

<https://doi.org/10.31891/2307-5732-2025-359-1>
УДК 004.4

АЗАРОВ МАКСИМ

Львівський національний університет імені Івана Франка
<https://orcid.org/0009-0003-5261-0468>
e-mail: maksym.azarov@lnu.edu.ua

ЛЯШКЕВИЧ ВАСИЛЬ

Львівський національний університет імені Івана Франка
<https://orcid.org/0000-0003-2810-6061>
e-mail: vasyliashkevych@lnu.edu.ua

ВИКОРИСТАННЯ МОЖЛИВОСТЕЙ JAVASCRIPT ПРИ ПОБУДОВІ ПРОГРАМНИХ ІНСТРУМЕНТІВ ДЛЯ ФІНАНСОВИХ ПОМІЧНИКІВ

Дана робота присвячена розробці та впровадженню інтелектуальної системи для ефективного управління грошовими активами користувачів. Основною метою роботи є створення багатофункціонального інструмента, який допоможе користувачам ефективніше управляти як криптовалютними, так і фіатними активами. Платформа пропонує рекомендації в реальному часі на основі автоматизованого аналізу ринкових даних та інформації з гаманців користувачів. Також система підтримує інтеграцію з криптовалютними гаманцями, Monobank API та іншими фінансовими інструментами, що дозволяє користувачам контролювати всі активи в єдиному інтерфейсі. У реалізації використано API Binance для отримання актуальних ринкових даних у режимі реального часу, CoinGecko API для збору додаткової аналітики та історичних даних, Infura для безпечного та зручного читання даних з криптовалютних гаманців, а також Monobank API для моніторингу банківських рахунків і транзакцій.

Особливістю платформи є використання натренованої моделі машинного навчання, яка прогнозує зростання або падіння обраної криптовалюти, спираючись на історичні та поточні ринкові дані. Це дає змогу формувати персоналізовані інвестиційні рекомендації та своєчасно попереджати користувача про потенційні зміни ринку.

Розроблене рішення вирізняється можливістю об'єднання традиційних банківських рахунків і криптовалютних активів у єдиній платформі з простим та інтуїтивним інтерфейсом. Окрім оптимізації процесів управління фінансами, система спрямована на підвищення фінансової грамотності користувачів, надаючи аналітичні дані та поради, що допомагають зменшити інвестиційні ризики й підвищити дохідність портфеля.

Ключові слова: фінансові пропозиції, прийняття рішень, великі мовні моделі, машинне навчання, фінансові інструменти, програмні моделі.

AZAROV MAKSYM

LYASHKEVYCH VASYL

Ivan Franko National University of Lviv

USING JAVASCRIPT CAPABILITIES IN BUILDING SOFTWARE TOOLS FOR FINANCIAL ASSISTANTS

This work is dedicated to the development and implementation of an intelligent system for the effective management of own financial assets. There are a lot of different financial assistants on the market, but it is hard to trust them because, as usual, there is no scientific background behind. A real financial decision-making process is complex and can involve different techniques in parallel to guarantee the best decision. In the meantime, the applicability of AI technology is increasingly growing, proposing new opportunities for investigation. The main goal of the project is the creation of a multifunctional tool that helps users manage both cryptocurrency and fiat assets more efficiently. The platform provides real-time recommendations based on automated analysis of market data and user wallet information. Additionally, the system supports integration with cryptocurrency wallets, the Monobank API, and other financial tools, enabling users to manage all their assets within a single interface. In the implementation, Binance API is used to obtain real-time market data, CoinGecko API to gather additional analytics and historical information, Infura for secure and convenient reading of data from cryptocurrency wallets, and Monobank API to monitor bank accounts and transactions.

A distinctive feature of the platform is the use of a trained machine learning model that predicts the growth or decline of a selected cryptocurrency based on historical and current market data. This enables the generation of personalised investment recommendations and timely alerts about potential market changes.

The developed solution stands out for its ability to combine traditional bank accounts and cryptocurrency assets into a single platform with a simple and intuitive interface. In addition to optimising financial management processes, the system aims to improve users' financial literacy by providing analytical data and advice that help reduce investment risks and increase portfolio profitability.

Keywords: financial recommendations, decision-making, large language models, machine learning, financial instruments, software models.

Стаття надійшла до редакції / Received 03.10.2025

Прийнята до друку / Accepted 15.11.2025

Постановка проблеми у загальному вигляді

та її зв'язок із важливими науковими чи практичними завданнями

В умовах сучасної глобалізованої економіки управління фінансовими активами стає все більш складнішим завданням через динамічність ринкових змін, нестабільність валют та розвиток нових фінансових інструментів, зокрема криптовалют. Зростання популярності криптовалют, а також поширення технологій блокчейну значно змінили традиційні підходи до інвестування та управління капіталом. Це створює нові можливості для індивідуальних інвесторів, але водночас вимагає глибшого аналізу і швидких рішень, які часто виходять за межі людських можливостей через великий обсяг інформації та постійні коливання ринку.

Завдяки впровадженню машинного навчання, фінансові платформи мають можливість надавати

рекомендації щодо купівлі та продажу активів, що дозволяє мінімізувати ризики та підвищити дохідність портфеля користувача. Проте, такі платформи не надають відповідних рекомендацій у вільному доступі, що пов'язано з багатьма правилами, які регулюють діяльність у фінансовій сфері. Також, не кожна платформа підтримує інтеграцію криптогаманців і банківських рахунків через API банку, що дозволяє користувачам отримувати актуальні дані про свої фінансові операції в єдиному інтерфейсі.

