

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

## Кваліфікаційна робота

на здобуття ступеня вищої освіти «Бакалавр»

**«Розробка інформаційного веб-сайту криптовалют з використанням технологій ASP.NET Core, Angular та Entity»**

(тема кваліфікаційної роботи українською мовою)

**«Development of an Informational Cryptocurrency Website Using ASP.NET Core, Angular, and Entity Framework»**

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

Мухін Михайло Олександрович

(прізвище, ім'я, по-батькові здобувача)

Керівник асистент Молчанова А.Ю.

(науковий ступінь, вчене звання, прізвище, ініціали)

  
(підпис)

Рецензент д.т.н., доцент Соколов А.В.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:  
Протокол засідання кафедри  
Інформаційних технологій

№ 1 від 09 червня 2024 р.

Завідувачка кафедри

  
(підпис) КАЗАКОВА Надія  
(прізвище, ім'я)

Захищено на засіданні ЕК № 13,  
протокол № 20 від 20 червня 2024 р.

Оцінка задовільно / Е / 60  
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

  
(підпис) КОПИЧЕНКО Іван  
(прізвище, ім'я)

Одеса 2024

<b>ЗМІСТ</b>	
<b>СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ</b>	5
<b>ВСТУП</b>	6
<b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМНИХ СИСТЕМ</b>	8
1.1 Аналіз предметної області	8
1.2 Аналіз існуючих програмних систем	10
<b>2 ВИБІР ТА ОБГРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ</b>	15
2.1 Функціональні та не функціональні вимоги	15
2.2 Обґрунтування вибору операційної системи	15
2.3 Вибір середовища розробки	17
<b>3 ТЕХНОЛОГІЯ ІНФОРМАЦІЙНИХ ВЕБ-САЙТІВ</b>	21
3.1 Поняття технології інформаційного веб-сайту	21
3.2 Сфери використання технології інформаційних веб-сайтів	22
3.3 Основні принципи роботи технології інформаційного веб-сайту	23
3.4 Інструменти для розробки програмного забезпечення з використанням інформаційного веб-сайту	24
<b>4 ПРОЕКТУВАННЯ ВЕБ-САЙТУ</b>	27
4.1 Створення UML-діаграми варіантів використання системи	27
4.2 Створення діаграми діяльності (активностей)	28
4.3 Створення діаграми послідовності	30
4.4 Проектування макетів інтерфейсу застосунку	30
<b>5 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОГО ВЕБ-САЙТУ</b>	33
5.1 Робота з ASP.NET Core	33
5.2 Використання Angular	40
5.3 Опис та тестування застосунку	45
<b>ВИСНОВКИ</b>	52
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ</b>	54

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ОС – операційна система

ПЗ – програмне забезпечення

API – Application Programming Interface – програмний інтерфейс програми

IDE – Integrated Development Environment – інтегроване середовище розробки

ORM (Object-Relational Mapping) – це технологія, яка дозволяє розробникам працювати з базами даних, використовуючи об'єктно-орієнтований підхід замість традиційного SQL-запитів. ORM автоматично перетворює дані з бази даних в об'єкти в програмному кодї та навпаки, що спрощує роботу з базами даних та зменшує кількість ручного коду. Прикладами ORM є Entity Framework для .NET, Hibernate для Java, та Django ORM для Python.

СУБД (Система Управління Базами Даних) – це програмне забезпечення, яке дозволяє створювати, управляти та працювати з базами даних. СУБД забезпечує інструменти для збереження, оновлення, вилучення та управління даними, а також для підтримки безпеки, надійності та ефективності доступу до даних. Прикладами СУБД є Microsoft SQL Server, MySQL, PostgreSQL, Oracle Database та MongoDB.

SQL (Structured Query Language) – це стандартизована мова запитів, яка використовується для управління та маніпулювання даними в реляційних базах даних. SQL дозволяє виконувати різноманітні операції з даними, такі як вибірка (SELECT), вставка (INSERT), оновлення (UPDATE) та видалення (DELETE) даних. Крім того, SQL підтримує створення та управління структурами баз даних, такими як таблиці та індекси.

MVC (Model-View-Controller) – архітектурний шаблон для розділення додатка на три основні компоненти: модель (Model), представлення (View) та контролер (Controller). Це допомагає організувати код, роблячи його більш підтримуваним та зрозумілим.

DI (Dependency Injection) – патерн проектування, який використовується для управління залежностями об'єктів. DI дозволяє підвищити модульність коду, спрощуючи його тестування та розширення.

EF (Entity Framework) – ORM (Object-Relational Mapping) бібліотека для .NET, яка дозволяє розробникам працювати з базами даних, використовуючи об'єктно-орієнтований підхід. Entity Framework спрощує взаємодію з базами даних, автоматично генеруючи SQL-запити на основі коду C#.

JWT (JSON Web Token) – стандарт для створення токенів доступу, які використовуються для аутентифікації та авторизації користувачів у веб-додатках. JWT дозволяє безпечно передавати інформацію між клієнтом і сервером.

CORS (Cross-Origin Resource Sharing) – механізм, який дозволяє веб-додаткам на одному домені взаємодіяти з ресурсами на іншому домені. CORS використовується для забезпечення безпечного доступу до ресурсів між різними джерелами.

CI/CD (Continuous Integration/Continuous Deployment) – практики автоматизації інтеграції та розгортання програмного забезпечення. CI/CD допомагає забезпечити постійне тестування, інтеграцію нових змін та автоматизоване розгортання додатків.

REST (Representational State Transfer) – архітектурний стиль для розробки мережевих додатків, який використовує стандартні методи HTTP. RESTful сервіси широко використовуються для створення веб API, що забезпечують взаємодію між клієнтськими та серверними додатками.

## ВСТУП

У світі, де цифрові технології стрімко розвиваються, інформаційні веб-застосунки для криптовалют стають все більш популярними та актуальними. Вони дозволяють користувачам стежити за курсами, аналізувати ринкові тенденції та здійснювати транзакції безпосередньо через Інтернет, мінімізуючи потребу в особистих зустрічах або контактах. Такі платформи, як і мистецтво під час карантину, адаптуються до нових умов, пропонуючи користувачам віртуальні можливості для взаємодії.

Як швидко та безпечно отримати актуальну інформацію про криптовалюту, яка цікавить? Можна звернутися до фінансового консультанта або провести власний аналіз ринку. Але чи не буде зручніше мати доступ до інформаційного веб-застосунку, який надасть всю необхідну інформацію в один клік? Саме тому актуальною є розробка веб-застосунку для аналізу та торгівлі криптовалютами.

Сьогодні, коли майже кожна людина має доступ до Інтернету, створення веб-застосунку є важливим завданням. Метою цієї роботи є розробка інформаційного веб-застосунку “CryptoCurrencies info” для аналізу та торгівлі криптовалютами. Для досягнення цієї мети були сформульовані наступні завдання:

- провести аналіз ринку криптовалют;
- порівняти існуючі веб-застосунки для торгівлі криптовалютами;
- обґрунтувати вибір технологій та інструментів для розробки;
- спроектувати систему, використовуючи мову UML;
- реалізувати веб-застосунок;
- підготувати інструкцію для користувачів;
- провести тестування веб-застосунку.

Структура кваліфікаційної роботи бакалавра складається з вступу, 5 розділів, висновків, переліку посилань на 18 найменувань, 1 додатку. Повний обсяг проекту становить 90 сторінок, містить 31 рисунок і 1 таблицю.

## **1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМНИХ СИСТЕМ**

### **1.1 Аналіз предметної області**

У світі, де цифрові активи стають все більш важливими, криптовалюти відіграють ключову роль у формуванні нової економічної реальності. Цей аналіз ринку криптовалют має на меті висвітлити основні тенденції, виклики та можливості, які вони пропонують.

#### Тенденції ринку

З появою Bitcoin у 2009 році, криптовалютний ринок зазнав значного зростання. За останніми даними, загальна ринкова капіталізація криптовалют перевищує трильйони доларів, що свідчить про значний інтерес інвесторів. Криптовалюти пропонують альтернативу традиційним фіатним валютам та фінансовим системам, пропонуючи децентралізацію, безпеку та прозорість завдяки блокчейн технології.

#### Виклики ринку

Однак, разом з можливостями, криптовалюти також пропонують виклики. Волатильність цін є однією з основних проблем, яка може призвести до значних фінансових втрат для недосвідчених інвесторів. Регуляторні питання також залишаються невирішеними у багатьох країнах, що створює правову невизначеність.

#### Можливості ринку

Незважаючи на виклики, криптовалюти пропонують унікальні можливості. Вони можуть сприяти фінансовій інклюзії, надаючи доступ до фінансових послуг людям без банківських рахунків. Криптовалюти також можуть сприяти інноваціям у фінансовому секторі, пропонуючи нові форми інвестицій та способи переказу коштів.

Ринок криптовалют продовжує розвиватися, пропонуючи як можливості, так і виклики. Його вплив на глобальну економіку та фінансові системи буде залежати від того, як він буде регулюватися та інтегруватися у ширший економічний контекст. Для глибшого розуміння цього ринку необхідно проводити подальші дослідження та аналіз.

## 1.2 Аналіз існуючих програмних систем

### CoinGecko: Глибокий Аналіз Ринку Криптовалют

CoinGecko[5] є однією з найбільш комплексних аналітичних платформ у світі криптовалют. Вона надає детальний огляд ринку, включаючи не тільки ціни та обсяги торгів, але й глибокий аналіз фундаментальних показників, які впливають на ринкову динаміку.

- **Ціни та Обсяги Торгів:** CoinGecko відстежує ціни на тисячі криптовалют у реальному часі, надаючи користувачам можливість бачити актуальні ціни та денні обсяги торгів.
- **Ринкова Капіталізація:** Платформа відображає ринкову капіталізацію для кожної криптовалюти, що дозволяє користувачам оцінити загальну вартість ринку.
- **Розвиток Спільноти та Відкритого Коду:** CoinGecko аналізує зростання спільноти навколо кожної криптовалюти та її відкритий код, що є важливими індикаторами здоров'я та стійкості проекту.
- **Події та Метрики Блокчейну:** Платформа також відстежує важливі події, які можуть вплинути на ринкові ціни, та надає аналіз метрик блокчейну, таких як хешрейт та транзакції на блок.

