

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка web-застосунку для оцінки криптоактивів»

на здобуття освітнього ступеня бакалавра

зі спеціальності 121 Інженерія програмного забезпечення

освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Ілля ЛУПУ
(підпис)

Виконав: здобувач вищої освіти групи ПД-44

_____ Ілля ЛУПУ

Керівник: _____ Ірина ЩЕРБИНА
к.т.н., доцент

Рецензент: _____

Київ 2025

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Лупу Іллі Ігоровичу

1. Тема кваліфікаційної роботи: «Розробка web-застосунку для оцінки криптоактивів»

керівник кваліфікаційної роботи к.н.т., доцент Ірина ЩЕРБИНА,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 р. № 56.

2. Строк подання кваліфікаційної роботи «02» червня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: теоретичні відомості про основні метрики криптоактивів, ончейн-аналіз, визначення блокчейн рівня 1 , таблиця популярних криптовалют , статистика прибутковості за ROI, технічна документація з описом бібліотек.

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

1. Аналіз застосунку для оцінки криптоактивів .
2. Проектування web-застосунку для оцінки криптоактивів.
3. Розробка застосунку для оцінки криптоактивів.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до програмного забезпечення.

3. Програмні засоби реалізації.
4. Діаграма архітектури застосунку.
5. Діаграма послідовності роботи клієнт-сервер-база даних.
6. Діаграма варіантів використання.
7. Діаграма структури застосунку.
8. Екранні форми.
9. Апробація результатів дослідження.

6. Дата видачі завдання «25» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	10.04.-18.04.2025	
2	Аналіз та дослідження існуючих аналогів	19.05-20.04.2025	
3	Визначення ключових метрик блокчейнів та ринку	21.04 -24.04.2025	
4	Аналіз аналогів і створення функціональних вимог до застосунку для оцінки криптоактивів	25.04 -01.05.2025	
5	Проектування архітектури застосунку для оцінки криптоактивів	02.05 - 03.05.2025	
6	Програмна реалізація та тестування застосунку для оцінки криптоактивів	04.05 - 12.05.2025	
7	Оформлення роботи: вступ, висновки, реферат	13.05 - 16.05.2025	
8	Розробка демонстраційних матеріалів	16.05 - 18.05.2025	
9	Попередній захист роботи	19.05 - 30.05.2025	

Здобувач вищої освіти

(підпис)

Ілля ЛУПУ

Керівник

кваліфікаційної роботи

(підпис)

Ірина ЩЕРБИНА

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: __53__ стор., __2__ табл., __15__ рис., __15__ джерел.

Мета роботи – є розробка web-застосунку для надання інвесторам оцінки блокчейн-проектів на основі ключових метрик.

Об'єкт дослідження процес оцінки криптоактивів.

Предмет дослідження – web-застосунок для оцінки метрик криптовалют.

Короткий зміст роботи: У роботі досліджено проблему оцінки криптоактивів у сучасних умовах ринку. Проаналізовано ключові метрики (TVL, ринкова капіталізація, активні користувачі тощо) та існуючі аналітичні платформи. Розроблено веб-застосунок на Vue.js з використанням Chart.js та Element Plus, який надає інтерактивну оцінку блокчейн-проектів через систему ChainScore, візуалізацію даних та AI-чат для аналітики. Реалізовано функції порівняння проектів, сортування, топ-100 криптоактивів та визначення рейтингу криптоактиву за шкалою 1-100.

Сферою використання застосунку є надання інвесторам, аналітикам та розробникам інструменту для оцінки блокчейн-проектів на основі ключових метрик.

КЛЮЧОВІ СЛОВА: КРИПТОАКТИВИ, BLOCKCHAIN, WEB-ЗАСТОСУНОК, БЛОКЧЕЙН-МЕТРИКИ, DEFI, TVL, CHAINSCORE

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП.....	10
1 АНАЛІЗ ЗАСТОСУНКУ ДЛЯ ОЦІНКИ КРИПТОАКТИВІВ.....	12
1.1 Класифікація криптоактивів та вибір категорій для аналізу	12
1. Дослідження існуючих аналогів.....	17
2 ПРОЕКТУВАННЯ WEB-ЗАСТОСУНКУ ДЛЯ ОЦІНКИ КРИПТОАКТИВІВ...24	
2.1 Визначення функціональних і нефункціональних вимог до застосунку.....	24
2.2. Вибір технічних засобів для розробки застосунку для оцінки криптоактивів	24
2.2.1. Вибір мови програмування та середовище розробки для оцінки криптоактивів.....	24
2.2.2 Вибір веб-фреймворків та програмних бібліотек	26
2.2.3 Система контролю версій	27
2.3. Архітектура застосунку для оцінки криптоактивів.....	29
3 РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ОЦІНКИ КРИПТОАКТИВІВ	35
3.1 Проектування інтерфейсу користувача	35
3.2 Реалізація серверної частини	37
3.3 Реалізація клієнтської частини	38
3.4 Завантаження проекту на github.....	49
3.5 Тестування	49

ВИСНОВКИ	52
ПЕРЕЛІК ПОСИЛАНЬ	53
ДОДАТОК А. ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ	55
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – Програмне забезпечення

IDE – Integrated Development Environment

API – Application Programming Interface

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

TVL – Total Value Locked

MAU – Monthly Active Users

ROI – Return on Investment

L1 – Layer1

DeFi – Decentralized Finance

dApps – Decentralized Applications

ВСТУП

Актуальність: У сучасних умовах швидкого розширення ринку криптоактивів і стрімкого розвитку блокчейн-технологій інвестори та розробники стикаються з безпрецедентними обсягами даних і різноманіттям блокчейн-мереж. Щодня з'являються багато криптовалют, кожна з яких має свої унікальні технічні характеристики, економічні моделі та показники ефективності.

Відсутність спеціалізованих аналітичних інструментів ускладнює оперативну оцінку та порівняння цих параметрів, що значно ускладнює процес ухвалення обґрунтованих інвестиційних і технологічних рішень.

З розвитком ринку стає все більш очевидною потреба у створенні комплексних аналітичних інструментів, що забезпечують систематизацію та візуалізацію ключових метрик блокчейн-проектів.

Швидкий ріст ринку криптовалют викликає великий інтерес серед інвесторів, проте навіть за умов зростання інвестиції в блокчейн-проекти потребують ретельного аналізу та обережності". Згідно з аналітичними дослідженнями, інвестиції в блокчейн-проекти можуть приносити високу прибутковість, але вимагають детального аналізу численних факторів і показників.[1]

Об'єктом дослідження є процес оцінки криптоактивів.

Предметом дослідження є web-застосунок для оцінки метрик криптовалют.

Метою роботи є надання інвесторам швидкої оцінки блокчейн-проектів на основі ключових метрик.

Методи дослідження: було проаналізовано існуючі рішення для оцінки криптопроектів, було здійснено пошук відкритих web api і визначення головних метрик для оцінки криптовалют. Було проведено огляд технологій розробки було обрано Vue.js для клієнтської частини Supabase для серверної частини, бібліотек Chart.js для реалізації графіків і Element plus для реалізації таблиць

Наукова новизна роботи визначається наступними функціональними складовими: можливість оцінки криптоактиву за метриками у шкалі від 1-100 ; чат-бот для допомоги з визначення метрик; графічна візуалізація метрик

Практична значущість результатів полягає в можливості використання реалізованого веб-застосунку для ефективної оцінки криптоактивів на основі популярних метрик.

1 АНАЛІЗ ЗАСТОСУНКУ ДЛЯ ОЦІНКИ КРИПТОАКТИВІВ

1.1 Класифікація криптоактивів та вибір категорій для аналізу

Потреба в платформі для аналітики криптоактивів стає дедалі актуальнішою через постійні зміни на ринку та появу нових проєктів.

Станом на квітень 2025 року існує 17 134 криптовалюти. Однак не всі криптовалюти є активними або цінними. Якщо не враховувати багато «мертвих» криптовалют, залишається лише близько 10 385 активних криптовалют.[2]

Аналізуючи аналітичні дані ресурсу CoinGecko щодо перших 100 криптовалют за ринковою капіталізацією, можна встановити, що понад 40 % з них становлять блокчейн-проєкти, які належать до рішень першого (Layer 1) та другого (Layer 2) рівнів.[3]

Блокчейни рівня 1 — це мережа, яка функціонує як інфраструктура для інших протоколів, застосунків і мереж, які можна створити. Рівень 1 відповідає за програмування мов і свідомих процесів, а також за правила й параметри, які дозволяють блокчейну функціонувати належним чином. Найвідомішими прикладами блокчейнів рівня 1 є Ethereum, Bitcoin і Litecoin.[4]

Блокчейни рівня 2 - це додаткові протоколи, які будуються поверх базових блокчейнів (L1) для поліпшення їхньої швидкості, масштабованості або вартості транзакцій.

Також можна побачити що серед інвестиції у найбільших криптофондів, високий прибуток отримали базові блокчейни (L1), такі як Solana Avalanche, Sui , які за останні роки показали ROI 1000-4000%+.[5]

Ринок криптоактивів постійно змінюється, з'являються нові проєкти, що створює потребу в вебзастосунку для їх оцінки. Велика кількість метрик і криптовалют ускладнює аналіз, тому необхідний інструмент, який дозволяє швидко оцінити проєкт за основними метриками та визначити його рейтинг.

Вебзастосунок для оцінки криптоактивів — це спеціалізований онлайн-ресурс, який надає користувачам можливість аналізувати блокчейн-проєкти за

рядом метрик та приймати рішення на основі комплексної оцінки. Унікальною особливістю застосунку є можливість визначити ChainScore — числовий показник від 1 до 100, що відображає перспективність криптоактиву на основі фундаментальних показників. Він розраховується за спеціальною формулою, яка враховує головні показники блокчейну.

Основні цілі вебсайту для оцінки криптоактивів:

- Надання об'єктивної оцінки криптоактивів за шкалою від 1 до 100
- Відображення основних блокчейн-метрик і ринкових показників у зручному для аналізу вигляді;
- Порівняння блокчейнів за ключовими метриками (транзакції, активність користувачів, комісії тощо); аналітика та допомога користувачеві завдяки AI-асистенту.
- Надання огляду топ-100 криптоактивів за оцінкою та метриками;
- Допомога у прийнятті рішень на основі актуальної та аналітичної інформації.

Переваги:

- Комплексна оцінка: ChainScore дає змогу швидко оцінити ефективність активу за багатьма параметрами.
- Простий інтерфейс: Користувачі можуть швидко знайти потрібну інформацію завдяки зручному дизайну та навігації.
- Інтерактивність та візуалізація: графіки, таблиці, прогрес-бари та AI-чат покращують користувацький досвід.
- Персоналізована допомога: AiHelper дозволяє адаптувати допомогу під потреби користувача.
- Доступність даних: користувач може самостійно аналізувати метрики і приймати рішення без потреби у зовнішніх джерелах.

Недоліки:

- Залежність від якості даних: Успішність оцінки залежить від актуальності та достовірності даних.

Не завжди враховуються суб'єктивні фактори: ChainScore базується на числових метриках і не може повністю відображати репутацію чи потенціал проєкту.

Потреба у розумінні метрик: Новачкам може бути складно з першого разу зрозуміти значення кожної метрики, хоча AiHelper частково вирішує цю проблему.

1.2 Визначення основних метрик оцінки криптовалют

Популярним методом визначення ключових метрик являється ончейн-аналіз. Ончейн-аналіз — це метод, що використовується у світі криптовалют для оцінки та інтерпретації даних блокчейну. [6]

Розглянемо основні метрики ончейн аналізу.

Поява та популяризація yield farming під час «DeFi-літа» 2020 року сприяли становленню показника TVL як одного з основних метричних інструментів у Web3. TVL почали використовувати для порівняння конкуруючих проєктів. Наприклад, 21 жовтня 2020 року проєкт Aave мав TVL у розмірі 899 мільйонів доларів США, а вже за рік цей показник зріс до 18,97 мільярда доларів США.[7]

Дохід dApp (fees) – це прибуток, який протокол отримує від комісій користувачів. Дохід dApp набуває ваги у Web3 тому що Відображає які сервіси мають стійкий попит і можуть бути прибутковими у довгостроковій перспективі.

Активні адреси (MAU). Аналіз активних адрес у мережі блокчейн може дати уявлення про залученість користувачів. Для їхнього відстеження необхідно переглядати унікальні адреси, що надсилали або отримували активи в мережі.

Також є декілька важливих показників які вимірюють соціальну активність людей та розробників.

Twitter являється важливою соціальною метрикою, тому що twitter – це дуже популярна соцмережа серед крипто спільноти тому варто враховувати соціальну активність проєктів на цій платформі. У своєму проєкті я використовую число підписників для відстеження активності блокчейн-проєктів проте існують сервіси

які враховують всі показники твіттер аккаунту: репости, лайки, кількість згадок та інші, проте вони мають закритий або платний API.

Активність розробників показує, скільки людей реально працюють над проектом: додають новий функціонал, виправляють помилки, оновлюють код. Це один із найкращих показників того, що проєкт живий і розвивається

Окрім технічних і соціальних метрик, важливу роль відіграють ринкові показники. Ціна Bitcoin слугує основним драйвером ринку - її падіння зазвичай тягне за собою зниження інших криптовалют.

Bitcoin Dominance (BTC.D) відображає співвідношення ринкової капіталізації Біткоїна до загального ринку: високе значення свідчить про зростання довіри до BTC як "безпечного активу" та можливий відтік капіталу з альткоїнів.

Індекс страху і жадібності (Fear & Greed Index) вимірює ринкові емоції - від паніки (масові продажі) до ейфорії (стрімкий ріст).

Ринкова капіталізація (market cap) дає зрозуміти реальний масштаб проєкту чим вона вища, тим більш стабільний актив. У таблиці 1.1 зображено основні метрики оцінки криптовалют.

Таблиця 1.1

Зведена таблиця основних метрик оцінки криптовалют

Категорія	Основні метрики	Приклади
Ончейн	TVL (Total Value Locked) – загальна сума активів, заблокованих у протоколі	Aave: \$18,97 млрд (2021)
	Дохід dApp (Fees) – комісії, які отримує протокол.	Uniswap

Продовження таблиці 1.1

Зведена таблиця основних метрик оцінки криптовалют

Категорія	Основні метрики	Приклади
Ончейн	MAU (Monthly Active Users) – унікальні адреси, що здійснюють транзакції.	Біткоїн має понад 1 млн активних користувачів щомісяця
Ринкові	Ціна Bitcoin – основний індикатор ринкових тенденцій.	BTC росте → ріст інших криптовалют
	Bitcoin Dominance (BTC.D) – частка BTC у загальній ринковій капіталізації	BTC.D > 50% → зростання інтересу до BTC
	Fear & Greed Index – індикатор ринкових емоцій (0–100).	Extreme Fear → можливий ріст, Extreme Greed → можлива корекція
	Ринкова капіталізація (Market Cap) – загальна оцінка проекту.	Ethereum: понад \$400 млрд
Соціальні	Активність у Twitter – кількість підписників, згадок, взаємодій.	Криптовалюта dogecoin має 3М+ фоловерів, високий соціальний тренд
	Активність розробників – частота оновлень коду, коміти в репозиторіях.	Polkadot, Solana – понад 100 комітів щотижнево

Вебзастосунок для оцінки криптоактивів орієнтований на широке коло користувачів з різним рівнем досвіду у сфері криптовалют. Цільова аудиторія додатку включає такі категорії користувачів:

– Досвідчені інвестори та трейдери — користувачі з глибокими знаннями ринку, які активно керують своїми портфелями. Їхня мета — оперативно оцінювати

потенціал блокчейн-проектів, виявляти тренди та приймати зважені інвестиційні рішення на основі ключових метрик.

- Початківці у сфері криптовалют — користувачі, які лише починають знайомство з крипторинком. Їхня мета — зрозуміти базові показники проектів та навчитися відрізняти перспективні активи від сумнівних, використовуючи інтуїтивно зрозумілий інтерфейс.

- Розробники та крипто-проекти — фахівці, які шукають перспективні блокчейн-екосистеми з великим потенціалом зростання та залучення великої кількості користувачів. Їхня мета — вибрати перспективну блокчейн-платформу для запуску та розробки свого додатку на основі ключових показників (TVL, активність користувачів, рівень екосистемної підтримки тощо, щоб забезпечити найкраще середовище для розвитку та масштабування проекту).

1.1 Дослідження існуючих аналогів

Існує багато сервісів для оцінки криптоактивів.

DefiLlama — це популярний аналітичний сервіс для відстеження децентралізованих фінансів. Він надає дані про загальну вартість активів у протоколах (TVL), ринкову статистику, інформацію про блокчейни, стейблкоїни, NFT та інші аспекти DeFi-екосистеми.

Основні функції DefiLlama:

- Аналіз фінансових показників протоколів: інформація про річні доходи, комісії та прибутковість.

- Інструменти для розробників: API для інтеграції даних DefiLlama у власні проекти.

- Моніторинг загальної вартості заблокованих активів (TVL): перелік даних по різних DeFi-протоколам.

- Аналіз фінансових показників протоколів: інформація про річні доходи, комісії та прибутковість.

Переваги:

- Відкритий API: має відкритий API який дасть змогу розробникам використовувати DeFi метрики у додатках.
- Обширна база даних про протоколи: Надає користувачам актуальну інформацію про TVL у реальному часі, що дозволяє відслідковувати зміни на ринку.
- Функція сортування таблиці: Користувачі можуть зручно організовувати дані за основними метриками, такими як TVL та комісії, що полегшує їхній аналіз. Недоліки:
 - Не враховується якість TVL: Платформа не оцінює якість заблокованих активів, що може призвести до помилкових висновків.
 - Не розрізняє "реальне" блокування коштів і фальшиві дії: Відсутність модерації може призвести до того, що користувачі отримають дані, які включають wash-trading або штучно завищений TVL, ускладнюючи точність аналізу.

На рисунку 1.1 показано приклад сторінки DefiLama.

