовуючи певну кількість бібліотек, можна відтворити певну функціональність багатопоточності. Тим не менш, механізм інтерпретатора - Gil, який використовує Python, не дозволяє виконувати певні завдання одночасно, але дозволить виконувати операції лише над одним потоком одночасно [39].

Під час вибору мови програмування, яка буде використана для розробки програмного рішення проектованої системи, перевага була надана JavaScript.

Завдяки особливостям архітектури Node.js, ця платформа, оскільки вона основана на подіях, найкраще підходить для розробки застосунків, які часто виконуватимуть численні паралельні запити та взаємодіятимуть між клієнтом і сервером у режимі реального часу. Хоча мови JavaScript та Python мають динамічну типізацію, використання мікросервісного підходу в Node.js, на відміну від монолітної моделі в Python, є більш доцільним варіантом для масштабованості розробленої системи. Крім того, цей підхід буде кращим для розробки розширюваних проектів, що полегшить роботу та дасть вам краще розуміння системи під час її обслуговування. Платформа Node.js, на відміну від Python, є багатопотоковою, і запити обробляються набагато швидше, що робить цю платформу, завдяки кращій продуктивності, улюбленим вибором для створення веб-системи [40].

Використання допоміжних модулів

Express.js. Expressjs - один із найпопулярніших серверних фреймворків для платформи Node.js, що використовується в розробці веб-додатків. Він є своєрідним «хребтом» у розробці під час створення серверної частини веб-системи. Надає широкий спектр можливостей для створення одно- та багатосторінкових веб-додатків. Цей фреймворк є гнучким і не нав'язує якогось конкретного правильного способу виконання певного завдання. Цей модуль

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє швидко розробляти інтерфейс прикладного програмного забезпечення (API) та створювати механізми для обробки різних методів запитів, таких як GET, POST, PUT, DELETE та інші [35].

Request.Request для Node.js користується попитом при роботі з HTTP-запитами. Хоча він є обгорткою навколо нативного HTTP-модуля Node.js, він набагато простіший в експлуатації. Модуль надає можливість надсилати різні типи запитів, використовувати проксі-сервери, інші мережеві адаптери під час надсилання запитів, TSL/SSL та вибирати різні конфігурації. Враховуючи його функціональність, цей модуль є гарним рішенням для реалізації методів збору даних сервером та створення API для клієнтської частини системи [36].

Fs. Модуль fs (від file system) є одним з найбазовіших і найнеобхідніших модулів Node.js. Ця бібліотека функцій використовується під час роботи з файлами операційної системи. Набір методів у цьому модулі дозволяє виконувати різні маніпуляції з файлами та з файловою системою комп'ютера: читати, записувати, доповнювати Файл, створювати, редагувати та видаляти каталоги тощо. Кожен метод має дві версії для роботи: синхронну та асинхронну, залежно від потреби. Навіть з таким лаконічним набором планів цей модуль надає програмі можливість виконувати всі основні операції над файлами операційної системи або їх каталогами [37].

Середовище розробки Atom

Створення та редагування коду – як текстовий редактор було обрано Atom, середовище розробки для створення проектованої інформаційної системи. Перша версія цього програмного забезпечення з'явилася у 2014 році та зазнала кількох оновлень, з того часу здобувши значну популярність. Програма розроблена однією з найбільших спільнот розробників програмного забезпечення – GitHub. Цей програмний продукт є повністю безкоштовним та може використовуватися на таких операційних системах, як Windows, Linux, Mac та інших.

Головною перевагою Atom є можливість встановлення сторонніх пакетів, які суттєво змінюють редактор, роблячи його повноцінним середовищем розробки. Сьогодні у відкритому доступі є близько 9000 модулів для Atom, які

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

допомагають розробникам вирішувати різні завдання та розробляти програмне забезпечення. Крім того, редактор також може безпосередньо змінювати будь-які налаштування розробника, оскільки програма має відкритий вихідний код та написана з використанням таких технологій, як JavaScript, CSS та HTML.

Ця програма підходить для розробки графічного інтерфейсу користувача та створення серверної частини веб-застосунків. Atom вже має базовий широкий набір цінних інструментів та функцій, а також можливість створювати свій функціонал або додавати існуючий, встановивши необхідний пакет.

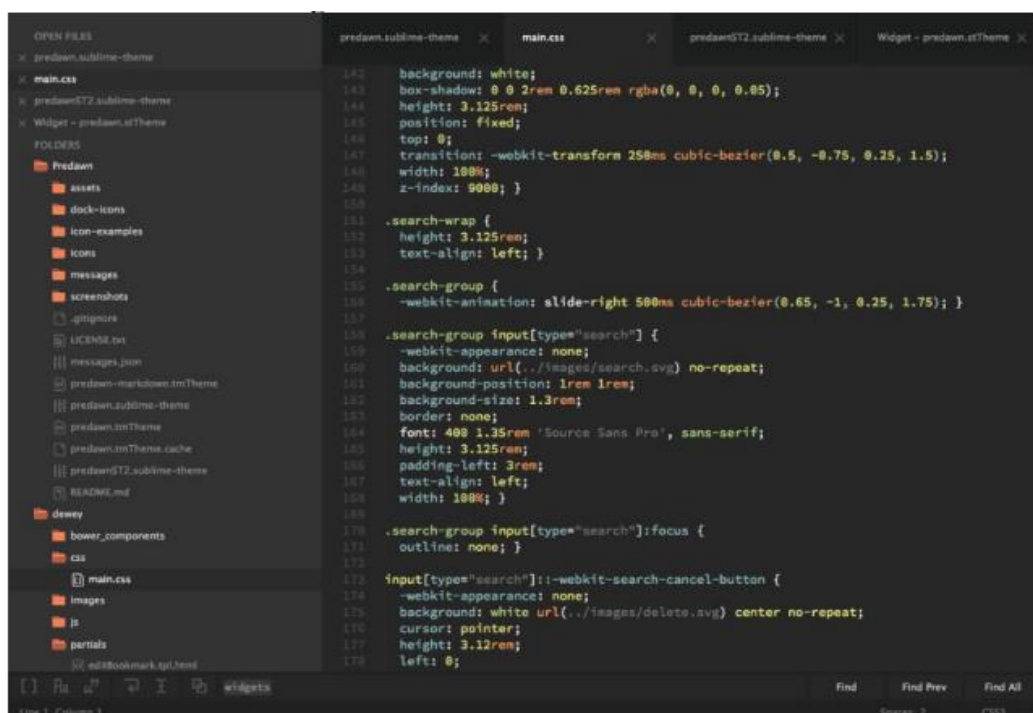


Рисунок 3.1 - Інтерфейс середовища розробки Atom

Atom надає можливість легко та швидко переміщатися по файлах проекту, використовувати технологію автозаповнення (також відому як IntelliSense) для популярних мов програмування, швидко редагувати файли HTML та CSS та переглядати зміни на сторінці в режимі реального часу, редагувати багато файлів одночасно, виконувати консольні команди та запускати програми, не виходячи з програмного середовища тощо [38]. Наприклад, інтерфейс програми показано на рис. 3.1.

Програмне забезпечення для тестування листонш

					БР.ІП – 67.00.00.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки програмне рішення проектованої системи передбачає наявність та розробку інтерфейсу прикладного програмного забезпечення API, що надсилає багато запитів, виникає потреба у використанні програмного забезпечення для тестування для таких завдань. Як таке програмне забезпечення було обрано Postman – програму, яка є хорошим інструментом для тестування готових зроблених або власноруч створених API.

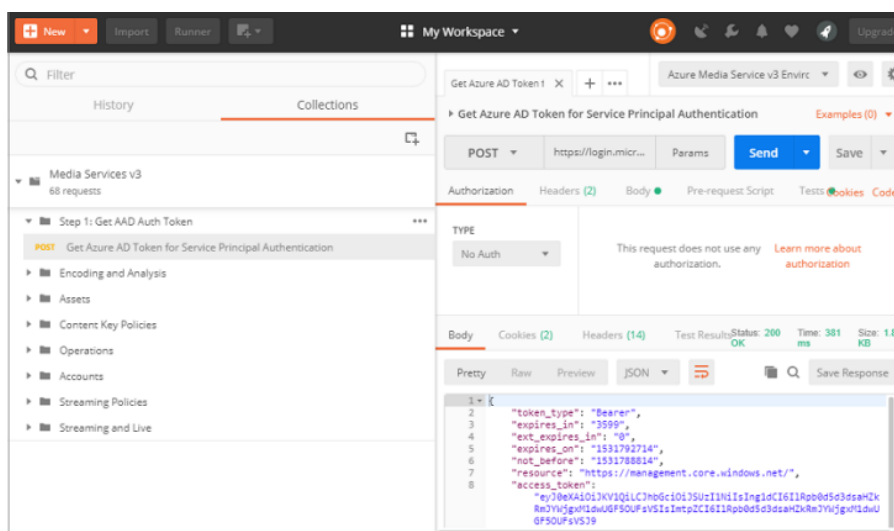


Рисунок 3.2 - Інтерфейс програми Postman

Основна кількість необхідних функцій у цьому програмному забезпеченні є безкоштовною. Програма надає зручний інтерфейс з широким функціоналом, що дозволяє проводити тестування через API, а саме надсилати HTTP-запити без попереднього написання програмного коду. Принцип роботи в Postman досить простий: введіть посилання, за яким буде надіслано запит, виберіть метод, заповніть форму або заголовки, якщо необхідно, та надішліть пропозицію, яка поверне код стану, час відповіді та відповідь від сервера з детальними даними. Також є можливість встановити моніторинг [39]. Інтерфейс програми показано на рис. 3.2.