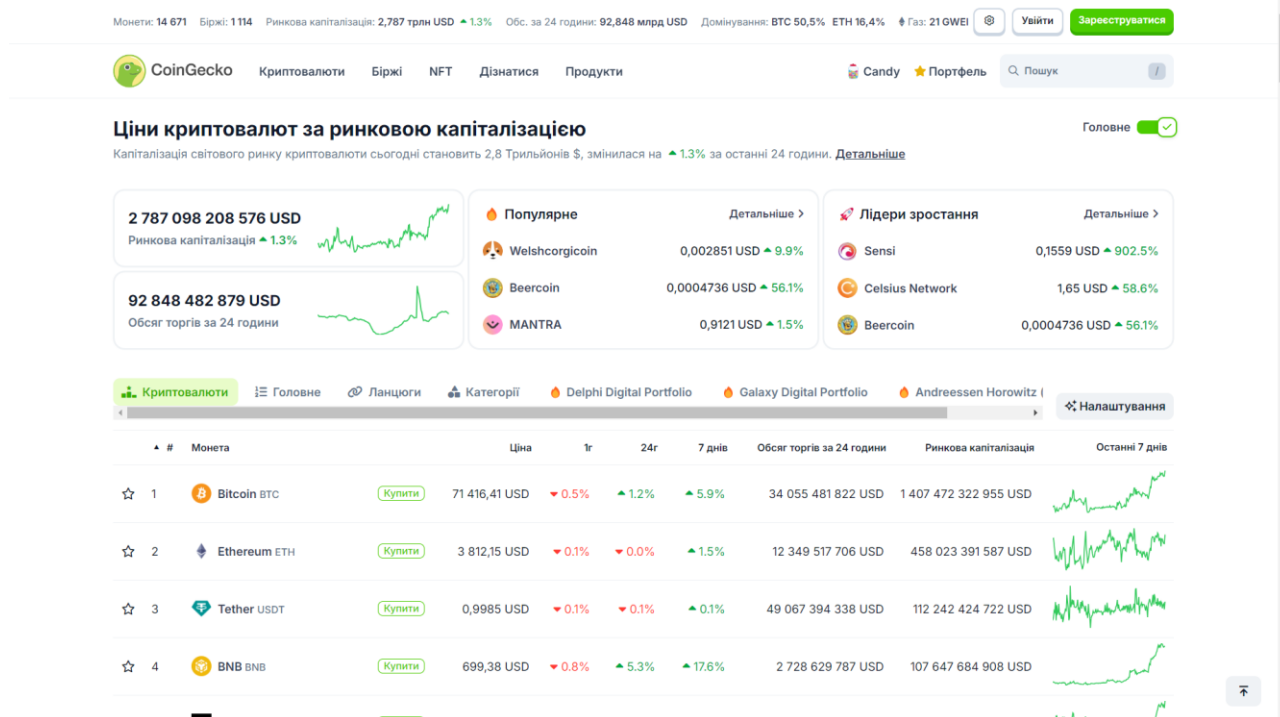


Рисунок 1.1 - Знімок веб-сайту CoinGecko

## CoinCar: Інструменти для Реального Часу Аналізу

CoinCar[6] пропонує інтуїтивно зрозумілі інструменти для відстеження ринкових даних у реальному часі. Ця платформа зосереджена на наданні актуальної інформації, що дозволяє користувачам швидко реагувати на ринкові зміни.

- **Актуальні Дані:** CoinCar надає інформацію про ціни та ринкові капіталізації криптовалют, оновлюючи дані без затримок.
- **Графіки та Тренди:** Користувачі можуть переглядати графіки для аналізу історичних та поточних трендів цін.
- **Порівняння Криптовалют:** Платформа дозволяє порівнювати криптовалюту за різними параметрами, що є корисним для оцінки потенціалу інвестицій.



Rank	Name	Price	Market Cap	VWAP (24hr)	Supply	Volume (24hr)	Change (24hr)
1	Bitcoin BTC	\$71,455.66	\$1.41t	\$70,892.30	19.71m	\$9.38b	1.23%
2	Ethereum ETH	\$3,816.74	\$458.52b	\$3,806.58	120.15m	\$5.03b	-0.10%
3	BNB BNB	\$700.80	\$116.89b	\$694.17	166.80m	\$919.67m	5.27%
4	Tether USDT	\$0.99	\$112.31b	\$0.99	112.32b	\$16.26b	0.01%
5	Solana SOL	\$174.29	\$80.17b	\$172.52	459.92m	\$736.28m	3.36%
6	USDC USDC	\$0.99	\$32.41b	\$0.99	32.41b	\$1.43b	0.01%
7	XRP XRP	\$0.52	\$24.02b	\$0.52	45.40b	\$515.91m	0.26%
8	Dogecoin DOGE	\$0.16	\$23.56b	\$0.16	144.59b	\$327.69m	1.30%
9	Cardano ADA	\$0.46	\$16.48b	\$0.46	35.71b	\$112.95m	0.05%
10	Shiba Inu SHIB	\$0.000025	\$15.29b	\$0.000025	589.27t	\$374.89m	6.16%
11	Avalanche AVAX	\$36.60	\$14.39b	\$36.16	393.24m	\$115.30m	2.08%

Рисунок 1.2 - Знімок веб-сайту CoinCap

Обидві платформи, CoinGecko та CoinCap, є важливими ресурсами для аналітиків та інвесторів, які прагнуть зрозуміти та аналізувати ринок криптовалют. Вони надають глибокий аналіз та інструменти для відстеження ринкових умов, що дозволяє користувачам приймати обґрунтовані рішення на основі даних. CoinGecko та CoinCap відіграють ключову роль у забезпеченні прозорості та доступності інформації, що є необхідним для здорового та ефективного функціонування криптовалютного ринку.

Pricebar – це додаток для Mac, який інтегрується з панеллю меню і допомагає користувачам стежити за цінами на різноманітні товари та послуги. Він пропонує зручний спосіб для моніторингу знижок та цінових змін, дозволяючи користувачам бути в курсі найкращих пропозицій без необхідності постійно перевіряти різні сайти вручну.

Основні функції Pricebar:

1. Інтеграція з панеллю меню:

- Додаток працює у фоновому режимі, розташовуючись у панелі меню Mac. Це забезпечує швидкий доступ до інформації про ціни, не займаючи багато місця на екрані.

2. Відстеження цін:

- Користувачі можуть додавати товари, які вони хочуть відстежувати. Додаток автоматично перевіряє ціни на цих товарах і повідомляє про будь-які зміни.

### 3. Сповіщення про знижки:

- Pricebar відправляє сповіщення користувачам, коли ціна на товар знижується. Це дозволяє швидко реагувати на знижки та придбати товар за вигідною ціною.

### 4. Підтримка різних магазинів:

- Додаток підтримує широкий спектр онлайн-магазинів, що дозволяє відстежувати ціни на товари з різних джерел. Серед популярних магазинів – Amazon, eBay, Walmart та інші.

### 5. Історія цін:

- Користувачі можуть переглядати історію цін на відстежувані товари. Це допомагає зрозуміти, чи є поточна пропозиція дійсно вигідною, порівнюючи її з минулими цінами.

### 6. Фільтри та налаштування:

- Додаток дозволяє налаштовувати фільтри для відстежуваних товарів, такі як мінімальна та максимальна ціна, частота перевірки цін та інші параметри.

### 7. Зручний інтерфейс:

- Інтерфейс Pricebar спроектований таким чином, щоб бути інтуїтивно зрозумілим і зручним у використанні. Всі основні функції доступні з панелі меню, що робить додаток легким у використанні навіть для новачків.

### 8. Синхронізація:

- Pricebar може синхронізувати дані між кількома пристроями, що дозволяє користувачам мати доступ до своїх відстежуваних товарів з будь-якого Mac-пристрою.

### Переваги використання Pricebar:

- Економія часу: Користувачам не потрібно вручну перевіряти ціни на різних сайтах, що економить час і зусилля.

- Ефективність: Швидке сповіщення про знижки дозволяє користувачам не пропустити вигідні пропозиції.
- Зручність: Інтеграція з панеллю меню Mac забезпечує зручний доступ до інформації в будь-який момент.

Недоліки та обмеження:

- Обмежена підтримка магазинів: Незважаючи на широкий спектр підтримуваних магазинів, деякі менші або локальні магазини можуть не бути підтримані.
- Залежність від інтернет-з'єднання: Додаток потребує стабільного інтернет-з'єднання для перевірки цін і надсилання сповіщень.

Pricebar є корисним інструментом для тих, хто хоче бути в курсі найкращих цінових пропозицій без постійного моніторингу різних сайтів. Завдяки своїм функціям, він допомагає користувачам економити гроші та робити вигідні покупки.

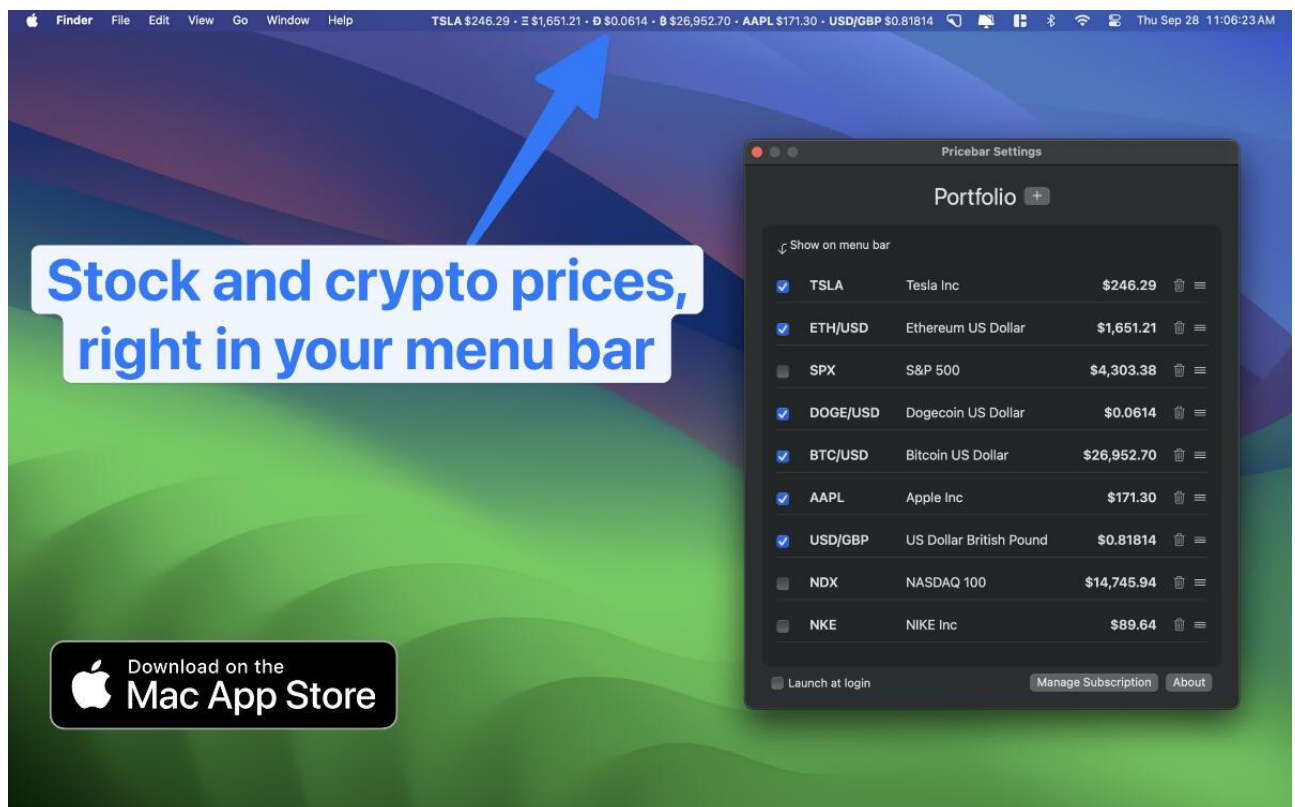


Рисунок 1.3 - Знімок додатку Pricebar

Wallmine – це потужний фінансовий інструмент, який надає широкий спектр послуг для інвесторів і трейдерів. Він поєднує в собі аналітичні можливості, дані про ринки та інструменти для прийняття інвестиційних рішень, забезпечуючи користувачів всебічною інформацією та інструментами для управління їхніми портфелями. Ось детальний огляд Wallmine та його функцій.

### Основні функції Wallmine

#### 1. Аналітика та дослідження ринків:

- Фінансові показники: Wallmine надає доступ до детальної фінансової інформації про компанії, включаючи звіти про доходи, баланси, звіти про грошові потоки та інші фінансові метрики.

- Показники рентабельності: Можливість перегляду показників рентабельності, таких як ROE (рентабельність власного капіталу), ROA (рентабельність активів), P/E (ціна до прибутку) та інших.

- Аналітичні графіки: Платформа надає інтерактивні графіки, які допомагають аналізувати історичні дані про ціни, обсяги торгів, індикатори та інші важливі метрики.

#### 2. Управління портфелем:

- Трекінг портфеля: Користувачі можуть створювати та відслідковувати свої інвестиційні портфелі, додавати активи, відстежувати їхні зміни в реальному часі.

- Аналіз портфеля: Інструменти для аналізу продуктивності портфеля, включаючи дохідність, волатильність, ризики та інші ключові показники.

