

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-застосунку для відстеження курсу криптовалют

Виконав: студент IV курсу, групи СПс-42
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Дацко А.І.
(підпис) (прізвище та ініціали)

Керівник Гащин Н.Б.
(підпис) (прізвище та ініціали)

Нормоконтроль Стоянов Ю.М.
(підпис) (прізвище та ініціали)

Завідувач кафедри Петрик М.Р.
(підпис) (прізвище та ініціали)

Рецензент
(підпис) (прізвище та ініціали)

Тернопіль - 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Петрик М.Р.
(підпис) (прізвище та ініціали)

«__» _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

Студенту Дацку Андрію Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-застосунку для відстеження курсу криптовалют

Керівник роботи Гащин Надія Богданівна, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «15» 04 2024 року № 4/7-344

2. Термін подання студентом завершеної роботи 24.06.2024 р.

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

1. Огляд предметної області.

2. Аналіз вимог та проектування архітектури застосунку

3. Реалізація та тестування застосунку

4. Безпека життєдіяльності, основи охорони праці

Висновки

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

1. Титулка. 2. Актуальність, мета, задачі дослідження. 3. Окремі програми для відстеження курсів криптовалют 4. Діаграма варіантів використання веб-застосунку. 5. Діаграма компонентів веб-застосунку. 6. Діаграма потоків даних для прецеденту «Дізнатися деталі криптовалюти». 7. Діаграма діяльності для прецеденту «Додати у вибране».

8. Програмні засоби. 9. Інтерфейс сторінки списку криптовалют.

10. Скріншоти роботи створеного веб-застосунку. 11, 12. Результати мануального тестування. 13. Висновки. Основні результати проведеного дослідження

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці			

7. Дата видачі завдання _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	16.04 – 18.04.24	Виконано
2.	Підбір джерел про веб-застосунки для відстеження курсу криптовалют	19.04 – 22.04.24	Виконано
3.	Опрацювання джерел про веб-застосунки для відстеження курсу криптовалют	23.04 – 29.04.24	Виконано
4.	Проведення дослідження щодо розробки веб-застосунку для відстеження курсу криптовалют	30.04 – 04.05.24	Виконано
5.	Розроблення програмного коду	05.05 – 12.05.24	Виконано
6.	Оформлення розділу «Огляд предметної області»	13.05 – 18.05.24	Виконано
7.	Оформлення розділу «Аналіз вимог та проектування архітектури застосунку»	19.05 – 24.05.24	Виконано
8.	Оформлення розділу «Реалізація та тестування застосунку»	25.05 – 31.05.24	Виконано
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи хорони праці»	01.06 – 05.06.24	Виконано
10.	Оформлення кваліфікаційної роботи	06.06 – 09.06.24	Виконано
11.	Нормоконтроль	10.06 – 13.06.24	Виконано
12.	Перевірка на плагіат	12.06 – 16.06.24	Виконано
13.	Попередній захист кваліфікаційної роботи	17.06 – 20.06.24	Виконано
14.	Захист кваліфікаційної роботи	25.06.24	

Студент

(підпис)

Дацко А.І.

(прізвище та ініціали)

Керівник роботи

(підпис)

Гашин Н.Б.

(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра містить: 51 сторінка, 23 рисунки, 44 джерел та 1 додаток.

БЛОКЧЕЙН, КРИПТОБІРЖА, КРИПТОВАЛЮТА, РЕНДЕРІНГ, ТРЕКІНГ, FLUX, TYPESCRIPT

У кваліфікаційній роботі бакалавра виконано огляд предметної області дослідження, проаналізовано альтернативні програмні рішення.

Розглянуті основні вимоги до веб- застосунку, реалізовано діаграму його варіантів використання. Описано архітектуру розробки, наведені діаграми потоків даних, компонентів та діяльності. Наведені засоби та підходи до реалізації веб-застосунку. Також були описані особливості реалізації деяких ключових модулів системи, представлені проміжні результати реалізації програми.

Здійснено мануальне функціональне тестування розробки. У процесі тестування програми не було виявлено зауважень, всі тести в межах представлених тестових середовищ були пройдені.

ABSTRACT

BLOCKCHAIN, CRYPTO EXCHANGE, CRYPTOCURRENCY,
RENDERING, TRACKING, FLUX, TYPESCRIPT

In the bachelor's thesis, an overview of the subject area of the study was performed, alternative software solutions were analyzed.

The main requirements for the web application were considered, and a diagram of its use options was implemented. The development architecture is described, diagrams of data flows, components and activities are given. Means and approaches to the implementation of a web application are given. Features of implementation of some key modules of the system were also described, intermediate results of program implementation were presented.

Manual functional testing of the development was carried out. In the process of testing the program, no comments were found, all tests within the presented test environments were passed.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

DOM (Document Object Model) – об'єктна модель документа.

DOM -дерево – інтерфейс, що обробляє HTML -документ як деревоподібну структуру в якій кожен вузол є об'єктом, що містить частину документа.

MPA (multi page application) - багатосторінковий додаток, сторінка повністю оновлюється при найменшій зміні станів на клієнті.

PWA (progressive web applications) - прогресивний веб-застосунок, що дозволяє запускати себе як у браузері, так і як десктопний або мобільний застосунок.

SPA (single page application) - односторінковий застосунок, що завантажується на одну HTML -сторінку, перехід між сторінками в таких застосунках реалізований засобами мови JavaScript.

UI (User Interface) – інтерфейс користувача.

UUID (Universally Unique Identifier) — стандарт ідентифікації, який використовується при створенні програмного забезпечення,

Блокчейн – незмінна, децентралізована база даних, доступ до якої мають усі вузли комп'ютерної мережі.

Лінтер (Linter) — це інструменти для дослідження програмного коду з метою виявлення помилок програмування, вад, порушень стилю, а також сумнівних чи підозрілих виразів.

Майнінг – видобуток блоків у мережі блокчейн для забезпечення криптовалютних платформ.

Монтування – вставка елементів у DOM- дерево.

Реактивність – спосіб автоматично оновлювати систему залежно від зміни потоку даних.

Рендеринг – процес отримання зображення за моделлю за допомогою компіляції програми.

ЗМІСТ

Вступ.....	8
1 Огляд предметної області.....	9
1.1 Поняття блокчейн.....	9
1.2 Поняття криптовалюти та їх види	9
1.3 Біржі криптовалют	11
1.4 Програми для відстеження курсів криптовалют.....	12
1.5 Платформи для розробки веб- застосунків.....	16
2 Аналіз вимог та проектування архітектури застосунку	19
2.1 Опис системи «Cryptocurrency».....	19
2.2 Вимоги до системи.....	19
2.3 Варіанти використання системи.....	20
2.4 Діаграма компонентів веб- застосунку Cryptocurrency	22
2.5 Архітектура веб- застосунку	23
2.6 Діаграма діяльності.....	25
3 Реалізація та тестування застосунку	27
3.1 Програмні засоби	27
3.2 Пакування та конфігурація застосунку.....	28
3.3 Реалізація сторінки списку криптовалют	31
3.4 Реалізація модуля відтворення графіків	34
3.5 Реалізація модуля додавання у вибране	36
3.6 Реалізація модуля завантаження графіків	38
3.7 Тестування розробки.....	39
4 Безпека життєдіяльності, основи охорони праці	42
4.1 Навчання працюючих і інструктажі з охорони праці.....	42
4.2 Санітарно-гігієнічні вимоги до умов праці	44
Висновки	47
Перелік джерел посилання	48
Додатки	

ВСТУП

Актуальність теми дослідження. У сучасному світі існує безліч видів зберігання грошових активів - фінітні валюти, нерухомість, фондовий ринок, дорогоцінні метали та ін. Одним з таких видів зберігання є криптовалюта. Це досить популярний спосіб зберігання коштів у тимчасовому світі, завдяки його надійності. Однак курси окремих криптовалют постійно змінюються, що вводить недосвідченого користувача в ступор.

Популярність криптовалют як виду зберігання грошових активів призвела до створення множини різних криптовалют. У зв'язку з цим звичайному користувачеві, недостатньо зануреному в блокчейн технології, але котрий бажає доторкнутися до світу блокчейн, стає дуже складно відстежувати котирування криптовалют на біржах, розбиратися в їх видах, а також почати заробляти на криптобіржах.

Метою роботи є розробка веб-програми для надійного відстеження курсів криптовалют.

Для досягнення мети потрібно вирішити наступні завдання:

- виконати аналіз предметної галузі;
- спроектувати веб- застосунок;
- реалізувати веб- застосунок;
- провести тестування розробки.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття блокчейн

Блокчейн зберігає у собі дані у цифровому форматі, при цьому кожен наступний вузол у мережі має хеш попереднього. Блокчейн гарантує безпеку та точність у записі та зберіганні даних, без третіх осіб. Ключова відмінність блокчейна від стандартних баз даних полягає в певному підході в їхньому структуруванні [4].

Блокчейн збирає інформацію групи, звані «блоки», що зберігають множину даних. "Блоки" мають певні можливості для їх зберігання, коли вони переповнюються, вони закриваються і посилаються на попередній заповнений вузол (блок), вся наступна інформація записується за аналогічним принципом [4].

Блокчейн працює за принципом видобутку цифрової інформації для запису та розповсюдження, але не для зміни. Усі записи в блокчейні не можна змінити, видалити чи іншим чином знищити. У результаті блокчейн прийнято називати DLT (з англ. «технологія розподіленого реєстру »).

Внаслідок високого ступеня безпеки баз даних блокчейну, у багатьох користувачів виникла потреба зберігати свої грошові заощадження в блокчейні. Внаслідок цього блокчейн має ключову роль у розвитку криптовалютних систем, наприклад, таких як Bitcoin, Ethereum та інших [5].

Таким чином, блокчейн - це незмінна та надійна база даних, завдяки якій користувачі можуть зберігати свої заощадження та кошти.

1.2 Поняття криптовалюти та їх види

Криптовалюта - особливий вид коштів у цифровій формі, що використовує шифрування для захисту транзакцій. Випуск криптовалюти відбувається

децентралізовано. Усі учасники мережі рівноправні і можуть у будь-який момент переводити чи отримувати криптовалюту [7].

Прийнято виділяти такі ключові категорії криптовалют:

- Bitcoin (виділяється окремо, оскільки це одна з перших крипто валют у світі з відкритим вихідним кодом [8]);
- альткоїни;
- токени;
- стейблкоїни;
- № NFT;
- DeFi.

Альткоїни (або альтернативні коїни) - це будь-які криптовалюти зі своїм блокчейном (за винятком Bitcoin). У зв'язку з тим, що у Bitcoin відкритий вихідний код, розробники альткойнів можуть прискорювати транзакції, оптимізувати процес майнінгу, створювати різні автоматизовані контракти, формувати базу для роботи з крипто- застосунками.

Токени - цифрові активи, які мають свого власного блокчейна. Замість майнінгу токени випускаються у повній емісії. Найчастіше такі активи випускають різні компанії з метою залучення коштів у розвиток своїх продуктів. Інвестори, у свою чергу, отримують гарантії того, що компанія виконає перед ними свої зобов'язання.

Стейблкоїни - цифрові гроші, ціна яких прив'язана до матеріальних активів (звичайна валюта, золото тощо). Курс Bitcoin і альткоїнів досить нестабільний і змінюється щодня, тоді як стейблкоїни гранично стабільні, і незважаючи на те, що їх котирування можуть змінюватися, відбувається це набагато рідше і без різких стрибків [9].