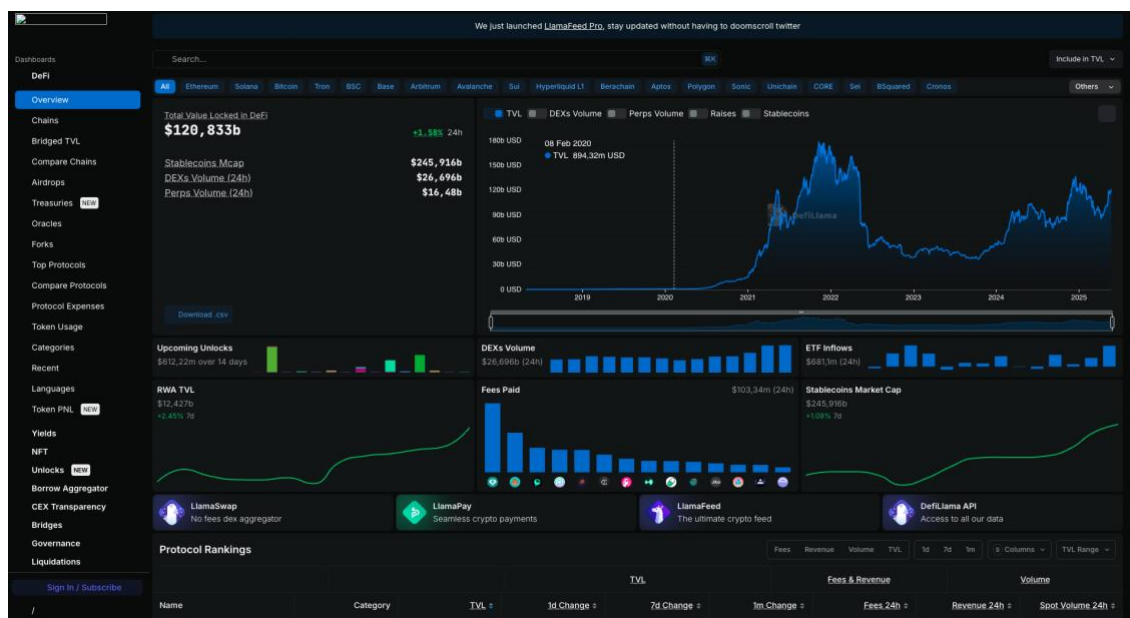


Рис 1.1 Приклад сторінки DefiLama

Token Terminal — це аналітична платформа, яка спеціалізується на фінансових показниках децентралізованих протоколів та tokenів. Вона надає користувачам можливість отримувати детальну інформацію про доходи, витрати,

ринкову капіталізацію та інші ключові метрики для різних криптовалютних проєктів.

Token Terminal є повноцінною платформою для роботи з даними на блокчейні, яка зосереджена на стандартизації фінансових та альтернативних даних для найбільш використовуваних блокчейнів і децентралізованих додатків.

Основні функції Token Terminal включають :

- Аналіз фінансових показників протоколів, що дозволяє інвесторам оцінювати їхню прибутковість та стійкість.
- Моніторинг ринкових трендів та порівняння різних блокчейн проєктів.
- Дані про блокчейн: комісії, кількість активних користувачів на день, обсяги торгів, активні позики та перекази стейблкоїнів.
- Проста візуалізація даних за допомогою дашбордів та панелей, що полегшує розуміння складних фінансових показників та допомагає користувачам приймати обґрунтовані рішення.

Переваги:

- Швидка зміна даних у реальному часі: Користувачі можуть швидко отримувати оновлену інформацію про ключові показники, такі як TVL, дохід та кількість активних користувачів.
- Зручні візуалізації: Платформа надає зрозумілі аналітичні панелі, що спрощують дослідження фінансових та альтернативних даних.
- Глибокий аналіз проєктів: Можливість легко досліджувати фундаментальні дані для різних блокчейнів і децентралізованих додатків.
- Різноманітні набори даних: Платформа пропонує широкий вибір наборів даних для аналізу активності в блокчейні та ринкових тенденцій.

Недоліки:

- Недоступний API: Відсутність публічного API може бути перешкодою у інтеграції даних для персонального використання.
- Завелика кількість непотрібних метрик: користувачі можуть заплутатися або бути перевантаженими надмірною кількістю метрик, які можуть не мати значення.

— Обмежена доступність для новачків: сайт має складну навігацію для початківців, оскільки він дуже насиченим великою кількістю різних показників.

— Вартість підписки: ціна підписки на повну функціональність платформи може бути дорогою для деяких користувачів.

На рисунку 1.2 зображено приклад сторінки Token Terminal.

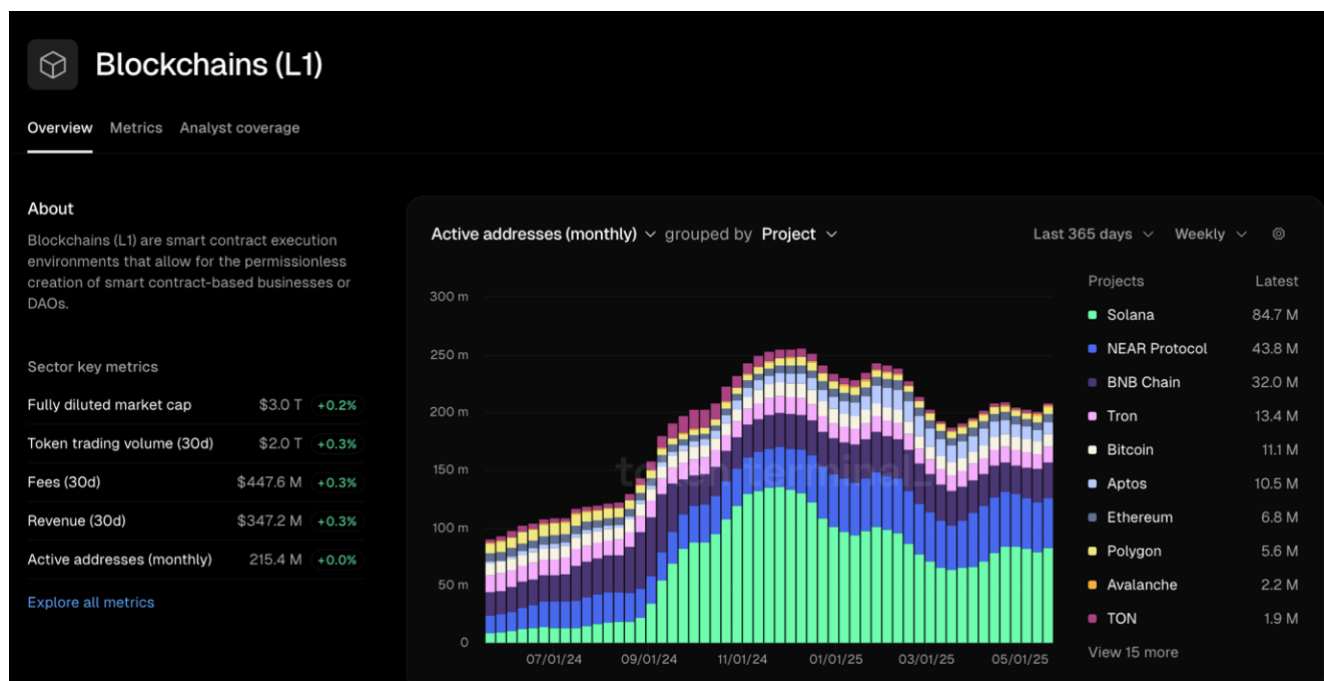


Рис 1.2 Приклад візуалізації метрики Token Terminal

DappRadar — це платформа яка дозволяє знаходити нові децентралізовані додатки, отримувати аналітику про колекції DeFi та NFT,. DappRadar включає багато інструментів для управління токенами та аналізу цінових рухів

Основні функції DappRadar включають :

— Рейтинги Dapps: DappRadar пропонує користувачам переглядати рейтинги децентралізованих додатків в різних категоріях, таких як DeFi, NFT та GameFi. Це спрощує процес пошуку популярних і нових проектів.

— Портфоліо: функція дозволяє користувачам контролювати свої токени, інвестиції в DeFi та NFT-активи, забезпечуючи зручний спосіб моніторингу їхніх фінансових вкладень.

– Аналітика та звіти: DappRadar пропонує детальний аналіз ринку, що включає дані про обсяги торгів, кількість активних користувачів та інші ключові показники. Це дозволяє користувачам бути в курсі останніх тенденцій у сфері Web3.

Переваги:

– Широкий вибір dapps: великий вибір проектів за категоріями, таких як DeFi, NFT та GameFi.

– Інтуїтивна система рейтингів: Користувачі можуть легко відстежувати популярність dapps завдяки зручним рейтингам, які можна фільтрувати за блокчейн-екосистемами та категоріями.

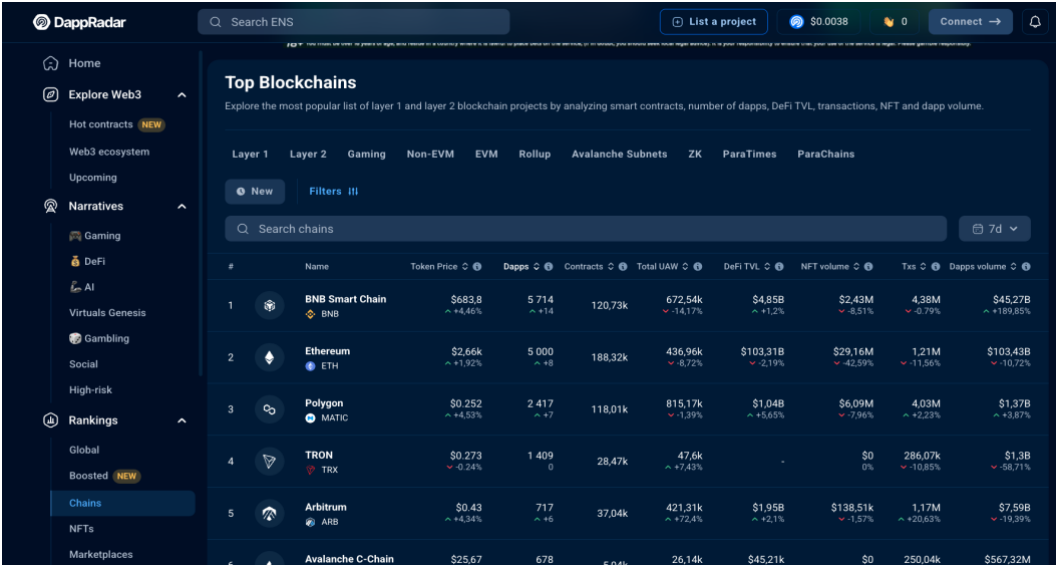
– Керування портфелем: Функція Portfolio дозволяє користувачам ефективно управляти своїми токенами, позиціями в DeFi та колекціями NFT.

Недоліки:

– Складний інтерфейс: є дуже багато зайвих параметрів які можуть перешкоджати користувачеві в аналізі проектів.

– Низький рівень активності окремих проектів: Існує безліч проектів з недостатньою активністю, і відсутність модерації може спричинити те, що користувачі стикаються з неперспективними чи застарілими dapps.

На рисунку 1.3 показано приклад таблиці блокчейнів на сторінці DappRadar.



#	Name	Token Price	Dapps	Contracts	Total UAW	DeFi TVL	NFT volume	Txs	Dapps volume
1	BNB Smart Chain BNB	\$683.8 +4.49%	5 714 +14	120,73k	672.54k -14.17%	\$4.85B +1.2%	\$2.43M -8.51%	4.38M -0.79%	\$45,27B +189.83%
2	Ethereum ETH	\$2,66k +1.92%	5 000 +8	188,32k	436,96k -8.72%	\$103.31B -2.19%	\$29,16M -42.59%	1.21M -11.56%	\$103,43B -10.72%
3	Polygon MATIC	\$0.252 +4.53%	2 417 +7	118,01k	815,17k -1.39%	\$1,04B +5.65%	\$6,09M -7.96%	4,03M +2.23%	\$1,37B +3.87%
4	TRON TRX	\$0.273 -0.24%	1 409 0	28,47k	47.6k +7.43%	-	\$0 0%	286,07k -10.85%	\$1,3B -58.71%
5	Arbitrum ARB	\$0.43 +4.34%	717 +8	37,04k	421,31k +72.4%	\$1.95B +2.1%	\$138,51k -1.57%	1,17M +20.63%	\$7,59B -19.39%
6	Avalanche C-Chain	\$25,67	678	5,04k	26,14k	\$45,21k	\$0	250,04k	\$567,32M

Рис 1.3 Приклад таблиці блокчейнів на сторінці DappRadar

У таблиці 1.2 зведено результати аналізу існуючих застосунків для оцінки криптовалют.

Таблиця 1.2

Порівняльна таблиця аналітичних застосунків для оцінки криптоактивів

Критерій	DefiLlama	Token Terminal	DappRadar
Призначення	Відстеження TVL, ринкової активності та інших DeFi-метрик	Стандартизована фінансова аналітика децентралізованих проєктів	Виявлення та аналітика DeFi, NFT та GameFi застосунків
Основні функції	- Моніторинг TVL у DeFi-протоколах. - API для інтеграцій. - Річні доходи та комісії	- Фінансова звітність (дохід, капіталізація) - Метрики користувачів та активності	- Рейтинги DApps по категоріях - Портфоліо користувача - Метрики по активності
API	Відкритий	Платний	Платний
Якість даних	Є ризик викривлень через необроблені або "накручені" дані	Дані структуровані та стандартизовані	Дані не завжди фільтруються від «мертвих» або неактивних dapps
Візуалізація	Графіки та таблиці з сортуванням	Аналітичні панелі та графіки	Обмежена візуалізація, але є портфоліо
Цільова аудиторія	Аналітики, розробники, дослідники	Інвестори, аналітики, фінансисти	Інвестори, NFT-ентузіасти, геймери

Продовження таблиці 1.2

Порівняльна таблиця аналітичних застосунків для оцінки криптоактивів

Критерій	DefiLlama	Token Terminal	DappRadar
Унікальні особливості	- Сильний фокус на TVL - Відкриті дані - Широке охоплення DeFi	- Фінансові KPI як у традиційній економіці - Стандартизація даних - Детальні метрики	- Каталог NFT, DeFi, GameFi - Портфоліо - Система рейтингів
Недоліки	- Не оцінює якість TVL - Не відфільтровує фальшиві метрики	- Немає публічного API - Перевантаження інформацією - Висока вартість підписки	- Забагато неактуальних проєктів - Складність аналізу для новачків

Проаналізувавши кожен аналог для оцінки криптоактивів, можна дійти висновку, що кожен з них заточений під конкретні потреби:

DefiLlama — спеціалізується на TVL та ринковій активності DeFi, пропонуючи відкриті дані та широке охоплення протоколів, але не фільтрує недостовірні метрики.

Token Terminal — орієнтований на фінансову аналітику у стандартизованому форматі, подібному до традиційних фінансів, але має закритий API та високу вартість підписки.

DappRadar — охоплює DeFi, NFT та GameFi, пропонуючи каталоги та рейтинги, але страждає від наявності неактивних проєктів і складного інтерфейсу для новачків.

2 ПРОЕКТУВАННЯ WEB-ЗАСТОСУНКУ ДЛЯ ОЦІНКИ КРИПТОАКТИВІВ

2.1 Визначення функціональних і нефункціональних вимог до застосунку

Дослідивши популярні вебсайти для оцінки криптовалют, цільову аудиторію, основні метрики та категорію криптовалют було сформовано наступні вимоги.

Функціональні вимоги:

- Відображення блокчейнів у вигляді таблиці з можливістю сортування за зростанням або спаданням за ключовими метриками
- Отримання оцінки блокчейну на основі метрик
- Візуалізація результатів у вигляді шкали (від 0 до 100).
- Інтерактивний чат-бот для запиту метрик.
- Виведення таблиць та графіків за метриками.
- Створення таблиці топ 100 криптовалют за рейтингом

Нефункціональні вимоги:

- Загрузка сторінки має бути не більше 3 секунд
- Підтримка основних браузерів (Chrome, Firefox, Safari, Edge).
- Модульна структура застосунку (Assets, Components, Router...)
- Інтерфейс користувача повинен бути адаптивний під мобільні пристрої

2.2. Вибір технічних засобів для розробки застосунку для оцінки криптоактивів

2.2.1. Вибір мови програмування та середовище розробки для оцінки криптоактивів

Для розробки застосунку використовувались основна мова для веб програмування – JavaScript та мови розмітки HTML і CSS.

JavaScript - динамічна, об'єктно-орієнтована прототипна мова програмування.

JavaScript використовується у моєму проекті з метою надання вебсторінкам інтерактивності.

JavaScript стала стандартом в індустрії завдяки своїй універсальності, оскільки працює в усіх сучасних веб-браузерах, що робить її доступною для розробників по всьому світу.

Вона дозволяє додавати динамічний контент на веб-сторінки, значно покращуючи досвід користувачів. Розвиток екосистеми, включаючи численні бібліотеки та фреймворки, такі як jQuery, React, Angular та Vue.js, спрощує розробку складних веб-додатків.

Крім того, величезна спільнота розробників активно підтримує JavaScript, забезпечуючи її постійний розвиток.

HTML — це мова розмітки для створення веб-сторінок. Основний застосунок HTML у моєму веб-додатку, є те що він використовується для форматування структури та вмісту веб-сторінки. HTML працює з CSS та JavaScript, тому сайти можуть бути естетично приємними і зручними у використанні. Ці та інші фактори, такі як підтримка спільноти та процеси, є причинами, чому HTML став домінуючою мовою розмітки для вебу та ключем до написання сучасних веб-додатків.

CSS (Cascading Style Sheets) — це мова стилів, яка використовується для опису зовнішнього вигляду та форматування веб-сторінок, написаних на HTML або XML.

Основна функція CSS у моєму проекті полягає в управлінні кольорами, шрифтами, відступами, розміщенням елементів та іншими аспектами дизайну веб-сторінок. Завдяки CSS розробники можуть легко змінювати стиль і макет веб-сайтів, не змінюючи сам HTML-код, що значно спрощує процес оновлення та підтримки. CSS підтримує каскадність, що дозволяє застосовувати стилі на різних рівнях — від глобальних до локальних, а також використовувати різні медіа-запити для адаптації дизайну під різні пристрої, такі як мобільні телефони, планшети та

настільні комп'ютери. Це робить веб-сторінки більш адаптивними та зручними для користувачів.