- Автоматизовані звіти: Wallmine генерує автоматизовані звіти про стан портфеля, допомагаючи користувачам розуміти його поточний стан та історичні зміни.

#### 3. Інформація про ринки:

- Акції та ETF: Інформація про тисячі акцій і ETF з різних бірж по всьому світу.

- Криптовалюти: Платформа надає дані про ціни, обсяги та інші метрики для різних криптовалют.

- Облігації та інші активи: Інформація про облігації, товари та інші фінансові активи.

#### 4. Новини та аналітика:

- Фінансові новини: Агрегація новин з різних джерел, включаючи інформаційні агенції, блоги та фінансові портали.

- Аналітичні статті: Власні аналітичні статті Wallmine, які охоплюють важливі ринкові події, тенденції та рекомендації.

#### 5. Інструменти для технічного аналізу:

- Індикатори та осцилятори: Великий набір технічних індикаторів та осциляторів, таких як SMA, EMA, MACD, RSI та інші.

- Малювання на графіках: Можливість малювання ліній тренду, каналів, рівнів підтримки та опору прямо на інтерактивних графіках.

#### 6. Скринери акцій та криптовалют:

- Фільтри та критерії: Різноманітні фільтри для пошуку акцій і криптовалют за певними критеріями, такими як ринкова капіталізація, P/E, дохідність дивідендів, обсяг торгів та інші.

- Збережені скринери: Користувачі можуть створювати та зберігати власні скринери для швидкого доступу до них у майбутньому.

#### Переваги Wallmine

- Широкий спектр даних: Wallmine надає доступ до великої кількості фінансових даних, включаючи акції, ETF, криптовалюти, облігації та інші активи.

- Інтерактивність: Інтерактивні графіки та аналітичні інструменти роблять аналіз більш зручним та ефективним.

- Управління портфелем: Інструменти для створення, аналізу та трекінгу портфелів дозволяють інвесторам ефективно управляти своїми інвестиціями.

- Інтеграція новин: Агрегація новин з різних джерел забезпечує користувачам доступ до актуальної інформації про ринки та компанії.

- Підтримка багатьох активів: Wallmine підтримує широкий спектр фінансових інструментів, що робить його корисним для інвесторів з різними потребами та стратегіями.

## Недоліки та обмеження

- Платні функції: Деякі функції Wallmine можуть бути доступні лише у платних версіях, що може бути обмеженням для користувачів, які шукають безкоштовні інструменти.
- Складність для новачків: Широкий спектр функцій та інструментів може бути складним для освоєння новачками, які не мають досвіду в інвестиціях та фінансовому аналізі.

Wallmine є потужним інструментом для інвесторів і трейдерів, які хочуть мати доступ до детальної фінансової інформації, аналітики та інструментів для управління своїми портфелями. Завдяки своїм можливостям, він може стати незамінним помічником у прийнятті інвестиційних рішень та моніторингу ринкових тенденцій.

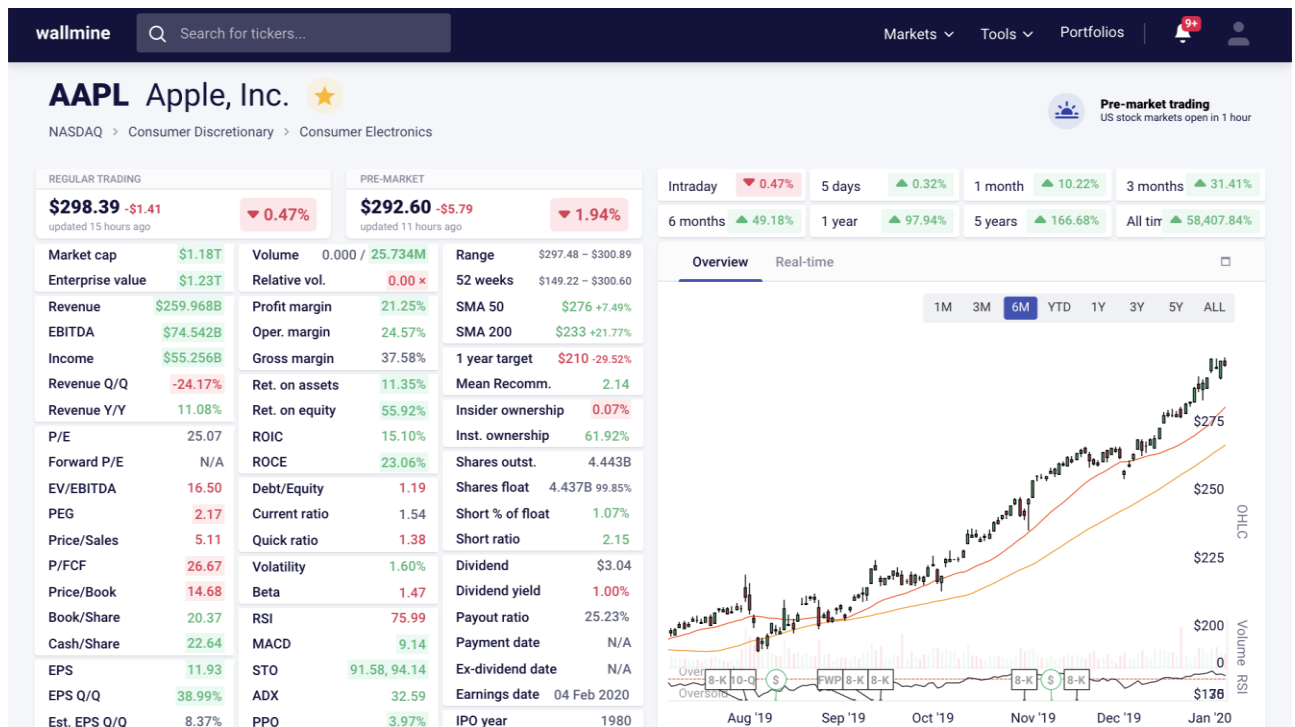


Рисунок 1.4 - Знімок додатку Wallmine

RocketFuel Crypto – це платформа для криптовалютних платежів, яка надає бізнесам можливість легко інтегрувати криптовалютні платежі у свої операції. Платформа прагне спростити процес прийому криптовалютних платежів та забезпечити безпеку транзакцій як для бізнесів, так і для споживачів. Ось детальний огляд сайту та його можливостей.

## Основні функції та можливості RocketFuel Crypto

### 1. Платіжні шлюзи для криптовалют:

- Інтеграція з електронною комерцією: RocketFuel Crypto пропонує платіжні шлюзи, які легко інтегруються з популярними платформами електронної комерції, такими як Shopify, WooCommerce, Magento та іншими. Це дозволяє онлайн-магазинам приймати криптовалютні платежі без необхідності значних технічних змін.

- Підтримка різних криптовалют: Платформа підтримує широкий спектр криптовалют, включаючи Bitcoin, Ethereum, Litecoin, Bitcoin Cash та інші. Це дає змогу клієнтам вибирати зручний для них спосіб оплати.

- Автоматичний конвертер валют: RocketFuel Crypto надає інструменти для автоматичної конвертації криптовалют у фіатні валюти (USD, EUR, тощо), що спрощує процес бухгалтерського обліку для бізнесів.

### 2. Безпека транзакцій:

- Шифрування даних: Платформа використовує сучасні методи шифрування для забезпечення безпеки транзакцій та захисту персональних даних клієнтів.

- Захист від шахрайства: RocketFuel Crypto застосовує різноманітні заходи для виявлення та запобігання шахрайським транзакціям, забезпечуючи безпечне середовище для покупок.

### 3. Користувацький досвід:

- Зручний інтерфейс: Платформа має інтуїтивно зрозумілий інтерфейс, що робить процес оплати зручним і простим як для бізнесів, так і для споживачів.

- Мобільні платежі: RocketFuel Crypto підтримує мобільні платежі, дозволяючи клієнтам здійснювати покупки за допомогою мобільних пристроїв.

### 4. Аналітика та звітність:

- Інформаційна панель: Платформа надає бізнесам доступ до інформаційної панелі, де можна відстежувати транзакції в режимі реального часу, переглядати аналітичні дані та генерувати звіти.

- Історія транзакцій: RocketFuel Crypto зберігає історію всіх транзакцій, що дозволяє бізнесам легко переглядати минулі платежі та аналізувати їхню ефективність.

## 5. Технічна підтримка:

- Підтримка клієнтів: Платформа надає цілодобову технічну підтримку для бізнесів і споживачів, допомагаючи вирішувати будь-які проблеми або питання, що можуть виникнути.

- Документація та ресурси: RocketFuel Crypto пропонує детальну документацію та навчальні матеріали для полегшення процесу інтеграції та використання платформи.

### Переваги використання RocketFuel Crypto

- Простота інтеграції: Легка інтеграція з популярними платформами електронної комерції дозволяє швидко додати можливість прийому криптовалютних платежів.

- Підтримка різноманітних криптовалют: Широкий вибір криптовалют робить платформу привабливою для клієнтів з різними уподобаннями.

- Безпека та захист: Сучасні методи шифрування та захисту від шахрайства забезпечують безпечне середовище для транзакцій.

- Зручність використання: Інтуїтивно зрозумілий інтерфейс і підтримка мобільних платежів забезпечують позитивний досвід для користувачів.

### Недоліки та обмеження

- Залежність від криптовалют: Криптовалютні платежі можуть бути нестабільними через волатильність криптовалют, що може створювати ризики для бізнесів.

- Регуляторні питання: В деяких регіонах можуть існувати регуляторні обмеження щодо використання криптовалют, що може вплинути на можливість використання платформи.

RocketFuel Crypto є потужним інструментом для бізнесів, які хочуть інтегрувати криптовалютні платежі у свої операції. Завдяки своїм функціям та можливостям, платформа забезпечує зручний і безпечний спосіб прийому криптовалютних платежів, що може залучити нових клієнтів і підвищити конкурентоспроможність бізнесу.



## Invoice Details

Invoice ID: 123424234234

Total Amount: USD 30000.00

Item 1		
Item Description	Quantity	Price Per Unit
Car	1 pieces	USD 30000
Unit ID / SKU	Amount	
165349621	USD 30000.00	

Payment Link

<https://payments.rocketfuelblockchain.com/hostedPage/68c56d39-6b2c-47e7-a773-665213ae68ef>



Please use your Wallet to Scan



COPY QR CODE

Рисунок 1.5 - Знімок додатку RocketFuel Crypto

Messari – це провідна платформа для аналітики, досліджень і даних про криптовалюту. Вона надає інвесторам, трейдерам і дослідникам детальну інформацію про ринки криптовалют, аналітичні звіти, дослідження та інструменти для управління інвестиціями. Ось детальний огляд Messari та його можливостей.

Основні функції та можливості Messari

1. Дані про ринки криптовалют:

- Ціни та обсяги торгів: Messari надає актуальні дані про ціни та обсяги торгів для тисяч криптовалют з різних бірж по всьому світу.
- Капіталізація ринку: Платформа відображає загальну капіталізацію ринку для кожної криптовалюти, а також загальну капіталізацію ринку криптовалют.

- Графіки цін: Інтерактивні графіки цін дозволяють користувачам аналізувати історичні дані про ціни, обсяги торгів та інші метрики.

## 2. Аналітика та дослідження:

- Аналітичні звіти: Messari публікує аналітичні звіти про ринки криптовалют, тренди, події та прогнози. Ці звіти допомагають інвесторам приймати обґрунтовані рішення.

- Дослідницькі статті: Платформа надає доступ до численних дослідницьких статей про криптовалюту, блокчейн-технології та пов'язані теми.

- Інформація про проекти: Детальна інформація про криптовалютні проекти, включаючи опис технологій, команду, історію фінансування та майбутні плани.