NFT - невзаємозамінні токени. Найчастіше їх використовують для перенесення блокчейн-прав на володіння унікальними активами. Наприклад, твори мистецтва, внутрішньоігрові предмети в онлайн-іграх і навіть нотаріально засвідчені документи.

DeFi – децентралізовані фінансові сервіси. Це не окремі криптовалюти, а

платформи для поєднання різних видів цифрових активів та їх функцій.

Найпопулярнішими криптовалютами є:

- Bitcoin (BTC) [10];
- Ethereum (ETH) [11];
- Binance Coin (BNB) [12].

Таким чином, криптовалюта - це один із способів зберігати грошові активи, а також мати можливість заробити на стрибках і падіння їх цін.

1.3 Біржі криптовалют

Криптовалютна біржа - це майданчик реалізації торгівлі та обміну однієї цифрової валюти в іншу чи валюти певної держави (долари, євро та інших.). Біржа, поряд з майнінгом, один із способів отримання криптовалюти [13].

Завдяки високій волатильності криптовалюти використовують як спекулятивний інструмент, це дозволяє витягти з неї максимальний прибуток.

Біржі поділяються на два типи майданчиків.

- на яких передбачено можливість обмінювати криптовалюту на реальні валюти;
- на яких можна обмінювати лише криптовалюту на криптовалюту.

На даний момент існує безліч бірж зі своїми перевагами та недоліками. Нижче наведені найпопулярніші представники криптовалютних бірж:

- Poloniex [14];
- Exmo [15].

Poloniex – американська криптовалютна біржа, що включає в свій функціонал крім обмінника криптовалют детальну статистику.

Exmo - універсальний сервіс, що займається торгівлею найбільш популярних коїнів. Нижче наведені найпопулярніші криптовалюти, представлені на майданчику Exmo:

- BitCoin [8, 10];
- ETH [11];
- DogeCoin [16].

Основною перевагою цієї біржі є можливість обмінювати валюту на гривні, долари, євро та ін.

Таким чином, криптовалютна біржа – це місце, де користувач може обміняти або купити крипто-активи за фінітну або криптовалюту.

1.4 Програми для відстеження курсів криптовалют

Для того, щоб зробити правильний вибір криптовалюти при інвестуванні, а також дізнатися тенденції у світі блокчейну, користувачі використовують веб-застосунки з трекерами криптовалют. Більшість таких застосунків мають докладну статистику з графіками, статистикою, можливістю пошуку та сортування за певними параметрами (наприклад, за типом криптовалюти). Однак часто вони не мають функціоналу аутентифікації на платформі.

Онлайн трекер криптовалют Forklog [17] містить у собі повний набір інформації про криптовалюту, біржі та блокчейни в цілому. На головній сторінці програми розміщено список статей про блокчейн. Інтерфейс головної сторінки представлений на рис. 1.1.

Крім цього, Forklog має сторінки з котируваннями коїнів, докладною інформацією про них, сторінку з прогнозами тенденцій у світі блокчейна. Усередині сторінки немає можливості відсортувати коїни за найбільшою зміною, а також дізнатися важливі деталі коїна такі як історія появи, або до якого типу він відноситься. Також, у Forklog немає особливих можливостей, які могли б його виділити на тлі конкурентів - наприклад, порівняння двох коїнів на графіках. Крім цього, у застосунку Forklog відсутня можливість додавання криптовалюти до обраного, що не дозволить користувачам належним чином збирати за нею

статистику простим і зручним способом. Однак, Forklog пропонує користувачеві детальну інформацію на гнучких і кастомізованих графіках, дає можливість знайти бажаний коін за допомогою поля введення.

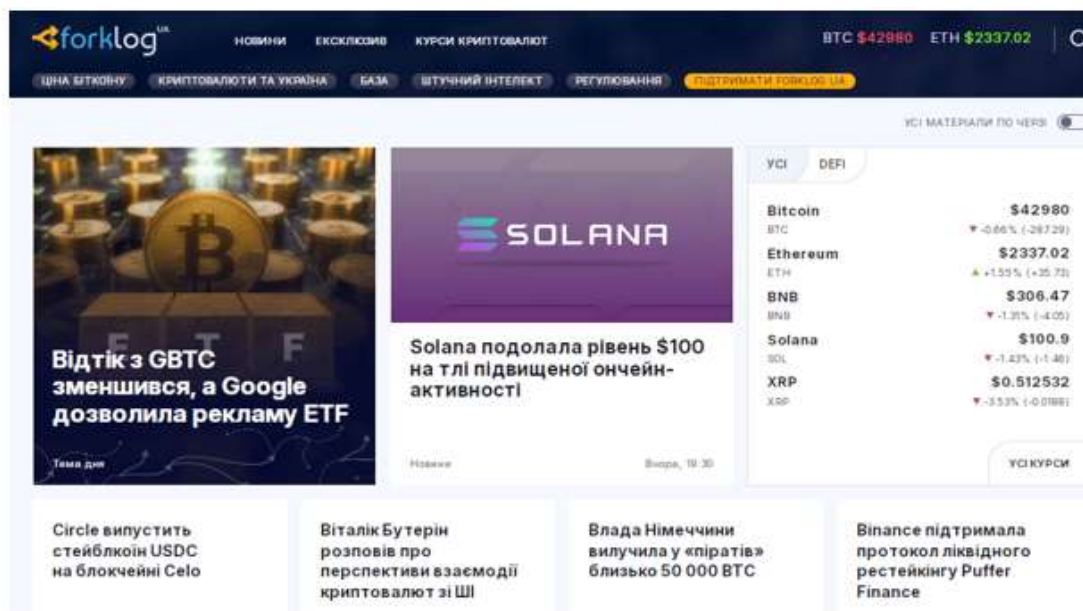


Рисунок 1.1 – Головна сторінка програми Forklog

Однак, застосунок має недоліки. Один з найголовніших недоліків - це відсутність додавання криптовалюти в обране, так як це спрощує роботу користувача з відстеження котирувань криптовалют всередині застосунку.

Разом з цим, що відрізняють від аналогів можливостей у веб- застосунку немає, в ньому немає ні можливості завантажити графіки в будь -якому форматі користувачеві на пристрій, ні можливості порівняти дві схожі криптовалюти.

Інтерфейс сторінки котирувань криптовалют представлений на рис. 1.2.

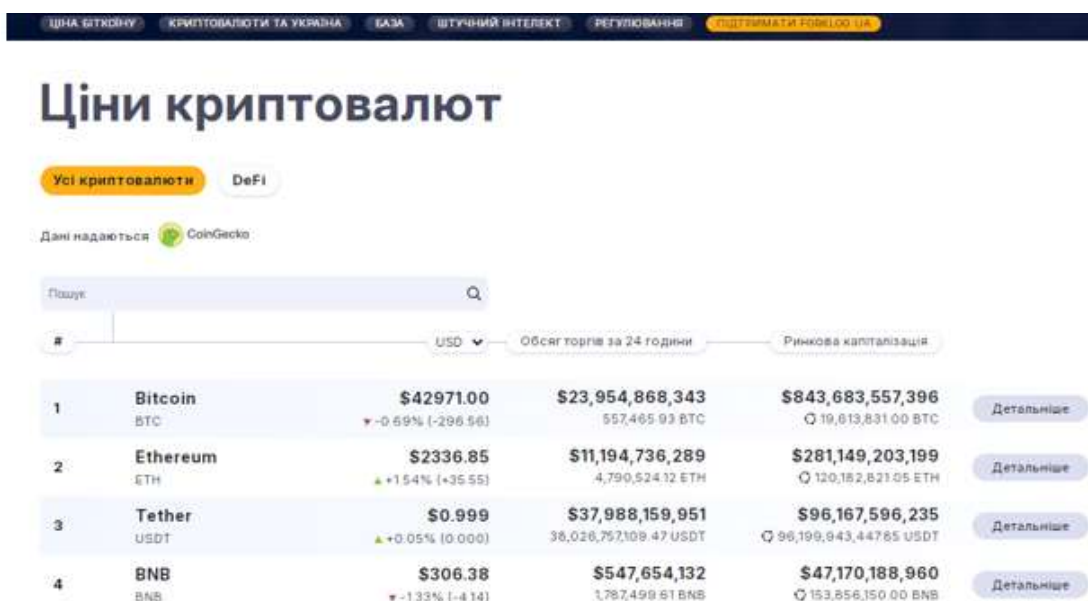


Рисунок 1.2 – Інтерфейс сторінки "ціни криптовалют" на ForkLog

Веб- застосунок BitInfoCharts [18] містить у собі детальну статистику по криптовалютах, що включає:

- капіталізацію на біржі;
- ціну на поточний момент у американському доларі;
- кількість блоків, отриманих майнінгом;
- нагороду за блок, здобутий майнінгом;
- цікаву інформацію про коін (дата першого видобутого блоку, кількість зірок на GitHub, останній коміт у GitHub репозиторій).

Також, як і Forklog, BitInfoCharts в інтерфейсі має пошук і сортування коїнів по різних полях. У застосунку є окрема сторінка зі списком усіх криптовалют, на якій відображена базова інформація про них, додатково при натисканні на будь-який із коїнів, програма перенаправить користувача на окреме посилання з більш докладною інформацією про криптовалюту. Інтерфейс сторінки BitInfoCharts представлений на рис. 1.3.



Рисунок 1.3 – Інтерфейс сторінки про BitCoin у веб- застосунку «BitInfoCharts»

На сторінці з детальною інформацією про конкретну криптовалюту відображається графік зміни ціни валюти за останній місяць, ціна вказана в доларах США. Крім цього, відображається кількість криптовалюти в цілому, капіталізація криптовалюти. Також, відображається інформація про середню ціну транзакцій, кількість блоків у блокчейні, обсяг блокчейну та гістограма комісії криптовалюти за 24 години. Гістограма BitCoin наведена на рис. 1.4.



Рисунок 1.4 – Гістограма комісії для BitCoin

Основна перевага «BitInfoCharts» - це детальна статистика та детальна інформація щодо криптовалют. У той час як основний недолік - застарілий дизайн.

Таким чином, основна мета застосунків для відстеження курсів криптовалют - відображення статистики з наочними графіками та тенденціями зростання або

спаду конкретного коїна.

1.5 Платформи для розробки веб- застосунків

У сучасному світі веб-розробки існує 3 різні види підходу до розробки застосунків - SPA, MPA, PWA.

Підхід SPA дозволяє спростити розробку, а також збільшити швидкість роботи програми. Проте головним мінусом такого підходу є повна відсутність SEO оптимізації, оскільки рендеринг сторінки відбувається на клієнті, тому пошукові роботи Google (та інші) бачать сторінку застосунку порожньою. Однак, у випадках, де контент статичний і важливий для SEO, цей недолік SPA може закрити серверний рендеринг, що дає змогу писати свій власний API для рендерингу на сервері [2].

MPA - один із перших підходів у розробці застосунків. Головний плюс такого підходу, на відміну від SPA – чудова SEO оптимізація, оскільки рендер програми відбувається на сервері. Недоліків такий підхід має багато. Перший з них - складність розробки. Для розробки MPA потрібно використовувати фреймворки, заточені на фронтенд і бекенд одночасно, наприклад, Angular, Flask або Ruby on rails, що ускладнює розробку і робить її менш підтримуваною та варіативною. Другий великий недолік MPA - погана швидкість роботи, внаслідок того, що перемальовування сторінки відбувається щоразу при зміні станів у застосунку [2].