З огляду на ці виклики, з'являється потреба в інтелектуальних фінансових помічниках, які можуть автоматизувати процес збору, аналізу та обробки ринкових даних для надання рекомендацій щодо управління активами. Такі системи можуть допомогти користувачам ефективніше управляти своїм капіталом, зменшуючи ризики та збільшуючи дохідність інвестицій за рахунок своєчасних та обґрунтованих рішень. Такі платформи могли б надати користувачам зручний і ефективний доступ до управління своїми фінансами як об'єднання традиційних банківських рахунків та криптовалютних гаманців, що дозволить користувачам легко відслідковувати фінансові активи в реальному часі.

Аналіз досліджень та публікацій

Грошовий капітал - це ресурси, що перебувають у грошовій формі і використовуються для створення прибутку, інвестицій та забезпечення фінансових потреб. Важливим аспектом грошового капіталу є його здатність трансформуватися в інші форми активів (наприклад, матеріальні чи фінансові) залежно від інвестиційних потреб і стратегій. [1]

Також слід звернути увагу, що є два типи криптовалютних гаманців [2]:

- Кастодіальний криптовалютний гаманець – це гаманець, в якому приватні ключі зберігаються третьою стороною. Це означає, що третя сторона буде зберігати та керувати вашими приватними ключами від вашого імені, тобто, у користувача не буде ні повного контролю над своїми коштами, ні можливості підписувати транзакції.

- Некастодіальний гаманець – це гаманець, в якому тільки власник володіє і контролює приватні ключі. Цей варіант найкраще підходить для користувачів, яким важливо мати повний контроль над своїми коштами. Оскільки посередників немає, можна торгувати криптовалютою прямо зі своїх гаманців - варіант для досвідчених трейдерів та інвесторів, які знають, як керувати та захищати свої приватні ключі і seed фрази.

Фінансовий помічник - це цифровий інструмент або застосунок, який допомагає користувачам керувати грошовими ресурсами, відстежувати витрати, планувати бюджет та здійснювати інвестиції. Він може включати поради з управління активами, аналітику та інструменти для відстеження ринкових трендів. [3]

Основні функції фінансових помічників:

- аналіз витрат і доходів. Автоматизоване відстеження транзакцій і генерація звітів, що допомагає бачити загальну картину фінансів;
- бюджетування. Створення індивідуальних бюджетів для контролю витрат у різних категоріях, що сприяє ефективному управлінню грошима;
- планування заощаджень і інвестицій. Дозволяє встановлювати цілі заощаджень, відстежувати їхній прогрес та пропонувати шляхи інвестування коштів для їхнього збільшення;
- інтеграція з банківськими та інвестиційними акаунтами. Завдяки доступу до реальних фінансових даних, такі помічники можуть автоматично отримувати інформацію з різних фінансових джерел і відображати їх в одному інтерфейсі.

На сьогодні, великі мовні моделі (LLM) набули широкого використання як джерело знань для сучасних фінансових помічників. Розрізняють багато різних LLM, але GPT, розробленою компанією OpenAI, є найбільш популярною з них. На основі GPT моделей, OpenAI розробила ChatGPT - генеративний чат-бот, який призначений для ведення природних розмов, розуміння складних питань і генерації детальних, послідовних відповідей практично на будь-яку тему. [4]

Компанія Google розробила свою модель Gemini. У ширшому сенсі це сімейство мультимодальних моделей штучного інтелекту, призначених для обробки різних типів даних, включаючи аудіо, зображення, програмний код, текст і відео. [5]

Claude AI (Claude) - це чат-бот на основі сімейства LLM, розроблених дослідницькою компанією Anthropic. Claude вирізняється високою ефективністю в обробці природної мови (NLP) та є мультимодальним: він приймає текстові, аудіо- та візуальні вхідні дані і може відповідати на запитання, узагальнювати документи та генерувати довгі тексти, діаграми, анімації, програмний код тощо. [6]

В статті "A First Look at Financial Data Analysis Using ChatGPT-4o" автор дослідив застосування ChatGPT-4o для таких завдань, як zero-shot prompting, аналіз часових рядів, аналіз ризику та дохідності, а також оцінка ARMA-GARCH, показавши її ефективність, порівнянню з традиційним програмним забезпеченням на кшталт Stata. Проте, дослідження також виявило невирішені питання, зокрема потребу в детальному аналізі конкретних помилок моделі, відсутність чітких методологій для людського нагляду та валідації результатів, обмеження моделі у прийнятті рішень в реальному часі [7].

Формулювання цілей статті

Метою роботи є: створення багатофункціонального інструменту-платформи, який допоможе користувачам ефективніше управляти як криптовалютними, так і фіатними активами. Платформа повинна пропонувати рекомендації в реальному часі на основі автоматизованого аналізу ринкових даних та інформації

з гарантів користувачів.

Об'єктом дослідження є програмні засоби для управління фінансовими ресурсами, що використовуються для моніторингу, аналізу, оптимізації та прийняття рішень щодо керування грошовим капіталом в інтелектуальних інформаційних системах.