## 3. Інструменти для управління інвестиціями:

- Трекінг портфеля: Messari пропонує інструменти для створення та трекінгу інвестиційних портфелів, що дозволяє користувачам відстежувати продуктивність своїх інвестицій в реальному часі.

- Аналіз портфеля: Інструменти для аналізу продуктивності портфеля, включаючи дохідність, ризики та інші ключові показники.

## 4. Новини та події:

- Агрегація новин: Платформа агрегує новини з різних джерел, забезпечуючи користувачам доступ до актуальної інформації про ринки та події в криптовалютному світі.

- Календар подій: Messari має календар подій, який включає важливі події в криптовалютній індустрії, такі як конференції, оновлення протоколів, релізи продуктів та інші.

## 5. Аналітичні інструменти та індикатори:

- Індикатори та осцилятори: Платформа надає широкий набір технічних індикаторів та осциляторів для технічного аналізу криптовалют.

- Порівняння криптовалют: Інструменти для порівняння різних криптовалют за різними параметрами, такими як ринкова капіталізація, обсяг торгів, продуктивність та інші.

## 6. Підтримка інституційних інвесторів:

- Інституційна аналітика: Messari пропонує інституційним інвесторам розширені аналітичні інструменти та дані для прийняття стратегічних інвестиційних рішень.

- API доступ: Платформа надає API доступ до своїх даних, що дозволяє інституційним клієнтам інтегрувати дані Messari у власні системи.

#### Переваги Messari

- Комплексні дані: Messari надає всебічну інформацію про ринки криптовалют, аналітику, дослідження та новини.

- Аналітичні звіти: Платформа публікує детальні аналітичні звіти, що допомагають інвесторам приймати обґрунтовані рішення.

- Інтерактивні інструменти: Інтерактивні графіки, аналітичні інструменти та індикатори роблять аналіз більш зручним та ефективним.

- Трекінг портфеля: Інструменти для трекінгу та аналізу портфелів допомагають інвесторам управляти своїми інвестиціями.

- Підтримка інституційних клієнтів: Розширені інструменти та API доступ роблять Messari привабливим для інституційних інвесторів.

#### Недоліки та обмеження

- Платні функції: Деякі функції Messari доступні лише у платних версіях, що може бути обмеженням для користувачів, які шукають безкоштовні інструменти.

- Складність для новачків: Широкий спектр функцій та інструментів може бути складним для освоєння новачками, які не мають досвіду в інвестиціях та фінансовому аналізі.

Messari є потужним інструментом для інвесторів, трейдерів і дослідників, які хочуть мати доступ до детальної інформації, аналітики та інструментів для управління своїми інвестиціями в криптовалюті. Завдяки своїм можливостям, він може стати незамінним помічником у прийнятті інвестиційних рішень та моніторингу ринкових тенденцій.

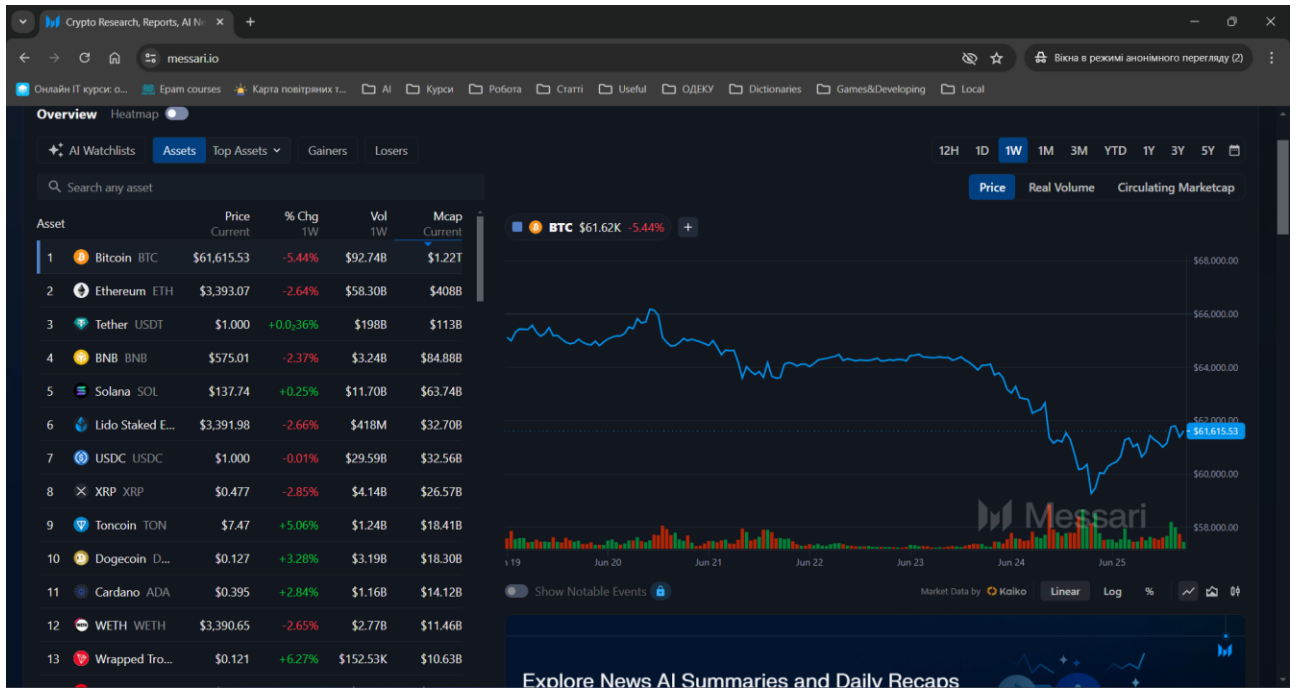


Рисунок 1.6 - Знімок додатку Pricebar

## 2 ВИБІР ТА ОБҐРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ

### 2.1 Функціональні та нефункціональні вимоги

Програма має включати такі важливі функції:

1. Додаток має мати багатосторінкову структуру з чіткою та інтуїтивно зрозумілою навігацією, щоб забезпечити зручну роботу користувача.

2. На головній сторінці має бути помітно відображено N найкращих валют на основі їх популярності на ринку або, як альтернатива, представлено 10 найкращих валют, отриманих із виклику API.

3. Слід включити спеціальну сторінку для надання вичерпної інформації про кожну валюту, включаючи її ціну, обсяг торгів, зміну ціни та доступні ринки для покупки, а також відповідні ціни. Крім того, включення прямих посилань на валютні сторінки на ринках було б вигідним.

4. Користувачам слід надати можливість легко шукати конкретну валюту, використовуючи її назву чи код, щоб отримати зручний доступ до відповідної інформації.

5. Додатку рекомендується дотримуватися архітектурного шаблону Model-View-Controller (MVC), з використанням фреймворків для прискорення розробки.

6. Незважаючи на те, що дизайн інтерфейсу користувача програми може відрізнитися, він повинен підтримувати естетичні стандарти, і заохочується розумне впровадження шаблонів дизайну.

Крім того, вкрай важливо, щоб архітектура та код програми були ретельно структуровані, щоб відповідати найкращим практикам і забезпечувати чистоту. Включення додаткових функцій, таких як відображення діаграм котирувань для валют (наприклад, діаграми японських свічок), сприяння конвертації валют без урахування методу та потенційної комісії, підтримка світлих/темних тем і увімкнення кількох локалізацій, значно покращить функціональність програми та звернення користувачів.

### 2.2 Обґрунтування вибору операційної системи

.Net є кросплатформним фреймворком. Це означає, що додаток легко скомпілювати для будь-якої платформи без розрахунку на інші платформи. Проте компіляція ASP.NET Core[1] проєктів для Linux має кілька переваг, які роблять це вартим вибором для розробників:

1. Економічна вигода: Linux часто є більш економічно вигідним варіантом порівняно з іншими операційними системами, особливо коли мова йде про сервери та хмарні рішення.
2. Підтримка контейнерів: Linux є ідеальною платформою для розгортання додатків у контейнерах, що сприяє легшому масштабуванню, підвищенню безпеки та використанню сучасних інструментів кластеризації, таких як Kubernetes.
3. Продуктивність та надійність: Linux відомий своєю високою продуктивністю та надійністю, особливо в середовищах з високим навантаженням, що робить його привабливим вибором для розгортання веб-додатків.
4. Спільнота та підтримка: Велика спільнота користувачів та розробників Linux забезпечує широкую підтримку та велику кількість ресурсів для вирішення проблем та навчання.
5. Сумісність з хмарними сервісами: Багато хмарних провайдерів пропонують оптимізовані рішення для Linux, що може спростити розгортання та управління ASP.NET Core[2] додатками.
6. Легкість розгортання: Розгортання ASP.NET Core додатків на Linux може бути здійснено за допомогою Nginx, який є потужним та гнучким веб-сервером, що підтримує високу продуктивність та безпеку.

Ці переваги роблять Linux привабливою платформою для розробки та розгортання ASP.NET Core додатків, забезпечуючи розробникам більше можливостей та гнучкості у виборі технологічного стеку.

Проте можливе й розгортання на Windows.

Розгортання ASP.NET Core додатків на Windows має ряд переваг, зокрема:

Інтеграція з Windows середовищем

### 1. Active Directory:

- Аутентифікація та авторизація: Легка інтеграція з Windows Active Directory (AD) дозволяє використовувати корпоративну інфраструктуру безпеки для управління доступом до додатків.

- Групові політики: Використання групових політик для централізованого управління користувачами та правами доступу.

### 2. Сервіси Windows:

- Windows Service: Можливість запуску ASP.NET Core додатків як Windows Service, що забезпечує надійний та автоматизований старт при завантаженні операційної системи.

- Task Scheduler: Використання Task Scheduler для планування та автоматизації завдань, таких як резервне копіювання, оновлення додатків та інші.

### Інтеграція з Microsoft екосистемою

#### 1. Microsoft SQL Server:

- Легка інтеграція: Пряма інтеграція з Microsoft SQL Server для ефективного управління базами даних.

- Реплікація та масштабування: Використання функцій реплікації та масштабування SQL Server для високої доступності та продуктивності.

#### 2. Azure DevOps:

- CI/CD процеси: Інтеграція з Azure DevOps для реалізації безперервної інтеграції та доставки (CI/CD), що спрощує автоматизацію розгортання та тестування.

- Моніторинг та логування: Використання Azure Monitor та Application Insights для моніторингу продуктивності додатків та логування.

### Розширені можливості IIS (Internet Information Services)

#### 1. Продуктивність та масштабованість:

- Продуктивність: IIS забезпечує високу продуктивність завдяки використанню асинхронного оброблення запитів та багатопотоковості.

- Масштабованість: Підтримка автоматичного балансування навантаження та масштабування веб-додатків.

## 2. Безпека:

- SSL/TLS: Легка реалізація SSL/TLS для шифрування трафіку, що забезпечує захист даних між клієнтами та сервером.

- Ізоляція додатків: Можливість ізоляції додатків за допомогою Application Pools, що підвищує безпеку та стабільність.

Зручність у розгортанні та управлінні

### 1. Windows Server:

- Графічний інтерфейс: Інтуїтивно зрозумілий графічний інтерфейс для управління сервером, додатками та службами.

- PowerShell: Використання PowerShell для автоматизації задач розгортання, налаштування та управління.

### 2. Підтримка Legacy систем:

- Зворотна сумісність: Підтримка старих версій .NET Framework та ASP.NET додатків, що спрощує міграцію та інтеграцію нових рішень з існуючими системами.

Переваги для розробників

### 1. Зручне середовище розробки:

- Visual Studio[7]: Використання Visual Studio з розширеними можливостями налагодження, інтеграції з Git, та підтримкою різних мов програмування.

- Visual Studio Code: Легка інтеграція з легким та потужним редактором коду Visual Studio Code[8] для розробки ASP.NET Core додатків.