Середовище розробки (IDE, від англ. Integrated Development Environment) — це програмне забезпечення, яке надає розробникам зручний набір інструментів для створення, тестування та налагодження програмного забезпечення. Воно об'єднує в одному інтерфейсі різні компоненти, які можуть включати: компілятор, редактор коду, дебагер, системи контролю версій та інструменти для роботи з базами даних.

Visual Studio Code (VS Code) — легкий, розширюваний та кросплатформенний редактор коду, розроблений компанією Microsoft. Вибір саме VS Code обумовлений його численними перевагами для мого проєкту:

- Широкою підтримкою мов програмування і фреймворків, зокрема JavaScript та Vue.js.
- Наявністю великої кількості розширень для роботи з Git, API, базами даних та автодоповненням коду.
- Вбудованим терміналом для виконання команд; зручним і зрозумілим інтерфейсом, а також ефективною підтримкою модульної структури проєкту, що є критично важливою для організації масштабованого веб-застосунку.

Завдяки цим характеристикам, VS Code забезпечив ефективну та комфортну розробку програмного продукту.[8]

2.2.2 Вибір веб-фреймворків та програмних бібліотек

Веб-фреймворк — це програмне забезпечення, яке надає структуру та набір інструментів для розробки веб-додатків. Він спрощує процес створення веб-сайтів і веб-сервісів, забезпечуючи готові рішення для поширених задач, таких як обробка запитів, управління сесіями, маршрутизація, безпека та доступ до бази даних.

Мною був обраний фреймворк Vue.js, який є прогресивним JavaScript-фреймворком, який пропонує численні переваги, зокрема простоту використання

Одна з ключових характеристик Vue.js — це компонентний підхід, який сприяє створенню модульних і повторно використовуваних компонентів.

Це значно спрощує управління кодом, підвищує його читабельність і дозволяє легко масштабувати проекти. Крім того, реактивна система Vue.js автоматично оновлює інтерфейс при зміні даних, що забезпечує плавну взаємодію з користувачем.

Vue.js забезпечує чітку структуру компонентів, у якій стилі, скрипти та HTML-код об'єднані в одному файлі. Це дозволяє уникнути необхідності створювати безліч окремих файлів у проекті.[9]

Проте, незважаючи на свої переваги, Vue.js має і деякі недоліки. Порівняно з іншими фреймворками, такими як React, Vue.js може мати меншу підтримку в деяких випадках, що може вплинути на доступність ресурсів і бібліотек. Нарешті, у великих проектах, де використовується багато сторонніх плагінів, може виникнути проблема з інтеграцією, що ускладнює підтримку коду.

Розглянемо основні бібліотеки проекту.

Axios — бібліотека для надсилання HTTP-запитів до зовнішніх API.

Chart.js — популярна бібліотека для створення інтерактивних графіків, яка підтримує багато типів діаграм (лінійні, стовпчикові, кругові тощо).

Vue Router — офіційна бібліотека Vue.js для організації маршрутизації між сторінками застосунку.

Element Plus — це популярна UI-бібліотека компонентів для Vue 3, яка надає набір готових елементів інтерфейсу, зокрема таблиці, форми, діалогові вікна.

2.2.3 Система контролю версій

В якості системи контролю версій був обраний GitHub — веб-платформа, що базується на Git, яка дозволяє зберігати репозиторії з кодом в хмарі. GitHub надає інструменти для спільної роботи, такі як пул-реквести, проблеми та вікна для обговорення, що робить його популярним серед розробників для ведення відкритих та приватних проектів може бути повільнішим і більш ресурсоємним.

Git — це безкоштовна розподілена система контролю версій з відкритим кодом, створена Лінусом Торвальдсом. Git вимагає низьких операційних витрат, але при цьому є достатньо гнучким і потужним, щоб задовольнити потреби

складних і масштабних проєктів з розробки розподіленого програмного забезпечення.

Простіше кажучи, Git - це трекер контенту. З огляду на це, Git поділяє загальні принципи більшості систем контролю версій. Однак відмінною рисою, яка робить Git унікальним серед розмаїття інструментів, є те, що це розподілена система контролю версій. Ця відмінність означає, що Git є швидким і масштабованим, має багатий набір команд, який надає доступ до високорівневих і низькорівневих операцій, а також оптимізований для локальних операцій.[10]

Важливо дотримуватись єдиного стилю повідомлень комітів (наприклад, Conventional Commits), щоб забезпечити зрозумілу історію змін та полегшити командну роботу.

2.2.4 Система зберігання та керування даними

У межах реалізації проєкту особливу увагу було приділено вибору ефективної системи зберігання даних. З огляду на характер інформації (ринкові метрики, дані з API, CSV-експорти тощо), було обрано NoSQL-базу даних Supabase.

Supabase — це сучасна хмарна платформа з відкритим кодом, яка поєднує у собі функціональність Firebase-альтернативи та можливості традиційних реляційних та нереляційних СУБД. Supabase базується на PostgreSQL, але підтримує гнучку роботу з JSON-полями, що дозволяє ефективно зберігати напівструктуровані дані, отримані з аналітичних сервісів (наприклад, Coingecko, Artemis тощо).[11]

Використання Supabase зумовлено такими чинниками:

- Простота інтеграції з JavaScript-фреймворками, включно з Vue.js, що використовується в даному проєкті.
- Наявність RESTful API для прямого доступу до даних.
- Висока швидкість зчитування і запису, що є критичним при обробці оновлюваних ринкових метрик.
- Гнучкість зберігання: можливість роботи з масивами, об'єктами та JSON-структурами без необхідності створення складної схеми.

Таким чином, Supabase задовольняє вимоги до масштабованості, швидкодії та зручності доступу до даних, що робить її оптимальним вибором для реалізації проєкту в умовах динамічного оновлення інформації.

2.3. Архітектура застосунку для оцінки криптоактивів

Мій додаток використовує клієнт-серверну архітектуру.

Модель клієнт-сервер – одна з найпопулярніших моделей для веб-розробки, де система поділена на дві основні частини – клієнт (користувацький інтерфейс) та сервер (джерело/обробник даних). Ця парадигма дозволила логічно розрізняти обов'язки і мати масштабованість, повторне використання частин, а також полегшене обслуговування.

Клієнтська частина реалізована за допомогою фреймворку Vue.js, який відповідає за відображення інтерфейсу, взаємодію з користувачем та ініціацію запитів. Всі HTTP-запити реалізовано через бібліотеку Axios.

Серверна частина використовує зовнішні API:

- CoinGecko API для даних ринку криптовалют та отримання метрик market cap , bitcoin price, btc.d.
- DeFiLlama API для отримання fees , tvl метрик.
- TokenMetrics API для отримання відповіді від ai чат боту.
- Також використовується Supabase , хмарна платформа даних, яка включає командний інтерфейс для створення, управління та використання бази даних PostgreSQL з REST API для відстеження метрик користувача і внутрішніх метрик, з якими клієнт працює через HTTP-запити.

Можна виділити ключові переваги та недоліки клієнт серверної архітектури.

Переваги:

- Чітке розділення функцій: клієнт відповідає за взаємодію з користувачем, а сервер — за обробку та зберігання даних.
- Сервер можна оновлювати або змінювати незалежно від клієнта, що спрощує підтримку та масштабування.

Недоліки:

- Якщо сервер недоступний або несправний, клієнт не може виконувати свою роботу.
- Затримки у комунікації: Відповідь сервера може займати час, що впливає на швидкість роботи додатка, особливо при великій кількості користувачів або складних запитах.

Архітектура застосунку представлена на рисунку 2.1.

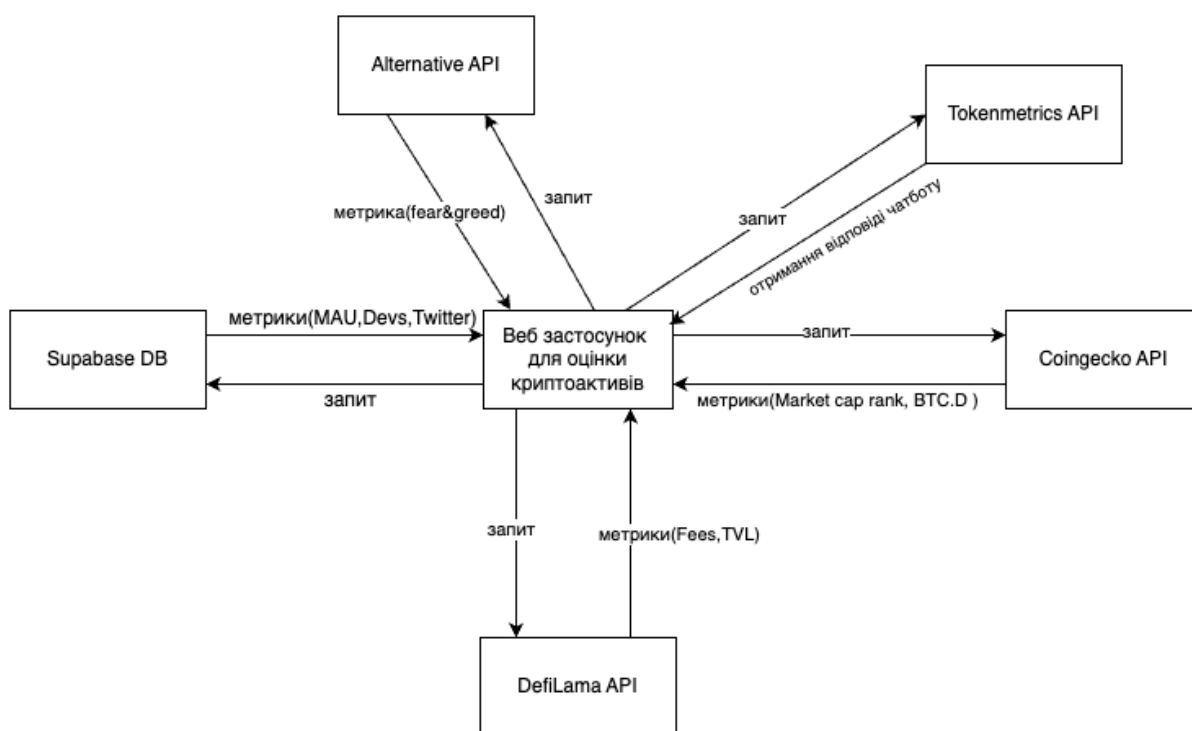


Рис 2.1 Архітектура застосунку

Діаграма послідовності роботи клієнт-сервер-база даних представлена на рисунку 2.2.

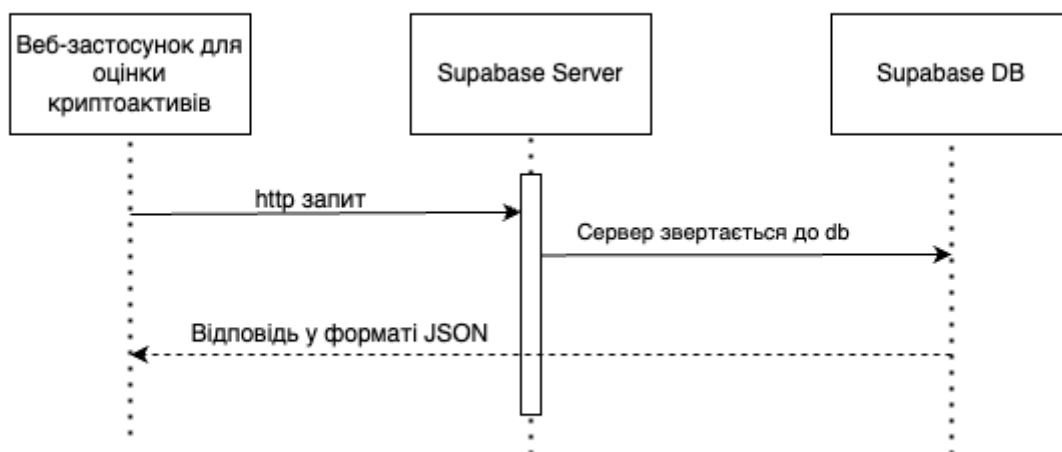


Рис 2.2 Діаграма послідовності Клієнт-Сервер-База даних для веб-застосунку для оцінки криптоактивів

2.4. Розробка функціональних вимог та діаграма варіантів використання

Оглянемо ключові функції веб-застосунку. Діаграма варіантів використання описує основні дії користувача та взаємодію з системою.

Основні дії, доступні користувачеві системи, наведені нижче:

Сортування та фільтрація даних: користувач може самостійно сортувати таблиці або графіки за вибраними метриками, а також застосовувати фільтри для вибірки потрібної інформації.

Інтерактивні графіки: перегляд метрик у візуальній формі з можливістю масштабування, аналізу історії зміни показників тощо.

Огляд ринкових метрик: користувач має змогу переглядати основні показники криптовалютного ринку, зокрема дані про Bitcoin, BTC.D та індекс страху.

Аналіз криптовалют: введення назви криптовалюти для отримання її комплексного рейтингу за шкалою ChainScore з урахуванням TVL, активності розробників, ринкової капіталізації, MAU, соціальних метрик та інших факторів.

Використання AI-чат-бота: користувач має змогу звернутися до інтегрованого AI-асистента для отримання аналітики, порад або інформації про криптовалюту, відправивши відповідний запит.

Перегляд топ-100 криптоактивів: можливість ознайомлення з найрейтинговішими криптовалютами, відсортованими за різними показниками, такими як ціна, капіталізація, зміна курсу тощо

Мовні налаштування та зміна теми: перемикання інтерфейсу між українською та англійською мовами.

Діаграма варіантів використання представлена на рисунку 2.3.



Рис 2.3 Діаграма варіантів використання для веб застосунку для оцінки криптоактивів

2.5 Аналітичні системи для оцінки криптоактивів

У межах даного проекту використовуються сучасні аналітичні сервіси для збору, структурування та інтерпретації даних про криптоактиви. Такі сервіси надають доступ до численних фінансових і ринкових метрик через публічні API або інтерактивні інтерфейси, що значно спрощує реалізацію системи оцінки цифрових активів.

Зокрема, у роботі застосовано ресурси CoinGecko, DefiLlama, TokenMetrics та Alternative.me, кожен з яких виконує специфічну функцію — від моніторингу ринкової капіталізації та обсягу торгів до надання індексу страху та жадібності (Fear & Greed Index) або показників Total Value Locked (TVL) у децентралізованих протоколах.

Сервіс CoinGecko є одним із найпопулярніших агрегаторів ринкової інформації про криптовалюту. Він надає дані про ринкову капіталізацію, обсяг торгів, динаміку цін, соціальні сигнали та інші метрики. API CoinGecko є безплатним, підтримує широкий спектр запитів і активно використовується в дослідницьких і комерційних проєктах.

DefiLlama спеціалізується на зборі даних із DeFi-протоколів. Його ключова метрика — Total Value Locked (TVL) — відображає загальний обсяг заблокованих коштів у смарт-контрактах. Платформа також підтримує порівняння між блокчейнами та протоколами, надаючи точну інформацію про ефективність і масштаби DeFi-проєктів.

TokenMetrics поєднує аналітику з елементами штучного інтелекту та машинного навчання для оцінки перспективності криптоактивів. Платформа надає прогностні оцінки, рейтинги, індекси й аналітичні звіти. На відміну від попередніх сервісів, TokenMetrics є комерційним продуктом із закритим API.

Alternative.me . Платформа Alternative.me відома насамперед через індекс страху та жадібності (Fear & Greed Index), що використовується для оцінки

загального емоційного стану ринку. Індекс ґрунтується на кількох чинниках: волатильності, обсягах торгів, соціальних мережах, домінуванні Bitcoin тощо. Це дозволяє використовувати його як психологічний індикатор для прогнозування ринкових рухів.

Платформа Artemis.xyz є сучасним аналітичним інструментом для моніторингу широкого спектра блокчейн-метрик, зокрема активності користувачів, обсягу транзакцій, доходу протоколів, кількості активних розробників тощо. На відміну від інших сервісів, Artemis не надає відкритого API для інтеграції, проте дозволяє експортувати дані у форматі CSV безпосередньо з інтерфейсу платформи. У межах реалізації даного проєкту було використано CSV-файли, експортовані з Artemis, для формування локальної бази даних. Це дало змогу інтегрувати специфічні метрики, які були відсутні або неповні в інших джерелах (зокрема, дані про developer activity, MAU, revenue та інші).[12]

3 РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ОЦІНКИ КРИПТОАКТИВІВ

3.1 Проектування інтерфейсу користувача

UX — це досвід користувача, який охоплює всі аспекти взаємодії користувача з продуктом або послугою. Це включає в себе не лише візуальні елементи, але й функціональність, зручність, задоволення та ефективність використання.[13]

UI — це інтерфейс користувача, який включає всі візуальні елементи, з якими користувач взаємодіє. Це можуть бути кнопки, меню, іконки, текстові поля, графіки та інші елементи, які формують зовнішній вигляд застосунку або веб-сайту.

UI/UX дизайн — це процес створення зручного, інтуїтивного інтерфейсу користувача та забезпечення позитивного досвіду взаємодії із застосунком. У рамках проектування було здійснено аналіз дизайну популярних рішень оцінки криптоактивів таких як DappRadar, Token Terminal, DefiLama.

На основі цього аналізу були сформовані функціональні вимоги до майбутнього застосунку, а також визначені основні сторінки та відповідні UI-елементи:

На рисунку 3.1 зображено реалізацію ui елементів у макеті Figma.