Предмет дослідження: алгоритми, методи та інші процеси прийняття рішення щодо керування грошовим капіталом.

Отже, завдання та цілі:

- дослідження існуючих засобів керування грошовим капіталом;
 - розроблення програмних моделей фінансових помічників для інтеграції з існуючими фінансовими сервісами і платформами;
 - розробити узагальнену модель фінансового помічника на основі аналізу користувачів;
 - розробити архітектуру та інтерфейс користувача для фінансового помічника;
- надати рекомендації для вибраної категорії людей.

Виклад основного матеріалу

В процесі дослідження програмних технологій для побудови нашої платформи було обрано три найближчі аналоги: "F.I.R.E on track", "Wally: AI Personal Finance" та "Magnifi" – в таблиці 1.

Таблиця 1

Результати порівняння фінансових помічників

Функції помічника	F.I.R.E on track	Wally: AI Personal Finance	Magnifi
Chatbot	+	+	+
Графіки	+	+	+
Банківські акаунти	-	+	-
Криптогаманці	-	-	-
IOS	-	-	-
Android	-	+	-
Web app	+	-	+
Криптопоради	-	-	-
Платна	-	-	+
Інвестиції	-	-	+

"F.I.R.E on track" - це платформа, що допомагає користувачам досягти фінансової незалежності та раннього виходу на пенсію (FIRE). Вона пропонує персоналізовані фінансові інсайти, відстеження цілей, нагадування про прогрес та AI-помічника FIRE Mate. Користувачі можуть встановлювати цілі заощаджень, контролювати доходи і витрати, а також тестувати фінансові сценарії через зручну панель. Платформа також підтримує масовий імпорт даних і обіцяє майбутню підтримку спільноти. [8]

Wally - ваш особистий помічник з фінансів, який спрощує керування грошима. Легко відстежуйте витрати, створюйте бюджети та дотримуйтеся їх, а також отримуйте цінні інсайти щодо своїх фінансових звичок. Додаток може під'єднуватися до банківських рахунків, щоб автоматично відстежувати витрати, а також надає аналітику щодо фінансових тенденцій. Wally також підтримує можливість створення спільних бюджетів для родин чи партнерів, що спрощує координацію спільних витрат. [9]

"Magnifi" - це III, розроблений, щоб допомогти інвестувати - насправді перший у світі. З Magnifi ми можемо швидше й краще досліджувати, отримувати допомогу в прийнятті рішень, планувати цілі, керувати своїм портфелем і вчитися, інвестуючи. Magnifi допомагає знайти та купити інвестиції на ринку з понад 15 000 акцій і фондів, проаналізувати наявні інвестиції в інших брокерських компаніях. [10]

Magnifi є експертом з інвестування зі III, системою пошуку інвестицій, даних за запитом, інтерактивних інструментів планування та брокерських послуг без комісії. Ми можемо інвестувати безпосередньо через Magnifi, пов'язати зовнішні інвестиційні рахунки, такі як E*Trade, Robinhood або Charles Schwab для Magnifi. Передплатники Magnifi також мають доступ до високоприбуткових ощадних рахунків, які допоможуть заробляти відсотки на неінвестовані кошти, поки інвестор чекає на інвестування.

Сьогодні важко уявити будь-якого помічника без використання LLM. Такі моделі забезпечують не тільки зручний інтерфейс для комунікації з користувачем, але і формують необхідні пояснення щодо ситуації, тенденції та ін. Завдяки високій популярності у застосуванні, на сьогодні, запропоновано дуже велику кількість LLM від провідних IT компаній, але їх призначення може бути різним. Насправді, дуже складно вибрати найкращу модель без правильного тестового сету і відповідного середовища для тестування. Саме тому, в даній статті, ми зосередилися на програмних технологіях побудови фінансових помічників, щоб в подальшому зробити повноцінне дослідження LLM та вибір найкращої.

Для інтеграції в систему було обрано ChatGPT, як одного з найпопулярніших представників LLM і таким, що викликає довіру у більшості користувачів. Модель має зручний API та бібліотеки від OpenAI, які підтримують швидку інтеграцію, багатомовність і безплатні запити для тестування. У порівнянні з альтернативами на кшталт Gemini чи Claude, ChatGPT є легшим у налаштуванні та використанні, зокрема в контексті fine-tuning, що надає йому перевагу в практичному застосуванні.

Попри високу доступність, гнучкість та зручність використання ChatGPT, модель має фундаментальні

обмеження, які не дозволяють покладатися на неї для прийняття оптимальних фінансових рішень. У поточному вигляді вона може виступати лише як асистент, а не як повноцінна система прийняття рішень. Модель GPT-4o, що була обрана для наукових досліджень, демонструє значний потенціал у сфері фінансового аналізу, проте має низку критичних недоліків. Зокрема, модель може допускати помилки у валідації фінансових даних, що здатне призвести до хибних висновків під час прийняття рішень. Також зафіксовані труднощі у моделюванні складних часових рядів, зокрема таких, як ARMA-GARCH, що є особливо важливим для точного фінансового прогнозування.

Насправді, багато людей очікують, що ШІ візьме на себе справу та зробить інвестування за них. Незалежно від того, який додаток чи платформа використовується для пошуку нових інвестицій, дослідження можливостей, порівняння акцій і фондів чи створення й керування повним портфелем - ШІ краще допоможе купувати інвестиції з персоналізованою інформацією, а ніж це зробить людина.