### 2. Підтримка різних версій .NET:

- .NET Framework та .NET Core: Підтримка одночасного розгортання додатків, створених на основі .NET Framework та .NET Core, що спрощує перехід на нові технології.

Переваги для бізнесу

### 1. Економія коштів:

- Зниження витрат: Використання існуючої інфраструктури Windows Server та SQL Server для розгортання ASP.NET Core додатків, що знижує витрати на навчання та підтримку.

### 2. Стабільність та підтримка:



- Тривала підтримка: Підтримка від Microsoft з регулярними оновленнями безпеки та новими функціями, що забезпечує стабільність та надійність додатків.

Розгортання ASP.NET Core додатків на Windows забезпечує інтеграцію з корпоративними системами, високу продуктивність, безпеку та зручність управління, що робить цю платформу привабливою для бізнесів будь-якого масштабу.

### 2.3 Вибір середовища розробки

Проєкт підтримувався протягом декількох місяців, тому середовище розробки змінювалось декілька разів протягом розробки. Протягом розробки використовувались наступні IDE:

Visual Studio — це інтегроване середовище розробки (IDE), розроблене компанією Microsoft. Воно використовується для створення комп'ютерних програм, включаючи веб-сайти, веб-додатки, веб-сервіси та мобільні додатки. Visual Studio використовує платформи розробки програмного забезпечення від Microsoft, такі як Windows API, Windows Forms, Windows Presentation Foundation (WPF), Windows Store та Microsoft Silverlight. Воно може генерувати як нативний код, так і керований код.

Основні можливості Visual Studio включають:

- Редактор коду з підтримкою IntelliSense: Інтелектуальний аналіз коду та автодоповнення.
- Вбудований відлагоджувач: Працює як на рівні вихідного коду, так і на рівні машинного коду.
- Профілювання коду: Допомогає виявити проблеми продуктивності.
- Дизайнер для GUI-додатків, веб-дизайнер, дизайнер класів та дизайнер схеми бази даних.
- Підтримка версій контролю: Співпраця над кодом з командою.
- Підтримка багатьох мов програмування: Включаючи C++, C#, JavaScript, TypeScript, Python, XML, XSLT, HTML та CSS.

Visual Studio є потужним інструментом для розробки, який дозволяє вам завершити весь цикл розробки в одному місці — від написання коду та відлагодження до збірки та розгортання додатків. Від першої програми "Hello World" до розробки та розгортання додатків — Visual Studio надає розробникам багато можливостей та гнучкості у виборі технологічного стеку.

Visual Studio Code (VS Code) — це легкий, але потужний редактор вихідного коду, розроблений компанією Microsoft. Він доступний для Windows, macOS та Linux. Основні можливості VS Code включають:

- Підтримка багатьох мов програмування: VS Code підтримує різні мови, такі як JavaScript, TypeScript, Node.js, C++, C#, Java, Python, PHP, Go, .NET та інші.
- Інтелектуальне завершення коду: VS Code надає підказки для змінних, методів та імпортованих модулів.
- Відлагодження: Ви можете відлагоджувати код безпосередньо в VS Code за допомогою терміналу.
- Розширення: Велика кількість розширень дозволяє розширити функціональність редактора.
- Теми та налаштування: Ви можете змінювати теми, гарячі клавіші та налаштування за своїм смаком.
- Безкоштовний та відкритий код: VS Code є безкоштовним редактором, який можна використовувати для розробки різних типів додатків.

Загалом, Visual Studio Code є популярним інструментом серед розробників завдяки своїм можливостям, легкості використання та активній спільноті.

Rider — це кросплатформене інтегроване середовище розробки (IDE) від компанії JetBrains, створене спеціально для розробки на .NET та C#. Воно підтримує різноманітні типи проектів, такі як .NET Framework, Mono та .NET Core, а також мови програмування, включаючи C#, VB.NET, F#, ASP.NET, XAML, XML, JavaScript, TypeScript, JSON, HTML, CSS та SQL.

Основні характеристики Rider включають:

- Підтримку різних фреймворків та технологій: ASP.NET, Unity, Xamarin.
- Єдине середовище для розробки: Rider надає інструменти для написання, тестування, відлагодження та розгортання .NET додатків.
- Розширені можливості: Інтеграція з іншими інструментами JetBrains, такими як ReSharper, що забезпечує додаткові можливості для аналізу коду та рефакторингу.

Rider є потужним інструментом для розробників .NET, який дозволяє ефективно працювати з кодом на різних платформах, включаючи Windows, Mac та Linux.

SonarLint — це розширення для інтегрованих середовищ розробки (IDE), яке допомагає розробникам виявляти та виправляти проблеми в коді в реальному часі. Він працює подібно до правописного перевіряльника, підкреслюючи проблеми в коді та надаючи рекомендації щодо їх виправлення. Основні можливості SonarLint:

- Підтримка багатьох мов програмування: SonarLint аналізує код на багатьох мовах, включаючи C#, Java, JavaScript, Python та інші.
- Реальний час: Він надає зворотний зв'язок під час написання коду, допомагаючи виявляти проблеми на льоту.
- Єдина конфігурація: Правила та налаштування синхронізуються з SonarQube або SonarCloud, забезпечуючи єдиний стандарт чистого коду для всієї команди.

SonarQube — це самостійний інструмент для автоматичного перевірки якості коду. Він інтегрується в робочий процес розробки та виявляє проблеми в коді, допомагаючи забезпечити постійний контроль якості. Основні можливості SonarQube:

- Аналіз більш як 30 мов програмування: SonarQube перевіряє якість коду на різних мовах, включаючи Java, C#, JavaScript, Python та інші.
- Інтеграція з CI/CD: Він використовується в CI/CD-пайплайнах для автоматичної перевірки коду перед релізом.
- Синхронізація налаштувань: Правила та налаштування синхронізуються з SonarLint[10], забезпечуючи єдиний стандарт чистого коду.

SonarCloud — це хмарний сервіс для аналізу коду, який виявляє проблеми в кодї на багатьох мовах програмування. Він інтегрується з платформами CI/CD та допомагає забезпечити постійний контроль якості коду.

Основні можливості SonarCloud схожі на можливості SonarQube.

OmniSharp[11] — це проект з відкритим вихідним кодом, спрямований на покращення досвіду[12] розробки для розробників, що працюють з .NET. OmniSharp надає різноманітні інструменти та сервіси для інтеграції з текстовими редакторами та IDE, зокрема Visual Studio Code, Sublime Text, Emacs, Vim та іншими.

Основні функції OmniSharp включають:

1. Інтелектуальні підказки (IntelliSense) — автозавершення коду, що допомагає швидко писати код.
2. Навігація по коду — можливість швидко переходити до визначень, оголошень та посилань на символи в кодї.
3. Аналіз коду — виявлення помилок та пропозиції щодо їх виправлення в реальному часі.
4. Рефакторинг коду — інструменти для автоматичного перетворення коду, такі як перейменування змінних, виділення методів тощо.
5. Підтримка запуску та налагодження — інтеграція з дебаггером для запуску та налагодження .NET додатків.

OmniSharp розробляється та підтримується спільнотою розробників та широко використовується серед професіоналів, що працюють з технологіями .NET.

## 3 ТЕХНОЛОГІЯ ІНФОРМАЦІЙНИХ ВЕБ-САЙТІВ

### 3.1 Поняття технології інформаційних веб-сайтів

Веб-сайти стали невід'ємною частиною нашого повсякденного життя, надаючи інформацію, розваги та послуги користувачам по всьому світу. Інформаційні веб-сайти займають особливе місце серед них, оскільки вони надають користувачам доступ до актуальної та корисної інформації в різних сферах діяльності, включаючи новини, освітні ресурси, аналітичні дані та багато іншого. Технологія створення інформаційних веб-сайтів постійно розвивається, забезпечуючи більш високу швидкість доступу, зручність використання та безпеку.

#### Основні компоненти інформаційних веб-сайтів

##### 1. Серверна частина (бекенд):

- Веб-сервери: Apache, Nginx, IIS.
- Мови програмування: PHP, Python, Ruby, Java, C#, Node.js.
- Фреймворки: Django, Ruby on Rails, Spring, ASP.NET Core.
- Бази даних: MySQL, PostgreSQL, MongoDB, SQLite.
- API: RESTful API, GraphQL для взаємодії між сервером та клієнтом.

##### 2. Клієнтська частина (фронтенд):

- Мови програмування: HTML, CSS, JavaScript.
- Фреймворки та бібліотеки: Angular[2], React, Vue.js.
- UI/UX дизайн: Адаптивний дизайн, інтерфейс користувача, досвід користувача.

##### 3. Засоби управління контентом (CMS):

- Популярні CMS: WordPress, Joomla, Drupal.
- Функціональність CMS: Управління статтями, медіафайлами, користувачами та коментарями.

##### 4. Інтеграція з зовнішніми сервісами:

- API зовнішніх сервісів: соціальні мережі, платіжні системи, аналітичні сервіси.

- Реклама та аналітика: Google Analytics, Facebook Pixel, рекламні мережі.

## Основні етапи розробки інформаційного веб-сайту

### 1. Проектування:

- Визначення цілей та завдань сайту: Які завдання має вирішувати веб-сайт? Яка його цільова аудиторія?

- Створення технічного завдання: Детальний опис функціональних вимог до сайту, включаючи дизайн, функціонал, інтеграції та безпеку.

### 2. Дизайн та розробка прототипу:

- Wireframes та mockups: Створення ескізів сторінок для візуалізації структури та взаємодії елементів.

- UI/UX дизайн: Створення інтерфейсу, що забезпечує зручність та ефективність використання сайту.

### 3. Розробка:

- Фронтенд-розробка: Верстка сторінок, реалізація клієнтської логіки.

- Бекенд-розробка: Реалізація серверної логіки, налаштування бази даних, створення API.

### 4. Тестування та налагодження:

- Функціональне тестування: Перевірка працездатності всіх функцій сайту.

- Тестування безпеки: Перевірка наявності вразливостей, тестування захисту від атак.

- Тестування продуктивності: Перевірка швидкості завантаження та обробки запитів.

### 5. Розгортання та підтримка:

- Розгортання на сервері: Налаштування серверного оточення, завантаження файлів сайту.

- Моніторинг та підтримка: Регулярне оновлення контенту, виправлення помилок, оновлення безпеки.

## Інноваційні технології та тренди

1. Прогресивні веб-додатки (PWA): Поєднують найкращі властивості веб-сайтів та мобільних додатків, забезпечуючи швидкий доступ та автономну роботу.

2. Веб-складальники (Web Assembly): Дозволяють виконувати високопродуктивний код безпосередньо в браузері.

3. Інтелектуальні системи рекомендацій: Використання машинного навчання для надання персоналізованого контенту користувачам.

4. Голосові інтерфейси: Інтеграція з голосовими помічниками та технологіями розпізнавання мови.

Технологія інформаційних веб-сайтів постійно еволюціонує, адаптуючись до нових вимог користувачів та розвитку технологій. Важливість якісного веб-сайту для надання актуальної та корисної інформації важко переоцінити, адже він забезпечує ефективну комунікацію між ресурсами та їх аудиторією. Інформаційні веб-сайти стають все більш інтерактивними, адаптивними та інтелектуальними, забезпечуючи користувачам зручний доступ до інформації будь-де та будь-коли.

### 3.2 Сфери використання технології інформаційних веб-сайтів

Інформаційні веб-сайти відіграють ключову роль у наданні користувачам доступу до різноманітних даних і послуг. Завдяки постійному розвитку технологій створення веб-сайтів, можливості їх використання значно розширилися. Від новинних порталів до освітніх платформ, інформаційні веб-сайти стали невід'ємною частиною різних сфер життя та діяльності.