Головна особливість і перевага PWA - кросплатформність, застосунок, написаний під персональний комп'ютер, може бути використаний як мобільний або веб- застосунок. Однак, мінусом такого підходу є підтримка деяких можливостей таких застосунків у деяких браузерях (Safari, Edge).

На даний момент веб- застосунки пишуться на різних фреймворках та бібліотеках мови JavaScript, в силу їх зручності та швидкості розробки. Вони не вимагають роботи з DOM- деревом [19], натомість у них є спеціальні методи та

функції для роботи з подіями, референсами та куки.

Найпопулярніші JavaScript -фрейморки на даний момент.

- React [19];
- Vue [20];
- Svelte [21].

Також найважливішим трендом веб-розробки є типізація. Для типізації у веб-застосунках використовують розроблена компанією Microsoft мова TypeScript. TypeScript компілюється в JavaScript і формально є корисною надбудовою над мовою, роблячи її строгішою і стабільнішою [22]. Типізація дуже корисна для командної розробки, а також у випадках, де потрібно працювати з великою кількістю запитів та валідацією.

Крім цього, в сучасних веб- застосунках використовують бібліотеки станів для зберігання даних та усунення проблеми сильної зв'язаності (prop-drilling) між множиною компонентів. Найпопулярніші бібліотеки для роботи зі станами:

- Redux (Redux Toolkit) [23];
- Vuex [20];
- MobX [24];
- Zustand [25].

Найчастіше використовувана - Redux, через хорошого налагоджувача як Redux DevTools. Однак у спільноті розробників багато скептично ставляться до Redux через сильну нагромадженість окремих частин коду [23]. Проблема була вирішена з випуском Redux Toolkit, проте в деяких проектах досі використовується стара версія Redux.

У силу того, що бібліотека React дозволяє розробнику не думати про роботу з DOM- деревом і сконцентрувати зусилля на роботі з даними було використано саме її. Крім цього, для роботи з API та зовнішніми сервісами [26] потрібна сувора типізація, щоб уникнути непередбачених ситуацій з наведенням типів даних. Також буде використано Redux-toolkit та Redux-saga [23] для роботи зі станами та асинхронними запитами до бази даних криптовалют. Для роботи зі стилізацією HTML елементів та компонентів буде використано CSS-in-JS бібліотеку styled-

components [27].

Підсумовуючи по першому розділі, слід зазначити, що моніторинг криптовалют - дуже важливе і потрібне завдання на даний момент, тому в рамках кваліфікаційної роботи буде реалізовано веб- застосунок для відстеження криптовалют.

2 АНАЛІЗ ВИМОГ ТА ПРОЕКТУВАННЯ АРХІТЕКТУРИ ЗАСТОСУНКУ

2.1 Опис системи «Cryptocurrency»

Метою даної роботи є розробка веб- застосунку для трекінгу криптовалют - Cryptocurrency.

Головне призначення даної програми - допомога в пошуку та систематизації актуальних даних про криптовалюту. Застосунок «Cryptocurrency» дозволить додавати цікаві для користувача криптовалюти та біржі до обраного для спрощеного моніторингу даних. Користувач буде мати можливість відсортувати криптовалюти за зростанням, падінням, популярністю та іншими важливими для нього параметрами, також застосунок передбачає можливість пошуку криптовалют за допомогою введення назви або ідентифікатора криптовалюти в спеціальну панель пошуку. Разом з цим користувач матиме можливість знайти криптовалюту з великого списку всіх можливих криптовалют. На окремій сторінці кожної криптовалюти для зручності користувача зміни котирувань будуть представлені на графіках. Також у користувача буде окрема сторінка обраного, на якій він зможе детальніше вивчити котирування цікавої його криптовалюти або біржі.

Ключовими особливостями Cryptocurrency будуть:

- можливість порівнювати дві криптовалюти на одному графіку;
- додавання криптовалюти або біржі у вибране;
- можливість зберегти картинку з графіком та інформацією про криптовалюту у форматі png.

2.2 Вимоги до системи

Визначення функціональних вимог. Веб- застосунок Cryptocurrency повинен відповідати функціональним вимогам, наведеним нижче, застосунок повинен:

- відображати список криптовалют та криптобірж на окремих сторінках;
- мати функціонал пошуку та сортування криптовалют;
- мати можливість переходу на окрему сторінку з криптовалютою для детального аналізу;
- володіти функціоналом відображення статистики криптовалют на графіках;
- містити в собі функціонал завантажувати графіки з обраною криптовалютою;
- містити в собі функціонал для додавання криптовалют та бірж до «вибраного».

Визначення нефункціональних вимог. В рамках аналізу вимог до веб-застосунку Cryptocurrency було висунуто нефункціональні вимоги, наведені нижче:

- веб- застосунок має під'єднуватися до зовнішнього веб-сервісу криптовалют та криптобірж [14];
- програма має бути адаптивною (підтримувати дозволи екрана всіх сучасних пристроїв);
- програма повинна підтримуватися в останніх версіях (останні 2) всіх популярних браузерів (Google Chrome, Safari, Opera).

2.3 Варіанти використання системи

Для проектування програми було використано мову графічного опису об'єктного моделювання UML. За висунутими до системи вимогами, була складена діаграма варіантів використання. Діаграма наведена на рис. 2.1.

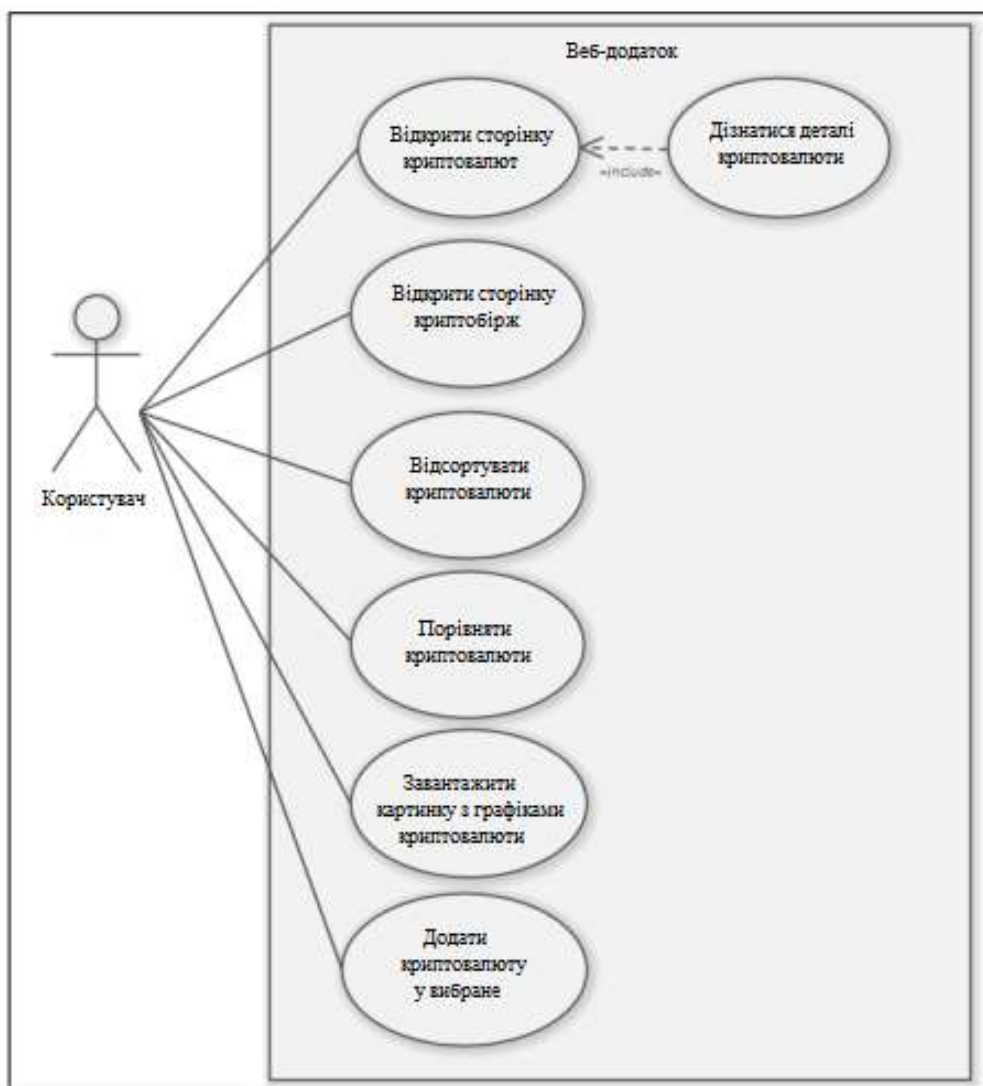


Рисунок 2.1 – Діаграма варіантів використання для веб-програми для відстеження курсів криптовалют

З системою взаємодіє один актор - безпосередній користувач системи. Користувач може виконувати дії, наведені нижче.

1. "Відкрити сторінку криптовалют". Користувач може відкрити окрему сторінку зі списком криптовалюту.

2. «Дізнатися деталі криптовалюту». Користувач, зайшовши на сторінку криптовалют, може дізнатися про деталі конкретного представника зі списку.

3. "Відкрити сторінку криптобірж". Користувач може відкрити окрему сторінку зі списком криптобірж.

4. "Відсортувати криптовалюту". Користувач може відсортувати

криптовалюти на відповідній сторінці.

5. "Порівняти криптовалюти". Користувач може на окремій сторінці з обраною криптовалютою порівняння криптовалюти з будь-яким іншим представником криптовалют і побачити різницю в їх ціні та зміні на графіках, а також в їх деталях.

6. "Завантажити картинку з графіками криптовалюти". Користувач може завантажити картинку з графіком криптовалюти, а також графіком порівняння криптовалют.

7. «Додати криптовалюту у вибране». Користувач може додати криптовалюту у вибране, для відстеження детальної інформації про неї.

2.4 Діаграма компонентів веб- застосунку Cryptocurrency

На рис. 2.2. представлена діаграма компонентів веб-програми Cryptocurrency.

У процесі проектування було прийнято рішення створити діаграму компонентів для веб-програми, що розробляється. Застосунок складається із семи компонентів, описаних нижче.

1. Application - компонент, який є тілом програми, у ньому знаходяться всі провайдери та нащадки. Крім цього, Application є точкою входу до програми.

2. Redux Provider - компонент, за допомогою якого застосунок додається можливість роботи зі сховищем.

3. Store - компонент глобального сховища програми.

4. Router - компонент, за допомогою якого в SPA - застосунок додається мультисторінковість за допомогою роутингу.

5. Routes - компонент, що зберігає в собі сторінки SPA.

6. Page - компонент, в якому реалізується бізнес-логіка застосунку.

7. UI - компонент, що не містить у собі бізнес-логіки, що відповідає тільки за малювання даних переданих у нього з батька.