Для машинного навчання обрано бібліотеку Brain.js, яка використовується для створення нейронних мереж, що здатні навчатись на базі історичних даних і робити прогнози – в таблиці 2. Даний пакет є простим у використанні, легко інтегрується з JavaScript і також дає можливість тренувати моделі прямо у веб-додатках без потреби у серверній інфраструктурі.

Таблиця 2

Результати порівняння бібліотек машинного навчання

	Brain.js	Tensorflow.js	ML5.js
Основний фокус	Прості нейронні мережі на JavaScript	Глибоке навчання, складні моделі	Доступне ML для початківців
Вага бібліотеки	4.55mb	147mb	4.55mb
Кількість скачувань в npm (за тиждень)	3200	112345	7500
Випадки використання	Прості прогнози, навчальні проекти	Складні моделі, висока продуктивність	Проекти для початківців, креативне ML

Для прогнозування майбутніх значень використовується Brain.js модель, з певними налаштуваннями. Функція активації leaky-relu, яка допомагає краще передати інформацію під час навчання з коефіцієнтом “пропуску” негативних значень 0.001. Параметр binaryThresh проставлений як 0.5, що досить зручно для такої типи задачі.

Використання React як UI бібліотеки має кілька переваг, які роблять його привабливим вибором для веб-розробки: компонентний підхід, віртуальний DOM, JSX, велика екосистема.

У якості ORM (Object-Relational Mapping) для бази даних було вибрано бібліотеку TypeORM, яка дозволяє зручно працювати з базами даних у Nest.js – в таблиці 3. Вона забезпечує зручне визначення моделей-сутностей з використанням TypeScript-декораторів, що робить код більш структурованим і читабельним. Також передбачена можливість генерувати міграції за допомогою CLI. TypeORM пропонує більш традиційний підхід ORM, дозволяючи розробникам визначати моделі сутностей за допомогою декораторів і забезпечуючи детальний контроль над запитамі SQL і зв'язками бази даних. Він краще підходить для тих, хто надає перевагу тісній роботі зі структурою бази даних і обробляє складні міграції.

Таблиця 3

Результати порівняння різних ORM бібліотек для Node.js

	TypeORM	Prisma	Sequelize
Опис	Найпопулярніша ORM для TypeScript і NestJS. Підтримує декоратори та працює з реляційними БД.	ORM нового покоління з генерацією коду, підходить для складних структур даних.	ORM для Node.js з підтримкою SQL- та NoSQL-баз, добре підходить для більшості баз даних.
Особливості	Декоратори, міграції, підтримка PostgreSQL, MySQL та ін.	Автогенерація, типізація, підтримка кількох баз даних	Моделі класи, інтеграція з різними базами
Переваги	Легка інтеграція з NestJS, багато документації	Зручний API, швидка продуктивність	Гнучкість, підтримка складних запитів
Недоліки	Обмежена продуктивність на великих базах	Потребує окремого Prisma сервера, обмежена підтримка декораторів	Декоратори не підтримуються "з коробки", менш строгий TypeScript
К-ть скачувань за місяць	~2100000	~2200000	~170000
Розмір бібліотеки	20.4mb	10.7mb	20.4mb

Враховуючи особливості розглянутих аналогів та обраних програмних технологій, розроблена високорівнева архітектура рішення зображена на рис. 1. Вона включає в себе опис кожної частини програми, взаємодію з іншими та технологіями, що використовуються.

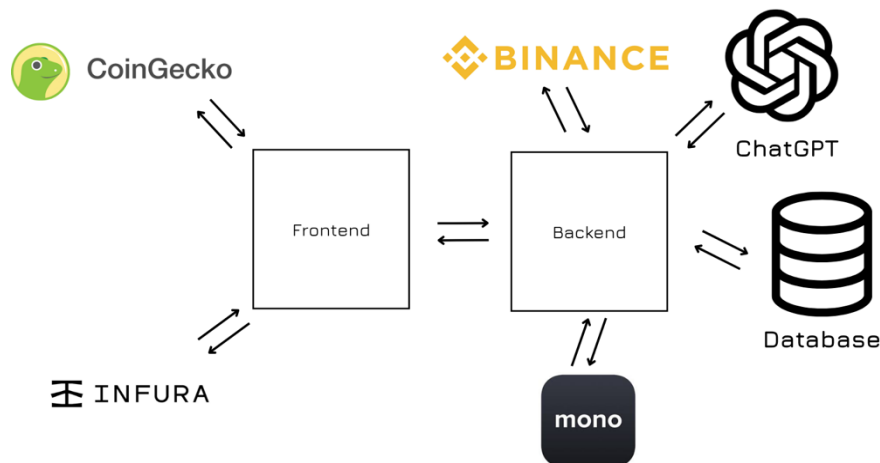


Рис. 1. Високорівнева архітектура рішення

Основні компоненти та їх взаємодія:

● Frontend - це частина додатку, з якою безпосередньо взаємодіє користувач. Це веб-інтерфейс, який користувач бачить у браузері. Він відповідає за відображення інформації, прийняття вводу від користувача та передачу даних на серверну частину. Технології: HTML, CSS, Javascript, React.js, Typescript, Brain.js, MaterialUI, lightweight-charts, axios, ethers, recoil, socket.io-client.