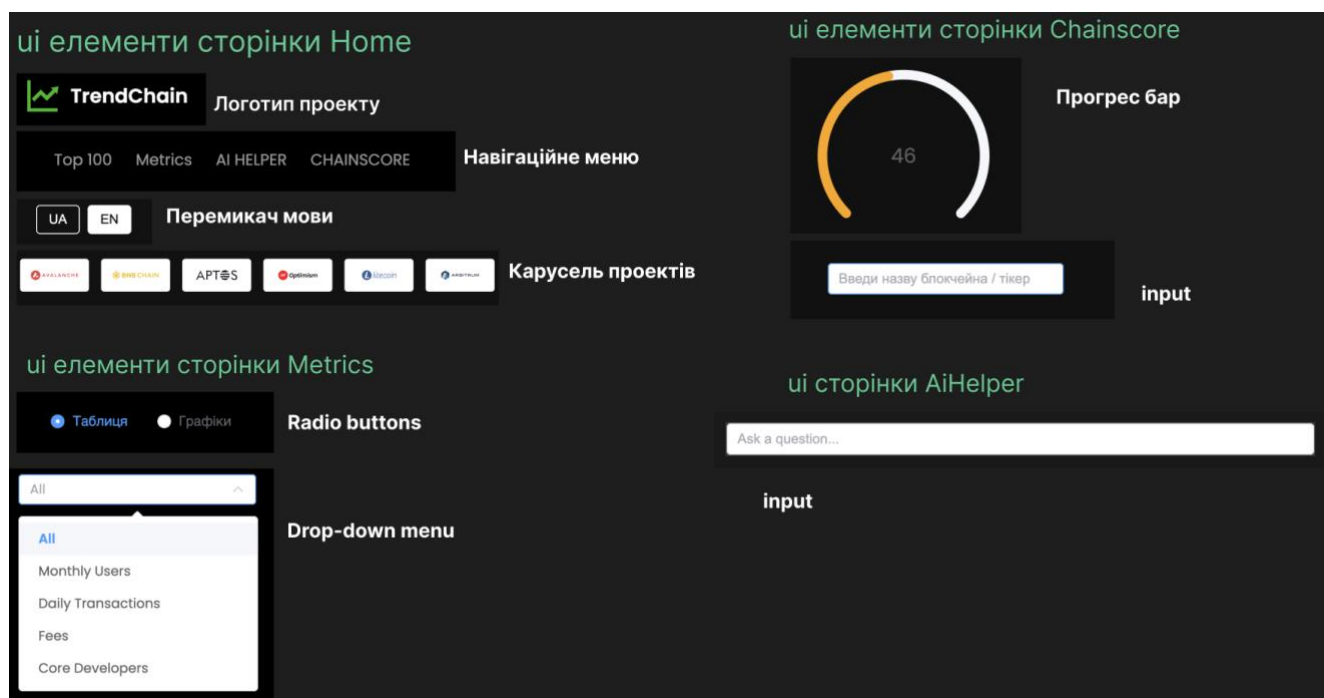


Рис 3.1 Реалізація ui елементів.

Головна сторінка (Home): відображає основну інформацію про проєкт; містить логотип, верхній хедер з навігаційним меню (Top 100, Metrics, AI Helper, ChainScore), перемикач мови (UA/EN), а також UI-елемент у вигляді автокаруселі логотипів блокчейнів, нижній футер з іншою інформацією про контакти, сторінки у мережах та інше.

Сторінка Metrics: включає radio buttons для перемикання між режимами відображення (графіки / таблиці), а також drop-down menu для вибору конкретної метрики для візуалізації; реалізовані bar charts, адаптивні графіки з кастомізацією, а також інтерактивні таблиці з функцією сортування.

Сторінка ChainScore: включає input, в який користувач вводить назву блокчейна; після введення система обробляє дані та виводить результат у вигляді кругового прогрес-бара з анімацією.

Сторінка AiHelper: включає input у який користувач водить своє питання і отримує відповідь від сервера(чат бота).

Сторінка Top 100: подана у готовій таблиці за допомогою бібліотеки ElementPlus де можна сортувати блокчейни за метриками.

3.2 Реалізація серверної частини

Серверна частина застосунку побудована на основі інтеграції з відкритими зовнішніми API та кастомною базою даних. Основні метрики, такі як загальна вартість заблокованих активів (TVL), щоденні комісії (Fees) та ринковий ранг (Market Cap Rank), отримуються в реальному часі з публічних джерел — зокрема, через API платформ DeFiLlama та CoinGecko. Це забезпечує стабільне надходження фінансових і статистичних показників для оцінки криптоактивів.

Однак низка важливих метрик, таких як щомісячна активність користувачів, кількість щоденних транзакцій, кількість активних розробників за тиждень та кількість підписників у Twitter), не надається цими сервісами у придатному для автоматичного оброблення форматі. З метою збереження та регулярного оновлення таких даних було реалізовано окреме серверне сховище на базі Supabase, яке виступає як хмарна реляційна база даних і дозволяє здійснювати зчитування, оновлення та вибірку необхідної інформації.

Джерелом цих розширених метрик є аналітична платформа Artemis, яка надає файли у форматі JSON та CSV з повним набором даних по блокчейн-проектах. Отримані дані обробляються та періодично завантажуються у Supabase, де вони стають доступними для подальшого використання на клієнтській частині застосунку.

На рисунку 3.2 представлено ERD-схему зовнішніх та внутрішніх даних.

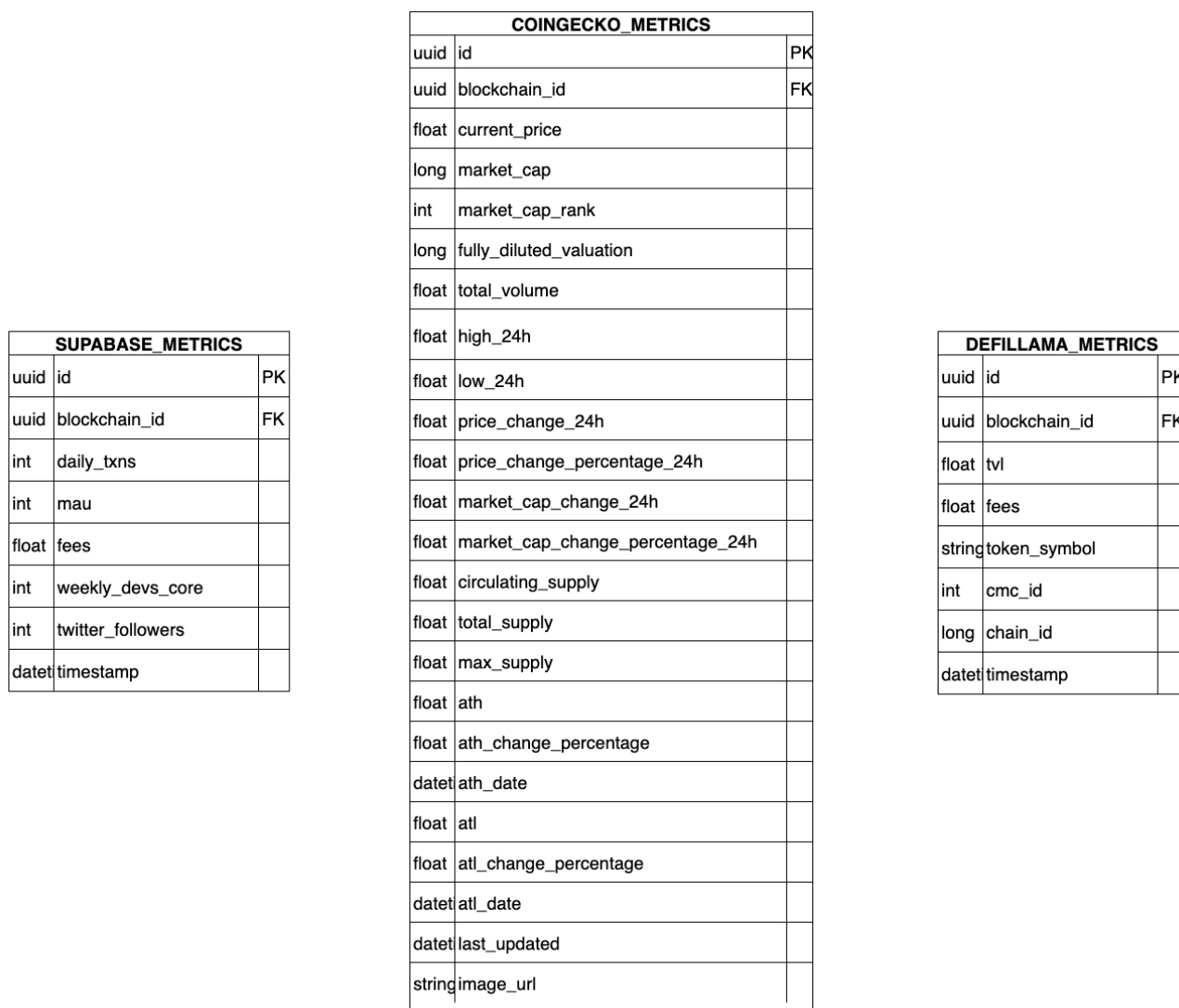


Рис 3.2 ERD-схема зовнішніх та внутрішніх даних

3.3 Реалізація клієнтської частини

Ініціалізуємо проект Vue.js на vite що забезпечить швидкість розробки завдяки застосуванню нативного ESM (ES Modules) і попередньому компілюванню залежностей. Для запуску проєкту використовувалась команда `npm init vite@latest`, далі було обрано шаблон `vue`.

На наступному етапі доцільно встановити додаткові бібліотеки, які забезпечать функціональність веб-застосунку. Серед них: Vue.js 3, Vue Router, Element Plus для компонентів інтерфейсу, а також Chart.js для візуалізації даних.

Встановлення цих бібліотек було здійснено за допомогою команди `npm install vue-router element-plus chart.js`.

Відповідно до нефункціональних вимог, була створена модульна структура проекту, що дозволяє організувати код у зрозумілій та логічній формі.

Нижче наведено діаграму структури застосунку, що ілюструє взаємозв'язки між сторінками, компонентами та зовнішніми бібліотеками.

На рисунку 3.3 показано діаграму структури застосунку.

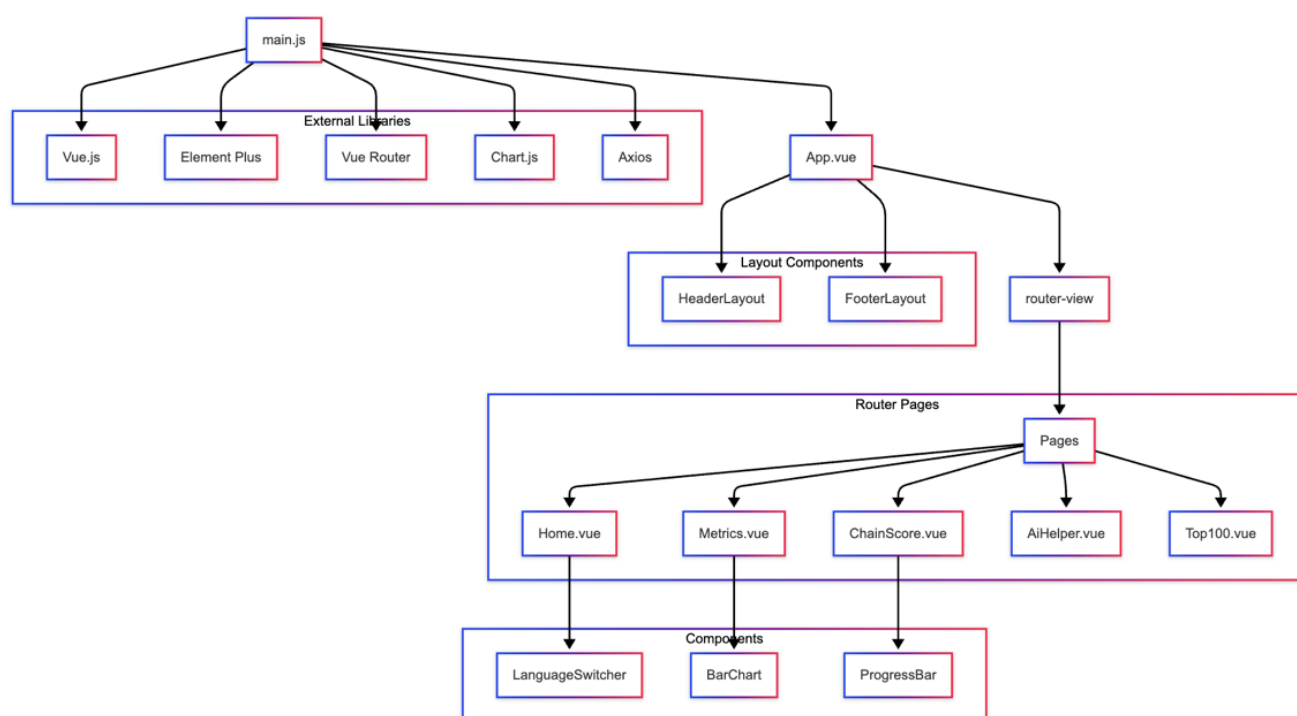


Рис 3.3 Діаграму структури застосунку

Модульна структура сприяє легшій підтримці, масштабованості та повторному використанню коду. Основні функціональні блоки, зокрема компоненти інтерфейсу, сторінки, маршрути, утиліти та зовнішні сервіси, винесено у відповідні директорії. Це дозволяє чітко відокремити логіку роботи з інтерфейсом від бізнес-логіки.[14]

На рисунку 3.4 показано приклад модульної структури у вигляді ієрархії папок, реалізованої у середовищі Visual Studio Code.

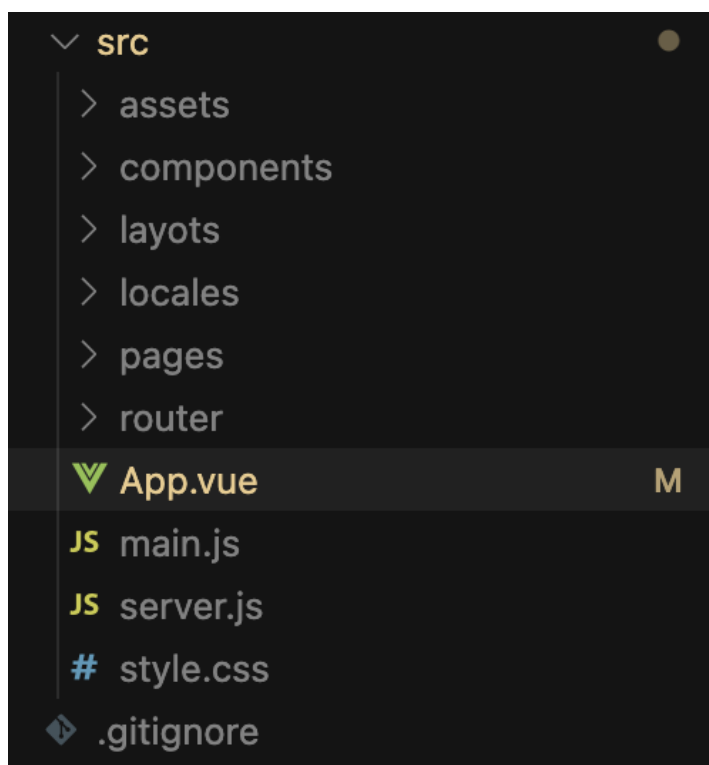


Рис 3.4 Приклад модульної структури

Опис папок:

assets/: Ця папка містить статичні ресурси, такі як зображення, шрифти та стилі, які використовуються в проекті. Вона забезпечує централізоване місце для зберігання всіх медіа-елементів.

components/: У цій папці розміщуються багаторазові Vue компоненти, які можуть використовуватися в різних частинах застосунку. Це дозволяє зберігати код чистим і організованим, а також сприяє повторному використанню компонентів.

layouts/: Тут зберігаються шаблони сторінок, які визначають загальну структуру та оформлення для різних частин застосунку. Це може включати заголовки, футери та інші загальні елементи, що повторюються на різних сторінках.

pages/: У цій папці містяться окремі сторінки застосунку, що відповідають маршрутам. Кожна сторінка може бути представлена окремим компонентом, що полегшує управління навігацією.

`router/`: Ця папка містить конфігурацію маршрутизації, яка визначає, як користувачі переміщуються між різними сторінками застосунку. Це важливий аспект для забезпечення зручності навігації.

`locales/`: У цій папці зберігаються файли перекладу, необхідні для реалізації багатомовного інтерфейсу. Це дозволяє підтримувати різні мови, такі як українська (UA) та англійська (EN), забезпечуючи доступність застосунку для ширшої аудиторії.

Після процесу ініціалізації автоматично створено основний шаблон програми — компонент `App.vue`, який виступає в ролі контейнера для всієї системи. У ньому реалізовано базову розмітку, підключено компонент маршрутизації `<router-view>`, а також загальні стилі та конфігурації. Наступним кроком стала розробка головних макетів (`layout-компонентів`), що повторюються на кожній сторінці застосунку.

`HeaderLayout.vue` — верхня панель навігації з кнопками переходу між сторінками та логотипом

`FooterLayout.vue` — нижній колонтитул із загальною інформацією про проєкт та посиланнями. Після цього було налаштовано маршрутизацію у файлі `router/index.js`, де визначено основні маршрути до сторінок, а також реалізовано механізм завантаження відповідних компонентів.

Кожна сторінка застосунку (наприклад, головна, `ChainScore`, `Top-100`, `Metrics`, `aiHelper`) була створена як окремий компонент у директорії `pages/`. Завдяки цьому забезпечується розділення відповідальності та можливість гнучко налаштовувати логіку для кожного розділу інтерфейсу

Імплементація системи локалізації стала важливим етапом у розробці інтерфейсу застосунку. Для забезпечення підтримки української та англійської мов було створено компонент `LanguageSwitcher.vue`, який відповідає за перемикання між мовами. У директорії `translations/` розміщено два основні файли локалізації: `ua.js` та `en.js`, що містять об'єкти з перекладами всіх текстових елементів інтерфейсу.

Для реалізації багатомовності на головній сторінці (Home page) було впроваджено систему локалізації. Основним компонентом став LanguageSwitcher.vue, розміщений у хедері застосунку, який надає користувачу можливість перемикання між українською та англійською мовами. У файлі translations.js створено об'єкт, що експортується, з структурованими перекладами для головної сторінки.

У компоненті Home.vue реалізовано логіку відображення контенту відповідно до обраної мови.

На головній сторінці (Home page) реалізовано три основні секції.

Перед першою секцією, безпосередньо під логотипом, розміщено інформаційну смужку з ключовими ринковими метриками, зокрема: поточною ціною Bitcoin (BTC Price), домінуванням Bitcoin (BTC Dominance) та індексом страху та жадібності (Fear & Greed Index). Під час завантаження сторінки ініціюються HTTP-запити до відповідних аналітичних сервісів, зокрема CoinGecko (для отримання значень BTC Price та BTC Dominance) та Alternative.me (для отримання Fear & Greed Index). Отримані дані відображаються у реальному часі, забезпечуючи користувача актуальною інформацією про стан ринку вже при першому візуальному контакті з інтерфейсом.