Технічні характеристики обраних програмних засобів розробки

Для розробки програмного рішення для інформаційної системи арбітражного пошуку [40] було обрано мову програмування JavaScript та

					БР.ІП – 67.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

платформу Node.js, що є найоптимальнішим вибором на основі проведеного аналізу. Система буде реалізована як веб-додаток з клієнт-серверною архітектурою. Можливість асинхронного виконання коду, подієво-кероване середовище платформи, висока продуктивність у завданнях клієнт-серверної комунікації в режимі реального часу та підтримка багатопоточності платформи node.js роблять її найкращим вибором для програмної реалізації системи. Значною перевагою також є популярність мови та розмір її спільноти, що забезпечує подальші оновлення та наявність великої кількості готових рішень у відкритому доступі [17]. Для роботи було обрано такі основні модулі:

- експрес - для створення інтерфейсу прикладного програмного забезпечення та забезпечення зв'язку між клієнт і сервер;
- запит – для надсилання та обробки заявок серверною частиною системи;
- fs - для роботи та маніпулювання файлами операційної системи.

Як середовище розробки програмного забезпечення було обрано Atom, безкоштовний текстовий редактор з відкритим вихідним кодом від GitHub з можливістю встановлення додаткових пакетів. Ця програма забезпечує весь необхідний функціонал для розробки та часткового налагодження серверної складової системи, можливість автозаповнення коду, зручну навігацію по проєкту, створення веб-сторінок та перегляд їх змін у режимі реального часу тощо. Для проведення тестування на власноруч розроблених інтерфейсах додатків програми та перевірки надсилання запитів до існуючих сторонніх API було обрано програму Postman, безкоштовний функціонал якої повністю справляється з поставленими завданнями. У цьому розділі було проведено порівняльний аналіз та розглянуто мову програмування для її подальшого застосування у створенні програмного рішення системи. Було обрано тип архітектури проєктованої системи - клієнт-серверна веб-платформа. Встановлено необхідні модулі та фреймворки для обраної мови програмування, які будуть використовуватися під час розробки. Нарешті, вибрано середовище розробки та описано його основні

					БР.ІІІ – 67.00.00.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

функції та переваги.

3.2 Розроблення графічного інтерфейсу користувача

Веб-застосунок було розроблено за допомогою ReactJS та спроектовано за допомогою Semantic-UI. ReactJS - це бібліотека JavaScript з відкритим кодом, розроблена Facebook для створення користувацьких інтерфейсів. Вона в основному використовується для обробки рівня перегляду для веб-застосунків та дозволяє розробникам створювати компоненти інтерфейсу користувача, які можна повторно використовувати. Мета react - створювати прості, масштабовані та швидкі веб- та мобільні застосунки. Semantic-UI - це фреймворк для адаптивного користувацького інтерфейсу, який дозволяє розробникам спростити процес проектування фронтенду. Semantic-UI повністю інтегровано в ReactJS, що призводить до швидкого та красивого дизайну веб-застосунку.

Цільова сторінка сайту показана на рисунку 3.3. Дизайн був натхненний іншими сайтами-агрегаторами цін на криптовалюту, де валюти та пов'язана з ними статистика відображаються в табличній формі. Цільова сторінка надає огляд поточного ринку криптовалют зі статистикою соціальних мереж поряд з цінами. Під час першого завантаження сторінки дані запитуються з сервера, і користувачеві відображається символ завантаження, доки сторінка не буде готова. Згадки та значення настроїв у соціальних мережах відображаються протягом 24 годин та 7 днів, а також у крайньому правому стовпці також наведено діаграму згадувань, щоб швидко ознайомитися з ажіотажом у соціальних мережах щодо цієї валюти за останні 7 днів.

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

CryptoAnalytics										
#	Name	Symbol	Price	Market Cap	Social Mentions			Social Sentiment		
					Change (24h)	Total (24h)	Total (7d)	Change (24h)	Total (24h)	Total (7d)
1	Bitcoin	BTC	\$8,603.31	\$146,523,617,858	-44%	12966	96939	2%	17	118
2	Ethereum	ETH	\$698.19	\$69,404,686,962.0	-39%	4519	37445	7%	13	96
3	Ripple	XRP	\$0.701185	\$27,479,417,880.0	-49%	3271	29401	-3%	16	110
4	Bitcoin Cash	BCH	\$1,477.69	\$25,305,884,557.0	-45%	4558	35641	-1%	15	107
5	EOS	EOS	\$14.5776	\$12,433,023,033.0	72%	1657	13794	-9%	12	89
6	Litecoin	LTC	\$140.642	\$7,947,871,066.0	-44%	420	2999	8%	6	40
7	Cardano	ADA	\$0.275225	\$7,135,777,989.0	46%	484	2620	5%	11	75

Рисунок 3.3 - Цільова сторінка

Кожен стовпець таблиці можна сортувати, що дозволяє користувачам сортувати за згадками, настройми або ціною. Сортування за замовчуванням здійснюється за ринковою капіталізацією. Кожен стовпець у таблиці також містить спливаюче вікно при наведенні на нього курсора, яке пояснює дані, що містяться в стовпці. У заголовку головної сторінки також є рядок пошуку, який дозволяє користувачам шукати певну криптовалюту.

Окремий розділ криптовалюти

Натискання на рядок криптовалюти на головній сторінці перенаправить користувача на сторінку огляду, де відобразиться масив інформації про вибрану криптовалюту. Вся сторінка розділена на сегменти з сітчастим макетом, що робить її повністю адаптивною. Для кожної криптовалюти передбачено чимало сегментів, тому для цієї демонстрації я розбив сторінку огляду на 3 частини.

Основна статистика та графік ціни/обсягу/настроїв



Рисунок 3.4 - Основні статистичні дані та графік

					БР.ІП – 67.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

Графік надає можливість перегляду історичних даних для вибраної криптовалюти. Випадаюче меню масштабування дозволяє користувачам вибирати період перегляду: 7 днів, 1 місяць або 3 місяці даних. Графік складається з лінії та площі, де лінія представляє ціну, а площа – обсяг у соціальних мережах. Наведення курсора на певну точку графіка відкриє легенду, що показує дату, ціну та обсяг у соціальних мережах за цю дату. Дві кнопки в заголовку графіка дозволяють легко перемикатися між відображенням настроїв у соціальних мережах та обсягу на графіку, а також відображенням ціни в доларах США або BTC.

Під діаграмою та основною статистикою розташовані два сегменти для пов'язаних валют та соціальної статистики. Сегмент пов'язаних валют відображає стовпчасту діаграму з усіма іншими валютами, які згадуються в обговореннях певної криптовалюти.



У цьому випадку це всі валюти, згадані в обговореннях Ethereum, і Bitcoin посідає перше місце серед згаданих з великим відривом. Таблиця соціальної статистики надає огляд настроїв у соціальних мережах та обсягу для певної криптовалюти. Стовпці "Всього" та "Зміна" надаються для згадок та настроїв протягом 24 годин, 7 днів та 30 днів. Знову ж таки, наведення курсора на заголовок стовпця надасть користувачеві більше деталей щодо даних у кожному стовпці. Стовпець змін як для згадок, так і для настроїв відображається різним кольором залежно від того, чи є зміна позитивною чи негативною. Зміна базується на значенні в стовпцях "Всього" порівняно зі значенням за попередній період. У цьому прикладі згадки Ethereum у соціальних мережах зросли на 87% порівняно з попередніми 30 днями.

Найактивніші користувачі, слова та біграми за датою

На рисунку 3.6 показано сегменти для найактивніших користувачів, а також кількість слів і біграм. Таблиця «Найактивніші користувачі» містить список користувачів, які найчастіше обговорюють певну криптовалюту. У стовпці активності показано, скільки постів і коментарів було отримано від цього користувача, що обговорюють певну криптовалюту.