#### Основні сфери використання технології інформаційних веб-сайтів

##### 1. Новинні портали

- Приклад: BBC, CNN, The New York Times.
- Функціональність: Оперативне оновлення новин, мультимедійний контент, інтерактивні елементи (опитування, коментарі).
- Вплив: Швидке інформування громадськості, висвітлення важливих подій у реальному часі.

##### 2. Освітні платформи

- Приклад: Coursera, Khan Academy, EdX.

- Функціональність: Онлайн-курси, вебінари, інтерактивні вправи, системи управління навчанням.
  - Вплив: Забезпечення доступу до освіти для широкої аудиторії, підтримка безперервного навчання.
3. Аналітичні та фінансові сервіси
- Приклад: Bloomberg, Yahoo Finance, CoinGecko.
  - Функціональність: Аналітичні інструменти, графіки, показники ринків, інвестиційні поради.
  - Вплив: Підтримка прийняття обґрунтованих фінансових рішень, аналіз ринкових тенденцій.
4. Медичні інформаційні сайти
- Приклад: WebMD, Mayo Clinic, Healthline.
  - Функціональність: Статті про захворювання, симптоми, лікування, бази даних лікарів та медичних установ.
  - Вплив: Підвищення обізнаності населення щодо здоров'я, підтримка самодіагностики та вибору лікування.
5. Соціальні мережі та платформи обміну контентом
- Приклад: Facebook, Twitter, Instagram.
  - Функціональність: Створення та обмін контентом, комунікація, групи за інтересами.
  - Вплив: Залучення користувачів до соціальних взаємодій, поширення інформації.
6. Електронна комерція
- Приклад: Amazon, eBay, Alibaba.
  - Функціональність: Каталоги товарів, системи замовлень, платежі онлайн, відгуки користувачів.
  - Вплив: Забезпечення доступу до товарів і послуг для глобальної аудиторії, підтримка малого та середнього бізнесу.
7. Громадські та урядові сайти
- Приклад: Gov.uk, USA.gov, Дія (Україна).



- Функціональність: Інформація про державні послуги, електронні сервіси, новини уряду.

- Вплив: Підвищення прозорості уряду, зручність доступу до державних послуг.

## 8. Розважальні ресурси

- Приклад: Netflix, YouTube, Spotify.

- Функціональність: Стрімінгові сервіси, бібліотеки медіа-контенту, рекомендаційні системи.

- Вплив: Забезпечення доступу до розважального контенту, підтримка творчих індустрій.

### Виклики та перспективи

#### 1. Безпека та конфіденційність

- Виклики: Захист персональних даних користувачів, запобігання кіберзлочинам.

- Рішення: Впровадження сучасних технологій безпеки, шифрування даних, постійний моніторинг систем.

#### 2. Оптимізація продуктивності

- Виклики: Швидке завантаження сторінок, обробка великих обсягів даних.

- Рішення: Використання CDN, оптимізація коду, застосування кешування.

#### 3. Адаптивний дизайн

- Виклики: Забезпечення зручності користування на різних пристроях (мобільних, планшетах, десктопах).

- Рішення: Використання адаптивного та респонсивного дизайну, тестування на різних платформах.

#### 4. Інтеграція з новими технологіями

- Виклики: Впровадження AI, машинного навчання, блокчейн-технологій.

- Рішення: Інвестування в дослідження та розвиток, навчання персоналу новим технологіям.

Технологія інформаційних веб-сайтів продовжує розвиватися, відкриваючи нові можливості для різних сфер діяльності. Від новин та освіти до медицини та електронної комерції, ці веб-сайти стали ключовими

інструментами для надання актуальної інформації та послуг. Зважаючи на швидкий розвиток технологій, інформаційні веб-сайти будуть і надалі адаптуватися до нових вимог і потреб користувачів, забезпечуючи ефективну та безпечну комунікацію в цифровому світі.

### 3.3 Основні принципи роботи технології доповненої реальності

Інформаційні веб-сайти надають користувачам доступ до важливої та актуальної інформації у різних сферах. Щоб ці сайти працювали ефективно та надійно, необхідно дотримуватися певних принципів їхньої розробки та функціонування. Ці принципи стосуються як серверної, так і клієнтської частин, а також взаємодії між ними.

#### Принципи роботи серверної частини

##### 1. Модульність та мікросервісна архітектура

- Пояснення: Розділення функціональності сайту на окремі модулі або сервіси, що можуть розроблятися, розгортатися та масштабуватися незалежно один від одного.

- Приклад: Окремі мікросервіси для аутентифікації, управління контентом, аналітики.

##### 2. Безпека

- Пояснення: Захист даних користувачів та серверної інфраструктури від несанкціонованого доступу та атак.

- Приклад: Використання HTTPS, захист від SQL-ін'єкцій, контроль доступу за допомогою ролей і прав.

##### 3. Оптимізація продуктивності

- Пояснення: Забезпечення швидкої обробки запитів і мінімальних затримок у роботі сайту.

- Приклад: Використання кешування, оптимізація запитів до бази даних, балансування навантаження.

##### 4. Надійність і стійкість

- Пояснення: Забезпечення безперебійної роботи сайту та його стійкості до збоїв.

- Приклад: Використання резервного копіювання даних, налаштування механізмів відновлення після збоїв, моніторинг роботи серверів.

### Принципи роботи клієнтської частини

#### 1. Адаптивний дизайн

- Пояснення: Забезпечення зручного перегляду та взаємодії з сайтом на різних пристроях (мобільних телефонах, планшетах, настільних комп'ютерах).

- Приклад: Використання CSS-медіазапитів, респонсивних фреймворків, таких як Bootstrap.

#### 2. Інтерактивність і зручність використання

- Пояснення: Надання користувачам інтерактивних елементів і зручного інтерфейсу для легкої навігації та отримання інформації.

- Приклад: Використання JavaScript та фреймворків, таких як Angular[4], React або Vue.js, для створення динамічних інтерфейсів.

#### 3. Швидкість завантаження

- Пояснення: Мінімізація часу завантаження сторінок для покращення користувацького досвіду.

- Приклад: Оптимізація зображень, використання кешування на стороні клієнта, зменшення кількості HTTP-запитів.

#### 4. Доступність

- Пояснення: Забезпечення можливості використання сайту людьми з різними фізичними можливостями.

- Приклад: Використання ARIA-атрибутів, забезпечення належного контрасту кольорів, підтримка навігації з клавіатури.

### Принципи взаємодії між клієнтом і сервером

#### 1. RESTful API

- Пояснення: Використання REST API для обміну даними між клієнтською та серверною частинами.

- Приклад: Надсилання запитів GET для отримання даних, POST для створення нових записів, PUT для оновлення та DELETE для видалення даних.

#### 2. Безпека передачі даних

- Пояснення: Забезпечення захищеної передачі даних між клієнтом і сервером.

- Приклад: Використання HTTPS для шифрування даних, захист від CSRF-атак.

### 3. Асинхронна обробка запитів

- Пояснення: Використання асинхронних запитів для забезпечення плавного користувацького досвіду без перезавантаження сторінки.

- Приклад: Використання технологій AJAX, Fetch API, WebSockets.

### 4. Кешування

- Пояснення: Використання механізмів кешування для зменшення навантаження на сервер і прискорення завантаження даних.

- Приклад: Кешування результатів запитів до бази даних, використання HTTP-заголовків для контролю кешування на стороні клієнта.

## Виклики та перспективи

### 1. Підтримка нових технологій

- Виклики: Постійний розвиток веб-технологій вимагає регулярного оновлення знань і навичок розробників.

- Рішення: Інвестування в навчання та професійний розвиток команди, впровадження нових технологій та інструментів.

### 2. Забезпечення масштабованості

- Виклики: Збільшення кількості користувачів і обсягу даних вимагає відповідного масштабування інфраструктури.

- Рішення: Використання хмарних сервісів, мікросервісної архітектури, контейнеризації.

### 3. Підвищення безпеки

- Виклики: Зростання кількості кіберзагроз вимагає постійного вдосконалення заходів безпеки.

- Рішення: Регулярне проведення аудиту безпеки, впровадження нових методів захисту, освіта користувачів щодо безпечної поведінки в Інтернеті.

Основні принципи роботи технології інформаційного веб-сайту забезпечують його ефективність, надійність та зручність для користувачів. Від

модульності та безпеки серверної частини до адаптивного дизайну та швидкості завантаження клієнтської частини – всі ці аспекти сприяють створенню якісного інформаційного ресурсу. Дотримання цих принципів дозволяє розробникам створювати веб-сайти, що відповідають сучасним вимогам та очікуванням користувачів, забезпечуючи безперебійний доступ до важливої інформації та послуг.

### 3.4 Інструменти для розробки програмного забезпечення

Для реалізації програми використовувались наступні технології:

- ASP.NET Core
- EntityFramework Core
- ADO.NET (at old sql class version)
- MSSQL\PostgresSQL (you can switch for you need)
- C# Language
- .Net Core 8
- CoinCap API
- CoinGecko API
- RestSharp
- Newtonsoft.JSON
- Git
- Docker
- XUnit
- Azure
- Angular

#### ASP.NET Core

Опис: ASP.NET Core – це високопродуктивний, кросплатформний фреймворк для створення сучасних веб-додатків та сервісів, розроблений компанією Microsoft. Він підтримує роботу на різних операційних системах, таких як Windows, macOS та Linux, що робить його надзвичайно гнучким.

Основні особливості:

- Кросплатформність: Можливість розгортання додатків на різних операційних системах.
- Висока продуктивність: Оптимізоване ядро забезпечує швидку обробку запитів.
- Відкритий код: Проект з відкритим вихідним кодом, що дозволяє спільноті робити внески та поліпшення.
- Інверсія управління (Dependency Injection): Підтримка DI з коробки, що спрощує управління залежностями.
- Розширюваність: Легка інтеграція з іншими технологіями та бібліотеками.

### EntityFramework Core

Опис: EntityFramework Core (EF Core) – це сучасний, легкий об'єктно-реляційний механізм відображення (ORM) для .NET. Він дозволяє розробникам працювати з базою даних, використовуючи об'єкти .NET, замість написання SQL-запитів.

Основні особливості:

- Підтримка різних СУБД: Підтримує різні системи управління базами даних, такі як SQL Server, PostgreSQL, MySQL, SQLite.
- Міграції: Зручний механізм для управління версіями бази даних.
- LINQ: Можливість використання LINQ (Language Integrated Query) для написання запитів до бази даних.
- Висока продуктивність: Оптимізований для швидкої роботи з великими обсягами даних.

### ADO.NET

Опис: ADO.NET – це набір компонентів для роботи з даними в .NET, що забезпечує доступ до різноманітних джерел даних. Використовується для взаємодії з базами даних через запити SQL.

Основні особливості:

- Висока продуктивність: Прямий доступ до бази даних забезпечує швидке виконання запитів.

- Гнучкість: Можливість роботи з різними джерелами даних (SQL Server, Oracle, ODBC).

- Низькорівневий доступ: Більший контроль над процесом взаємодії з базою даних порівняно з ORM.

MSSQL/PostgreSQL

MSSQL (Microsoft SQL Server):

Опис: MSSQL – це реляційна система управління базами даних (СУБД) від Microsoft. Вона надає потужні засоби для зберігання, управління та аналізу даних, підтримує транзакції та високий рівень безпеки.

Основні особливості:

- Висока продуктивність: Оптимізована для обробки великих обсягів даних.

- Інструменти адміністрування: Інтуїтивні інструменти для управління базою даних.

- Безпека: Розширені механізми безпеки та контролю доступу.

PostgreSQL:

Опис: PostgreSQL – це потужна, відкрита та безкоштовна СУБД, відома своєю стабільністю, масштабованістю та багатими можливостями для розширення. Підтримує різноманітні типи даних і розширені функціональні можливості.

Основні особливості:

- Масштабованість: Підтримка великих баз даних і високих навантажень.