Перша секція (Hero) містить основний заголовок та опис платформи, доповнений анімованим прогрес-баром, який демонструє динамічну зміну даних. У другій секції (Features) розміщено картки з описом особливостей платформи, організовані у вигляді grid-сітки для зручного відображення.

Третя секція (Why) включає графічне представлення даних та пояснення важливості метрик для аналізу блокчейн-мереж.

На рисунку 3.5 показано приклад сторінки Home web-застосунку для оцінки криптоактивів.

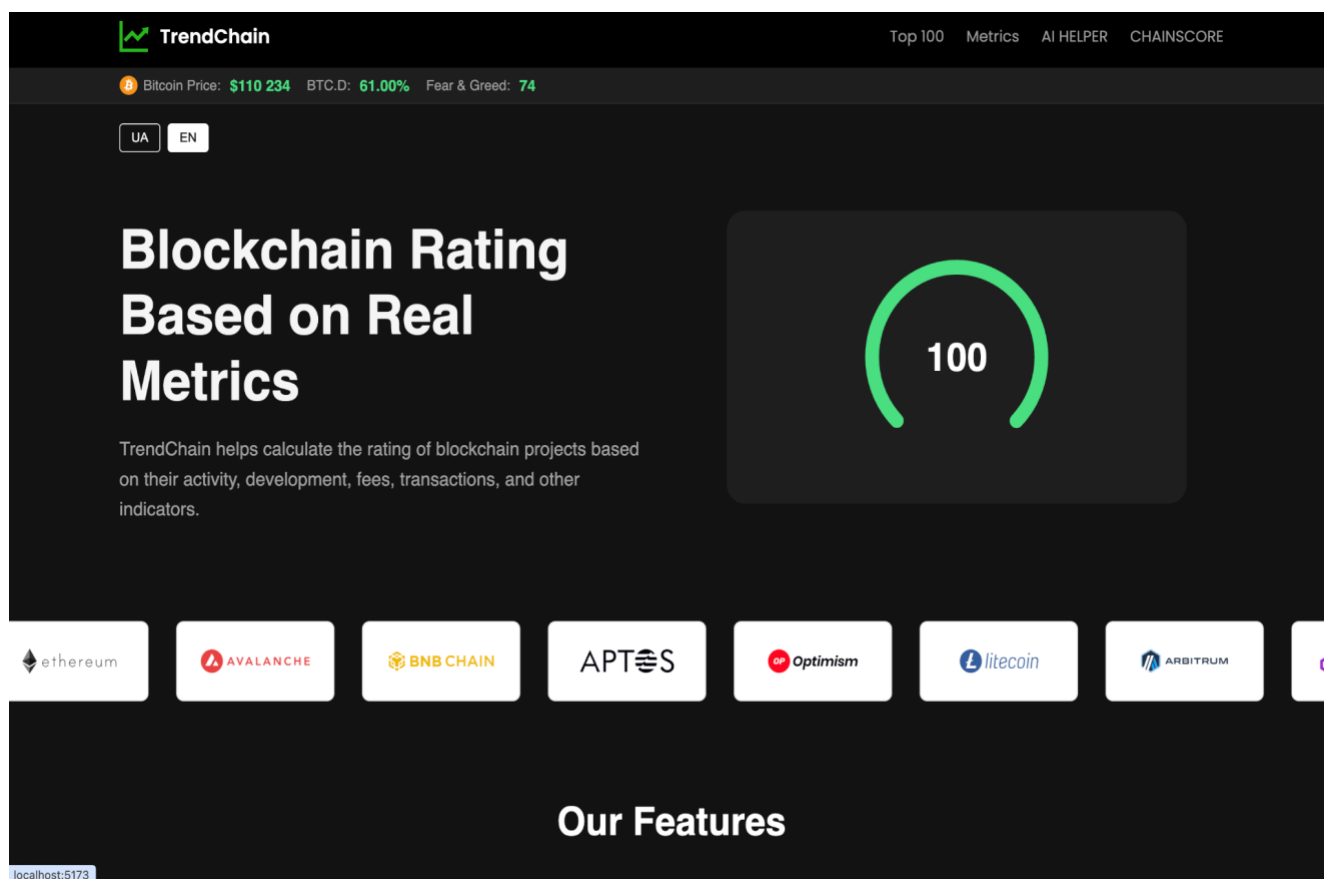
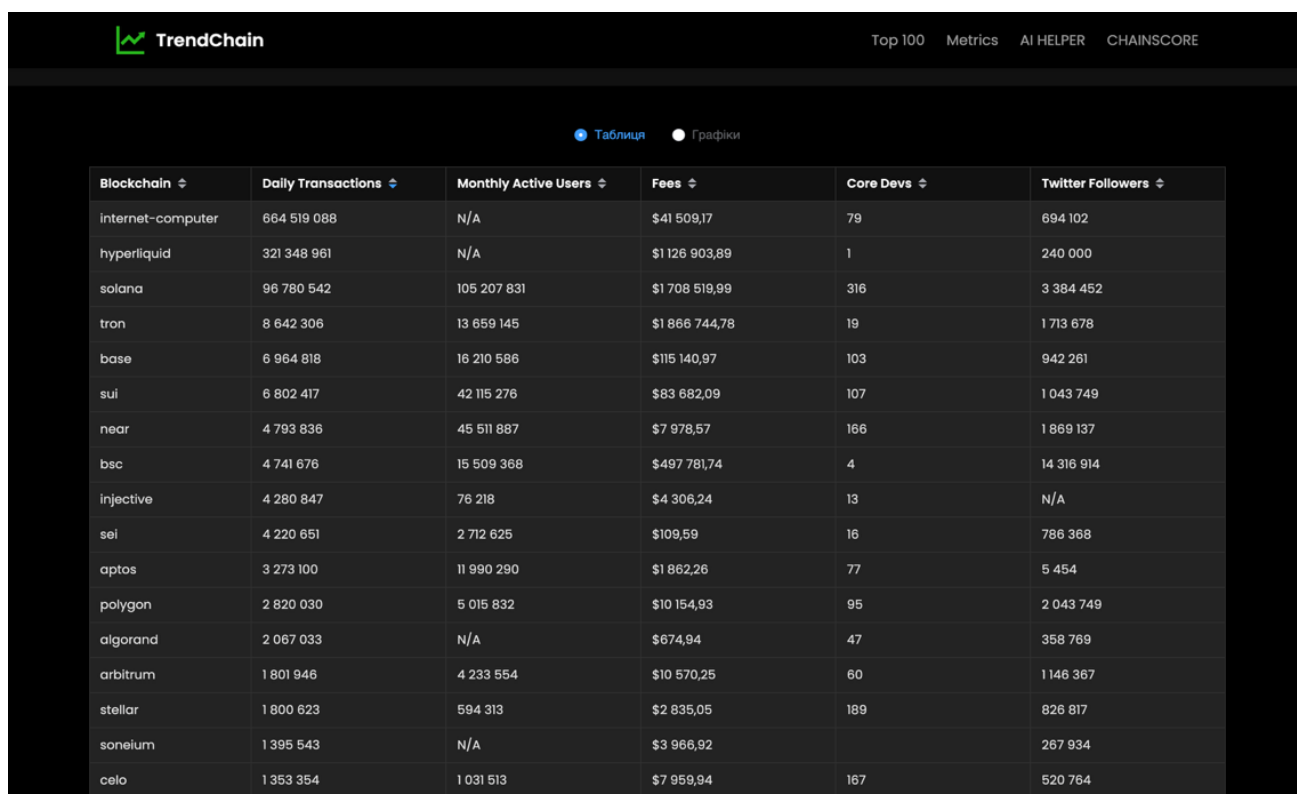


Рис 3.5 Головна сторінка Home

Сторінка Metrics відповідає за візуалізацію даних з метрик криптоактивів. На ній реалізовано перемикач режиму перегляду у вигляді radio buttons, що дозволяє обирати між таблицею та графіком. Для вибору конкретної метрики використовується випадаючий список (drop-down menu). Графіки створено за допомогою бібліотеки Chart.js, а саме — компонентів BarChart та RadarChart, які забезпечують кастомізовану візуалізацію обраних метрик. Для табличного подання використовується компонент MetricsTable, побудований на основі Element Plus Table, що підтримує сортування, фільтрацію та адаптивне відображення.

На рисунку 3.6 показано приклад сторінки Metrics web-застосунку для оцінки криптоактивів.



The screenshot shows the TrendChain website's Metrics page. At the top, there's a navigation bar with 'Top 100', 'Metrics', 'AI HELPER', and 'CHAINSCORE'. Below the navigation bar, there are tabs for 'Таблиця' (Table) and 'Графіки' (Charts). The main content is a table with 6 columns: Blockchain, Daily Transactions, Monthly Active Users, Fees, Core Devs, and Twitter Followers. The table lists various blockchain projects and their corresponding metrics.

Blockchain	Daily Transactions	Monthly Active Users	Fees	Core Devs	Twitter Followers
internet-computer	664 519 088	N/A	\$41 509,17	79	694 102
hyperliquid	321 348 961	N/A	\$1 126 903,89	1	240 000
solana	96 780 542	105 207 831	\$1 708 519,99	316	3 384 452
tron	8 642 306	13 659 145	\$1 866 744,78	19	1 713 678
base	6 964 818	16 210 586	\$115 140,97	103	942 261
sui	6 802 417	42 115 276	\$83 682,09	107	1 043 749
near	4 793 836	45 511 887	\$7 978,57	166	1 869 137
bsc	4 741 676	15 509 368	\$497 781,74	4	14 316 914
injective	4 280 847	76 218	\$4 306,24	13	N/A
sei	4 220 651	2 712 625	\$109,59	16	786 368
aptos	3 273 100	11 990 290	\$1 862,26	77	5 454
polygon	2 820 030	5 015 832	\$10 154,93	95	2 043 749
algorand	2 067 033	N/A	\$674,94	47	358 769
arbitrum	1 801 946	4 233 554	\$10 570,25	60	1 146 367
stellar	1 800 623	594 313	\$2 835,05	189	828 817
sonelium	1 395 543	N/A	\$3 966,92		267 934
celo	1 353 354	1 031 513	\$7 959,94	167	520 764

Рис 3.6 Сторінка Metrics

На сторінці ChainScore реалізовано механізм оцінювання блокчейн-мереж за сукупністю ключових метрик. Основним UI-елементом є текстове поле (input), в яке користувач вводить назву криптовалюти або тікер (коротка назви по типу btc). Після обробки запиту і використання формули, система формує загальну оцінку від 1 до 100, яка виводиться у вигляді кругового прогрес-бара з анімацією, також виводяться усі метрики за якими була оцінка криптоактиву. Для візуального подання рівня якості криптоактиву в застосунку використовується кольоровий прогрес-бар із фіксованими діапазонами значень. Колір індикатора змінюється відповідно до підсумкової оцінки за шкалою, що дозволяє швидко оцінити поточний стан блокчейн-проєкту:

Червоний колір (0–25) — індикатор критично низьких показників. Позначає високий рівень ризику, нестабільність проєкту, відсутність достатніх ончейн-метрик або низьку активність користувачів та розробників.

Помаранчевий колір (26–50) — індикатор середніх показників. Вказує на помірний рівень ризику та необхідність проведення додаткового аналізу проєкту, оскільки дані є неоднозначними або недостатніми.

Зелений колір (51–100) — індикатор високих та стабільних показників. Свідчить про наявність позитивних ончейн-метрик, активне використання мережі та низький рівень інвестиційного ризику.

На початковому етапі розробки була розглянута ідея простого додавання значень метрик для формування підсумкової оцінки криптоактиву. Однак такий підхід виявився неефективним з огляду на значні відмінності у шкалах значень різних показників. Зокрема, метрики на кшталт TVL можуть досягати мільярдів доларів, тоді як соціальні метрики або активність розробників вимірюються у тисячах. Це призводить до ситуації, коли окремі показники непропорційно впливають на фінальний результат, знецінюючи решту даних. Крім того, в процесі дослідження було виявлено, що після перших 20 позицій у рейтингу (топ 100 криптовалют) значення ключових метрик стрімко знижуються: наприклад, проєкти з топ-10 можуть мати TVL понад один мільярд доларів, тоді як далі цей показник падає до сотень або навіть десятків мільйонів.

Щоб вирішити цю проблему, для кожної метрики було застосовано нормалізацію до контрольованого діапазону за допомогою простих формул, які зменшують вплив абсолютних значень та дозволяють зберігати пропорції між показниками.

ChainScore розраховується як сума шести ключових метрик, кожна з яких нормалізується до певного діапазону значень:

Розберемо кожну складову:

- Метрика TVL формула: $\text{TVL} / 100,000,000$
- Метрика Market Cap Rank формула: $(1/\text{rank}) * 100$ (Інвертований ранг)
- Метрика Fees формула: $30\text{-денні комісії} / 1,000,000$
- Метрика MAU формула: $\text{MAU} / 1,000,000$ (Кількість користувачів в мільйонах)
- Метрика Twitter формула: $\text{Кількість підписників} / 100000$

– Метрика Developer Activity (за тиждень) формула: Кількість активних розробників/15

Таким чином, значення рейтингу ChainScore формується як сума результатів шести нормалізованих метрик, що дозволяє представити комплексну оцінку криптоактиву у межах шкали від 0 до 100.

На рисунку 3.7 показано приклад сторінки ChainScore web-застосунку для оцінки криптоактивів.

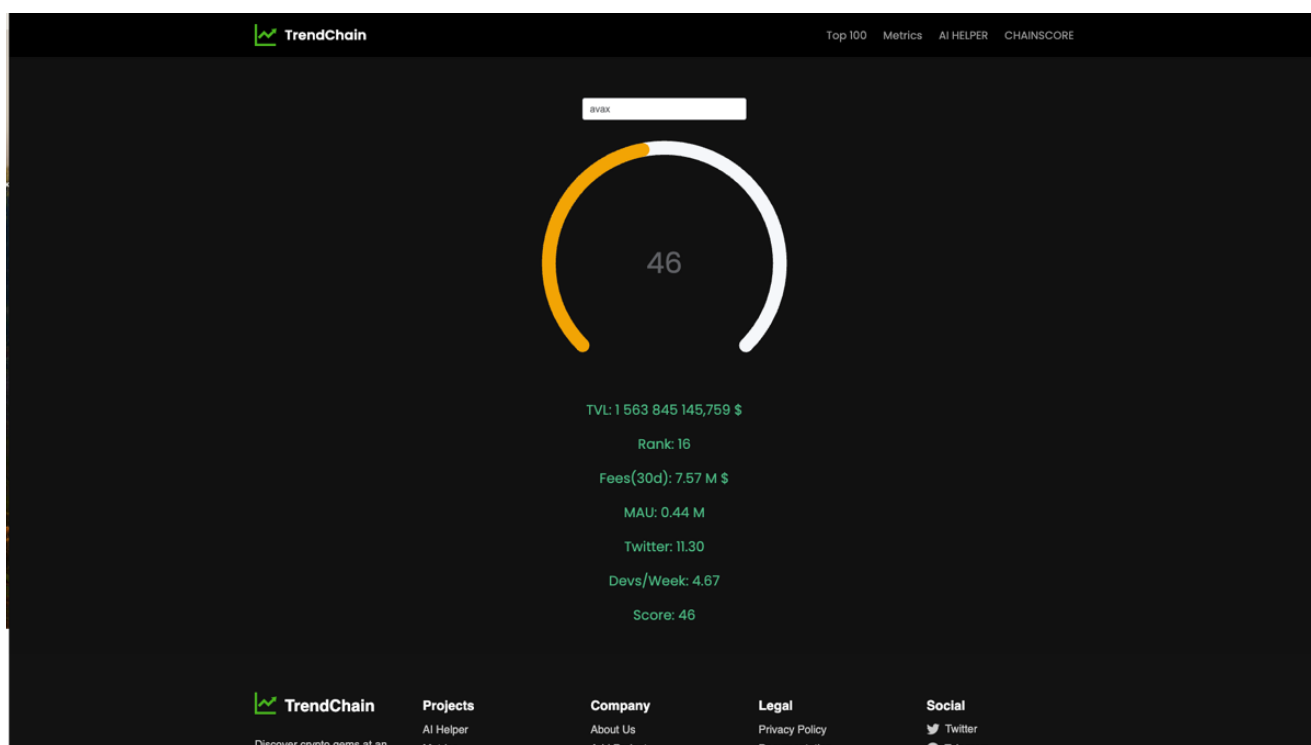
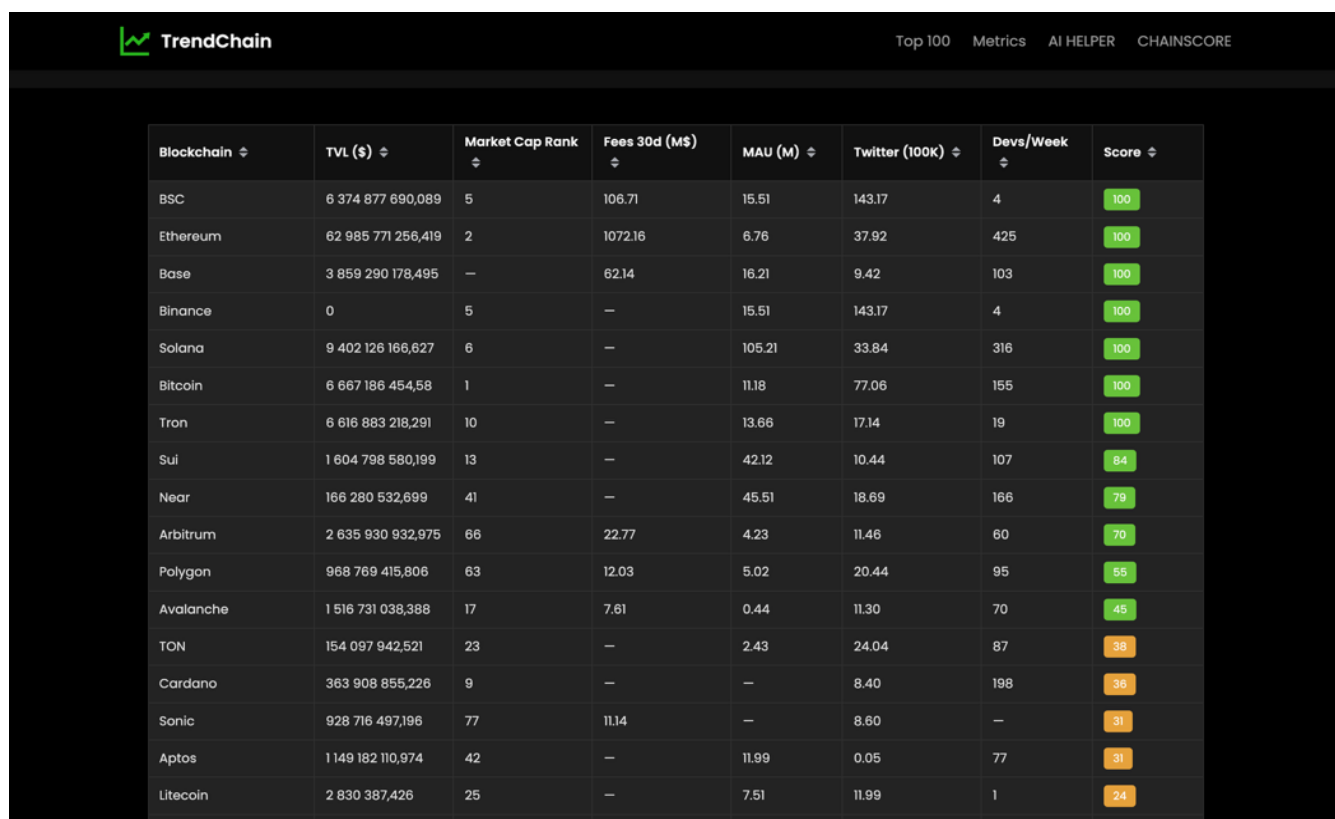


Рис 3.7 Сторінка ChainScore

Сторінка Top 100 реалізована як інтерактивна таблиця на основі Element Plus, з можливістю сортування за всіма доступними метриками: ринковою капіталізацією, TVL, комісіями, активними користувачами тощо.