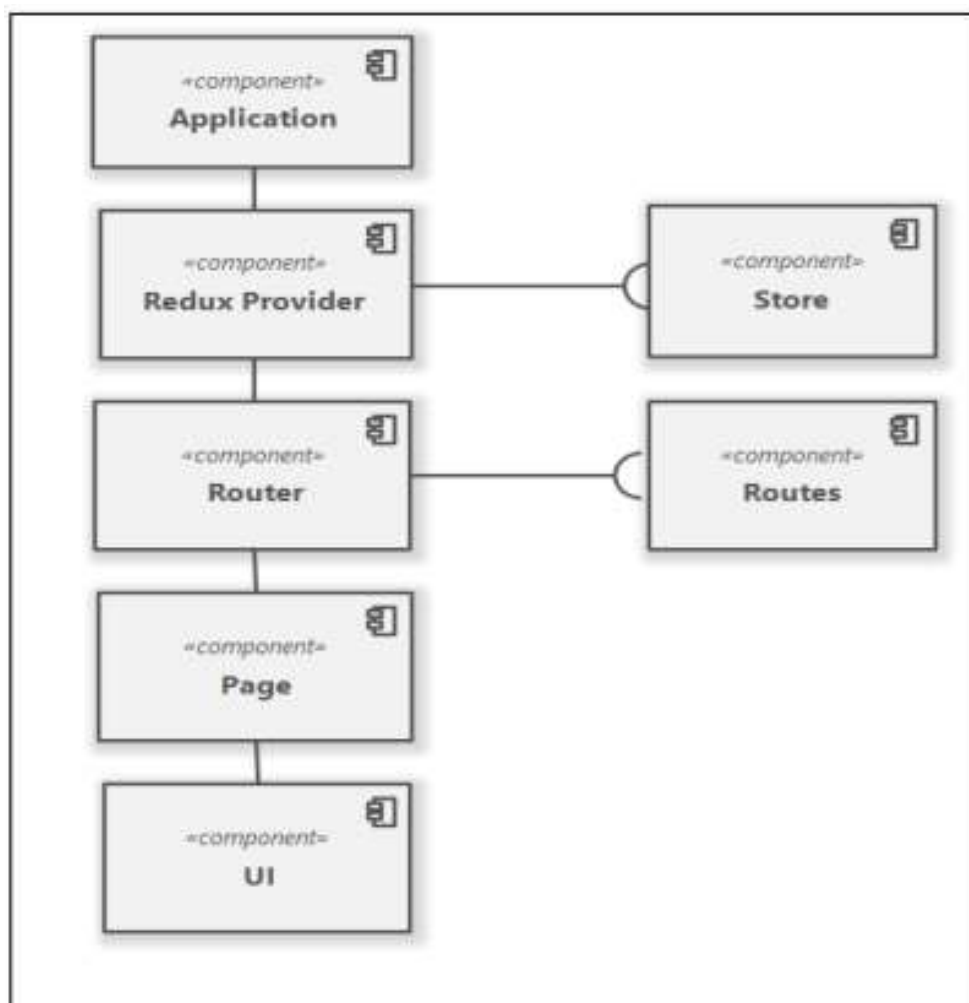


Рисунок 2.2 – Діаграма компонентів веб- застосунку

Вибір такого набору компонентів обумовлений сучасними підходами в архітектурі веб- застосунків, описаними в наступному пункті даного розділу.

2.5 Архітектура веб- застосунку

Веб- застосунок Cryptocurrency буде використовувати популярний на даний момент підхід до розробки програмного забезпечення – Flux. Це підхід, у якому вирішується проблема жорсткої зв'язаності компонентів, засобами реактивного програмування [28]. У такій архітектурі використовуються Flux сховища, що являють собою глобальний об'єкт, доступний на всіх етапах життєвого циклу

програми. На ці сховища підписуються окремі модулі або UI -компоненти, які за допомогою диспетчерів та селекторів передають, змінюють або отримують дані сховища. Реактивність допомагає окремим компонентам уникнути зайвого рендеру пов'язаних компонентів, за допомогою отримання оновлень даних безпосередньо зі сховища.

Веб- застосунок матиме статичний сервер і буде звертатися до стороннього веб-сервісу для отримання необхідних даних. Логіка роботи буде реалізована безпосередньо на клієнті завдяки описаній вище Flux архітектурі.

У зв'язку з цим, для прецеденту номер 2 з пункту 2.4 роботи - «Дізнатися деталі криптовалюти», була розроблена діаграма потоків даних [29]. Діаграма представлена на рис. 2.3.

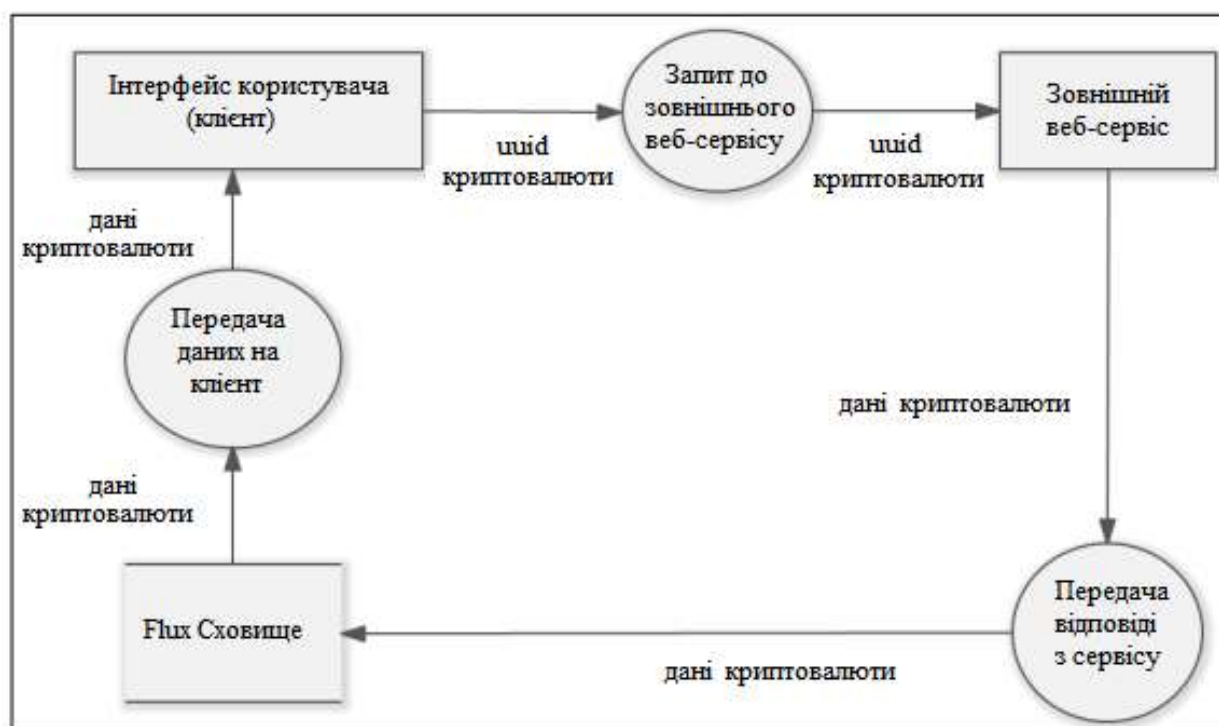


Рисунок 2.3 – Діаграма потоків даних для прецеденту «Дізнатися деталі криптовалюти»

Інтерфейс користувача ініціює надсилання запиту до зовнішнього веб-сервісу при монтуванні сторінки, потім надсилає запит до зовнішнього веб-сервісу криптовалют [26]. Після обробки запиту, зовнішній веб сервіс повертає веб-

застосунок дані по криптовалюті, потім він збирає дані та прибирає їх у глобальне сховище, а потім передає його в інтерфейс користувача.

2.6 Діаграма діяльності

Оскільки в розглянутих в п. 1.4 застосунках не було функціоналу для позначки цікавих користувачеві криптовалют, було вирішено додати таку відмінну можливість у веб- застосунок Cryptocurrency, що розробляється . Внаслідок цього була розроблена діаграма діяльності для прецедента «Додати до вибраного», згаданого в четвертому пункті другого розділу роботи. На рис. 2.4 представлена діаграма діяльності для прецеденту «Додати до вибраного».

Після того, як користувач натисне кнопку додати до обраного, ідентифікатор криптовалюти (uuid) буде переданий у локальне сховище браузера - – localStorage. Потім стан кнопки, на якому знаходиться слухач події додавання криптовалюти у вибране, перейде на стан видалення криптовалюти з обраного користувача. Варто відзначити, що видалення криптовалюти з обраного відбувається аналогічним чином.

Цей спосіб реалізації додавання до обраного є одним із найпростіших, так як він не вимагає звертатися до сервера для отримання даних по обраній криптовалюті, всі дії, пов'язані з додаванням до обраного, відбувається всередині браузера користувача. Разом з цим дані про обрану криптовалюту не будуть втрачені при перезавантаженні клієнта.

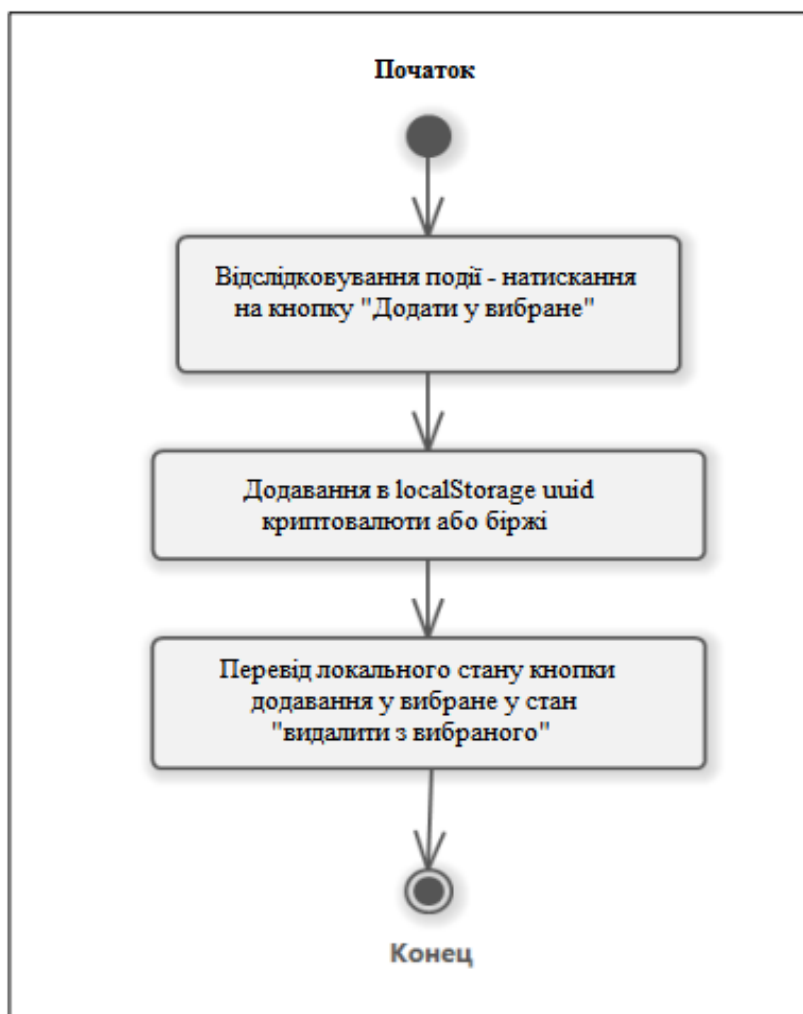


Рисунок 2.4 – Діаграма діяльності для прецеденту «Додати у вибране»

У цьому розділі були розглянуті основні вимоги до системи, було описано мету розробки програмного продукту та реалізовано діаграму варіантів використання веб- застосунку Cryptocurrency. Також була описана архітектура веб- програми, що розробляється, для трекінгу криптовалют. Разом з цим були наведені діаграма потоків даних для прецеденту «дізнатися деталі криптовалюти », діаграма компонентів та діаграма діяльності для прецеденту «додати криптовалюту до обраного».