Рисунок 3.6 - Найактивніші користувачі, слова та біграми.

Натискання на ім'я користувача перенаправить на його профіль на Reddit. Стовпчикові діаграми слів і біграм надають користувачеві найчастіше вживані слова та найчастіше вживані біграми (пари слів) під час обговорення певної криптовалюти. У верхній частині сегмента є кнопка, яка дозволяє користувачам вибирати між підрахунками за весь час або виберіть дату (рисунки 3.7 -3.10). Вибір конкретної дати відобразить найбільше слів та біграм, вимовлених за цю дату.

Рядок пошуку у верхній частині сторінки дозволяє користувачам швидко знайти будь-яку криптовалюту, навіть якщо вона не відображається на головній сторінці. Під час введення назви, тікеру або символу (наприклад, «Bitcoin», «BTC», «ETH») список результатів динамічно оновлюється в реальному часі. Пошук працює за частковим збігом, є нечутливим до регістру та підтримує одночасний пошук як за повною назвою, так і за тікером. Результати відображаються у випадаючому списку з релевантними підказками (назва + тікер + іконка монети), відсортованими за релевантністю. Клік по будь-якому рядку у списку миттєво переводить користувача на детальну сторінку огляду обраної криптовалюти.

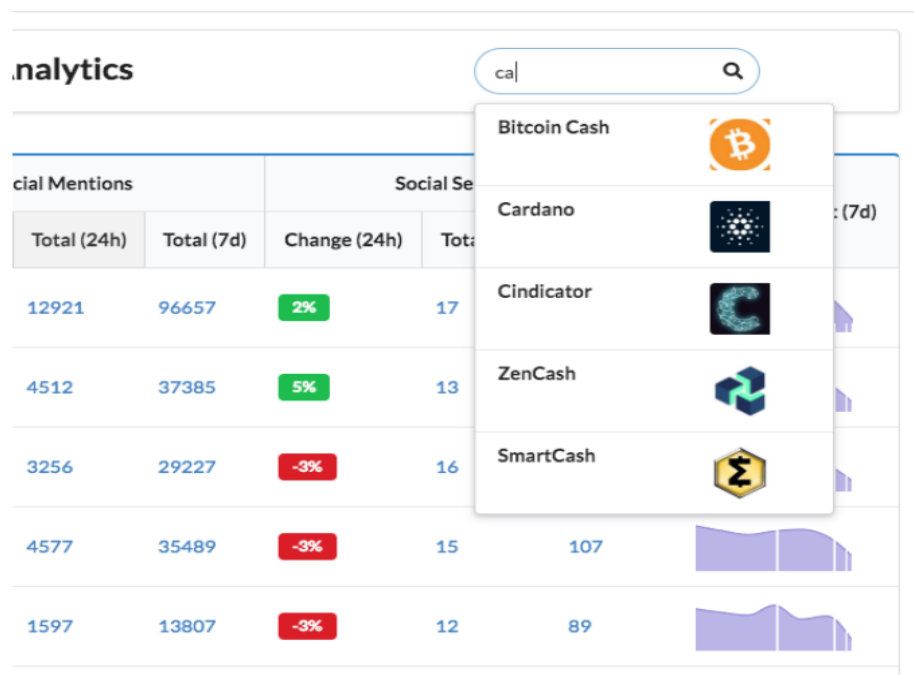


Рисунок 3.7 - Панель пошуку

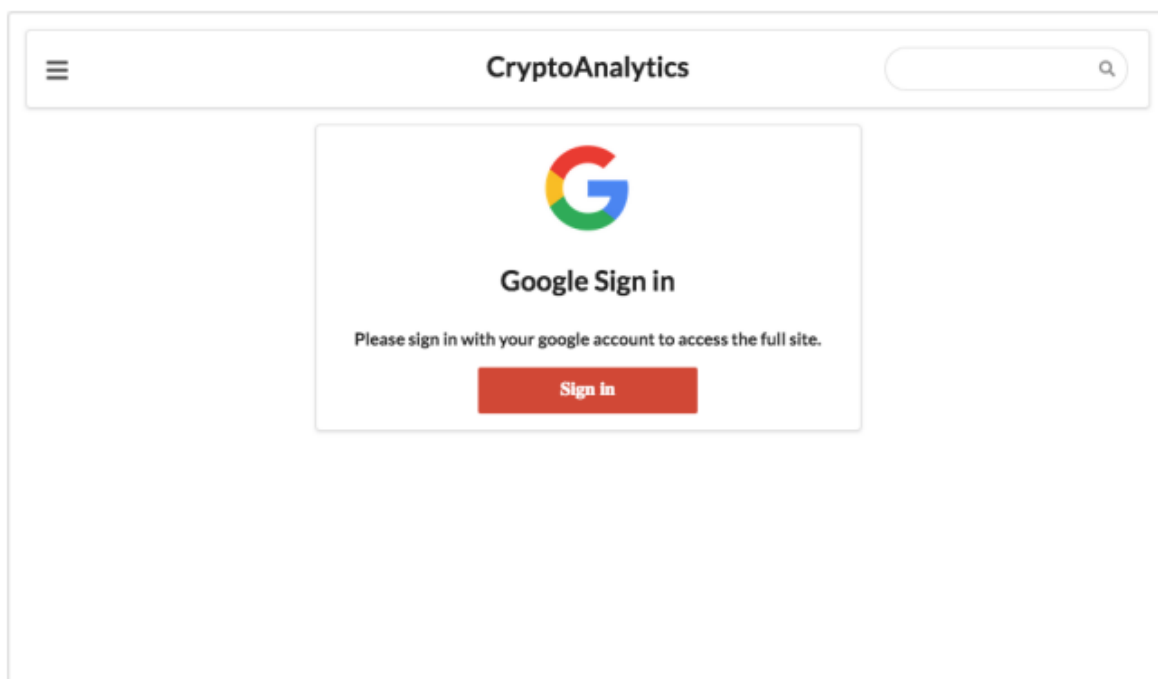


Рисунок 3.8 - Вхід

Перш ніж користувач зможе отримати доступ до огляду певної криптовалюти, йому потрібно увійти (рисунок 3.9), використовуючи свій обліковий запис Google. Після входу його буде перенаправлено до валюти, яку він вибрав раніше.

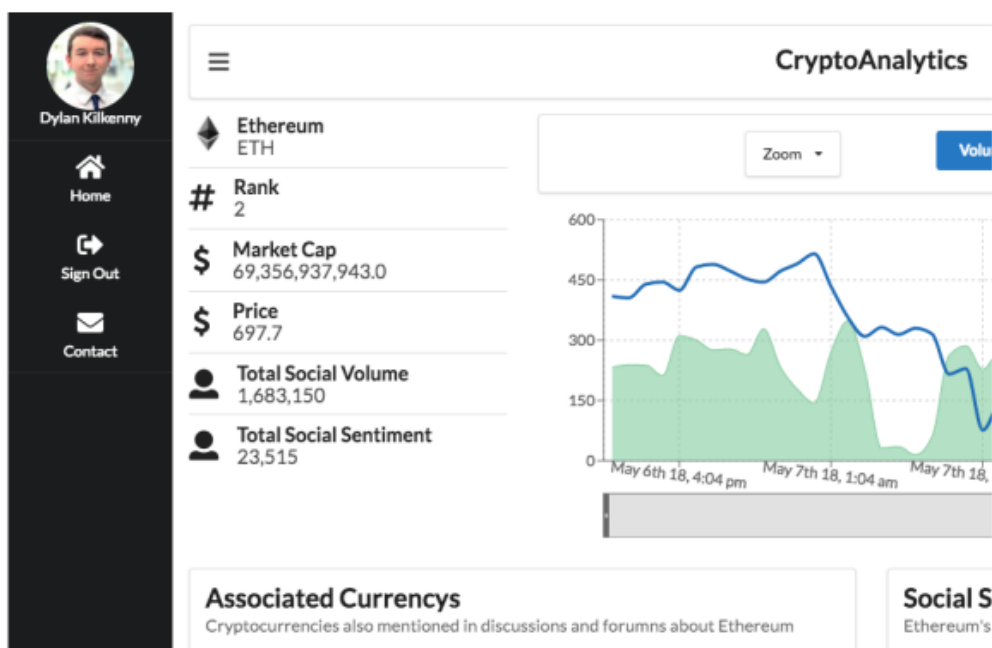


Рисунок 3.9 - Користувач увійшов у систему

					БР.ІП – 67.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

Після входу в систему користувач бачить своє фото профілю та ім'я при відкритті бічного меню

Home

Sign in

Contact



CryptoAnalytics

#	Name	Symbol	Price	Market Cap	Social Mentions		
					Change (24h)	Total (24h)	Total
1	 Bitcoin	BTC	€7,180	€122,288,162,954	-50%	12921	9665
2	 Ethereum	ETH	€583	€57,988,295,943.0	-30%	4512	3738
3	 Ripple	XRP	€1	€22,940,921,553.0	-52%	3256	2922
4	 Bitcoin Cash	BCH	€1,224	€20,954,162,803.0	-50%	4577	3548
5	 EOS	EOS	€12	€10,327,246,389.0	20%	1597	1380

Рисунок 3.10 - Користувач вийшов із системи

Якщо користувач не увійшов у систему, у бічному меню нічого не відображатиметься.