● Backend - це серверна частина додатку, яка обробляє запити від фронтенду, взаємодіє з базою даних та іншими зовнішніми сервісами. Бекенд виконує основну логіку програми та забезпечує безпеку даних. Технології: TypeORM, Nest.js, Typescript, axios, ethers, openai, socker.io, jest, Docker.

● БД - це сховище даних, яке використовується для зберігання інформації про користувачів, транзакції, конфігурації та іншу необхідну інформацію. Як сховище даних тут використовується популярна об'єктно-реляційна БД PostgreSQL.

● CoinGecko, Binance, Infura, Mono - це зовнішні сервіси, які надають різні API для отримання даних про криптовалюту, виконання транзакцій та взаємодії з блокчейном.

Функціональні можливості рішення, які пов'язані із криптовалютами, фінансами та чатботом зображено на діаграмі прецедентів (рис. 2). Отже, основні варіанти використання даної платформи:

- "Log in" - користувач входить у систему. Цей варіант використання включає перевірку пароля.
- "Sign up" - користувач реєструється в системі.
- "Add wallet" - користувач додає гаманець до свого профілю. Включає перевірку адреси гаманця.
- "Browse Cryptocurrency List" - користувач переглядає список криптовалют. Для цього здійснюється інтеграція з CoinGecko, що надає дані про криптовалюту.
- "Add monobank client" - користувач додає клієнта Monobank, що дозволяє зв'язати акаунт з Monobank.
- "Check Bank Balance" - користувач перевіряє свій баланс у банку, отримуючи інформацію з Monobank.
- "Bot Communication" - користувач може спілкуватися з ботом, що, використовує ChatGPT для відповідей.

● "CryptoCurrency Prediction" - користувач отримує прогнози щодо криптовалют. Цей варіант використовує прогнозну модель для оцінювання ринкових тенденцій за допомогою даних з Binance.

Для збереження необхідної інформації використовується реляційна база даних PostgreSQL. Структуру бази даних подано у вигляді діаграми "сутність-зв'язок" (рис. 3). Для комунікації з базою даних з серверної частини використовується ORM, а саме TypeORM. Саме тому в базі даних містяться додаткові таблиці, які створює задля внесення додаткової інформації про поточний стан бази. В базі даних представлено п'ять таблиць.

Migrations - це системна таблиця, яка автоматично створюється при використанні міграцій в TypeORM. Вона призначена для відстеження виконаних міграцій та їх поточного статусу. Зазвичай ця таблиця має стандартне ім'я "migrations". Під час виконання міграцій TypeORM перевіряє, чи вже була виконана певна міграція. Якщо міграція ще не була виконана, то після її виконання інформація про неї, така як ім'я, записується саме у цю таблицю. TypeORM автоматично керує цією таблицею, тому користувачу не потрібно втручатися у її роботу.

Typeorm_metadata - це ще одна системна таблиця, яка автоматично створюється в базі даних при використанні метаданих в TypeORM. Вона використовується для зберігання інформації про метадані сутностей, такі як стовпці, зв'язки, індекси та інші конфігураційні дані. Важливо зауважити, що таблиця typeorm_metadata також повністю керується автоматично і не потребує ручної модифікації. TypeORM відповідає за створення та оновлення цієї таблиці за необхідності, відповідно до визначених метаданих у програмі.

User - таблиця, створена для зберігання самого користувача та усіх його даних. Дана схема має певні обмеження (constraints), а саме: email користувача має бути унікальним в БД та пароль має зберігатись у зашифрованому вигляді задля безпеки.



Рис. 2. Діаграма прецедентів

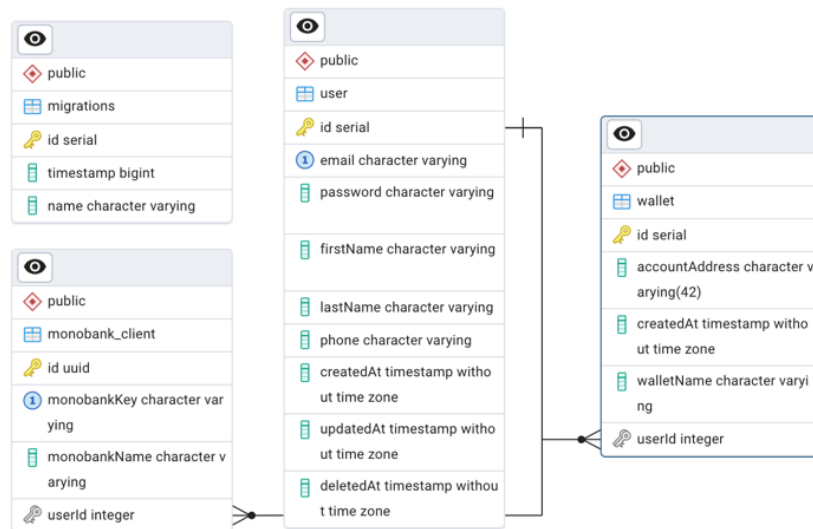


Рис. 3. Діаграма "сутність-зв'язок"

Серверна частина [11] як діаграму класів показано на рис. 4. Основні компоненти серверної частини:

- User - основний клас, що представляє користувача системи.
- Wallet - представляє криптовалютний гаманець користувача.
- MonobankClient - представляє обліковий запис Monobank.
- Services - класи, що забезпечують функціональність для різних частин системи:
 - UserService - працює з об'єктами User, має методи для створення, пошуку та оновлення профілів користувачів.
 - AuthService - відповідає за автентифікацію користувачів, забезпечуючи функції входу та виходу, а також отримання даних користувача з токенів.
 - MonobankService - забезпечує управління акаунтами Monobank користувача, з методами для створення, пошуку та отримання клієнтів Monobank.
 - WalletService - забезпечує управління гаманцями користувачів, дозволяючи створювати, шукати, оновлювати та видаляти гаманці.

- ```
graph TD
 subgraph "Domain Entities"
 User["User
createdAt: Date
deletedAt: Date
email: string
firstName: string
id: number
lastName: string
monobankAccounts: MonobankClient[]
password: string
phone: string
previousPassword: string
updatedAt: Date
wallets: Wallet[]
loadPreviousPassword(): async
setPassword(): async"]
 MonobankClient["MonobankClient
id: string
monobankKey: string
monobankName: string
user: User
userId: number"]
 Wallet["Wallet
accountAddress: string
createdAt: Date
id: number
user: User
userId: number
walletName: string"]
 end

 subgraph "Domain Services"
 UsersService["UsersService
constructor(entityManager)
create(createProfileDto): Promise
findOne(fields: Partial)!: Promise
update(id: updateProfileDto): async"]
 AuthService["AuthService
constructor(usersService, jwtService)
getUserFromToken(token: string): async
me(user: Express.User): async
signIn(email: password): async
signUp(loginDto: SignupDto): async"]
 AuthController["AuthController
constructor(authService)
me(request): async
signIn(loginDto): Promise
signUp(loginDto: SignupDto): User"]
 BinanceService["BinanceService
binance: Binance
CANDLE_REPEAT_INTERVAL: 200
getEmitter(pair: string): Observable"]
 BinanceGateway["BinanceGateway
logger: Logger
subscriptions: Map
server: DefaultEventsMap, any>
constructor(binanceService)
handleError(client: error)
afterIn(server)
handleConnection(client)
handleDisconnect(client)
handleSubscribeToCandle(client.data...)"]
 MonobankService["MonobankService
apiUrl: <string>
constructor(httpService, entityManager)
create(userId: number, monobankClientInfo?)
create(url: string, createMonobankClientInfo?)
findOneMonobankClient(id: string): async
findOneMonobankClient(userId: number)
getClientInfo(id: string): async"]
 MonobankController["MonobankController
constructor(monobankService)
create({ user }, createMonobankClientInfo?)
findAll({ user }): async
findClientInfo(id: string): async"]
 OpenaiService["OpenaiService
openai: OpenAI
constructor(monobankService)
askQuestion(userInput: question): async"]
 OpenaiController["OpenaiController
constructor(openaiService)
askCryptoQuestion({ user }, question?): ..."]
 WalletService["WalletService
constructor(entityManager)
create(userId: createWalletDto): async
findAll(userId: number): async
findOne(userId, id): async
remove(userId, id): async
update(id: updateWalletDto): async"]
 WalletController["WalletController
constructor(walletService)
create({ user }, createWalletDto): Wall...
findAll({ user })!: Wallet[]>
findOne({ user }, id): Wallet->
remove({ user }, id): Promise
update(id: updateWalletDto): Wallet"]
 end

 User --> MonobankClient
 User --> Wallet
 MonobankClient --> MonobankService
 MonobankService --> MonobankController
 MonobankController --> MonobankService
 MonobankController --> WalletService
 WalletService --> WalletController
 WalletService --> Wallet
 UsersService --> AuthService
 AuthService --> AuthController
 BinanceService --> BinanceGateway
```

Для передачі даних на клієнтську частину необхідно трансляції цінових даних (свічок) з Binance API. Тут створюється підключення до Binance через `binance-api-node` для доступу до даних про ринки. Функція `getCandleEmitter` має можливість передачі валютної пари (наприклад: `ETHUSDT`) і після цього через `WebSocket`-з'єднання отримується ціна `high` кожної хвилинної свічки та передається через `Observable`. Також встановлено інтервал повторної передачі значень, що надсилає останнє значення на клієнт з певною частотою, навіть якщо нові свічки не надійшли. Таким чином, цей сервіс передає актуальні ціни з Binance API для заданої валютної пари з підтримкою регулярного оновлення.

Клієнтська частина [12] написана з використанням бібліотеки React.js, але з використанням Vite для швидшої збірки та запуску проекту. Для створення базової структури проекту використовується команда `pnpm create vite@latest`, яка генерує TypeScript-шаблон.

Використання чат-бот системи ChatGPT (рис. 5) забезпечує комунікацію та підтримую користувача з порадами. Повідомлення у даному чаті відображаються умовно в залежності від того, чи вони від користувача чи від бота. Різні стилі застосовуються для візуального розрізнення двох відправників. В майбутньому, його можна буде замінити на більш компетентну мережу для роботи у фінансовому секторі.





Для функціонального тестування використовувались різні браузери (Google Chrome, Microsoft Edge та Mozilla Firefox), щоб переконатись, що програма коректно працює на різних платформах. Приклад сторінки графіку криптовалют наведено на рис. 6, де відображено графік криптовалют в реальному часі і також передбачення можливості росту/падіння криптовалюти за допомогою моделі машинного навчання. За допомогою індикаторів буде показуватись чи зараз тренд до зростання чи тренд до падіння.