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЗАСТОСУНКУ

3.1 Програмні засоби

Реалізація веб-застосунку для відстеження курсів криптовалют здійснювалася за допомогою наступного набору інструментів.

React версії 18.2. - бібліотека мови JavaScript, що дозволяє досить просто розробляти SPA, декомпонувати UI компоненти, звертатися до вузлів DOM- дерева та працювати з великими обсягами даних [19].

TypeScript версії 4.8.4 - мова програмування, що дозволяє додавати типізацію та об'єктно-орієнтоване програмування JavaScript. TypeScript при компіляції збирається в нативний JavaScript код, який зі свого боку успішно запускається у всіх сучасних браузерах [22].

Redux toolkit версії 1.8.4 - бібліотека станів JavaScript, що дозволяє зберігати стани у глобальному сховищі та відокремлювати логіку роботи з даними від UI компонентів [23].

Redux-saga версії 1.2.1 - бібліотека JavaScript для зв'язку серверної частини програми (API) безпосередньо з клієнтською частиною застосунку [30].

Styled-components версії 5.3.6 – бібліотека, що дозволяє писати CSS стилі та анімації всередині JavaScript [27].

ChartJS версії 3.9.1 – бібліотека JavaScript, що дозволяє будувати графіки на основі структурованих даних [31].

ESLint версії 7.25 - лінтер для JavaScript, для відстеження та усунення синтаксичних помилок та попереджень [32].

Vite версії 3.1 - пакувальник JavaScript - застосунків [33], що покращує швидкість складання програми в dev режимі.

3.2 Пакування та конфігурація застосунку

Налаштування пакувальника веб-програми. Сучасні браузерери здатні збирати і аналізувати інформацію тільки з HTML, CSS і JavaScript у чистому вигляді, в силу цього для роботи TypeScript, jsx (шаблонізатор HTML, що використовується в React), а також CSS-in-JS використовуються пакувальники, що перетворюють їх на зрозумілий формат для браузера.

Найпопулярнішими пакувальниками JavaScript -застосунків є:

- Webpack [35];
- ESBuild [36];
- Vite [33].

За даними npm, кількість завантажень Webpack близько 27 млн. завантажень на тиждень, у той час як у ESBuild і Vite кількість завантажень значно менше - у ESBuild 10 млн. завантажень, а у Vite 3 млн [35]. Висока кількість завантажень ESBuild обумовлена тим, що в якості dev залежності при складанні програми за допомогою Vite використовується ESBuild.

Конфігурація Vite досить легковажна і проста, на відміну конфігурацій Webpack, у яких зустрічається дуже багато складних і заплутаних залежностей [35]. Тому для спрощення розробки системи було обрано збирача Vite. Конфігурація пакування представлена кодом на рис. 3.1.

Vite підтримує роботу з готовими популярними плагінами, так у застосунку використовуються плагіни для роботи з React та ESLint. Крім цього, в конфігурацію пакування були додані порт для запуску програми локально і або, необхідний для більш простого звернення до абсолютних шляхів у коді.

```

import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';
import eslintPlugin from 'vite-plugin-eslint';
import path from 'path';
import { fileURLToPath } from 'url';

const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);

export default defineConfig({
  plugins: [
    eslintPlugin({
      exclude: ["/@react-refresh", "**/*.css"]
    }),
    react(),
  ],
  resolve: {
    alias: {
      '@': path.resolve(__dirname, './src'),
    },
  },
  server: {
    port: 3002,
  },
})

```

Рисунок 3.1 – Конфігурація пакування веб-застосунку

Налаштування TypeScript. Для розробки програми використовується мова TypeScript, яка так само, є обгорткою над мовою JavaScript. Для TypeScript використовуються спеціальні конфігурації для компілювання мови в JavaScript. Конфігурація TypeScript у застосунку представлена на рис. 3.2.

Конфігурація містить у собі набір правил компіляції коду в JavaScript точку входу в застосунок - src, які містять бібліотеки та спеціальні параметри, що виключають можливість нестрогої типізації з JavaScript.

Налаштування кросбраузерності веб-застосунку. Кожен сучасний браузер містить у собі свої особливі CSS - стилі, тому, для підтримки кросбраузерності верстки використовуються два підходи:

- нормалізація CSS;
- скидання стилів.

```

{
  "compilerOptions": {
    "target": "ESNext",
    "useDefineForClassFields": true,
    "lib": ["DOM", "DOM.Iterable", "ESNext"],
    "allowJs": false,
    "skipLibCheck": true,
    "esModuleInterop": false,
    "allowSyntheticDefaultImports": true,
    "strict": true,
    "forceConsistentCasingInFileNames": true,
    "module": "ESNext",
    "moduleResolution": "Node",
    "resolveJsonModule": true,
    "isolatedModules": true,
    "noEmit": true,
    "jsx": "react-jsx"
  },
  "include": ["src"],
  "references": [{ "path": "./tsconfig.node.json" }]
}

```

Рисунок 3.2 – Конфігурація TypeScript у застосунку

Нормалізація CSS - підхід, у якому задаються стилі поверх браузерних, цей підхід дозволяє використовувати потрібні стилі у всьому застосунку, без написання додаткових.

Скидання стилів - підхід, при якому всі реалізовані всередині браузера стилі скидаються до стилів зі специфікації HTML, такий підхід дозволяє робити верстку більш гнучкою і передбачуваною.

Всередині програми використовується скидання стилів, оскільки цей підхід є більш передбачуваним та масштабованим.

Налаштування лінера веб-програми. Для відстеження помилок та попереджень під час розробки у мові JavaScript та TypeScript використовуються лінери та форматувальники коду. Найпопулярнішим є ESLint [36]. За даними npm він має 31 млн завантажень за тиждень [34]. ESLint дозволяє відстежувати помилки та попередження за заданими правилами в конфігурації, а також усувати деякі з них. Код конфігурації лінера представлений на рис. 3.3.

```

module.exports = {
  env: {
    node: true,
    es2020: true
  },
  globals: { React: true },
  root: true,
  plugins: ["@typescript-eslint"],
  extends: [
    "eslint:recommended",
    "plugin:@typescript-eslint/recommended",
    "plugin:react/recommended",
    "plugin:react-hooks/recommended"
  ],
  settings: {
    react: {
      version: "detect"
    }
  },
};

```

Рисунок 3.3 – Конфігурація лінера в застосунку

3.3 Реалізація сторінки списку криптовалют

Для зручності розробки завдання по реалізації сторінки списку криптовалют була декомпонована на наступні пункти:

- верстка UI компонентів на сторінці;
- реалізація логіки зберігання даних клієнта;
- реалізація логіки відображення даних клієнта;
- реалізація логіки взаємодії із зовнішнім веб-сервісом.

У рамках першого пункту була реалізована анімація завантаження даних на клієнті, реалізовано поле введення тексту, перемикач опції, компонент криптовалют та пагінація. Детально в рамках випускної кваліфікаційної роботи будуть розглянуті компоненти пагінації та завантаження даних. Частина коду верстка та CSS компонента анімації завантаження даних на клієнта наведено на рис. 3.4.

```

export const Animation = keyframes`
  0%, 100% {
    margin-top: 0;
  }

  50% {
    margin-top: 2rem;
  }
`;

export const Dot = styled.div`
  background-color: ${COLORS.duckEggBlue};
  border-radius: 50%;

  animation: ${Animation} 1s infinite;
  animation-delay: ${(props: DotProps) => props.delay};
`;

const DOTS = [
  '0s',
  '0.05s',
  '0.1s',
  '0.15s',

```

Рисунок 3.4 – Фрагмент верстки компонента анімації

Для реалізації функції анімації був використаний CSS контролер `keyframes`, що дозволяє задавати правила анімації компонентів [3]. Для масштабованості компонента `Loader`, малювання компонента `Dot`, поміщена в ітератор масиву `DOTS`.

Компонент пагінації не містить логіки, а використовується лише як UI компонент і вся логіка відбувається без його участі, компонент лише відповідає за рендер і монтування вмісту на сторінці. Частина програмного коду компоненту пагінації наведений на рис. 3. 5.

Компонент пагінації приймає слухача події натисканням на кнопку (номер сторінки), кількість даних і поточну вибрану сторінку. За допомогою прапорця `isActive` визначається відображення вибраної та невибраної сторінки. Якщо прапорець `isActive` істинний, подія `onClick`, що відповідає за вибір сторінки, не спрацює.


```

const Pagination = (props: PaginationProps) => {
  const {
    onClick
  } = props;

  const totalCount = useSelector(makeSelectTotalCount);

  const currentPage = useSelector(makeSelectCurrentPage);

  const pagination: any[] = [];

  for (let i = 0; i < Math.floor(totalCount/10); i++) {
    pagination.push(i + 1);
  }

  return (
    <PaginationWrapper>
      {pagination.map((page) => {

```

Рисунок 3.5 – Фрагмент компонент пагінації

Для отримання даних із зовнішнього веб-сервісу використовується бібліотека `redux -saga` [30]. Ця бібліотека дозволяє накладати побічні ефекти на `actions`, підключених до `redux редьюсерам` [23], за допомогою такої конструкції, як генератори JavaScript [38]. На рис. 3.6 наведено логіку отримання даних із зовнішнього веб-сервісу.

```

export function* getCoinsFromApi({ payload }: any): any {
  try {
    const {
      limit,
      offset,
    } = payload;

    const search = yield select(makeSelectSearchParams);
    const orderBy = yield select(makeSelectOrderBy);
    const direction = yield select(makeSelectOrderDirection);

    const response = yield call(request, getCoinsApi(limit, offset, search,
      orderBy, direction), COINS_REQUEST_OPTIONS);
    if (response) {
      yield put(getCoinsResponse(response));
      yield put(updateTotalCount(response?.data?.stats?.total));
    }
  } catch (e) {
    console.warn('coins request error:', e);
  }
}

```

Рисунок 3.6 – Логіка отримання даних із зовнішнього веб-сервісу

При доставці події `getCoins` в сховище `redux`, спрацьовує побічний ефект - `getCoinsFromApi`, що створює GET запит до сервера. За допомогою вбудованого ефекту `select` з бібліотеки `redux-saga` збираються параметри [3] зі сховища `redux`. За допомогою вбудованого ефекту `call` створюється запит до зовнішнього веб-сервісу із заданими параметрами, після того, як клієнт отримує асинхронну відповідь із сервера, дані записуються в сховище за допомогою ефекту `put` та `redux actions`.

В результаті цієї реалізації була отримана сторінка списку криптовалют, що знаходиться на сторінці `/coins`. Зовнішній вигляд сторінки списку криптовалют показано на рис. 3.7.