3.3 Аналіз результатів використання та оцінка ефективності веб-застосунку

Головною метою цього проєкту було визначити, чи можна передбачити рух цін на криптовалюти за допомогою даних із соціальних мереж. Другою метою було створити вебзастосунок, подібний до сайту агрегації цін на криптовалюти, але з додатковими функціями вимірювання згадувань у соціальних мережах та настроїв щодо певної криптовалюти.

Сервер NodeJS було налаштовано для забезпечення API для веб-застосунку, щоб він міг надсилати запити до бази даних. ExpressJS — це фреймворк NodeJS, який дозволяє налаштовувати проміжне програмне

забезпечення для відповіді на HTTP-запити.

```
// Retrieves all currencies from db
app.get('/AllCurrencies', function (req, res) {
  // Assign quantity
  var quantity = req.query.quantity
  // Get live cryptocurrency data from coinmarketcap
  getCoins(null, quantity, function (coins) {
    // Retrieve social stats from database
    RedditAPI.Social(db, function (social) {
      // map retrieve cryptocurrency data and social stats on id
      const result = coins.map(val => {
        return Object.assign({ _id: val.id }, val, social.filter(v => v.id === val.symbol)[0]);
      });
      // filter out results without any social stats
      const filter = result.filter(obj => {
        if ('tiny_chart' in obj) {
          if (obj.tiny_chart.length > 0) {
            return true
          }
        }
      })
      // return json object
      res.json(filter)
    })
  })
});
```

Рисунок 3.11 - Кінцева точка /AllCurrencies

На рисунку 3.11 показано одну з кінцевих точок, реалізованих на сервері API.

Вищевказана кінцева точка використовується для головної сторінки веб-застосунку. До кінцевої точки надсилається HTTP-запит GET разом із параметром запиту для кількості валют, які потрібно повернути. Кількість передається як аргумент до

Функція getCoins, яка повертатиме вказану кількість криптовалют з API coinmarketcap.com. Звідти до бази даних запитується соціальна статистика для всіх криптовалют. Два масиви, криптовалют та соціальної статистики, згенеровані для криптовалют в системі аналізу, зіставляються в один масив за ідентифікатором ключа. Об'єднаний масив потім фільтрується, щоб видалити будь-які криптовалюти, де не було соціальної статистики, і отриманий масив повертається до веб-застосунку.

Щоб розпочати цей проєкт, мені довелося вирішити, які криптовалюти

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

відстежувати. Наразі на coinmarketcap.com представлено понад 1500 криптовалют, більшість з яких не демонструють значної цінової динаміки чи визнання користувачів у соціальних мережах.

Щоб спростити це завдання, я вирішив зосередитися на 200 найкращих криптовалютах на coinmarketcap.com. Протягом усього цього проєкту Python активно використовувався для економії часу на рутинних завданнях, таких як агрегування списків криптовалют та парсинг їхніх відповідних облікових записів соціальних мереж і URL-адрес [subreddit](https://www.reddit.com).

Щойно я визначився з тим, які валюти відстежувати, і зібрав необхідну для цього інформацію, таку як повні назви, тікери, скорочення, URL-адреси соціальних мереж і [subreddit](https://www.reddit.com), я почав збирати історичні та поточні дані про криптовалюту. Ще до початку розробки веб-сервера або веб-додатку система збору та аналізу даних мала працювати в хмарі цілодобово. Одна з цілей цього проєкту - визначити, чи можна передбачити рух ціни криптовалюти, використовуючи історичні дані про ціну та дані соціальних мереж, пов'язані з певною криптовалютою.

Вся система розміщена на сервері Ubuntu з 4 ГБ оперативної пам'яті та 2 процесорами, наданими DigitalOcean.com. Доступ до сервера здійснюється через віддалений вхід через SSH, а завантаження та вивантаження файлів здійснюється через SFTP. Для забезпечення безперебійної роботи системи було налаштовано багато `cronjob`. Збір даних відбувається в 3 частини: постійний потік твітів, які зберігаються у файли CSV, постійний потік постів та коментарів Reddit, які зберігаються у файли CSV, і, нарешті, збір історичних даних з Reddit, які також зберігаються у файли CSV. Два потокові скрипти для Reddit та Twitter, які підключаються до відповідного потокового API кожного сайту, працюють 24 години на добу, постійно зберігаючи твіти, пости та коментарі у файли CSV. Було налаштовано два `cronjob`, які викликають інший скрипт Python для впорядкування та передачі зібраних CSV-файлів до класів аналізатора. Ці `cronjob` виконуються раз на годину, в різний час, щоб уникнути перевантаження ресурсів.

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if __name__ == "__main__":

    hashtags = []
    # Load currencies
    CURRENCYS = pd.read_csv('../data/CurrencySymbols.csv')
    # Create hashtag list
    for i, row in CURRENCYS.iterrows():
        hashtags.append("#"+row["Name"])
        hashtags.append("#"+row["Symbol"])
    # Create OAuth instance
    AUTH = tweepy.OAuthHandler(PERSONAL_KEY, PERSONAL_SECRET)
    AUTH.set_access_token(ACCESS_TOKEN_KEY, ACCESS_TOKEN_SECRET)
    # Create stream instance
    TWITTER_STREAM = Stream(AUTH, MyListener())
    # Filter based on hashtags
    TWITTER_STREAM.filter(track=hashtags[:400])

```

Рисунок 3.12 - Екземпляр потоку та фільтр

Значна частина обговорень криптовалют відбувається в Твіттері. Зазвичай там робляться оголошення, які зрештою поширюються на Reddit та інших сайтах соціальних мереж. На щастя, Твіттер пропонує потоковий API, який дозволяє завантажувати твіти в режимі реального часу, і це дуже корисно для збору великої кількості твітів, пов'язаних з широким колом тем. На відміну від REST API, де дані витягуються самотійно, потоковий API Твіттера передає дані в постійний сеанс. Це дозволяє завантажувати більше даних у режимі реального часу, ніж це можна було б зробити за допомогою REST API.

API Twitter досить вичерпний і складний для навігації. Щоб пришвидшити розробку та зменшити кількість головних болів, було використано Твееру, обгортку для Twitter на Python.

Наведений вище фрагмент взято зі скрипта Python, який реалізує бібліотеку твееру для потокової передачі твітів. На рисунку 3.12 показано створення екземпляра OAuth, необхідного для зв'язку з API Twitter, та створення екземпляра потоку для отримання твітів у прямому ефірі. Щоб збирати лише твіти, пов'язані з криптовалютами, потік потрібно фільтрувати. Завантажується CSV-файл, що містить назви та тікерні символи для 200 криптовалют, а з назв та символів створюються хештеги, які додаються до списку та передаються до потоку Twitter для відстеження.

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

```

def on_data(self, data):
    # Load tweet json
    tweet = json.loads(data)
    # If a stand alone tweet and not a retweet
    if "extended_tweet" in tweet and (not tweet["retweeted"]) and ('RT @' not in tweet["text"]):
        # print and update counter
        print("Count: " + str(self.count), end="\r")
        self.count += 1
        # transform date
        date = time.strftime('%Y-%m-%d %H:00:00', time.strptime(tweet["created_at"],
                                                                '%a %b %d %H:%M:%S +0000 %Y'))
        # Append relevant data to tweets list
        self.tweets.append([
            date,
            tweet["user"]["screen_name"],
            tweet["extended_tweet"]["full_text"],
            tweet["retweet_count"],
            tweet["favorite_count"]
        ])
    # If length of tweets list == 5000, export to csv
    if len(self.tweets) == 5000:
        t = self.tweets
        self.tweets = self.tweets[:1]
        self.SaveTweet(t)

```

Рисунок 3.13 - Функція обробки даних у Twitter stream.

На рисунку 3.13 показано клас слухача потоку, реалізований з бібліотеки твеєру. Коли отримується твіт, викликається вищезгадана функція, і твіт перетворюється на об'єкт Python з JSON. Потім перевіряється, чи є твіт відповіддю, ретвітом чи стандартним твітом. Ідеальний твіт у цьому випадку – це той, який не є ретвітом чи відповіддю, а просто звичайним твітом. Дата твіту потім перетворюється на відповідний формат для етапу аналізу. Після того, як обов'язкові поля з твіту додаються до списку, що містить усі отримані твіти, перевіряється довжина списку. Якщо список містить 5000 твітів, список твітів зберігається у файлі CSV, а список очищується. Це робиться з кількох причин. Об'єм пам'яті на сервері Ubuntu, де розміщена система, досить обмежений. Запобігання надмірному збільшенню твітів, що зберігаються в пам'яті, та запобігання надмірному збільшенню розміру набору даних, який буде передано аналізатору даних, є необхідними запобіжними заходами для забезпечення безперебійної роботи системи.