Порядок у списку топ 100 визначається за допомогою спеціальної формули chainscore, яка агрегує різні метрики в єдиний комплексний показник.

На рисунку 3.8 показано приклад сторінки Top 100 web-застосунку для оцінки криптоактивів.



Blockchain ↕	TVL (\$) ↕	Market Cap Rank ↕	Fees 30d (M\$) ↕	MAU (M) ↕	Twitter (100k) ↕	Devs/Week ↕	Score ↕
BSC	6 374 877 690,089	5	106.71	15.51	143.17	4	100
Ethereum	62 985 771 256,419	2	1072.16	6.76	37.92	425	100
Base	3 859 290 178,495	—	62.14	16.21	9.42	103	100
Binance	0	5	—	15.51	143.17	4	100
Solana	9 402 126 166,627	6	—	105.21	33.84	316	100
Bitcoin	6 667 186 454,58	1	—	11.18	77.06	155	100
Tron	6 616 883 218,291	10	—	13.66	17.14	19	100
Sui	1 604 798 580,199	13	—	42.12	10.44	107	84
Near	166 280 532,699	41	—	45.51	18.69	166	79
Arbitrum	2 635 930 932,975	66	22.77	4.23	11.46	60	70
Polygon	968 769 415,806	63	12.03	5.02	20.44	95	55
Avalanche	1 516 731 038,388	17	7.61	0.44	11.30	70	45
TON	154 097 942,521	23	—	2.43	24.04	87	38
Cardano	363 908 855,226	9	—	—	8.40	198	36
Sonic	928 716 497,196	77	11.14	—	8.60	—	31
Aptos	1 149 182 110,974	42	—	11.99	0.05	77	31
Litecoin	2 830 387,426	25	—	7.51	11.99	1	24

Рис 3.8 Сторінка топ 100

Для взаємодії з користувачем створено сторінку AI Helper, яка містить інтерфейс чату для спілкування з AI-асистентом.

Користувач може вибрати карточку з популярних питань по метрикам або написати своє власне питання.

Реалізовано відображення історії повідомлень та форму введення запитів, історія чату зберігається протягом сесії та виводиться у хронологічному порядку.

На рисунку 3.9 показано приклад сторінки Ai Helper web-застосунку для оцінки криптоактивів.

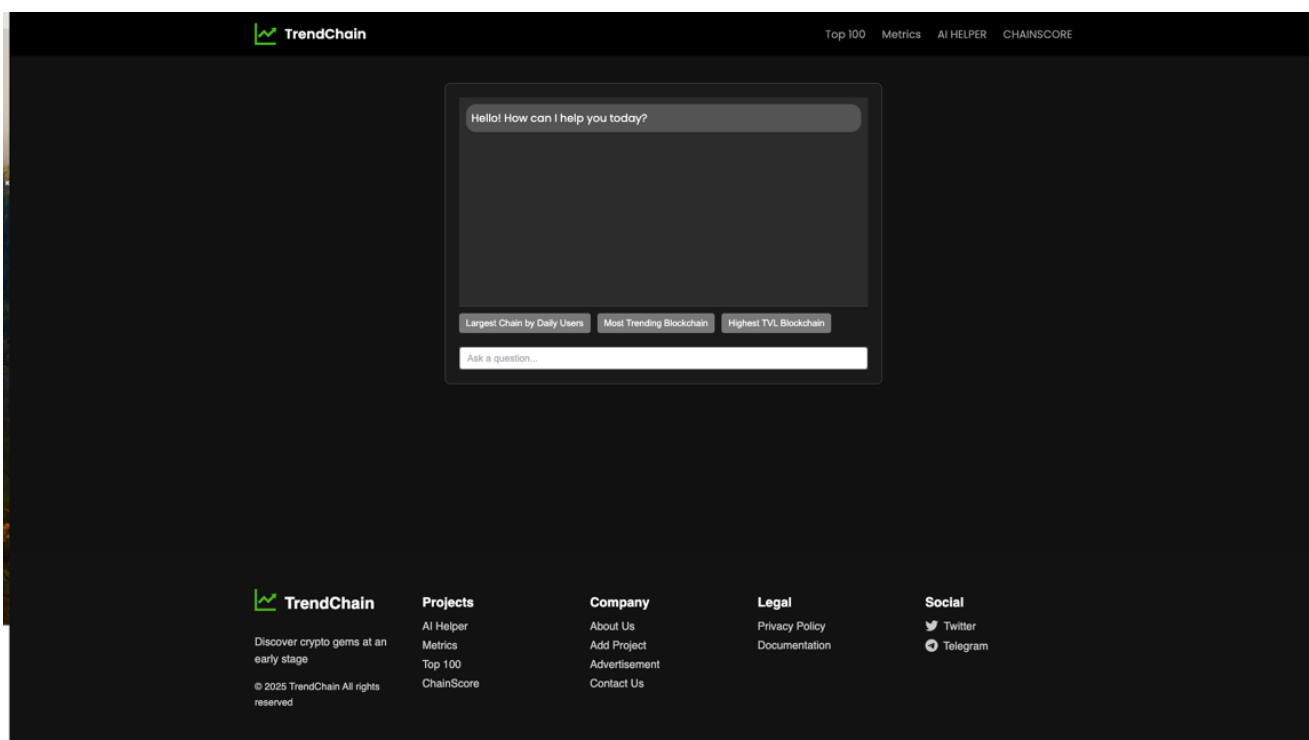


Рис 3.9 Сторінка взаємодії з користувачем AI Helper

Крім основних сторінок, у застосунку реалізовано кілька компонентів, які відповідають за покращення взаємодії користувача з інтерфейсом та забезпечують зручну візуалізацію даних. Вони інтегровані у відповідні частини сторінок і виконують конкретні функціональні задачі:

`LanguageSwitcher.vue` — компонент перемикавання мови інтерфейсу (UA/EN), інтегрований у шапку застосунку. Використовується на головній сторінці для забезпечення мультимовності.

`BarChart.vue` — компонент для візуалізації даних у вигляді стовпчикових діаграм на сторінці `Metrics`. Побудований на базі бібліотеки `Chart.js`, він наочно відображає ключові метрики блокчейнів — кількість транзакцій, активних користувачів, обсяг комісій тощо.

`ProgressBar.vue` — використовується на сторінці `ChainScore` для демонстрації результатів у вигляді кругових індикаторів прогресу. Реалізує кольорове кодування результатів (червоний — до 25%, помаранчевий — 26–50%, зелений — 51–100%) і дозволяє оцінити як загальний скор, так і окремі показники. Всі компоненти

побудовані з дотриманням принципів повторного використання та модульності.

3.4 Завантаження проекту на github

Для того, щоб додати Vue.js проект до існуючого репозиторію на GitHub, необхідно виконати кілька кроків. Вони включають ініціалізацію у проекті, додавання файлів до репозиторію та завантаження змін до GitHub. Для початку треба відкрити термінал у кореневому каталозі Vue.js проекті Далі виконати команду для ініціалізації Git-репозиторію:

- `git init`

Далі додаємо URL існуючого репозиторію GitHub як віддалений репозиторій:

`https://github.com/illyach/diploma-project`

Далі треба додати всі файли проекту до індексу Git:

- `git add .`

Наступним кроком створюємо коміт з повідомленням, яке описує зміни, внесені у проект:

- `git commit -m "Added all files"`

Останнім кроком є завантаження змін до віддаленого репозиторію на GitHub:

- `git push -u origin master`

3.5 Тестування

Сутність тестування полягає в перевірці веб-застосунку для виявлення помилок, дефектів або неправильного функціонування. Тестування є критично важливим для забезпечення якості програмного забезпечення та задоволення вимог користувачів. Основні цілі тестування включають перевірку правильності роботи програми, виявлення помилок, забезпечення відповідності вимогам та підтримку надійності та стабільності системи.[15]

Для веб-застосунку аналітичної платформи було вирішено використовувати ручне тестування компонентів та API інтеграцій.

Тестування охоплювало кілька ключових сценаріїв, спрямованих на перевірку функціональності різних частин застосунку.

Тестування розрахунку ChainScore

1. Перевірено коректність обробки та нормалізації даних з різних джерел (DeFiLlama, CoinGecko).
2. Виконано тестування формули розрахунку скору з різними вхідними параметрами.
3. Тестування обробки відсутніх даних та їх вплив на фінальний скор.
4. Виявлено та виправлено критичний баг, коли загальний рейтинг міг перевищувати 100 балів. Проблему було вирішено додаванням обмеження:
5. `return Math.min(Math.ceil(total), 100);`

Тестування візуалізації даних

1. Проведено тестування компонента ProgressBar:
 - Перевірка коректності відображення різних діапазонів значень.
 - Тестування кольорового кодування (червоний 0-25%, помаранчевий 26-50%, зелений 51-100%).
 - Перевірка анімації заповнення.
2. Тестування компонента BarChart:
 - Перевірка коректності відображення даних на графіках.
 - Тестування сортування та масштабування даних.
 - Перевірка адаптивності при різних розмірах екрану.

Тестування інтеграції з API

1. Тестування взаємодії з DeFiLlama API:
 - Перевірка отримання TVL даних.
 - Тестування отримання даних про комісії.
 - Обробка помилок при недоступності API.
2. Тестування взаємодії з CoinGecko API:

- Перевірка отримання ринкових даних.
- Тестування отримання рейтингу за капіталізацією.
- Обробка лімітів запитів.

Тестування локалізації

1. Перевірка роботи компонента LanguageSwitcher:
 - Тестування перемикання між українською та англійською мовами .
 - Перевірка коректності відображення перекладів.

Ручне тестування компонентів та API інтеграцій дозволило виявити та усунути критичні помилки, зокрема проблему з перевищенням максимального значення рейтингу. Кожен етап тестування включав ретельний аналіз поведінки системи та перевірку правильності обробки даних, що забезпечило стабільну роботу застосунку та коректне відображення інформації користувачам.

ВИСНОВКИ

В процесі виконання дипломної роботи було розроблено веб-застосунок для оцінки криптоактивів, що відповідає сучасним вимогам аналітики у сфері блокчейн-технологій. Мета дипломної роботи виконана в повному обсязі.

1. Проведено аналіз основних джерел інформації у сфері блокчейн-технологій, що дозволило визначити ключові категорії показників для аналізу блокчейн-проектів, включаючи метрики, такі як щоденні транзакції, активні користувачі, комісії, активність розробників та | соціальні показники.

2. Досліджено існуючі рішення для оцінки криптоактивів, визначено їх ключові функції, переваги та недоліки, що підтвердило необхідність розробки власного веб-застосунку, який би враховував ці недоліки.

3. Визначено основні засоби реалізації веб-застосунку, зокрема використані мови програмування JavaScript (Vue.js), HTML, CSS, а також бібліотеки для візуалізації даних, такі як Chart.js, та інтерфейсних компонентів, такі як Element Plus.

4. Розроблено веб-застосунок, що дозволяє швидко отримати інтерактивну оцінку блокчейн-проекту на основі ключових метрик, що значно спрощує процес аналізу для користувачів з різним рівнем досвіду.

5. Реалізовано систему метрик з візуалізацією, включаючи прогрес-бари, діаграми та рейтингову систему, що дозволяє користувачеві легко порівнювати проекти між собою та приймати обґрунтовані інвестиційні рішення.

Робота пройшла апробацію. За її результатами було опубліковано наступні тези доповідей:

1. Лупу І.І., Web-застосунок для оцінки криптоактивів. // Матеріали VI Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». Збірник тез. 24.04.2025, м. Київ. К.:ДУІКТ, С. 218-221.

2. Лупу І.І., Забезпечення безпеки даних у системах криптовалют. // Матеріали VI Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». Збірник тез. 24.04.2025, м. Київ. К.:ДУІКТ, С. 451-453.

ПЕРЕЛІК ПОСИЛАНЬ

1. Татаринцева Ю.Л., Колонтаєвський Р.Г. Методи інвестування у блокчейн-технології. *Ефективна економіка*. 2025. С. 13-14. URL: <https://doi.org/10.32702/2307-2105.2025.4.81> (дата звернення: 18.05.2024).
2. How Many Cryptocurrencies are There In 2025? [Електронний ресурс] / Fabio Duarte. – Режим доступу: <https://explodingtopics.com/blog/number-of-cryptocurrencies> (дата звернення: 18.05.2024).
3. CoinGecko [Електронний ресурс]. – Режим доступу: <https://www.coingecko.com/> (дата звернення: 18.05.2024).
4. Bybit Learn. Layer 1 Blockchain [Електронний ресурс]. – Режим доступу: <https://learn.bybit.com/uk/glossary/layer-1-blockchain/> (дата звернення: 18.05.2024).
5. DropsTab. a16z Portfolio [Електронний ресурс]. – Режим доступу: <https://dropstab.com/investors/a16z-andreessen-horowitz/portfolio> (дата звернення: 18.05.2024).
6. Що таке ончейн-аналіз і що він вимірює? [Електронний ресурс]. – Режим доступу: binance.com/uk-UA/blog/all/що-таке-ончейн-аналіз-і-що-він-вимірює-1616182433092262996 (дата звернення: 19.05.2025)
7. Key Metrics for the Rapidly Expanding Web3 Ecosystem [Електронний ресурс] / Chainlink– 2022. – Режим доступу: <https://blog.chain.link/web3-metrics/> (дата звернення: 19.05.2025)
8. Visual Studio Code Features [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.linkedin.com/pulse/visual-studio-code-featurespramuditha-madura>. (дата звернення: 21.05.2025).
9. Shavin M. Learning Vue: Core Concepts and Practical Patterns for Reusable, Composable, and Scalable User Interfaces. — Sebastopol, CA : O'Reilly Media, 2023. — 400 p. — ISBN 978-1-492-09882-9.