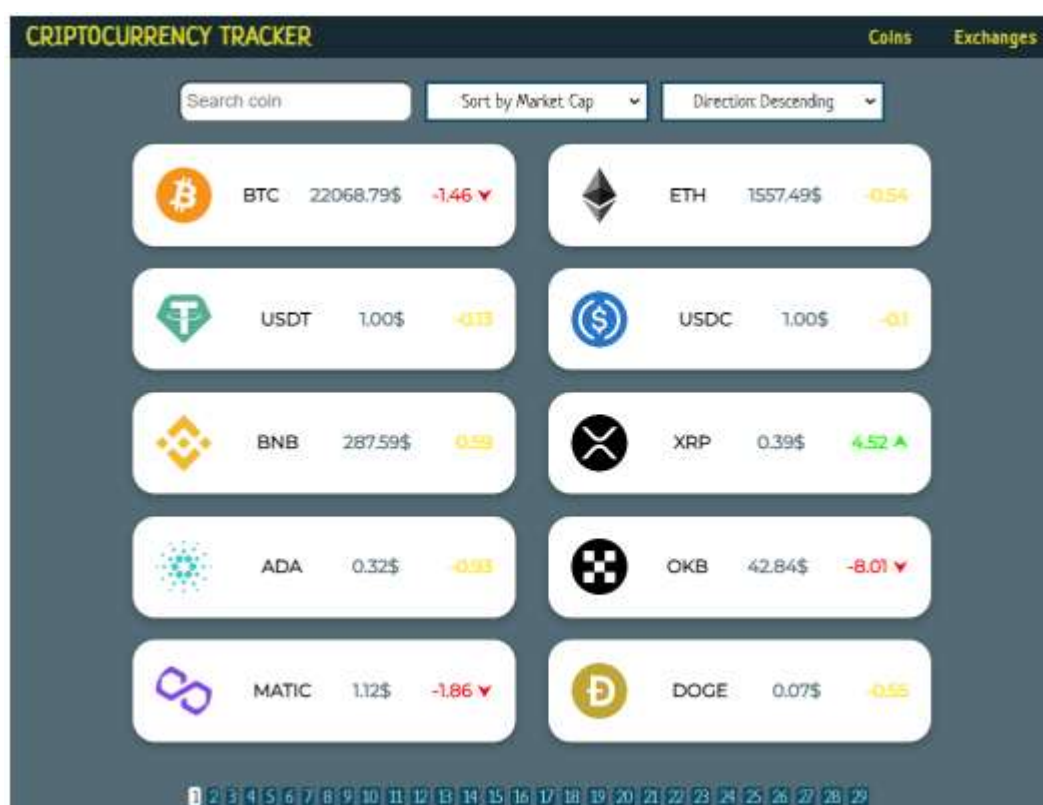


Рисунок 3.7 – Інтерфейс сторінки списку криптовалют

3.4 Реалізація модуля відтворення графіків

Модуль відтворення графіків реалізований за допомогою бібліотеки `ChartJS`

[31]. Серед UI компонентів цієї бібліотеки було обрано компонент Line, що здійснює рендер графіка із заданими опціями та даними. Верстка цього компонента показана на рис. 3.8.

```
<coinStyles.ChartWrapper>
  <Line
    options={OPTIONS}
    data={data}
  />
</coinStyles.ChartWrapper>
```

Рисунок 3.8 – Верстка компонента Line

Властивість data у компоненті Line наповнюється з redux сховища. У сховищі дані, необхідні для побудови графіка, потрапляють після надсилання запиту під час монтування сторінки. При отриманні відповіді з сервера дані збираються в потрібний для ChartJS формат. Реалізація картингу даних для відтворення графіків представлена на рис. 3.9.

```
const response = yield call(request, formatCoinRequest(uuid), options);

if (response) {
  yield put(getCoinResponse(response));

  const coin = yield select(makeSelectCoinData);

  const labels = coin?.sparkline
    .map((spark: string, sparkIndex: number) => `${sparkIndex}h`);

  yield put(updateData({
    labels,
    datasets: [{
      label: coin?.symbol,
      data: coin?.sparkline.map((spark: string) => Number(spark)),
      borderColor: COLORS.bioticGrasp,
      backgroundColor: COLORS.bioticGrasp,
    }],
  }));
}
```

Рисунок 3.9 – Реалізація картингу даних для відображення графіків

Для реалізації порівняння двох графіків було реалізовано додаткову подію,

яка при його виклику додає параметр `datasets` дані іншої криптовалюти, тим самим утворюється накладення двох графіків.

Приклад такої ситуації зображено на рис. 3.10.

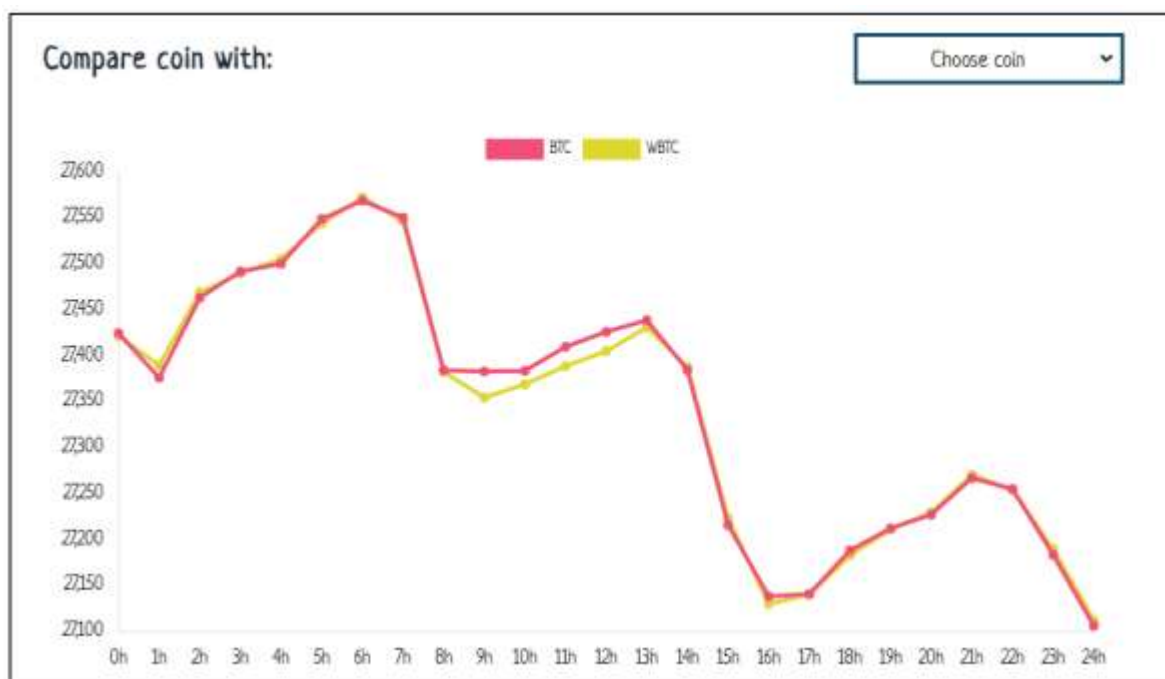


Рисунок 3.10 – Графік порівняння криптовалют BTC та Wrapped BTC

3.5 Реалізація модуля додавання у вибране

Для додавання у вибране був використаний об'єкт `localStorage` [38]. Це веб-сховище, яке дозволяє зберігати дані всередині браузера. За допомогою методів `getItem` та `setItem` здійснюється запис та читання даних об'єкта. Реалізація функцій додавання до обраного та видалення біржі з обраного наведена на рис. 3.11.

```
const [isFavorite, setIsFavorite] = useState(localStorage.getItem('favoriteExchange') === uuid);

const handleAddInFavorites = () => {
  localStorage.setItem('favoriteExchange', uuid);
  setIsFavorite(true);
}

const handleRemoveFromFavorites = () => {
  localStorage.removeItem('favoriteExchange');
  setIsFavorite(false);
}
```

Рисунок 3.11 – Функції додавання біржі до обраного та видалення біржі з обраного

При рендері елемента списку бірж задається значення стану `useState` обраної біржі з об'єкта `localStorage`, при спрацюванні події `handleAddInFavorites` додається `uuid` обраної біржі, а також стан відображення кнопки відображається перемикається на `true`. При спрацюванні `handleRemoveFromFavorites` з об'єкта `localStorage` знищується поле `favoriteExchange`, а стан кнопки перемикається на `false`.

Вибране відображається на домашній сторінці програми. Інтерфейс домашньої сторінки з доданими до обраного біржею та криптовалютою представлені на рис. 3.12.

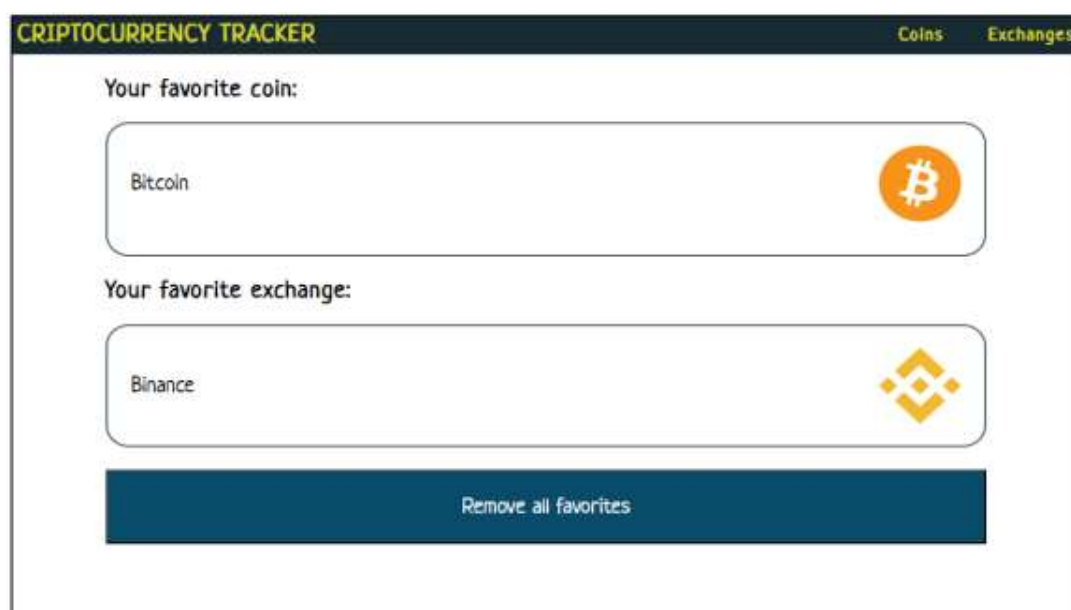


Рисунок 3.12 – Інтерфейс домашньої сторінки

3.6 Реалізація модуля завантаження графіків

Для реалізації завантаження графіків використовується npm пакет `html-to-image` [39]. Він дозволяє за допомогою конструкції `Promise` [38] асинхронно створювати скріншот DOM- вузла. Реалізація події завантаження графіка у форматі `png` наведена на рис. 3.13.

```
const handleDownloadGraphs = () => {
  if (!graphRef) return;

  setIsDisabledButton(true);

  toPng(graphRef.current, { cacheBust: true })
    .then((url) => {
      const link = document.createElement('a');

      const date = new Date();
      const currentTime = date.getHours() + ':'
        + date.getMinutes()
        + ':'
        + date.getSeconds();

      link.download = compareCoin
        ? `${currentTime}-${coin.name}-compared-${compareCoin.name}.png`
        : `${currentTime}-${coin.name}.png`;

      link.href = url;
      link.click();

      setIsDisabledButton(false);
    })
    .catch((e) => {
      setIsDisabledButton(false);
      alert('error', e);
    });
};
```

Рисунок 3.13 – Реалізація події завантаження графіка

`Promise toPng` переводить поточний вузол DOM дерева в `Canvas`, завдяки чому з'являється можливість завантажити його на пристрій користувача .