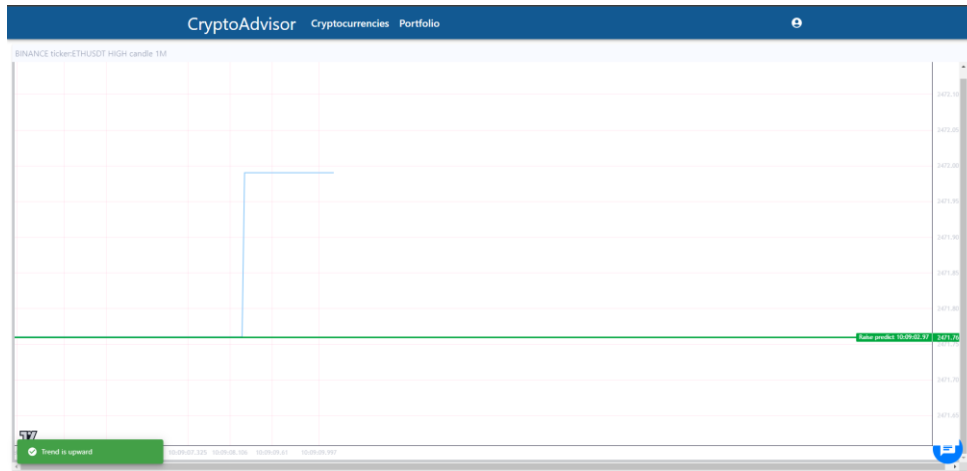


Рис. 6. Сторінка для прогнозування ціни

Для реалізації сторінки прогнозування ціни реалізована система (рис. 6), що використовує реактивне програмування для аналізу фінансових даних (наприклад, цін на криптовалюти) з метою виявлення тенденцій на основі нейронних мереж. Основна логіка організована через потоки даних (emitters) та оператори, які обробляють та трансформують ці дані.

Система потоку даних:

- вся система побудована на основі патерну "потоків даних", використовуючи Source та Operator з бібліотеки react-declarative;
- використовуються мультикаст-потоки (multicast) для одночасної підписки кількох споживачів на один і той же потік даних.

Потоки даних:

- trendEmitter: обробляє ціни, що надходять через priceEmitter, застосовує корекцію, визначає тенденції (зростання або падіння) і зберігає ці дані для подальшої обробки та визначає, чи відбувається підвищення чи зниження цін та генерує об'єкти, що містять дані про ціну та відповідну тенденцію.
- positiveSetEmitter і negativeSetEmitter: ці потоки відповідають за відбір даних з trendEmitter на основі виявлених тенденцій (позитивні або негативні).

Навчання нейронної мережі:

- netTrainsetEmitter: об'єднує позитивний та негативний набори даних з positiveSetEmitter і negativeSetEmitter, готуючи їх для навчання нейронної мережі;
- netEmitter: створює та навчає нейронну мережу з використанням brain.js, на основі даних, отриманих з об'єданого потоку. Використовує асинхронний процес для тренування, обробляючи навчальні дані та реєструючи статус навчання.

Вхідні дані для нейронної мережі netInputEmitter: Формує вхідні дані для нейронної мережі, застосовуючи корекцію цін і групування значень для навчання. Вхідні дані складаються з процентних змін між парами значень.

Прогнозування predictEmitter: Створює Subject, що може видавати різні стани прогнози (наприклад, "тренування", "зростання", "падіння" та інші).

### Висновки з даного дослідження

#### і перспективи подальших розвідок у даному напрямі

У роботі проведено ґрунтовне дослідження сучасних підходів до управління фінансовими активами, зокрема аналіз ринку криптовалют та інструментів для інтеграції з банківськими системами. Розглянуто існуючі рішення для аналізу ринкових трендів та автоматизації прийняття рішень щодо інвестицій. Особливу увагу приділено можливостям використання машинного навчання для створення рекомендацій на основі поточних ринкових даних. Проаналізовано ключові аспекти інтеграції традиційних банківських рахунків та криптовалютних гаманців в єдиному інтерфейсі, що дозволяє ефективніше керувати як криптовалютними, так і фіатними активами в одній платформі.

Розроблена платформа вирізняється можливістю інтеграції різних криптовалютних гаманців і банківських рахунків через API Monobank, що дозволяє користувачам в одному місці відстежувати та управляти як криптоактивами, так і фіатними рахунками. Завдяки ML-моделі для аналізу ринкових даних сервіс пропонує рекомендації щодо купівлі чи продажу активів, автоматизуючи прийняття фінансових рішень і знижуючи інвестиційні ризики.



Розроблене рішення забезпечує користувачам простий і ефективний доступ до керування фінансами, об'єднуючи традиційні банківські рахунки та криптовалютні гаманці в єдиній платформі. Система надає можливість відслідковувати фінансові активи в реальному часі. Завдяки впровадженню ML-моделі користувачі отримують рекомендації щодо оптимальних операцій з активами, що допомагає мінімізувати ризики та збільшити дохідність портфеля. Інтеграція криптогаманців і Monobank в одному інтерфейсі значно підвищує зручність і функціональність, забезпечуючи користувачів ефективним інструментом для управління капіталом.