10. Ponuthurai P.K., Loeliger J. Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development. — 2nd ed. — O'Reilly Media, 2022. — 502 p. — ISBN 978-1-492-09716-7.
11. Supabase: документація [Електронний ресурс] / Supabase. — Режим доступу: <https://supabase.com/docs> (дата звернення: 22.05.2025).
12. Artemis.xyz: документація [Електронний ресурс]. — Режим доступу: <https://app.artemis.xyz/docs/welcome/overview#what-does-artemis-offer>. (дата звернення: 22.05.2024).
13. Unger R., Chandler C. A Project Guide to UX Design: For user experience designers in the field or in the making / Russ Unger, Carolyn Chandler. — 1st ed. — New Riders Pub, 2009. — 267 p. — ISBN 978-0-321-60737-9.
14. Why Use a Modular Architecture in Software Design? [Електронний ресурс] // AppMaster.io : блог. — Режим доступу: <https://appmaster.io/blog/why-use-a-modular-architecture-in-software-design> (дата звернення: 22.05.2025).
15. Тестування програмного забезпечення : навч. посіб. / [уклад. О. В. Бойко, В. В. Гапон, С. О. Цибульник та ін.] ; за ред. О. В. Бойко. — Черкаси : ЧНУ імені Богдана Хмельницького, 2017. — 284 с. — ISBN 978-966-920-199-7

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка web-затосунку для оцінки криптоактивів

Виконав студент 4 курсу
ПД-44 Лупу Ілля Ігорович
к.т.н., доц., доцент кафедри ІПЗ Щербина Ірина Сергіївна

Київ – 2025

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

2

Мета роботи: розробка web-застосунку для надання інвесторам оцінки блокчейн-проектів на основі ключових метрик

Об'єкт дослідження: процес оцінки криптоактивів

Предмет дослідження: web-застосунок для оцінки метрик криптовалют

ЗАДАЧІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1. Визначити ключових метрик блокчейнів та ринку
2. Проаналізувати аналоги і створити функціональні та нефункціональні вимоги
3. Визначити засоби реалізації web-застосунку
4. Спроекувати архітектуру web-застосунку
5. Спроекувати інтерфейс користувача
6. Розробити web-застосунок відповідно до функціональних вимог
7. Провести тестування застосунку

1

АНАЛІЗ АНАЛОГІВ

Критерій / Застосунок	DappRadar	CoinGecko	DeFiLlama	Token Terminal	TrendChain
Оцінка за шкалою	-	-	-	-	+
Топ-100 за рейтингом	-	+	-	-	+
Аналітика метрик	+	+	+	+	+
Графіки та таблиці	+	+	+	+	+
AI-помічник	-	-	-	-	+
Оцінка за набором метрик	-	-	-	+	+
Фільтрація і сортування	+	+	+	+	+

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функціональні:

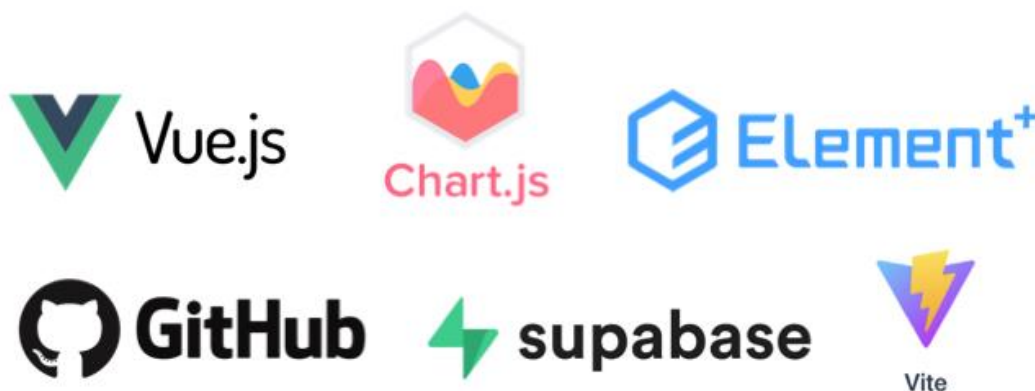
- Відображення блокчейнів у вигляді таблиці з можливістю сортування за зростанням або спаданням за ключовими метриками
- Отримання оцінки блокчейну на основі метрик
- Візуалізація результатів у вигляді шкали (від 0 до 100).
- Інтерактивний чат-бот для запиту метрик.
- Виведення таблиць та графіків за метриками.
- Створення таблиці топ 100 криптовалют за рейтингом

Нефункціональні:

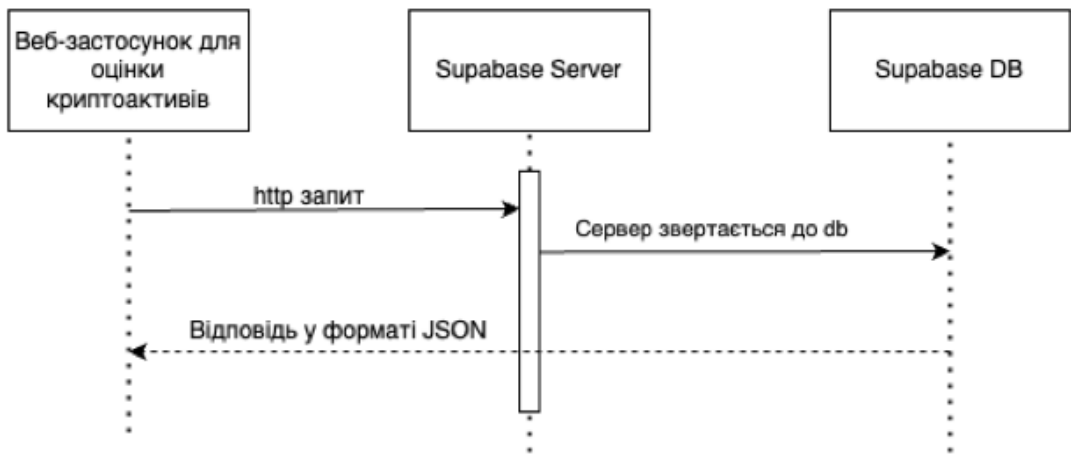
- Критичні дані (наприклад, API-ключі) мають зберігатися в зашифрованому вигляді.
- Підтримка основних браузерів (Chrome, Firefox, Safari, Edge).
- Модульна структура застосунку (Assets, Components, Router...)
- Інтерфейс користувача повинен бути адаптивний під мобільні пристрої

3

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

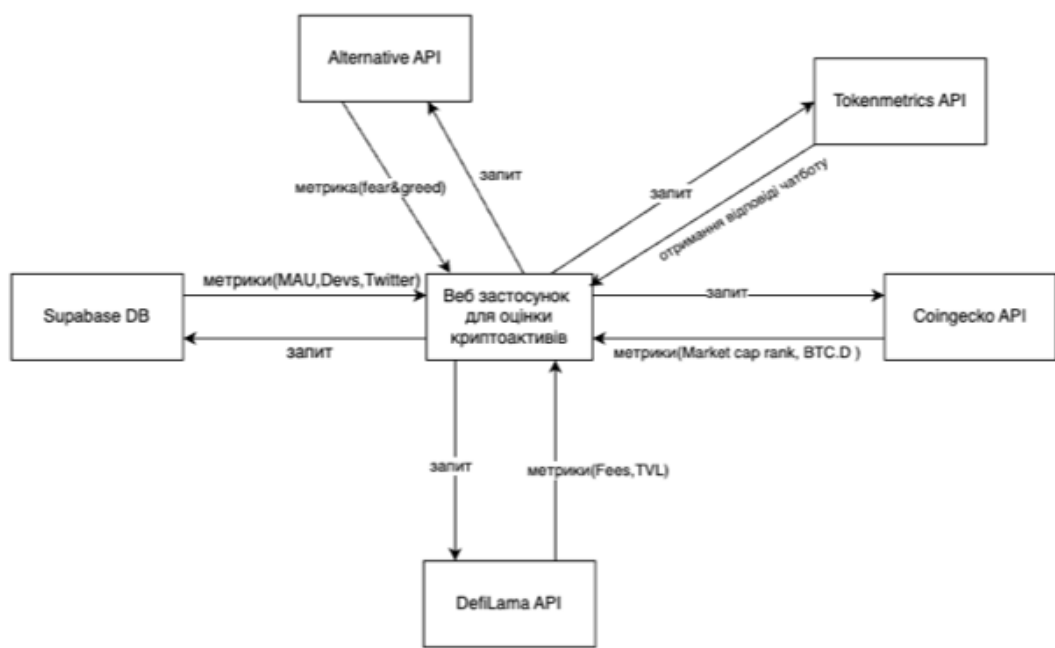


ДІАГРАМА ПОСЛІДОВНОСТІ РОБОТИ
КЛІЄНТ-СЕРВЕР-БАЗА ДАНИХ



8

АРХІТЕКТУРА ЗАСТОСУНКУ



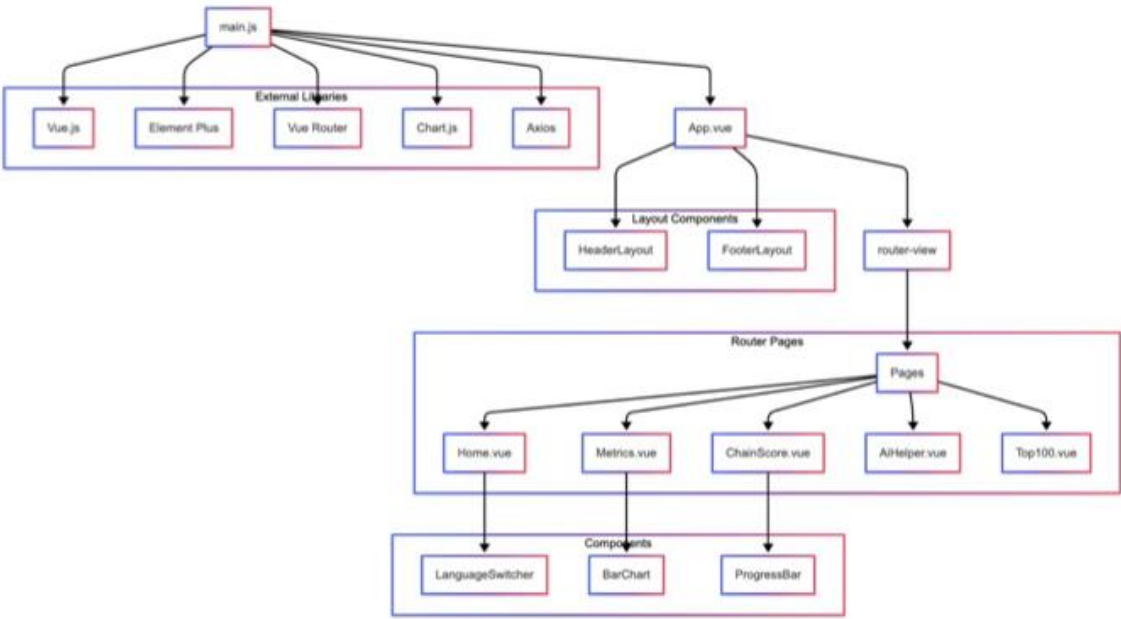
7

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



9

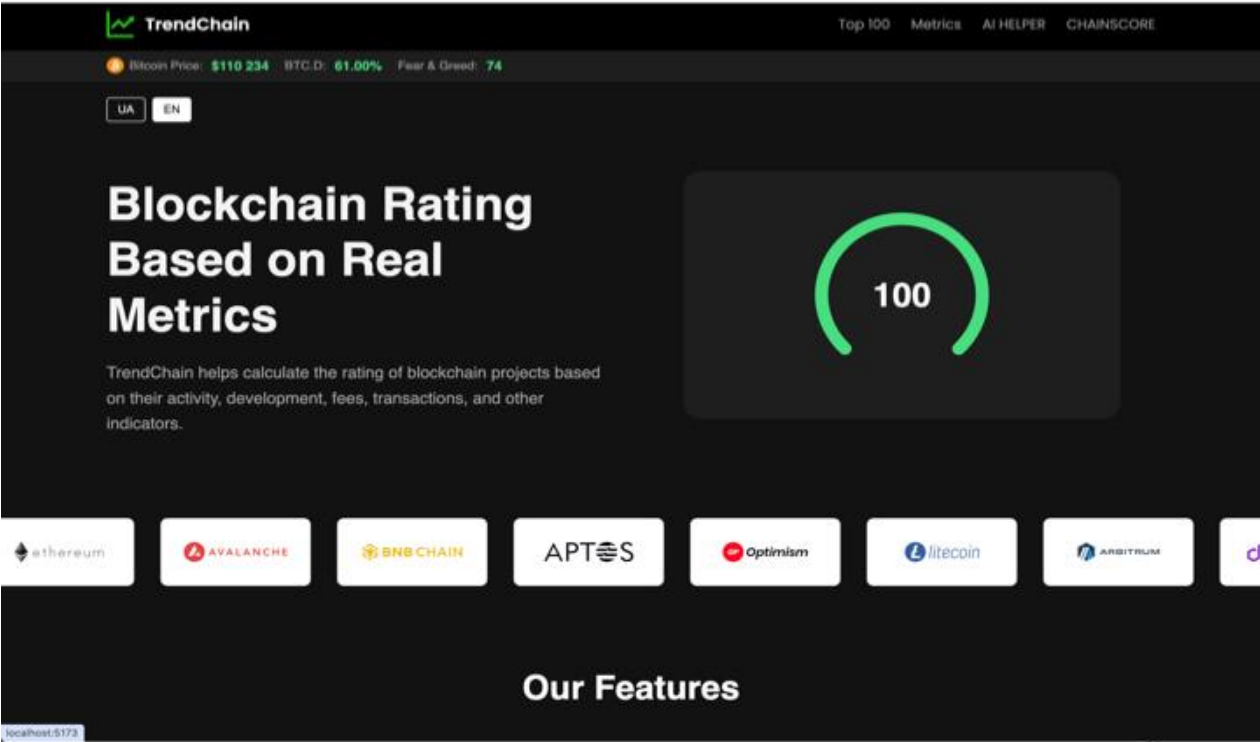
ДІАГРАМА СТРУКТУРИ ЗАСТОСУНКУ



10

ЕКРАННІ ФОРМИ

Головна сторінка з інформацією



ЕКРАННІ ФОРМИ

Сторінка метрик Metrics (таблиця)

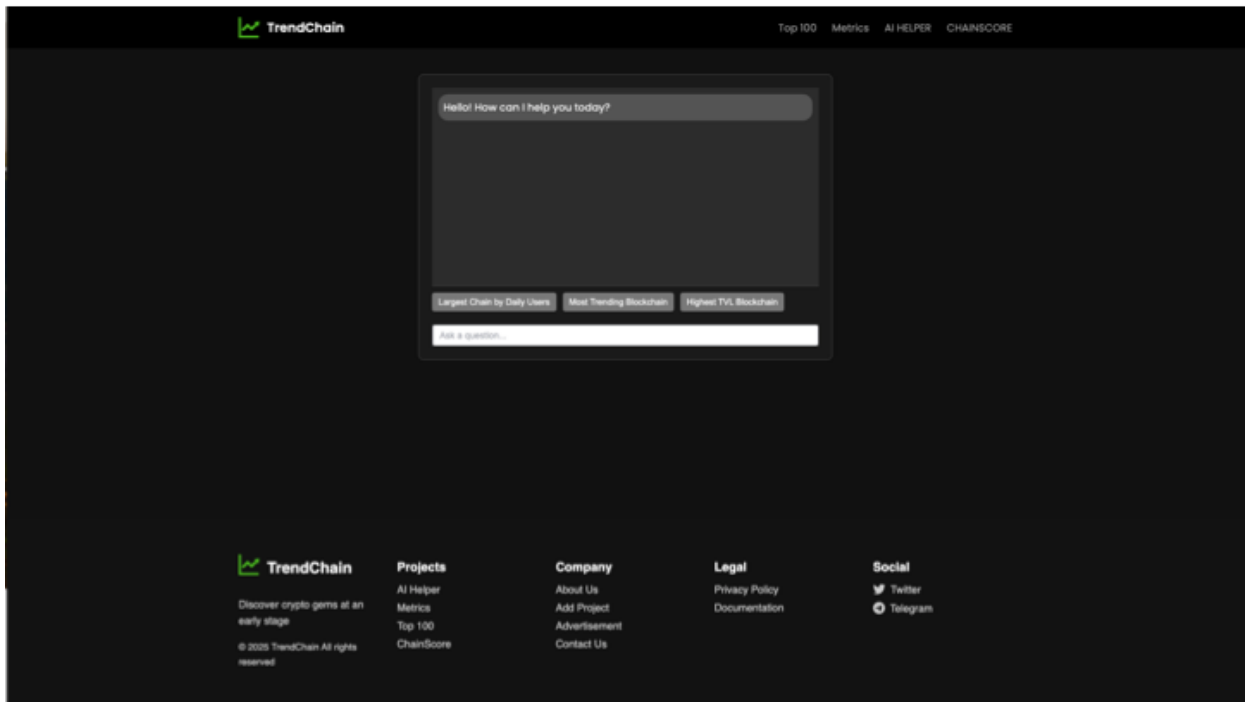
TrendChain Top 100 Metrics AI HELPER CHAINSCORE

Таблиця Графіки

Blockchain	Daily Transactions	Monthly Active Users	Fees	Core Devs	Twitter Followers
internet-computer	664 519 088	N/A	\$41 509,17	79	694 102
hyperliquid	321 348 961	N/A	\$1 126 903,89	1	240 000
solana	96 780 542	105 207 831	\$1 708 519,99	316	3 384 452
tron	8 642 306	13 659 145	\$1 866 744,78	19	1 713 678
base	6 964 818	16 210 586	\$115 140,97	103	942 261
sui	6 802 417	42 115 276	\$83 682,09	107	1 043 749
near	4 793 836	45 511 887	\$7 978,57	166	1 669 137
bsc	4 741 676	15 509 368	\$497 781,74	4	14 316 914
injective	4 280 847	76 218	\$4 306,24	13	N/A
sei	4 220 651	2 712 625	\$109,59	16	786 368
aptos	3 273 100	11 990 290	\$1 862,26	77	5 454
polygon	2 820 030	5 015 832	\$10 154,93	95	2 043 749
algorand	2 067 033	N/A	\$674,94	47	358 769
arbitrum	1 801 946	4 233 554	\$10 570,25	60	1 146 367
stellar	1 800 623	594 313	\$2 835,05	189	826 817
sonelium	1 395 543	N/A	\$3 966,92		267 934
celo	1 353 354	1 031 513	\$7 959,94	167	520 764

ЕКРАННІ ФОРМИ

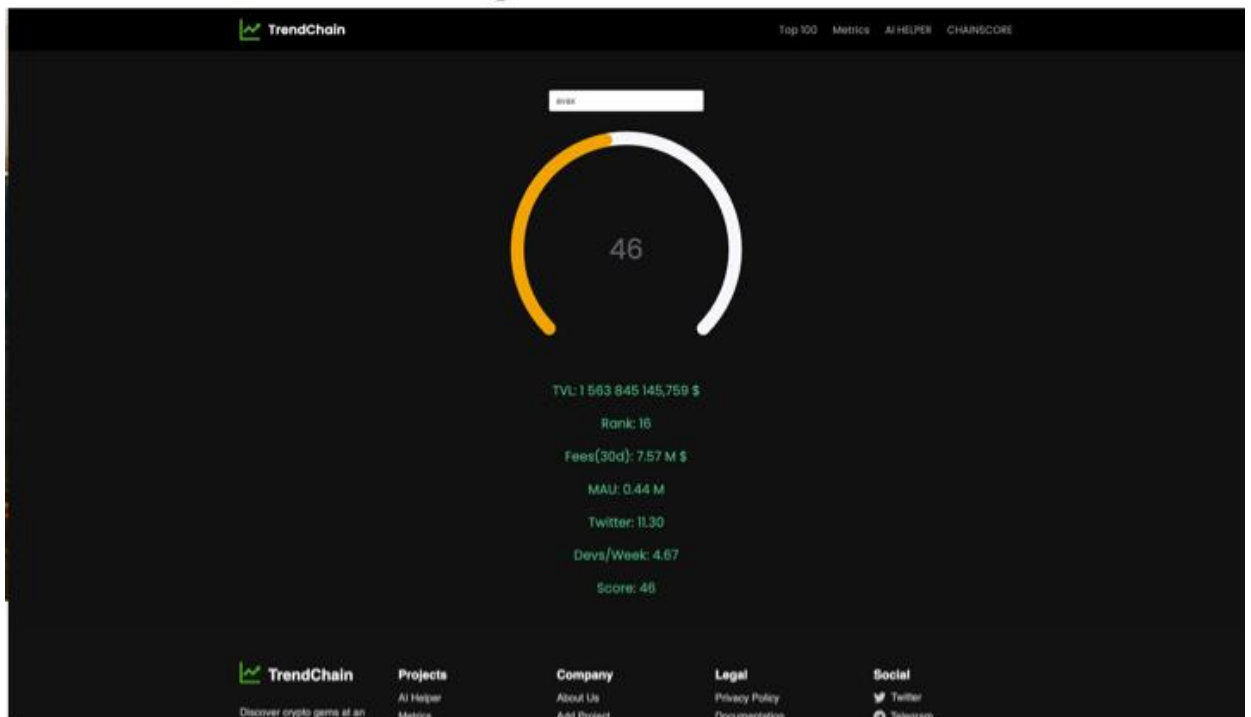
Сторінка метрик AI helper



14

ЕКРАННІ ФОРМИ

Сторінка Chain Score



15

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Лупу І.І., Web-застосунок для оцінки криптоактивів. // Матеріали VI Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». Збірник тез. 24.04.2025, м. Київ. К.:ДУІКТ, С. 218-221.