Для реалізації під'єднання до вузла DOM дерева був використаний нативний для React хук `useRef`, що дозволяє отримати подання елемента верстки [19].

При відловленні події `handleDownloadGraphs` створюється посилання, потім асинхронно викликається завантаження скріншота вузла, після цього

користувачеві завантажується файл, в назві якого вказано найменування криптовалюти та дата завантаження.

У випадку, якщо користувач зробив порівняння двох криптовалют, на картинці буде відображено спільний графік криптовалют та їх детальну інформацію. Приклад завантаженого файлу представлений на рис. 3.14.



Рисунок 3.14 – Завантажений файл у форматі png

3.7 Тестування розробки

Відповідно до вимог, висунутих у роботі, було вирішено провести функціональне мануальне тестування. Для реалізації поставленого завдання було

прийнято рішення розмістити застосунок за допомогою GitHub Pages [40]. Для цього конфігурацію збірки vite було додано поле base: '/graduate work' з назвою репозиторія кодової бази веб-застосунку [41]. Разом з цим у конфігурацію проекту були додані скрипти для складання та розміщення застосунку. Скрипти представлені на рис. 3.15.

```
"predeploy": "npm run build",
"deploy": "gh-pages -d dist",
```

Рисунок 3.14 – Скрипти для розміщення програми на GitHub Pages

Скрипт deploy відповідає за публікацію пакування програми на сервері GitHub Pages. Перед публікацією спрацьовує скрипт predeploy, що відповідає за пакування програми. Після пакування програми до папки dist, її вміст публікується на сервер GitHub Pages.

Виходячи з вимог до системи було складено набір тестів. У табл. 3.1 представлений набір тестів для перевірки коректної роботи веб-застосунку.

Таблиця 3.1 – Набір необхідних тестів для перевірки роботи веб-програми

№	Назва тесту	Кроки	Очікуваний результат
1	2	3	4
1	Тестування пагінації на сторінці криптовалют	З домашньої сторінки натиснути на посилання Coins. У сторінці, що відкрилася, потрібно натиснути на цифру 2 серед сторінок	Вміст сторінки переключається з 1 на 2 сторінку
2	Тестування за позовом на сторінці криптовалют	На сторінці /coins у полі введення "Search coin" ввести назву криптовалюти	У списку криптовалют з'являться тільки криптовалюти з такою назвою (або такі, що включають це слово у свою назву)
3	Тестування сортування криптовалют	На сторінці /coins зробити вибір однієї з опцій сортування та впорядкованості	Список криптовалют сортується за заданими правилами

Продовження таблиці 3.1

1	2	3	4
4	Тестування від покриття сторінки конкретної криптовалюти	На сторінці /coins або на домашній сторінці натиснути на криптовалюту	Відбудеться перенаправлення на сторінку /coin/:uuid, де uuid - ідентифікатор крип товалюти
5	Тестування до зменшення крипто валюти в обране	На сторінці вибраної криптовалюти натиснути кнопку "Add in favorites". Перейти на домашню сторінку	Відображення криптовалюти у вибраному
6	Тестування за навантаження даних за графіками конкретної криптовалюти	На сторінці вибраної криптовалюти натиснути кнопку "Download Graphs".	Після натискання на кнопку, вона блокується, і через деякий час (залежно від потужності клієнта) завантажується картинка графіків у форматі png
7	Тестування порівняння двох криптовалют на графіках	На сторінці вибраної криптовалюти переключити опцію Compare coin with	Після натискання на кнопку, на клієнті завантажиться обрана криптовалюта, після чого графік оновиться і буде показано порівняння двох графіків . Разом з цим додається додаткова інформація щодо додаткової обраної для порівняння криптовалют
8	Тестування відображення даних по обраній валюті на домашній сторінці	На домашній сторінці навести на обрану валюту	Після наведення мишею (утримування кнопки криптовалюти на мобільному пристрої) відкривається подрібнена інформація та графіки за різні часові періоди по валюті

Разом з цим було обрано набір середовищ для тестування програми. Він представлений у табл. 3.2.

Таблиця 3.2 - Набір середовищ тестування веб-застосунку

Пристрій та операційна система	Браузер	Версія браузера
Acer aspire 5, Windows 10	Chrome	111.0.2
Acer aspire 5, Mac OS Ventura (Віртуальна машина)	Safari	16.3
OPPO A83, Android 7.1.1	Chrome	111.0.2

Маючи дані MDN, Chrome має схожий браузерний API з більшістю браузерів, крім Safari [3]. Тому для тестування на Windows і Android буде використано Chrome, а на Mac OS буде використано Safari.

Грунтуючись на даних з табл. 1 та табл. 2, було проведено мануальне функціональне тестування веб-застосунку Cryptocurrency. Результати тестування наведено у додатку А до роботи.

У процесі тестування програми не було виявлено зауважень, всі тести в межах представлених тестових середовищ були пройдені.

У цьому розділі були описані засоби та підходи до реалізації веб- застосунку Cryptocurrency. Разом з цим були описані деталі реалізації деяких ключових модулів системи, а також наведені проміжні результати реалізації програми

Також було проведено мануальне функціональне тестування, а також були описані засоби тестування та його результати.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Навчання працюючих і інструктажі з охорони праці

Однією із складових ефективної роботи з профілактики виробничого травматизму є належна підготовка, навчання та підвищення кваліфікації працівників з питань охорони праці. Загальний порядок проведення навчання з питань охорони праці встановлений Законом України «Про охорону праці» (ст. 18. «Навчання з питань охорони праці»).

Виконання вимог Закону України «Про охорону праці» в частині проведення навчання та перевірки знань з питань охорони праці здійснюється відповідно до Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держкомітету України з нагляду за охороною праці 26 січня 2005 р. № 15 (далі — Типове положення).

Нагляд за дотриманням вимог Типового положення здійснюють органи державного нагляду за охороною праці, а координацію та методичний супровід — Головний навчально-методичний центр та навчальні підрозділи експертно-технічних центрів Держгірпромнагляду.

Вивчення предмета «Охорона праці» при підготовці, перепідготовці та підвищенні кваліфікації працівників, які залучаються до виконання робіт з підвищеною небезпекою, на підприємстві регламентується п. 2.3 Типового положення. На підприємствах згідно з п. 1.1 Додатку 3 Типового положення навчання та перевірку знань з питань охорони праці повинні проходити керівники, заступники керівників, головні спеціалісти, керівники основних виробничих та технічних служб, безпосередньо пов'язані з організацією безпечного ведення робіт. Крім цього, згідно з п. 5 Додатку 3, навчання та перевірку знань з питань охорони праці мають проходити керівники, спеціалісти служб охорони праці, члени комісій з перевірки знань з питань охорони праці, особи, відповідальні за технічний стан і безпечну експлуатацію об'єктів підвищеної небезпеки підприємств.

Типове положення встановлює порядок та місце проведення навчання та

перевірки знань з питань охорони праці посадових осіб (п. 5.2 та п. 5.3). Посадові особи, перелік яких наведено в п. 5.2, проходять навчання у Головному навчально-методичному центрі Держнаглядохоронпраці. Перевірка знань цієї категорії посадових осіб проводиться комісією, створеною наказом Держгірпромнагляду.

Організацію навчання та перевірки знань з питань охорони праці працівників на підприємстві здійснюють працівники служби кадрів або інші спеціалісти, яким роботодавець доручив організацію цієї роботи. Навчання та перевірка знань з питань охорони праці працівників (виконавців і посадових осіб), які не залучаються до виконання робіт підвищеної небезпеки, проводиться не рідше ніж один раз на три роки. Посадові особи та працівники, які виконують роботи підвищеної небезпеки, проходять спеціальне навчання та перевірку знань відповідних нормативно-правових актів з охорони праці не рідше одного разу на рік.

Посадові особи малих підприємств (п. 5.4), які не мають можливості створити власні комісії з перевірки знань з питань охорони праці та провести навчання з питань охорони праці, проходять навчання та перевірку знань в навчальних закладах, які мають відповідний дозвіл.

Спеціальне навчання з питань охорони праці може проводитись безпосередньо на підприємстві або навчальним закладом, який має відповідний дозвіл. При проведенні такого навчання на підприємстві навчальні плани та програми розробляються з урахуванням конкретних видів робіт, виробничих умов і функціональних обов'язків працівників і затверджуються наказом керівника підприємства.

Періодичність інструктажів, навчання та перевірки знань з питань охорони праці залежить від видів виконуваних робіт та встановлюється Типовим положенням. Перевірка знань з питань охорони праці після проведення спеціального навчання проводиться комісією підприємства.

Якщо на підприємстві неможливо створити комісію з перевірки знань з питань охорони праці (п. 4.4 Типового положення), перевірка знань проводиться комісією спорідненого підприємства або Теруправління Держгірпромнагляду.

Всі працівники та посадові особи підприємства, включаючи посадових осіб,

відповідальних за виконання робіт підвищеної небезпеки (крім зазначених в п. 5.2 та п. 5.3 Типового положення), проходять навчання та перевірку знань з питань охорони праці на підприємстві. Типове положення не зобов'язує, але й не забороняє проводити навчання всіх виконавців та посадових осіб (особливо тих, що виконують роботи підвищеної небезпеки) в навчальних закладах. У нашій країні є багато підприємств, де таке навчання проводиться, і це має позитивні наслідки. Ті витрати, які при цьому несуть підприємства, перекриваються створенням більш безпечних умов праці і в результаті збереженням життя та здоров'я працівників.

Також в навчальних закладах проходять навчання та перевірку знань із загальних питань охорони праці всі посадові особи та фахівці, які проводять інструктажі або навчання підлеглих працівників з питань охорони праці, виконують роботи з проектування об'єктів, а також інші працівники, незалежно від того, передбачено таке навчання Типовим положенням чи ні.

4.2 Санітарно-гігієнічні вимоги до умов праці

На робочих місцях працівників, які відповідальні за експлуатацію сервісу управління механізмом авторських прав на мультимедійні файли, необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Приміщення для роботи з ЕОМ мають бути обладнані системами опалення, кондиціонування повітря, або припливно-витяжною вентиляцією. У приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості та рухливості повітря відповідно до норм та правил, а також ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування», затверджених наказом Мінрегіону від 25.01.2013 р. № 24.

Відповідно до санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99 в офісних приміщеннях, обладнаних ЕОМ, температура повітря повинна становити 22-25°C, відносна вологість повітря – 40-60 %, швидкість руху повітря – не більше 0,1 м/с [42].