- Розширюваність: Можливість додавання власних типів даних, функцій, операторів.

- Відкритий код: Активна спільнота розробників і регулярні оновлення.

C# Language

Опис: C# – це сучасна об'єктно-орієнтована мова програмування, розроблена компанією Microsoft. Вона широко використовується для створення різноманітних додатків на платформі .NET.

Основні особливості:

- Об'єктно-орієнтованість: Підтримка основних принципів ООП (інкапсуляція, спадкування, поліморфізм).
- Сучасні можливості: Підтримка асинхронного програмування, лямбда-виразів, LINQ.
- Безпечність типів: Строгий контроль типів, що допомагає уникати багатьох помилок.

#### .NET Core 8

Опис: .NET Core 8 – це кросплатформний фреймворк для розробки різноманітних додатків, включаючи веб, мобільні, десктопні та хмарні рішення. Він забезпечує високу продуктивність, безпеку та масштабованість.

Основні особливості:

- Кросплатформність: Підтримка різних операційних систем.
- Висока продуктивність: Оптимізація роботи додатків.
- Розширюваність: Легка інтеграція з іншими технологіями та бібліотеками.

#### CoinCap API

Опис: CoinCap API – це програмний інтерфейс, що надає доступ до ринкових даних криптовалют у реальному часі. Він дозволяє отримувати інформацію про ціни, обсяги торгів, ринкові капіталізації та інші показники криптовалют.

Основні особливості:

- Дані в реальному часі: Оновлення цін і обсягів торгів у режимі реального часу.
- Різноманітні кінцеві точки: Доступ до різних типів даних про криптовалюту.
- Легка інтеграція: Простота використання у веб-додатках.

#### CoinGecko API

Опис: CoinGecko API – це API, що надає доступ до великої кількості даних про криптовалюту, таких як ціни, обсяги торгів, історичні дані, ринкові капіталізації та інші показники.

Основні особливості:



- Широкий охоплення криптовалют: Дані про велику кількість криптовалют.
- Розширені метрики: Доступ до аналітичних показників і історичних даних.
- Регулярне оновлення: Свіжі дані для точного аналізу.

### RestSharp

Опис: RestSharp – це популярна бібліотека для роботи з HTTP запитами в C#. Вона значно спрощує інтеграцію з RESTful сервісами та API.

Основні особливості:

- Простота використання: Інтуїтивно зрозумілий інтерфейс для створення HTTP запитів.
- Підтримка асинхронних операцій: Можливість виконання запитів без блокування основного потоку.
- Гнучкість: Легка обробка відповідей у різних форматах (JSON, XML).

### Newtonsoft.JSON

Опис: Newtonsoft.JSON (або JSON.NET) – це потужна бібліотека для роботи з JSON у C#. Вона забезпечує зручні засоби для серіалізації та десеріалізації даних у форматі JSON.

Основні особливості:

- Висока продуктивність: Швидке перетворення даних між JSON і об'єктами C#[16].
- Гнучкість налаштувань: Широкі можливості налаштування серіалізації та десеріалізації.
- Підтримка складних структур: Можливість обробки складних JSON структур і типів даних.

### Git

Опис: Git – це розподілена система контролю версій, яка використовується для управління змінами у вихідному коді проекту. Вона дозволяє розробникам зберігати історію змін, працювати в команді та зберігати різні версії проекту.

Основні особливості:

- Розподіленість: Кожен розробник має повну копію історії проекту.
- Підтримка гілок: Можливість створення та злиття гілок для паралельної роботи над різними функціональностями.
- Висока продуктивність: Швидке виконання основних операцій (коміти, злиття).

### Docker

Опис: Docker – це платформа для автоматизації розгортання додатків у контейнерах. Вона дозволяє пакувати додаток разом з усіма його залежностями в єдиний контейнер, який можна легко переносити між різними середовищами.

Основні особливості:

- Контейнеризація: Пакування додатків у ізольовані контейнери для забезпечення консистентності середовища.
- Масштабованість: Легка масштабованість додатків за рахунок контейнерів.
- Інтеграція з хмарними сервісами: Підтримка різних хмарних платформ для розгортання контейнерів.

### XUnit

Опис: XUnit – це сучасний тестовий фреймворк для .NET, який використовується для написання автоматизованих тестів. Він забезпечує підтримку юніт-тестів, інтеграційних тестів та інших видів тестування.

Основні особливості:

- Легка інтеграція: Простота інтеграції з .NET проектами та CI/CD пайплайнами.
- Асинхронні тести: Підтримка асинхронного тестування для сучасних додатків.
- Розширюваність: Можливість розширення фреймворку через створення власних атрибутів та налаштувань.

### Azure

Опис: Microsoft Azure – це хмарна платформа, яка надає широкий спектр сервісів для розробки, розгортання та управління додатками через

глобальну мережу дата-центрів. Azure підтримує різноманітні хмарні сервіси, включаючи обчислювальні ресурси, бази даних, аналітику та машинне навчання.

Основні особливості:

- Широкий вибір хмарних сервісів: Різноманітні інструменти для розробки та управління додатками.
- Глобальна доступність: Дата-центри по всьому світу забезпечують високу доступність і надійність.
- Інтеграція з DevOps: Підтримка CI/CD пайплайнів для автоматизації процесів розробки та розгортання.

Angular

Опис: Angular – це потужний фреймворк для створення динамічних веб-додатків, розроблений Google. Він забезпечує широкий набір інструментів для розробки клієнтської частини додатків з використанням мови TypeScript.

Основні особливості:

- Модульність: Структурований підхід до організації коду через модулі.
- Двостороннє зв'язування даних: Спрощує синхронізацію даних між моделлю та представленням.
- Компонентний підхід: Використання компонентів для побудови багаторазово використовуваних елементів інтерфейсу.
- Розширюваність: Можливість інтеграції з іншими бібліотеками та інструментами.

Ці інструменти забезпечують ефективний процес розробки, тестування, розгортання та підтримки інформаційного веб-сайту для криптовалют, роблячи його надійним, продуктивним та зручним для користувачів

## 4 ПРОЕКТУВАННЯ ВЕБ-САЙТУ

Після проведення аналізу предметної області, програмних продуктів-аналогів та вибору засобів розробки, необхідним етапом є проектування інформаційної системи. На цьому етапі необхідно створити UML-діаграми варіантів використання (прецедентів), активності і послідовності та розробити макети інтерфейсу застосунку.

### 4.1 Створення UML-діаграми варіантів використання системи

Діаграма варіантів використання (прецедентів) відображає функціональне призначення проектованої програмної системи. Суть діаграми прецедентів полягає в тому, що систему представляють як групу акторів, які за допомогою варіантів використання взаємодіють із нею.

Актор – це сутність, що взаємодіє з системою для вирішення деяких завдань. Актором може бути людина, інша система, пристрій або програмний засіб. В даному випадку акторами визначено користувача застосунку та веб-сайт. На діаграмі також зображена база даних.

Після визначення акторів системи, необхідно сформулювати перелік усіх варіантів використання, з якими будуть взаємодіяти визначені актори. Серед основних варіантів використання системи: з боку користувача переглядати криптовалюту та конкретну криптовалюту, з боку веб-сайту – пошук криптовалюту, пошук та додавання нових маркетів до бази данх.

За результатами сформованих варіантів використання та акторів системи було розроблено діаграму варіантів використання, яку наведено на рис.

4.1

### 4.2 Створення діаграми діяльності (активностей)

Для моделювання процесу виконання операцій в мові UML використовуються діаграми діяльності. На діаграмі діяльності відображається логіка або послідовність переходу від однієї діяльності до іншої, при цьому увагу

фіксується на результаті діяльності. Сам же результат може привести до зміни стану системи або повернення деякого значення. Для системи була створена діаграма діяльності, яка відображає основний процес проекрованої системи – збирання маркетів із сайту CoinGecko (рис. 4.2).