```

# Get subreddits
subreddits = reddit.subreddit(subs)
# Stream comments recieved to subreddits
for comment in subreddits.stream.comments():
    # Get post
    submission = comment.submission
    # Add comment to list
    comments.append(
        [comment.id, comment.id, comment.body, comment.created_utc, comment.score, comment.author])
    # add the comment to the appropriate csv file
    SaveComments(str(submission.subreddit), comments)
    # Get postIDs
    PostIDs = pd.read_csv('../data/reddit/PostIDs.csv')
    # Check if already processed
    if PostIDs['ID'].str.contains(submission.id).any():
        continue
    # Add current postID to data frame and save back to csv
    PostIDs.loc[len(PostIDs)] = submission.id
    PostIDs.to_csv('../data/reddit/PostIDs.csv', sep=',', index=False)
    # Add post to the list
    posts.append(
        [submission.id, submission.title,
         submission.created_utc, submission.score,
         submission.num_comments, submission.author])
    # Save post to appropriate csv file
    SavePosts(str(submission.subreddit), posts)

```

Рисунок 3.14 - Трансляція Reddit

Збір публікацій та коментарів на Reddit відбувається кількома способами. Найголовніше, що трансляція на Reddit налаштовується на отримання живої стрічки коментарів та публікацій. Подібно до API трансляції Twitter, дані надсилаються, а не витягуються, що дозволяє збирати більше даних одночасно. Для пришвидшення розробки було використано обгортку API Reddit. PRAW, що розшифровується як обгортка API Python Reddit, забезпечує простий доступ до API Reddit і навіть керує багатьма правилами API Reddit, такими як обробка OAuth, агента користувача та обмеження кількості запитів за хвилину.

Потік Reddit приймає список subreddits для завантаження як аргумент. Потім надсилання для цих subreddits отримуються та додаються до черги для розбору. Рисунок 3.14 – це фрагмент основної логіки потоку Reddit. Цикл for міститься в реченні try-ехсерт для обробки помилок, а також у циклі while для того, щоб потік залишався відкритим. Коли коментарі надходять через потік, вони додаються до списку коментарів, а потім до відповідного csv-файлу.

Кожен subreddit має власні CSV-файли для коментарів та публікацій, щоб не змішувати всі коментарі та публікації subreddits в одному файлі. Після додавання коментарів до CSV-файлу завантажується файл PostIDs. Цей файл містить список попередніх публікацій, які були перевірені. Оскільки люди можуть коментувати публікацію будь-коли, а тип відкритого потоку API запитує лише коментарі, ймовірність повторного розбору вже проаналізованої публікації надзвичайно висока. Щоб запобігти таким випадкам додавання дублікатів публікацій, необхідно відстежувати всі додані публікації, а поточний ідентифікатор публікації порівнювати з цим списком. Якщо його немає у списку, його ідентифікатор публікації додається до списку, а публікація зберігається у відповідному CSV-файлі для подальшого аналізу. Потік Reddit розроблений для безперервної роботи 24 години на добу. Публікації та коментарі додаються до CSV-файлів, і раз на годину викликається допоміжний скрипт для обробки файлів.

Одна з цілей цього проєкту -визначити, чи можна створити прогнозовану модель для визначення руху цін на криптовалюти на основі обсягу та настроїв у соціальних мережах. Стрічки Twitter та Reddit можуть збирати лише найновіші дані та не матимуть достатньо даних для створення прогнозованої моделі. Тому історичні дані також необхідно збирати через Reddits API. Коли скрипт для збору історичних даних спочатку розроблявся, він використовував кінцеву точку, доступну через Reddits API, яка дозволяла прямі запити щодо публікацій у певному діапазоні дат. Це було дуже зручно, оскільки, наприклад, я міг зробити запит на всі публікації, опубліковані протягом 1 місяця.

На жаль, reddit видалив цю кінцеву точку наприкінці березня. Це стало серйозною перешкодою для розвитку цієї системи, оскільки на цьому етапі було зібрано недостатньо історичних даних. Мені пощастило знайти сторонній сервіс pushshift.io, який відстежував публікації на reddit за допомогою потокового API Reddit. Зрештою, я розробив обхідний шлях, щоб запитувати pushshift.io на наявність усіх ідентифікаторів публікацій у певному діапазоні дат, додавати всі ідентифікатори до списку, а потім запитувати кожну окрему публікацію безпосередньо до API reddit. Це дозволяло безперервно збирати історичні дані без

					БР.ІІІ – 67.00.00.000 ІЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

необхідності використання пошукового API Reddit. Хоча в цього рішення є недолік. Використовуючи оригінальний пошуковий API Reddit, публікації отримувалися пакетно. Використовуючи цей новий метод з pushshift API, публікації доводилося запитувати з reddit окремо, і з затримкою в 1 секунду між запитами це значно уповільнювало швидкість збору історичних даних.

```
for ID in post_ids:
    # Get reddit post
    submission = reddit.submission(id=ID)
    # Add post to the list
    posts.append(
        [submission.id, submission.title, submission.subreddit_name_prefixed,
         submission.created_utc, submission.score, submission.num_comments,
         submission.author, submission.shortlink])
    # Add post to csv file
    SavePosts(subreddit, posts)
    # reset post list
    posts[:] = []
    #If post does not have any comments skip it
    if submission.num_comments == 0:
        continue
    #If comments have alot of nested replies this will load them also
    submission.comments.replace_more(limit=0)
    #Add comments to list
    for comment in submission.comments.list():
        comments.append(
            [comment.id, submission.id,
             comment.body, comment.created_utc, comment.score, comment.author])
    # add comments to csv file
    SaveComments(subreddit, comments)
    # reset comment list to empty
    comments[:] = []
```

Рисунок 3.15 - Історичні дані Reddit

Основна логіка скрипта збору історичних даних дуже схожа на логіку потоку Reddit. На рисунку 3.15 показано цикл for у функції main, який перебирає ідентифікатори публікацій, запитує окрему публікацію на Reddit та додає відповідну інформацію з цієї публікації до списку, а потім зберігає цей список у файлі CSV для цього subreddit. Публікації перевіряються на наявність коментарів і також зберігаються у файлі CSV.

Класи Reddit Analyser та Twitter Analyser містять поточний зібраний набір даних, і виклик будь-якої функції в кожному класі поверне об'єкт json, що містить узагальнені дані, які потім можна додати до бази даних. RedditDB та TwitterDB – це два допоміжні скрипти для керування потоком як старих, так і нових даних до та з класів аналізатора Reddit та Twitter. Знову ж таки, RedditDB та TwitterDB

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

досить схожі, але функції в кожному з них адаптовані до типу даних, які обробляються з кожного веб-сайту. У головній функції кожного скрипта встановлюється підключення до бази даних MongoDB, створюється екземпляр відповідного класу аналізатора з наборами даних, і викликаються різні функції, що містяться в скриптах, для додавання узагальнених даних з класу аналізатора до бази даних.

На рисунку 3.16 показано одну з функцій RedditDB, яка отримує масив криптовалют та кількість згадок про них у цьому конкретному сабреддіті. Вищезгаданій функції передаються 3 аргументи: екземпляр Reddit Analyser, підключення до бази даних та назва сабреддіту, який аналізується. До бази даних запитуються наявність будь-яких документів, що стосуються поточного сабреддіту, і перевіряється, чи існує вже це поле. Якщо так, дані, що зараз знаходяться в цьому полі, передаються класу аналізатора, де вони об'єднуються з найновішими даними. Старі дані потім видаляються з документа, а об'єднані дані додаються. Якщо старих даних не існує, що може статися з меншими сабреддітами, які не отримують багато постів, то клас аналізатора повертає лише найновіші дані, які потім додаються до бази даних.