Розроблена платформа дозволяє підвищити фінансову грамотність користувачів, забезпечуючи їм доступ до аналітичних даних і порад щодо оптимізації фінансових рішень. Загалом, проєкт має на меті створити потужний інструмент для управління капіталом, що значно полегшить процеси моніторингу та аналізу фінансових активів. Саме тому, наступні кроки для дослідження будуть присвячені питанням, що пов'язані із моделюванням та прийняттям фінансових рішень при різних обставинах.

### Література

1. Financial Capital [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Financial\\_capital](https://en.wikipedia.org/wiki/Financial_capital) (дата звернення: 10.08.2025). — Назва з екрана.
2. Кастодіальні та некастодіальні гаманці: у чому різниця? [Електронний ресурс] – Режим доступу до ресурсу: <https://academy.binance.com/uk/articles/custodial-vs-non-custodial-wallets-what-s-the-difference> (дата звернення: 10.08.2025). — Назва з екрана.
3. Financial Assistant Job Description, Duties & Skills. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.accountingtechniciansireland.ie/news/financial-assistant-job-description-duties-skills> (дата звернення: 10.08.2025). — Назва з екрана.
4. ChatGPT vs Chatbots: What's the Real Difference and Which One Should You Choose? [Електронний ресурс] – Режим доступу до ресурсу: <https://fastbots.ai/blog/chatgpt-vs-chatbots-what-s-the-real-difference-and-which-one-should-you-choose> (дата звернення: 10.08.2025). — Назва з екрана.
5. What is Google Gemini. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/think/topics/google-gemini> (дата звернення: 10.08.2025). — Назва з екрана.
6. What is Claude AI. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/think/topics/claude-ai> (дата звернення: 10.08.2025). — Назва з екрана.
7. A First Look at Financial Data Analysis Using ChatGPT-4o. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mdpi.com/1911-8074/18/2/99> (дата звернення: 10.08.2025). — Назва з екрана.
8. F.I.R.E on track [Електронний ресурс] – Режим доступу до ресурсу: <https://fireontrack.com>.
9. Wally. [Електронний ресурс] – Режим доступу до ресурсу: <https://play.google.com/store/apps/details?id=com.gilosher.osherwallet&hl=en> (дата звернення: 10.08.2025). — Назва з екрана.
10. Magnifi AI for your financial future [Електронний ресурс] – Режим доступу до ресурсу: <https://magnifi.com> (дата звернення: 10.08.2025). — Назва з екрана.
11. CryptoAdvisor: Серверна частина [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/MaxAzarov/crypto-monetary-advisor-backend> (дата звернення: 10.08.2025). — Назва з екрана.
12. CryptoAdvisor: Клієнтська частина [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/MaxAzarov/crypto-monetary-advisor-frontend> (дата звернення: 10.08.2025). — Назва з екрана.

### References

1. Financial Capital [Elektronnyy resurs] – Rezhym dostupu do resursu: [https://en.wikipedia.org/wiki/Financial\\_capital](https://en.wikipedia.org/wiki/Financial_capital) (data zvernennia: 10.08.2025). — Nazva z ekrana.
2. Кастодіальні та некастодіальні гаманці: у чому різниця? [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://academy.binance.com/uk/articles/custodial-vs-non-custodial-wallets-what-s-the-difference> (data zvernennia: 10.08.2025). — Nazva z ekrana.
3. Financial Assistant Job Description, Duties & Skills. [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://www.accountingtechniciansireland.ie/news/financial-assistant-job-description-duties-skills> (data zvernennia: 10.08.2025). — Nazva z ekrana.
4. ChatGPT vs Chatbots: What's the Real Difference and Which One Should You Choose? [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://fastbots.ai/blog/chatgpt-vs-chatbots-what-s-the-real-difference-and-which-one-should-you-choose> (data zvernennia: 10.08.2025). — Nazva z ekrana.
5. What is Google Gemini. [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://www.ibm.com/think/topics/google-gemini> (data zvernennia: 10.08.2025). — Nazva z ekrana.
6. What is Claude AI. [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://www.ibm.com/think/topics/claude-ai> (data zvernennia: 10.08.2025). — Nazva z ekrana.
7. A First Look at Financial Data Analysis Using ChatGPT-4o. [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://www.mdpi.com/1911-8074/18/2/99> (data zvernennia: 10.08.2025). — Nazva z ekrana.
8. F.I.R.E on track. [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://fireontrack.com> (data zvernennia: 10.08.2025). — Nazva z ekrana.
9. Wally. [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://play.google.com/store/apps/details?id=com.gilosher.osherwallet&hl=en> (data zvernennia: 10.08.2025). — Nazva z ekrana.
10. Magnifi AI for your financial future. [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://magnifi.com> (data zvernennia: 10.08.2025). — Nazva z ekrana.
11. CryptoAdvisor: Серверна частина. [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://github.com/MaxAzarov/crypto-monetary-advisor-backend> (data zvernennia: 10.08.2025). — Nazva z ekrana.
12. CryptoAdvisor: Клієнтська частина. [Elektronnyy resurs] – Rezhym dostupu do resursu: <https://github.com/MaxAzarov/crypto-monetary-advisor-frontend> (data zvernennia: 10.08.2025). — Nazva z ekrana.