2. Лупу І.І., Забезпечення безпеки даних у системах криптовалют. // Матеріали VI Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». Збірник тез. 24.04.2025, м. Київ. К.:ДУІКТ, С. 451-453.

16

ВИСНОВКИ

1. Проведено аналіз основних джерел інформації у сфері блокчейн-технологій, визначено ключові категорії показників для аналізу блокчейн-проектів
2. Досліджено існуючі рішення та визначено їх ключові функції, переваги та недоліки
3. Визначено основні засоби реалізації, бібліотеки, мови програмування
4. Розроблено вебзастосунок з використанням сучасних технологій що дозволяє швидко отримати інтерактивну оцінку блокчейн-проекту.
5. Реалізовано систему метрик з візуалізацією: прогрес-бари, діаграми, рейтингова система, що дозволяє користувачеві легко порівнювати проекти між собою.
6. Проведено тестування системи, що підтвердило її працездатність, зручність інтерфейсу та відповідність поставленим функціональним вимогам.

17

ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

Сторінка Home

```
<template>
  <div class="landing-container">
    <div class="price-bar" v-if="bitcoinPrice">
      <div class="price-content">
        
        <span class="price-label">Bitcoin Price:</span>
        <span class="price-value">${{
bitcoinPrice.toLocaleString() }}</span>
        <span class="dominance-label">BTC.D:</span>
        <span class="dominance-value">{{ btcDominance
}}%</span>
        <span class="fear-greed-label">Fear & Greed:</span>
        <span class="fear-greed-value">{{ fearGreedIndex
}}</span>
      </div>
    </div>
    <div class="language-switcher-container">
      <LanguageSwitcher @language-
change="changeLanguage" />
    </div>
    <!-- Хедер і онук -->
    <section class="hero">
      <div class="hero-content">
        <div class="hero-text">
          <h1>{{ t.hero.title }}</h1>
          <p>{{ t.hero.description }}</p>
        </div>
        <div class="score-widget">
          <div class="score-container">
            <el-progress
              type="dashboard"
              :percentage="percentage2"
              :width="200"
              :stroke-width="15"
              :show-text="true"
              :color="customColors"
            >
              <template #default>
                <div class="progress-content">
                  <span class="score-number">{{ displayNumber
}}</span>
                </div>
              </template>
            </el-progress>
          </div>
        </div>
      </div>
    </div>
    <!-- Bitcoin Price Bar -->

    <!-- Карусель логотипів -->
    <div class="carousel-wrapper">
```

```
      <div class="carousel-track" :style="{ animationDuration:
logos.length * 2 + 's' }">
        <div v-for="(logo, index) in repeatedLogos"
:key="index" class="carousel-logo">
          <div class="logo-bg">
            
          </div>
        </div>
      </div>

    <!-- Особливості -->
    <section class="features">
      <h2>{{ t.features.title }}</h2>
      <div class="features-grid">
        <div v-for="(card, index) in t.features.cards"
:key="index" class="feature-card">
          <div class="feature-icon">{{ ['💎', '🔍', '💡',
'🔗'][index] }}</div>
          <h3>{{ card.title }}</h3>
          <p>{{ card.description }}</p>
        </div>
      </section>

    <!-- Чому важливо -->
    <section class="why">
      <div class="why-card">
        <div class="why-content">
          <h2>{{ t.why.title }}</h2>
          <p>{{ t.why.description }}</p>
        </div>
        <div class="graph-container">
          
        </div>
      </div>
    </section>
  </div>
</template>
```

Сторінка Chainscore

```
<script setup>
import ProgresBar from '../components/ProgresBar.vue';
import { ref, watch, computed } from 'vue';
import chainsData from '../assets/supabase.json';

const input = ref("");
const showProgress = ref(false);
const targetTvl = ref(0);
const originalTvl = ref(null);
const marketCapRank = ref(null);
const fees30d = ref(null);

// Нові змінні для додаткових метрик
const mau = ref(null);
const twitterFollowers = ref(null);
const weeklyDevsCore = ref(null);
```

```

const finalScore = computed(() => {
  const tvlScore = targetTvl.value || 0;
  const rankScore = marketCapRank.value ? (1 /
marketCapRank.value) * 100 : 0;
  const feesScore = fees30d.value ? fees30d.value / 1_000_000
: 0;

  const mauScore = mau.value ? mau.value / 1_000_000 : 0;
  const twitterScore = twitterFollowers.value ?
twitterFollowers.value / 100_000 : 0;
  const devsScore = weeklyDevsCore.value ?
weeklyDevsCore.value / 15 : 0;

  console.log("fees", feesScore, "rank", rankScore, "tvl",
tvlScore, "mau", mauScore, "tw", twitterScore, "devs",
devsScore);

  const total = tvlScore + rankScore + feesScore + mauScore +
twitterScore + devsScore;
  return Math.min(Math.ceil(total), 100);
});

watch(input, async (newValue) => {
  if (!newValue) {
    showProgress.value = false;
    return;
  }

  try {
    const llamaRes = await fetch('/api/llama/v2/chains');
    const llamaData = await llamaRes.json();

    const found = llamaData.find((item) => {
      const nameMatch = item.name?.toLowerCase() ===
newValue.toLowerCase();
      const symbolMatch = item.tokenSymbol?.toLowerCase()
=== newValue.toLowerCase();
      return nameMatch || symbolMatch;
    });

    const layer1Res = await
fetch('/api/coingecko/api/v3/coins/markets?vs_currency=usd&c
ategory=layer-
1&order=market_cap_desc&per_page=250&page=1');
    const layer2Res = await
fetch('/api/coingecko/api/v3/coins/markets?vs_currency=usd&c
ategory=layer-
2&order=market_cap_desc&per_page=250&page=1');

    const [layer1Data, layer2Data] = await
Promise.all([layer1Res.json(), layer2Res.json()]);
    const combinedData = [...layer1Data, ...layer2Data];

    const market = combinedData.find((item) => {
      const nameMatch = item.name?.toLowerCase() ===
newValue.toLowerCase();
      const symbolMatch = item.symbol?.toLowerCase() ===
newValue.toLowerCase();
      return nameMatch || symbolMatch;
    });

    if (found && found.tvl) {
      originalTvl.value = found.tvl;
      targetTvl.value = Math.min(Math.floor(found.tvl /
100_000_000), 100);
      showProgress.value = true;
      const feesRes = await
fetch('/api/llama/overview/fees/${found.name}?excludeTotalD

```

```

ataChart=true&excludeTotalDataChartBreakdown=true&dataT
ype=dailyFees`);
    const feesData = await feesRes.json();
    fees30d.value = feesData.total30d || 0;
  } else {
    originalTvl.value = null;
    targetTvl.value = 0;
    showProgress.value = true;
  }

  marketCapRank.value = market?.market_cap_rank ?? null;

  // Зчитуємо дані з JSON
  const fromStats = chainsData.find(item =>
item.name?.toLowerCase() === newValue.toLowerCase() ||
item.symbol?.toLowerCase() === newValue.toLowerCase()
);

  if (fromStats) {
    mau.value = fromStats.MAU || 0;
    twitterFollowers.value =
fromStats.TWITTER_FOLLOWERS || 0;
    weeklyDevsCore.value =
fromStats.WEEKLY_DEVS_CORE || 0;
  } else {
    mau.value = twitterFollowers.value =
weeklyDevsCore.value = 0;
  }

  // Якщо нічого не знайдено
  if (!found && !market && !fromStats) {
    targetTvl.value = 0;
    marketCapRank.value = 50; // (1/50)*100 = 2
    showProgress.value = true;
  }

  } catch (err) {
    console.error('Error fetching data:', err);
    showProgress.value = false;
    originalTvl.value = null;
    targetTvl.value = 0;
    marketCapRank.value = null;
  }
});
</script>

<template>
<div class="chainscore-wrapper">
  <el-input
    v-model="input"
    style="width: 240px"
    placeholder="Введи назву блокчейна / тікер"
  />

  <div v-if="showProgress" class="score-section">
    <ProgresBar :target="finalScore" />
    <p class="tvl-text">TVL: {{ originalTvl?.toLocaleString()
|| '—' }} $</p>
    <p class="tvl-text">Rank: {{ marketCapRank || '—' }}</p>
    <p class="tvl-text">Fees(30d): {{ fees30d ? (fees30d /
1_000_000).toFixed(2) : '—' }} M $</p>
    <p class="tvl-text">MAU: {{ mau ? (mau /
1_000_000).toFixed(2) + ' M' : '—' }}</p>
    <p class="tvl-text">Twitter: {{ twitterFollowers ?
(twitterFollowers / 100_000).toFixed(2) : '—' }}</p>
    <p class="tvl-text">Devs/Week: {{ weeklyDevsCore ?
(weeklyDevsCore / 15).toFixed(2) : '—' }}</p>
    <p class="tvl-text">Score: {{ finalScore }}</p>
  </div>

```

```

</div>
</template>

<style scoped>
.chainscore-wrapper {
  margin-top: 40px;
  display: flex;
  flex-direction: column;
  align-items: center;
}

.score-section {
  margin-top: 30px;
  text-align: center;
}

.tvl-text {
  margin-top: 12px;
  font-size: 20px;
  font-weight: 500;
  color: #42b983;
}
</style>

```

Сторінка Metrics

```

<template>
  <div class="black-table-container">
    <!-- Перемикач між таблицею та графіками -->
    <div style="margin: 20px auto; text-align: center;">
      <el-radio-group v-model="viewMode" size="medium">
        <el-radio label="table">Таблиця</el-radio>
        <el-radio label="charts">Графіки</el-radio>
      </el-radio-group>
    </div>

    <!-- Селектор метрики для графіків -->
    <div v-if="viewMode === 'charts'" style="text-align:
center; margin-bottom: 20px;">
      <el-select v-model="selectedMetric"
placeholder="Оберіть метрику" style="width: 250px;">
        <el-option label="All" value="ALL" />
        <el-option label="Monthly Users" value="MAU" />
        <el-option label="Daily Transactions"
value="DAILY_TXNS" />
        <el-option label="Fees" value="FEES" />
        <el-option label="Core Developers"
value="WEEKLY_DEVS_CORE" />
      </el-select>
    </div>

    <!-- Графіки -->
    <div v-if="viewMode === 'charts'" class="charts-
container">
      <template v-if="selectedMetric === 'ALL'">
        <BarChart :data="chartData('MAU')" label="Monthly
Active Users" />
        <BarChart :data="chartData('DAILY_TXNS')"
label="Daily Transactions" />
        <BarChart :data="chartData('FEES')" label="Fees
(USD)" />
        <BarChart
:data="chartData('WEEKLY_DEVS_CORE')" label="Weekly
Core Developers" />
      </template>
      <template v-else>

```

```

        <BarChart :data="chartData(selectedMetric)"
:label="metricLabel[selectedMetric]" />
      </template>
    </div>

    <!-- Таблиця -->
    <el-table
v-if="viewMode === 'table'"
:data="filteredData"
style="width: 90%; margin: 20px auto;"
border
highlight-current-row
:default-sort="{ prop: 'DAILY_TXNS', order: 'descending'
}"
@sort-change="handleSortChange"
class="pure-black-table"
>
      <el-table-column prop="name" label="Blockchain"
width="180" sortable />
      <el-table-column prop="DAILY_TXNS" label="Daily
Transactions" sortable :formatter="formatNumber" />
      <el-table-column prop="MAU" label="Monthly Active
Users" sortable :formatter="formatNumber" />
      <el-table-column prop="FEES" label="Fees" sortable
:formatter="formatCurrency" />
      <el-table-column prop="WEEKLY_DEVS_CORE"
label="Core Devs" sortable />
      <el-table-column prop="TWITTER_FOLLOWERS"
label="Twitter Followers" sortable :formatter="formatNumber"
/>
    </el-table>
  </div>
</template>

<script>
import { ref, onMounted } from 'vue'
import chainsData from '../assets/supabase.json'
import BarChart from '../components/BarChart.vue'

export default {
  components: { BarChart },
  data() {
    return {
      tableData: chainsData,
      filteredData: [...chainsData],
      sortOptions: {
        prop: 'DAILY_TXNS',
        order: 'descending'
      },
      viewMode: 'table',
      selectedMetric: 'ALL',
      metricLabel: {
        MAU: 'Monthly Active Users',
        DAILY_TXNS: 'Daily Transactions',
        FEES: 'Fees (USD)',
        WEEKLY_DEVS_CORE: 'Weekly Core Developers'
      }
    }
  },
  methods: {
    formatNumber(row, column, cellValue) {
      return cellValue ? cellValue.toLocaleString() : 'N/A'
    },
    formatCurrency(row, column, cellValue) {
      return cellValue
        ? `$$${cellValue.toLocaleString(undefined, {
          maximumFractionDigits: 2 })}`
        : 'N/A'
    },
  },

```

```

    handleSortChange({ prop, order }) {
      this.sortOptions = { prop, order }
      this.filteredData.sort((a, b) => {
        const valA = a[prop] || 0
        const valB = b[prop] || 0
        return order === 'ascending' ? valA - valB : valB - valA
      })
    },
    chartData(metric) {
      const data = this.filteredData.map(item => ({
        name: item.name || 'Unknown',
        value: typeof item[metric] === 'number' ? item[metric] :
0      )))
      return [...data].sort((a, b) => b.value - a.value)
    },
    created() {
      this.handleSortChange(this.sortOptions)
    }
  }
</script>

<style scoped>
.black-table-container {
  background-color: #000;
  min-height: 100vh;
  padding: 20px;
}

/* Pure black table styles */
.pure-black-table {
  --el-table-border-color: #333;
  --el-table-header-bg-color: #111;
  --el-table-row-hover-bg-color: #222;
  --el-table-current-row-bg-color: #222;
  --el-table-bg-color: #000;
  --el-table-text-color: #e0e0e0;
  --el-table-header-text-color: #fff;
}

.pure-black-table ::v-deep .el-table__row {
  background-color: #232323;
  color: #e0e0e0;
}

.pure-black-table ::v-deep .el-table__header th {
  background-color: #111;
  color: #fff;
  font-weight: bold;
}

.pure-black-table ::v-deep .el-table__body tr:hover > td {
  background-color: #211338 !important;
}

.pure-black-table ::v-deep .el-table--border {
  border: 1px solid #333;
}

.pure-black-table ::v-deep .el-table--border th {
  border-bottom: 1px solid #333;
}

.pure-black-table ::v-deep .el-table--border td {
  border-bottom: 1px solid #333;
}
</style>

```

КОМПОНЕНТ BarChart

```

<!-- components/BarChart.vue -->
<template>
  <div style="width: 90%; margin: 40px auto;">
    <h3 style="color: white; text-align: center;">{{ label
  }}</h3>
    <Bar :data="chartData" :options="chartOptions" />
  </div>
</template>

<script>
import { Bar } from 'vue-chartjs'
import {
  Chart as ChartJS,
  Title,
  Tooltip,
  Legend,
  BarElement,
  CategoryScale,
  LinearScale
} from 'chart.js'

ChartJS.register(Title, Tooltip, Legend, BarElement,
CategoryScale, LinearScale)

export default {
  name: 'BarChart',
  components: { Bar },
  props: {
    data: Array,
    label: String
  },
  computed: {
    chartData() {
      const sortedData = [...this.data].sort((a, b) => b.value -
a.value)

      return {
        labels: sortedData.map(item => item.name),
        datasets: [{
          label: this.label,
          backgroundColor: '#a855f7',
          borderRadius: 5,
          data: sortedData.map(item => item.value)
        }]
      },
      chartOptions() {
        return {
          responsive: true,
          plugins: {
            legend: { display: false }
          },
          scales: {
            x: {
              ticks: {
                color: 'white',
                font: {
                  size: window.innerWidth < 1442 ? 9 : 14
                }
              },
            },
            y: { ticks: { color: 'white' } }
          }
        }
      }
    }
  }
}

```

```

    }
  }
</script>

```

Компонент ProgressBar

```

<script setup lang="ts">
import { ref, watch } from 'vue';

const props = defineProps<{ target: number }>();

const percentage = ref(0);

const animateTo = (to: number) => {
  percentage.value = 0;
  const step = () => {
    if (percentage.value < to) {
      percentage.value += 1;
      requestAnimationFrame(step);
    } else {
      percentage.value = to;
    }
  };
  step();
};

watch(
  () => props.target,

```

```

  (newVal) => {
    if (typeof newVal === 'number') animateTo(newVal);
  },
  { immediate: true }
);

```

```

const getColor = (value: number) => {
  if (value <= 25) return '#ff4d4f';
  if (value <= 50) return '#ffa500';
  if (value <= 75) return '#fadb14';
  return '#52c41a';
};
</script>

```

```

<template>
<div class="demo-progress">
  <el-progress
    type="dashboard"
    :percentage="percentage"
    :color="getColor"
    :width="360"
    :stroke-width="20"
    :format="(percent) => ` ${percent} ` "
  />
</div>
</template>

```