```
def SetCurrencyMentions(RA, db, subreddit):
    # Find subreddit document
    cursor = db.subreddits.find({"id": subreddit})
    for doc in cursor:
        # If already exists
        if "currency_mentions" in doc:
            # call reddit analyser function with old data
            cm = RA.CurrencyMentions(doc["currency_mentions"])
            # Remove old data from document
            db.subreddits.update(
                {"id": subreddit},
                {"$unset": {"currency_mentions": ""}})
            # Add new merged data
            db.subreddits.update_one(
                {"id": subreddit},
                {"$set": {"currency_mentions": cm}})
        # else it does not exist
        else:
            # call reddit analyser function without old data
            cm = RA.CurrencyMentions(None)
            # update document with summarised data
            db.subreddits.update_one(
                {"id": subreddit},
                {"$set": {"currency_mentions": cm}})
```

Рисунок 3.16 Встановлення згадок про валюту

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

Загалом у RedditDB та 9 у TwitterDB є 12 функцій, які запитують зведені дані від класу аналізатора, кожна з яких надає унікальне уявлення про заданий набір даних. Кожній функції в RedditDB та TwitterDB передаються ті самі аргументи, що й функції, показані на рисунку 3.16, і всі вони дотримуються тієї ж процедури перевірки старих даних перед запитом найновіших даних від класу аналізатора.

Найважливішими компонентами всієї системи є класи аналізаторів Reddit та Twitter. Вони обидва досить схожі, але знадобилося дві різні реалізації, щоб врахувати відмінності в структурі даних, а також тип інформації, яка мала бути вилучена з двох сайтів. Для цілей цього пояснення я посилаюся лише на аналізатор Reddit, але те, що я описую, стосується обох, і між цими двома класами є лише кілька логічних відмінностей.

Після запуску Reddit Analyser передаються 5 аргументів: зібрані публікації, зібрані коментарі, список криптовалют, список стоп-слів та список заблокованих користувачів. Для підготовки коментарів та публікацій до аналізу виконується функція очищення

```
def CleanseData(self, source, posts):
    #Change date format
    source["Date"] = pd.to_datetime(source["Date"],unit='s')
    #Create df object
    if posts:
        data = {"Author": source["Author"], "Text": source["Title"], "Date": source["Date"], "Score": source["Score"]}
    else:
        data = {"Author": source["Author"], "Text": source["Body"], "Date": source["Date"], "Score": source["Score"]}
    #Create df
    data = pd.DataFrame(data=data)
    #Convert datetime to date
    data["Date"] = data["Date"].dt.strftime('%Y-%m-%d %H:00:00')
    #Remove URLs
    data["Text"] = data["Text"].str.replace(r'http\S+', '', case=False)
    #Remove Na's
    data = data.dropna(how='any',axis=0)
    #Remove punctuation
    data["Text"] = data["Text"].str.replace('[^\w\s]','')
    #To lower case
    data['Text'] = data.Text.str.lower()
    #Remove Stop words
    stop = self.stopwords['word'].tolist()
    data["Text"] = data["Text"].apply(lambda x: ' '.join([word for word in x.split() if word not in stop]))
    with open(self.banned_path, "r") as jsonFile:
        users = json.load(jsonFile)
    data = data[~data['Author'].isin(users["users"])]
    return data
```

Рисунок 3.17 - Очищення даних

					БР.ІІ – 67.00.00.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

Наведений вище фрагмент коду на рисунку 3.17 показує функцію очищення, яка виконується для кожного екземпляра класу аналізатора. Основна мета цієї функції - перетворити дату у відповідний формат, який дозволяє легко групувати за годинами, видалити будь-які URL-адреси, розділові знаки та відсутні значення, а також перетворити весь текст у нижній регістр. Стоп-слова також видаляються з усього тексту на основі попередньо визначеного списку стоп-слів, що передаються аналізатору. Нарешті, користувачі, імена яких відображаються у файлі заблокованих користувачів, також видаляються з набору даних. Ці користувачі складаються виключно з облікових записів ботів, які не мають відношення до цього проекту та мали помітний негативний вплив на результати попереднього аналізу.

Після створення екземпляра Reddit Analyser можна викликати функції всередині класу для аналізу очищених даних та повернення зведених даних.

```
def SentimentByDay(self, oldsbd):
    # Copy of comments and posts
    cs = self.comments.copy()
    ps = self.posts.copy()
    # Create a new column with sentiment score for each piece of text
    cs['Comment_SA'] = np.array([ self.AnalyseSentiment(text) for text in cs['Text'] ])
    ps['Post_SA'] = np.array([ self.AnalyseSentiment(text) for text in ps['Text'] ])
    # Drop unneeded columns
    cs = cs.drop(['Author', 'Score', 'Text'], 1)
    ps = ps.drop(['Author', 'Score', 'Text'], 1)
    # Group and sum the scores by hour
    cs = cs.groupby('Date')['Comment_SA'].sum().reset_index()
    ps = ps.groupby('Date')['Post_SA'].sum().reset_index()
    # Merged the two dataframes, and drop and NA values
    sbd = pd.merge(cs, ps, on='Date', how='outer')
    sbd.fillna(0, inplace=True)
    sbd["Sentiment"] = sbd["Comment_SA"] + sbd["Post_SA"]
    # If old data
    if oldsbd != None:
        # Create data frame from records
        oldsbd = pd.DataFrame.from_records(data=oldsbd)
        # Merge old and new data
        oldnew_merged = pd.concat([sbd, oldsbd])
        # Group by hour and sum sentiment scores
        oldnew_merged = oldnew_merged.groupby('Date').sum().reset_index()
        # Convert dataframe to json
        oldnew_merged = oldnew_merged.to_json(orient='records', date_format=None)
        oldnew_merged= json.loads(oldnew_merged)
        # Return json array
        return oldnew_merged
    # Convert dataframe to json
    sbd = sbd.to_json(orient='records', date_format=None)
    # Return json array
    return json.loads(sbd)
```

Рисунок 3.18 - Настрої за днями

Прикладом однієї з функцій є функція SentimentByDay на рисунку 3.18, яка

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

обчислює погодинний настрій для всіх постів і коментарів, переданих класу під час створення екземпляра. Лексикон аналізу AFINN використовується для присвоєння словам оцінки від -5 до +5, причому позитивні оцінки вказують на позитивний настрій і навпаки. Пости та коментарі групуються за годинами та підсумовуються, щоб отримати оцінку настрою за цю годину. Потім два фрейми даних об'єднуються та повертаються. Якщо попередні дані про настрій за днями вже існують для даного subreddit, старі дані також можуть бути передані функції як аргумент. Старі дані потім об'єднуються з поточними даними, згрупованими за годинами і знову підсумовуються, щоб отримати нові оновлені значення. Усі функції в класі Reddit Analyser приймають 1 або більше аргументів, щоб дозволити об'єднання старих даних з поточними.

Після налаштування системи аналізу та збору достатньої кількості даних настав час створити деякі прогнознi моделі, щоб визначити, чи можна передбачити рух цін на криптовалюти на основі даних соціальних мереж. Протягом розробки системи аналізу виникло чимало проблем. По-перше, twitter.com не надає простого способу збору історичних твітів. Тому для цієї моделі прогнозування мені довелося покладатися виключно на дані Reddit. Reddit також вирішив трохи попрацювати та посеред цього проєкту видалив важливий API для збору історичних даних на своєму сайті. Щоб обійти цю проблему, мені довелося покладатися на сторонній сервіс для збору ідентифікаторів історичних постів, а потім запитувати API Reddit для кожного посту окремо. Цей метод був у 50 разів повільнішим за попередній, оскільки я міг запитувати лише 1 пост за секунду. Процес збору даних був набагато повільнішим, ніж спочатку передбачалося, тому для розробки підтвердження концепції для прогнозної моделі було налаштовано окремі скрипти поза системою аналізу для збору лише публікацій, пов'язаних з біткойном, тоді як основна система аналізу продовжує працювати з понад 200 валютами. Це дозволило мені отримати 18 місяців публікацій з Reddit, пов'язаних з біткойном.

Передбачувані моделі були створені в RStudio за допомогою R та бібліотеки каретки. Набір даних був попередньо очищений за допомогою систем

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

аналізу. Єдиним необхідним перетворенням було об'єднання історичних цін на біткойн, історичних згадок та настроїв щодо біткойна в один набір даних та створення нового стовпця, який би визначав рух ціни біткойна. Наприклад, якщо поточного дня ціна вища, ніж попереднього дня, стовпець руху ціни містив би значення "вище" для поточного дня. Цей стовпець був залежною змінною для всіх моделей. У наборі даних було загалом 7 незалежних змінних: Закриття, Максимум, Мінімум, Відкриття, Обсяг, Згадки та Настрої. Закриття, Максимум, Мінімум та відкриття – усі

пов'язані з ціною біткойна в певний момент дня. Обсяг – це сума торгів у доларах США, яку було продано біткойном у певний день, а згадки та настрої – це кількість згаданих біткойнів на Reddit та настрої в цей день.