Приміщення, призначені для роботи з ЕОМ, повинні мати природне освітлення. У виробничих приміщеннях, обладнаних ЕОМ, необхідно створити належне освітлення. При експлуатації сервісу управління механізмом авторських прав на мультимедійні файли, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18 [43]. Природне світло повинно бути бічним, зорієнтованим, як правило, на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості не нижче 1,5%. При виробничій потребі дозволяється експлуатувати ЕОМ у приміщеннях без природного освітлення за узгодженням з органами Держпромгірнагляду та органами й установами санітарно-епідеміологічної служби. Вікна приміщень повинні мати регульовальні пристрої для відчинення, а також жалюзі, штори тощо. Штучне освітлення приміщення з робочими місцями, обладнаними відеотерміналами ЕОМ загального та персонального користування, має бути всеосяжним і рівномірним. У випадку, коли переважають роботи з документами, допускається комбіноване освітлення (додатково до загального освітлення встановлюється світильники місцевого освітлення). Світильники розміщуються збоку від робочих місць (переважно ліворуч), або локально над робочим місцем (при розташуванні відеотерміналів ЕОМ за периметром приміщення). Як джерело світла при штучному освітленні застосовуються, як правило, люмінесцентні лампи. У світильниках місцевого освітлення допускається застосування ламп розжарювання. Рівень освітленості на робочому місці має становити 300-500 лк. При використанні комбінованого освітлення не допускається відблисків на поверхні екрана та збільшення освітлення екрана вище 300 лк.

Орієнтація вікон повинна бути на північ або північний схід, вікна повинні мати жалюзі, які можна регулювати, або штори; не дозволяється розміщувати

кабінети обчислювальної техніки у підвальних приміщеннях будинків; кабінети, обладнані комп'ютерною технікою, в навчальних закладах повинні розміщуватись в окремих приміщеннях з природним освітленням та організованим обміном повітря; стіни, стеля і підлога та обладнання кабінетів комп'ютерної техніки повинні мати покриття із матеріалів з матовою фактурою з коефіцієнтом відбиття: стін — 40- 50 %, стелі — 70 - 80 %, підлоги — 20-30 %, предметів обладнання — 40-60 % (робочого столу — 40-50 %, корпусу дисплею та клавіатури — 30-50 %, стелажів — 40-60 %); поверхня підлоги повинна мати антистатичне покриття та бути зручною для вологого прибирання; забороняється використовувати для оздоблення інтер'єру приміщень комп'ютерних кабінетів полімерні матеріали (дерев'яно-стружкові плити, шпалери, що придатні для миття, плівкові та рулонні синтетичні матеріали, шаровий паперовий пластик та ін.), що виділяють у повітря шкідливі хімічні речовини, які перевищують гранично допустимі концентрації; вміст шкідливих хімічних речовин в повітрі дошкільних та учбових приміщень з комп'ютерами не повинен перевищувати середньодобові концентрації [42].

Організація робочого місця фахівця із експлуатації сервісу повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидючи. Загальні ергономічні вимоги».

Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Рівень шуму не повинний перевищувати: на місцях, де працюють програмісти та оператори ЕОМ, 55 дБА, у лабораторіях, де складаються алгоритми та ведеться робота з документацією – 60 дБА, у машинному залі – 65 дБА, у приміщеннях, де розміщені гучні агрегати обчислювальних машин – 75 дБА.

ВИСНОВОК

В результаті розробки веб-програми для відстеження курсів криптовалют був проведений наступний набір робіт:

- було виконано аналіз предметної області;
- було зроблено проектування веб-застосунку;
- було реалізовано веб-застосунок для відстеження криптовалюти Cryptocurrency;
- було проведено тестування отриманого веб-застосунку.

Крім цього, планується розробити аналог цієї програми для мобільних пристроїв із використанням React Native, який має гарну сумісність із React.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Lewis A. The Basics of Bitcoins and Blockchains: An Introduction to Cryptocurrencies and the Technology that Powers Them. // Mango Media, 2018. – 408 с.
2. SPA vs. MPA: Pros, Cons & How To Make Final Choice. URL: <https://www.simicart.com/blog/spa-vs-mpa/> (дата звернення: 10.02.2024).
3. Документація браузерного API – MDN. URL: <https://developer.mozilla.org/en-US/> (дата звернення: 10.02.2024).
4. Когут Ю. Технології блокчейн та криптовалюта: ризики та кібербезпека // К.: Дакор, 2022. – 316 с.
5. Що таке блокчейн? URL: <https://egera.com/uk/shcho-take-blokcheyn> (дата звернення: 05.02.2024).
6. What is hashing in Blockchain. URL: <https://learn.bybit.com/blockchain/what-is-hashing-in-blockchain/> (дата звернення: 06.02.2024).
7. Що таке криптовалюта і як вона працює? URL: <https://blog.whitebit.com/uk/what-is-a-cryptocurrency/> (дата звернення: 06.02.2024).
8. Antonopoulos A. Mastering Bitcoin. // O'Really Media, 2014. – 298 с.
9. Stablecoins: definition, how they work, and types. URL: <https://www.investopedia.com/terms/s/stablecoin.asp> (дата звернення: 06.02.2024).
10. Веб-сайт криптовалюти BitCoin. URL: <https://bitcoin.org/en/> (дата звернення: 15.03.2024).
11. Веб-сайт криптовалюти Ethereum. URL: <https://ethereum.org/en/> (дата звернення: 15.03.2024).
12. Веб-сайт криптовалюти Binance. URL: <https://www.binance.com/en> (дата звернення 15.03.2024).
13. Crypto airdrop season: Why people are making thousands for 'free'. URL: <https://www.cnet.com/personal-finance/crypto/crypto-airdrop-season-why-people-are-making-thousands-for-free/> (дата звернення: 10.02.2024).
14. Криптовалютна біржа Poloniex. URL: <https://poloniex.com> (дата звернення: 15.03.2024).

15. Криптовалютна біржі Exmo. URL: <https://exmo.com> (дата звернення: 15.03.2024).

16. Веб-сайт криптовалюти Dogecoin. URL: <https://dogecoin.com> (дата звернення: 15.03.2024).

17. Веб-застосунок ForkLog. URL: <https://forklog.com/> (дата звернення: 15.03.2024).

18. Веб-застосунок BitInfoCharts. URL: <https://bitinfocharts.com> (дата звернення: 15.03.2024).

19. Документація JavaScript бібліотеки React. URL: <https://reactjs.org/> (дата звернення: 10.02.2024).

20. Документація JavaScript фреймворку Vue. URL: <https://vuejs.org/> (дата звернення: 15.03.2024).

21. Документація JavaScript фреймворку Svelte. URL: <https://svelte.dev/> (дата звернення: 15.03.2024).

22. Документація мови TypeScript. URL: <https://www.typescriptlang.org/docs/> (дата звернення: 10.02.2024).

23. Документація бібліотеки Redux. URL: <https://redux-toolkit.js.org/> (дата звернення: 10.02.2024).

24. Документація бібліотеки MobX. URL: <https://mobx.js.org/> (дата звернення: 15.03.2024).

25. Документація бібліотеки Zustand. URL: <https://docs.pmnd.rs/zustand/getting-started/introduction> (дата звернення: 15.03.2024).

26. Документація енд-поінтів зовнішнього веб-сервісу Coinranking. URL: <https://rapidapi.com/Coinranking/api/coinranking1> (дата звернення: 12.02.2024).

27. Документація CSS-in-JS бібліотеки styled-components. URL: <https://styled-components.com> (дата звернення: 30.01.2024).

28. Архітектура веб-застосунків Flux. URL: <https://medium.com/@marina.kovalyova/flux-the-react-jsapplication-architecture-773f515d068d> (дата звернення: 18.02.2024).

29. Діаграми потоків даних. <https://www.maxzosim.com/data-flow-diagrams/>

(дата звернення: 18.02.2024).

30. Документація бібліотеки Redux-saga. URL: <https://redux-saga.js.org/> (дата звернення: 15.03.2024).

31. Документація бібліотеки ChartJS. URL: <https://www.chartjs.org/> (дата звернення: 15.03.2024).

32. Документація лінера ESLint. URL: <https://eslint.org/> (дата звернення: 26.02.2024).

33. Документація пакувальника Vite. URL: <https://vitejs.dev/> (дата звернення: 26.02.2024).

34. Пакетний менеджер npm.js. URL: <https://www.npmjs.com/> (дата звернення: 26.02.2024).

35. Документація пакувальника Webpack. URL: <https://webpack.js.org/> (дата звернення: 26.02.2024).

36. Документація пакувальника ESBuild. URL: <https://esbuild.github.io/> (дата звернення: 26.02.2024).

37. Документація Rollup. URL: <https://rollupjs.org/> (дата звернення: 26.02.2024).

38. Flanagan D. JavaScript: The Definitive Guide, 7th Edition. // O'Really Media Inc 2020. – 1032 с.

39. Пакет html-to-image. URL: <https://www.npmjs.com/package/html-to-image> (дата звернення: 26.02.2024).

40. Сервіс для розміщення веб-сторінок GitHub Pages. URL: <https://pages.github.com/> (дата звернення: 02.04.2024).

41. Репозиторій веб-застосунків на GitHub. [URL: <https://github.com/ViaChessLove/graduate-work> (дата звернення: 09.04.2024).

42. Зеркалов Д.В. Безпека життєдіяльності та основи охорони праці. Навчальний посібник. К.: «Основа». 2016. – 267 с.

43. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. [Електронний ресурс] - Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18> (дата звернення: 10.05.2024).

ДОДАТКИ

ДОДАТОК А

Результати тестування веб-програми Cryptocurrency

№	Назва тесту	Пристрій	Браузер	Тест пройдений ?
1	Тестування пагінації на сторінці крипто валют	Acer aspire 5, Windows 10	Chrome	Так
		Acer aspire 5, Mac OS Ventura (Віртуальна машина)	Safari	Так
		OPPO A83, Android 7.1.1	Chrome	Так
2	Тестування пошуку на сторінці криптовалют	Acer aspire 5, Windows 10	Chrome	Так
		Acer aspire 5, Mac OS Ventura (Віртуальна машина)	Safari	Так
		OPPO A83, Android 7.1.1	Chrome	Так
3	Тестування сортування криптовалют	Acer aspire 5, Windows 10	Chrome	Так
		Acer aspire 5, Mac OS Ventura (Віртуальна машина)	Safari	Так
		OPPO A83, Android 7.1.1	Chrome	Так
4	Тестування відкриття сторінки конкретної криптовалюти	Acer aspire 5, Windows 10	Chrome	Так
		Acer aspire 5, Mac OS Ventura (Віртуальна машина)	Safari	Так
		OPPO A83, Android 7.1.1	Chrome	Так
5	Тестування додавання криптовалюти в обране	Acer aspire 5, Windows 10	Chrome	Так
		Acer aspire 5, Mac OS Ventura (Віртуальна машина)	Safari	Так
		OPPO A83, Android 7.1.1	Chrome	Так
6	Тестування завантаження даних за графіками конкретної криптовалюти	Acer aspire 5, Windows 10	Chrome	Так
		Acer aspire 5, Mac OS Ventura (Віртуальна машина)	Safari	Так
		OPPO A83, Android 7.1.1	Chrome	Так
7	Тестування порівняння двох криптовалют на графіках	Acer aspire 5, Win dows 10	Chrome	Так
		Acer aspire 5, Mac OS Ventura (Віртуальна машина)	Safari	Так
		OPPO A83, Android 7.1.1	Chrome	Так
8	Тестування відображення даних щодо обраної криптовалюти на домашній сторінці	Acer aspire 5, Win dows 10	Chrome	Так
		Acer aspire 5, Mac OS Ventura (Віртуальна машина)	Safari	Так
		OPPO A83, Android 7.1.1	Chrome	Так

ДОДАТОК Б
Диск з роботою