```
# Set seed
set.seed(1234)
#Splitting data as training and test set. Using createDataPartition() function from caret
indxTrain <- createDataPartition(y = bit_forecast$Diff, p = 0.8,list = FALSE)
training <- bit_forecast[indxTrain,]
testing <- bit_forecast[-indxTrain,]
#Checking distribution in origanl data and partitioned data
prop.table(table(training$Diff)) * 100
prop.table(table(bit_forecast$Diff)) * 100
```

Рисунок 3.19 - Розділ даних

На рисунку 3.19 показано перший крок перед створенням прогностичних моделей. Набір даних потрібно було розділити на два окремі набори даних: один для навчання моделі, а інший для тестування. Бібліотека карет надає корисний функціонал для розділення даних, який було реалізовано для розділення набору даних у співвідношенні 80:20. 80% набору даних використовувалося для навчання, а 20% залишалося для тестування. Після розділення наборів даних розподіл навчального набору даних порівнювався з вихідним набором даних, який не видав жодних невідповідностей.

Першою створеною моделлю була модель випадкового лісу. Випадковий ліс можна ефективно розглядати як велику кількість дерев рішень, які допомагають приймати рішення на основі класифікації. Значення *mtry* , яке є

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

випадково вибраним предиктором, використовується для визначення того, яка ітерація може бути використана для отримання найточнішого зчитування класифікації. Ця модель є надзвичайно точною.

```
set.seed(1234)
ctrl <- trainControl(method="repeatedcv", repeats = 3)
RFFit <- train(Diff ~ ., data = training,
               method = "rf", trControl = ctrl,
               preProcess = c("center", "scale"), tuneLength = 20)
RFPredict <- predict(RFFit, newdata = testing)
#Get the confusion matrix to see accuracy value and other parameter values
confusionMatrix(RFPredict, testing$Diff)
```

Рисунок 3.20 - Випадковий ліс - код R

Наведений вище фрагмент на рисунку 3.20 показує код R, який використовується для реалізації прогностичної моделі випадкового лісу. Змінна ctrl контролює кількість наборів перехресної перевірки k-кратного розміру. RFFit – це навчена модель, яка потім використовується для порівняння з тестовим набором даних на точність. Функція confusionMatrix створює перехресну таблицю спостережуваних та передбачуваних класів з відповідною статистикою.

```

      Reference
Prediction Higher Lower
Higher         9      15
Lower        20      29

Accuracy : 0.5205
 95% CI : (0.4004, 0.639)
No Information Rate : 0.6027
P-Value [Acc > NIR] : 0.9389

Kappa : -0.0315
McNemar's Test P-Value : 0.4990

Sensitivity : 0.3103
Specificity : 0.6591
Pos Pred Value : 0.3750
Neg Pred Value : 0.5918
Prevalence : 0.3973
Detection Rate : 0.1233
Detection Prevalence : 0.3288
Balanced Accuracy : 0.4847

'Positive' Class : Higher
```

Рисунок 3.21 - Матриця плутанини випадкового лісу.

На рисунку 3.21 показано результати прогностичної моделі випадкового лісу, згенерованої за допомогою матриці плутанини. Точність моделі становить лише 52%, що лише трохи краще, ніж підкидання монети. Коефіцієнт каппа дуже малий, -0,03, що вказує на не дуже надійність цієї моделі. Зі спостережень за перехресним табулюванням Спостережувані та очікувані значення, схоже, що модель значно гірше прогнозує зростання ціни порівняно з її зниженням. Модель випадкового лісу демонструє низьку передбачуваність цін на криптовалюту на основі згадок та настроїв у соціальних мережах.

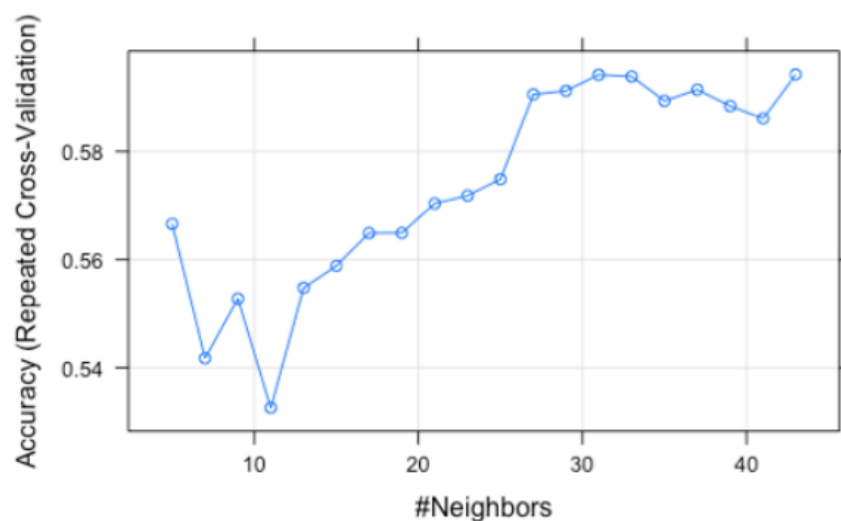


Рисунок 3.22 - Графік апроксимації методом KNN

Другим реалізованим алгоритмом був k-найближчих сусідів. Більш відомий як KNN, це також алгоритм контрольованої класифікації, який використовує значення k для класифікації подібних значень на основі значення k та виведення прогнозу. Значення k можна вибрати багатьма різними способами, наприклад, k може бути числом квадратного кореня з кількості екземплярів у наборі даних. Отже, якщо у нас є 100 екземплярів, значення k буде 10. Загальне використання k не має суттєвого значення, але його можна використовувати для підвищення ефективності моделі, якщо вибрати правильний. Алгоритм Knn був реалізований майже так само, як і Random forest. Усі алгоритми були реалізовані за допомогою бібліотеки каретки, тому код був дуже схожим.

Графік, показаний на рисунку 3.22, є результатом апроксимації KNN. З к-значенням 42 модель забезпечила найвищу прогностичну точність близько 60%.

Точність цієї моделі, здається, вища, ніж у випадкового лісу, але вона все ще лише трохи краща, ніж прогнозування лише випадковим чином.

```

Reference
Prediction Higher Lower
Higher      4      3
Lower     25     41

Accuracy : 0.6164
95% CI : (0.4952, 0.7279)
No Information Rate : 0.6027
P-Value [Acc > NIR] : 0.4557

Kappa : 0.0801
McNemar's Test P-Value : 7.229e-05

Sensitivity : 0.13793
Specificity : 0.93182
Pos Pred Value : 0.57143
Neg Pred Value : 0.62121
Prevalence : 0.39726
Detection Rate : 0.05479
Detection Prevalence : 0.09589
Balanced Accuracy : 0.53487

'Positive' Class : Higher

```

Рисунок 3.23 - Матриця плутанини для KNN

Аналізуючи матрицю плутанини, сформовану для моделі KNN (див. рисунок 3.23), можна побачити, що вона стикається з тими ж проблемами, що й модель «випадкового лісу». Зі специфічністю 0,93 модель правильно передбачила майже всі випадки зниження цін, але майже не вгадала жодного випадку їхнього зростання. Точність моделі становить 61%, але вона надзвичайно схильна до прогнозування зниження ціни, порівняно з її зростанням.

Основними цілями цього проєкту були розробка веб-застосунку, який би надавав уявлення про те, як сприймаються криптовалюти та скільки про них говорять у соціальних мережах, а також створення прогностичних моделей з використанням зібраних даних для веб-застосунку. Результати аналізу прогностичних моделей показують низьку прогностичну здатність, причому реалізація KNN показала найкращі результати з точністю 61%. Моделі були сильно упереджені в бік класифікації ціни як такої, що знижується, що могло бути

пов'язано з дисбалансом класів. Результати тестування клієнтами веб-застосунку були досить позитивними. Більшість респондентів вважали, що цей застосунок корисний для оцінки криптовалюти. Ринок криптовалют молодий, волатильний і базується на чистих спекуляціях. Хоча моделі показали слабку прогностичну здатність, я все ще вважаю, що цей тип сайту не тільки цікавий, але й надзвичайно корисний для трейдерів.

3.4 Висновок по розділу

У третьому розділі виконано реалізацію інтелектуального веб-застосунку для аналізу криптовалютного ринку з використанням сучасних програмних засобів і технологій веб-розробки. Розроблено графічний інтерфейс користувача, який забезпечує зручний доступ до аналітичної інформації та результатів обробки даних. Проведено аналіз роботи системи та оцінено її ефективність за показниками функціональності, швидкодії та зручності використання. Отримані результати підтвердили доцільність застосування розробленого веб-застосунку для моніторингу, аналізу та підтримки прийняття рішень на криптовалютному ринку.

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході виконання бакалаврської роботи було розроблено інтелектуальний веб-застосунок для аналізу криптовалютного ринку, побудований із використанням сучасних веб-технологій та інструментів розробки. Створена система забезпечує збір, обробку, аналіз і візуалізацію даних про криптовалютні активи, надаючи користувачам зручний інструмент для моніторингу ринкових тенденцій та підтримки прийняття інвестиційних рішень. У роботі було проаналізовано особливості функціонування криптовалютного ринку, досліджено сучасні підходи до створення аналітичних веб-систем, а також реалізовано програмне рішення, яке відповідає сучасним вимогам щодо продуктивності, надійності та зручності використання.

На початковому етапі дослідження було виконано аналіз предметної області та визначено особливості криптовалютного ринку як об'єкта дослідження. Розглянуто основні поняття криптовалют, принципи функціонування цифрових активів та існуючі платформи для моніторингу ринкових показників.

У процесі проєктування системи було сформульовано функціональні та нефункціональні вимоги до веб-застосунку. Особлива увага приділялася забезпеченню швидкого доступу до актуальної ринкової інформації, можливості аналізу змін курсів криптовалют у режимі реального часу, побудові інтерактивних графіків та забезпеченню високого рівня продуктивності. Також було розроблено архітектуру системи, яка забезпечує модульність, масштабованість та можливість подальшого розширення функціоналу.

Практична реалізація роботи охопила створення програмних модулів для отримання та обробки ринкових даних, реалізацію механізмів візуалізації статистичних показників та розроблення сучасного графічного інтерфейсу користувача. Інтерфейс було спроектовано таким чином, щоб забезпечити зручну навігацію, швидкий доступ до аналітичної інформації та комфортну взаємодію користувача із системою. Для відображення ринкових тенденцій

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

використовуються графіки, таблиці та інші засоби візуалізації, які сприяють кращому сприйняттю інформації.

Важливим етапом роботи стала перевірка функціонування розробленого веб-застосунку та аналіз результатів його використання. Проведене тестування підтвердило коректність роботи основних програмних модулів, стабільність функціонування системи та відповідність реалізованих можливостей визначеним вимогам. Отримані результати показали, що веб-застосунок забезпечує ефективний аналіз криптовалютного ринку та дозволяє оперативно отримувати актуальну інформацію щодо змін вартості цифрових активів.

Розроблений інтелектуальний веб-застосунок успішно виконує поставлені завдання та може використовуватися як інформаційно-аналітичний інструмент для моніторингу криптовалютного ринку. До основних переваг створеного рішення належать зручний інтерфейс користувача, можливість роботи з актуальними ринковими даними, наочна візуалізація інформації та модульна архітектура, яка дозволяє легко розширювати функціональні можливості системи.

Перспективами подальшого розвитку проекту є впровадження алгоритмів машинного навчання для прогнозування зміни курсів криптовалют, інтеграція додаткових джерел фінансових даних, розроблення системи персоналізованих рекомендацій для користувачів, автоматичне формування аналітичних звітів, а також реалізація мобільної версії веб-застосунку. Подальше вдосконалення цих компонентів дозволить підвищити точність аналізу, розширити функціональність системи та збільшити її практичну цінність для користувачів.

Отже, поставлена мета бакалаврської роботи досягнута, а всі визначені завдання виконані. Результатом роботи є інтелектуальний веб-застосунок для аналізу криптовалютного ринку, який забезпечує ефективну обробку та візуалізацію даних, підтримує процес прийняття рішень і може бути використаний як основа для подальшого розвитку сучасних аналітичних фінансових систем.

					БР.ІІ – 67.00.00.000 ІЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Angular Documentation. (2026). angular.dev. <https://angular.dev>
2. ASP.NET Core Documentation. (2026). Microsoft Learn. <https://learn.microsoft.com/aspnet/core>
3. Binance API Documentation. (2026). developers.binance.com. <https://developers.binance.com/docs>
4. Blockchain.com. (2026). Blockchain Explorer Documentation. <https://www.blockchain.com/explorer>
5. Bootstrap Documentation. (2026). getbootstrap.com. <https://getbootstrap.com/docs>
6. Buterin, V. (2014). A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum White Paper. <https://ethereum.org/en/whitepaper>
7. CoinGecko API Documentation. (2026). coingecko.com. <https://www.coingecko.com/en/api/documentation>
8. CoinMarketCap API Documentation. (2026). coinmarketcap.com. <https://coinmarketcap.com/api>
9. C# Documentation. (2026). Microsoft Learn. <https://learn.microsoft.com/dotnet/csharp>
10. Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley. <https://martinfowler.com/eaCatalog>
11. Freeman, E., & Robson, E. (2021). Head First Design Patterns (2nd ed.). O'Reilly Media. <https://www.oreilly.com>
12. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
13. Google Charts Documentation. (2026). developers.google.com. <https://developers.google.com/chart>
14. Grinberg, M. (2023). REST APIs with ASP.NET Core. O'Reilly Media. <https://www.oreilly.com>

					БР.ІІІ – 67.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		82

15. ISO/IEC 12207:2017. Systems and Software Engineering — Software Life Cycle Processes. ISO. <https://www.iso.org>
16. ISO/IEC 25010:2023. Systems and Software Quality Requirements and Evaluation (SQuaRE). ISO. <https://www.iso.org>
17. ISO/IEC 27001:2022. Information Security Management Systems. ISO. <https://www.iso.org>
18. JavaScript Documentation. (2026). MDN Web Docs. <https://developer.mozilla.org>
19. Kaspersky. (2025). Cryptocurrency Security and Cyber Threats. <https://www.kaspersky.com>
20. Martin, R. C. (2017). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Pearson Education.
21. Microsoft Azure Documentation. (2026). learn.microsoft.com. <https://learn.microsoft.com/azure>
22. MongoDB Documentation. (2026). mongodb.com. <https://www.mongodb.com/docs>
23. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. <https://bitcoin.org/bitcoin.pdf>
24. NIST. (2024). Blockchain Technology Overview. National Institute of Standards and Technology. <https://www.nist.gov>
25. OpenAPI Specification. (2026). Swagger. <https://swagger.io/specification>
26. OWASP Top 10 Web Application Security Risks. (2025). OWASP Foundation. <https://owasp.org/www-project-top-ten>
27. Osherove, R. (2019). The Art of Unit Testing (3rd ed.). Manning Publications.
28. PostgreSQL Documentation. (2026). postgresql.org. <https://www.postgresql.org/docs>
29. Postman API Platform Documentation. (2026). learning.postman.com. <https://learning.postman.com>
30. React Documentation. (2026). react.dev. <https://react.dev>

					БР.ІІІ – 67.00.00.000 ІІЗ	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

31. Richardson, C. (2021). Microservices Patterns. Manning Publications.
<https://www.manning.com/books/microservices-patterns>
32. Sass Documentation. (2026). sass-lang.com. <https://sass-lang.com/documentation>
33. Selenium Documentation. (2026). selenium.dev.
<https://www.selenium.dev/documentation>
34. Statista. (2025). Cryptocurrency Market Insights. <https://www.statista.com>
35. Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. First Monday Journal. <https://firstmonday.org>
36. TradingView Documentation. (2026). tradingview.com.
<https://www.tradingview.com/charting-library-docs>
37. Van Rossum, G., & Drake, F. (2024). Python Documentation.
<https://docs.python.org>
38. Verma, R., & Gupta, S. (2024). Machine Learning Techniques for Cryptocurrency Price Prediction. International Journal of Financial Technology, 18(2), 115–129.
39. Wang, J., & Li, X. (2025). Web-Based Cryptocurrency Analytics Systems: Architecture and Performance Evaluation. ACM Transactions on Web, 19(4), 1–22.
<https://doi.org/10.1145/3789123>
40. Zhao, Y., & Chen, L. (2024). Real-Time Data Processing for Cryptocurrency Market Analysis. IEEE Access, 12, 45871–45886.
<https://doi.org/10.1109/ACCESS.2024.3378456>

БІБЛІОГРАФІЧНА ДОВІДКА

Тема бакалаврської роботи: "Розроблення інтелектуального веб-застосунку для аналізу криптовалютного ринку "

Обсяг пояснювальної записки: 84 аркушів

Дата закінчення бакалаврської роботи 10 червня 2026р.

Підпис студента _____