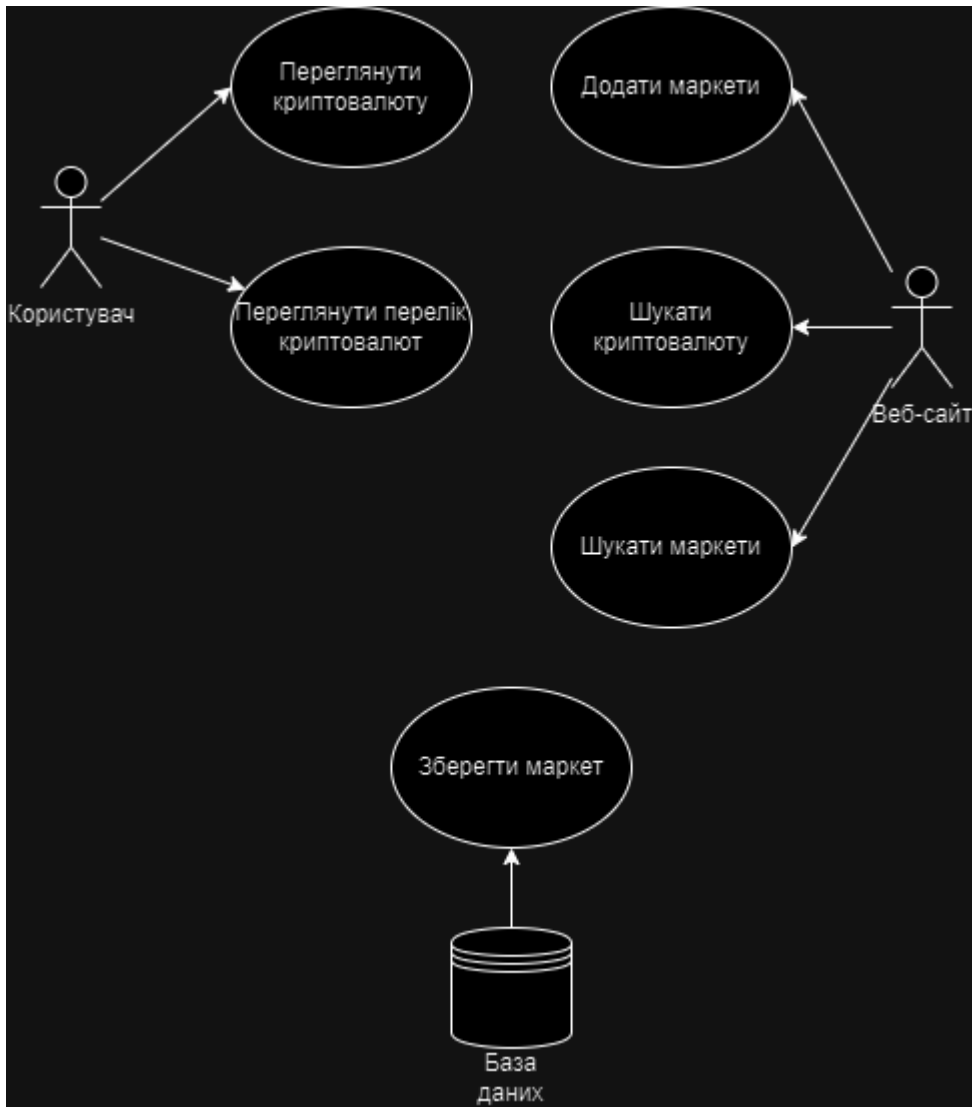


Рисунок 4.1 – Діаграма варіантів використання (прецедентів) систем

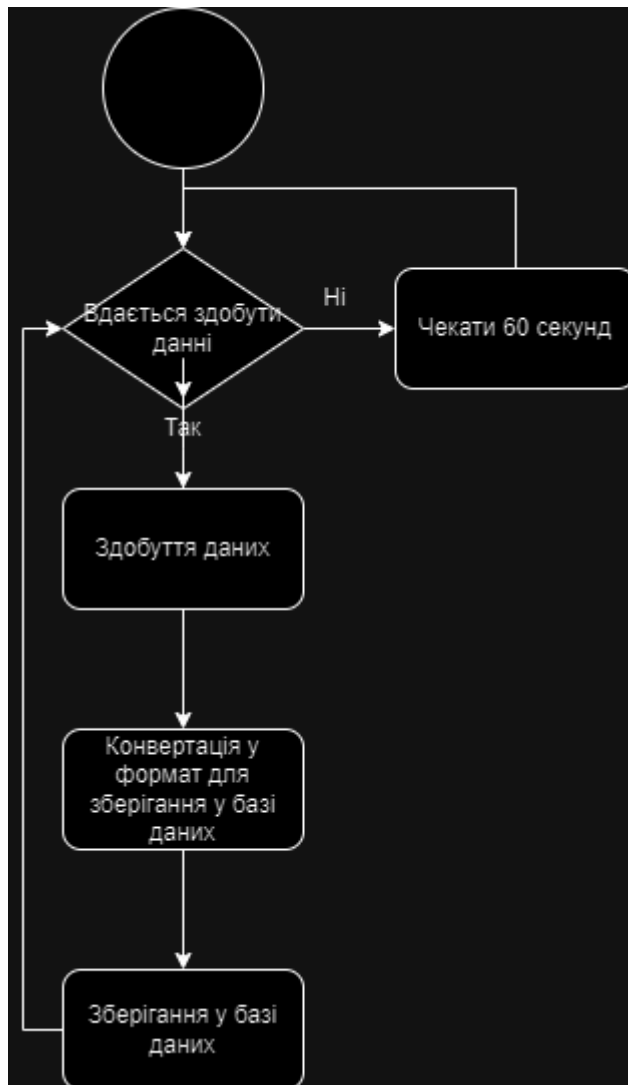


Рисунок 4.2 – Діаграма діяльності (активностей)

### 4.3 Створення діаграми послідовності

Також була створена діаграма послідовностей. Такі діаграми використовуються для уточнення діаграм прецедентів і більш детального опису логіки сценаріїв використання.

Головна послідовність сценарію розпізнавання зображення складається з взаємодій, які відображені на діаграмі на рис. 4.3: користувач створює та відправляє запит, отриманий запит обробляється API, яка знаходить збіги у

хмарній базі даних та повертає дані у застосунок, де ця інформація відображається для користувача.

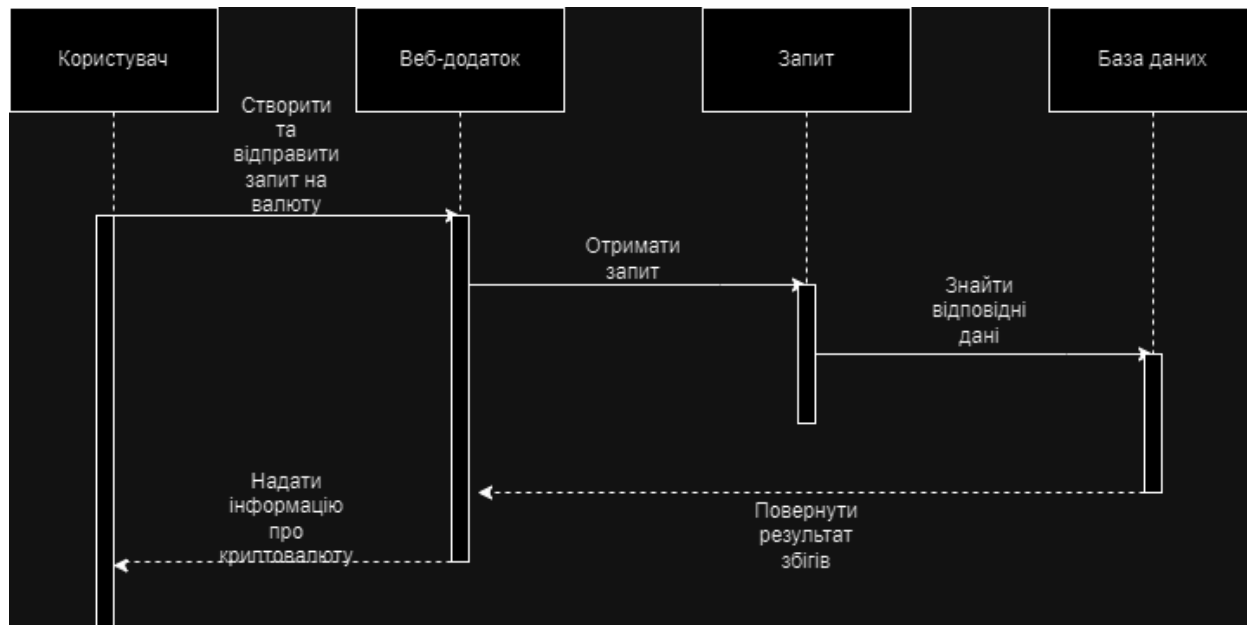


Рисунок 4.3 – Діаграма послідовностей

#### 4.4 Проектування макетів інтерфейсу застосунку

Після цього були створені інтерфейс за допомогою Scaffolding. Було вирішено розробити 3 сторінки у застосунку та зробити дизайн легким та простим. Прототипи інтерфейсу наведені на рис. 4.4, 4.5, 4.6

На головній сторінці було вирішено розмістити вітання користувача та топ 10 криптовалют.

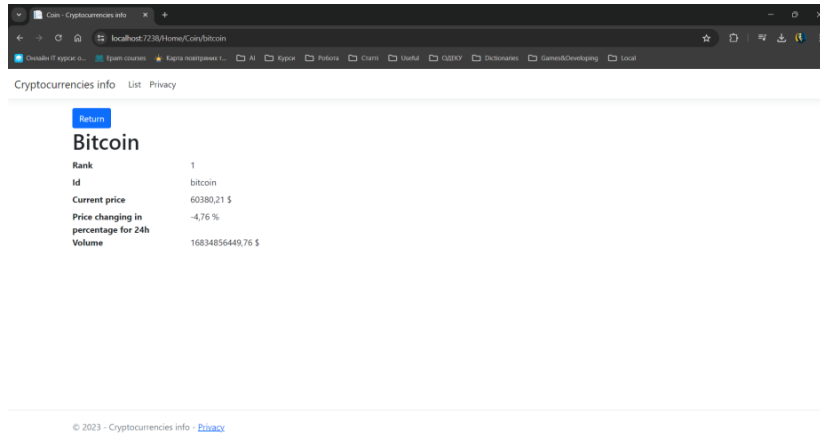


Рисунок 4.4 – Макети інтерфейсу застосунку



Cryptocurrencies info List Privacy

## Welcome

Rank	Name	Current price	Price changing in percentage for 24h	Volume
1	Bitcoin	60365,0972531622463102 \$	-4,8169915195599849 %	16824153385,5268352099771942 \$
2	Ethereum	3352,3248419124348669 \$	-2,0671163001126903 %	12479368266,3585136454608892 \$
3	Tether	1,0002671258319951 \$	0,0081821745426864 %	24468103318,8672050394679004 \$
4	BNB	567,5968239262921024 \$	-1,7015061730733994 %	450891066,9767046406980073 \$
5	Solana	132,2338691729014573 \$	2,8403781489644006 %	1316589859,6585663510365065 \$
6	USDC	1,0013192026284409 \$	0,0742790716491659 %	2645655995,0163995082095795 \$
7	XRP	0,4744396021814691 \$	-1,1040366355274024 %	578619412,0379115120403970 \$
8	Dogecoin	0,1190978358479211 \$	-2,6071588116104757 %	401376492,7642650359814156 \$
9	Cardano	0,3779615005519652 \$	-0,6052855946074976 %	160677174,5952392292970127 \$

Рисунок 4.5 – Макети інтерфейсу застосун

Cryptocurrencies info List Privacy

## Cryptocurrencies list

Search a coin  Find

Rank	Name	Current price	Price changing in percentage for 24h	Volume
1	Bitcoin	60380,9671318855962318 \$	-4,8169915195599849 %	16836426839,1295932237232733 \$
2	Ethereum	3352,2585654364477849 \$	-2,0671163001126903 %	12481459851,6594620045769427 \$
3	Tether	1,0003748081287492 \$	0,0081821745426864 %	2446821700,0863159182756321 \$
4	BNB	567,5909598908089290 \$	-1,7015061730733994 %	450976628,1366076380677285 \$
5	Solana	132,1504241299073174 \$	2,8403781489644006 %	1316728205,8566824667885817 \$
6	USDC	1,0013624568356256 \$	0,0742790716491659 %	2645919748,2937353285372411 \$
7	XRP	0,4745445415668316 \$	-1,1040366355274024 %	578624500,7511056846734099 \$
8	Dogecoin	0,1191207850419468 \$	-2,6071588116104757 %	401501153,4284587164629012 \$
9	Cardano	0,3780317388345054 \$	-0,6052855946074976 %	160665878,0902845867096622 \$

Рисунок 4.6 – Макети інтерфейсу застосунку

При натискання на відповідну валюту відображається коротка інформація про неї – назва, ID, поточна ціна, інтенсивність змін за останні 24 години та обсяг. Також сторінка валюти має таблиці з маркетами, що містять ID, валюти, з котрої та на котру міняються, ціна, статус довіри та посилання та обмін.

Таким чином, на етапі проектування системи були створені UML-діаграми варіантів використання (прецедентів), активності і послідовності та розроблені макети інтерфейсу застосунку. Це допомогло зрозуміти основні варіанти використання системи та логіку роботи застосунку

## 5 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОГО ВЕБ-САЙТУ

### 5.1 Робота з ASP.NET Core

Для роботи з ASP.NET Core потрібен набір інструментів та програмного забезпечення, що забезпечує ефективний процес розробки, тестування та розгортання додатків. По-перше, потрібна операційна система. ASP.NET Core підтримує кросплатформенність, тому ви можете розробляти та розгортати додатки на Windows, macOS і Linux. Зазвичай розробники використовують Windows 10 або новіші версії, останні версії macOS, або більшість дистрибутивів Linux, таких як Ubuntu, Debian, Fedora тощо.

Для розробки потрібно встановити .NET SDK, останню версію якого можна завантажити з офіційного сайту .NET. Також необхідно мати інтегроване середовище розробки (IDE). Найпопулярнішими є Visual Studio (для Windows і macOS), яке підтримує всі аспекти розробки на .NET, Visual Studio Code[8] (кросплатформене), яке є легким редактором коду з підтримкою розширень для .NET Core, та JetBrains Rider – кросплатформене IDE з підтримкою .NET Core та інтеграцією з різними інструментами розробки.

Для роботи з командним рядком можна використовувати Terminal або Command Prompt для виконання команд .NET CLI. Також підійдуть PowerShell (для Windows) або Bash (для Linux і macOS). Додатково необхідні такі інструменти, як Node.js для роботи з сучасними фронтенд фреймворками, такими як Angular, React або Vue.js. Завантажити Node.js можна з офіційного сайту. Пакетні менеджери npm або Yarn допоможуть в управлінні залежностями JavaScript. Docker потрібен для контейнеризації додатків і його можна завантажити з офіційного сайту Docker.

Для роботи з базами даних можна використовувати SQL Server, PostgreSQL або SQLite. Entity Framework Core є ORM для роботи з базами даних. Пакетний менеджер NuGet є основним для .NET і використовується для управління бібліотеками та інструментами .NET. NPM (Node Package Manager) використовується для управління фронтенд залежностями.

Git є системою контролю версій для управління вихідним кодом і його можна інсталиувати з офіційного сайту Git. Для хмарних сервісів і CI/CD можна використовувати Microsoft Azure для розгортання, управління та моніторингу додатків. GitHub Actions підходить для налаштування безперервної інтеграції та розгортання (CI/CD), а Azure DevOps для управління проектами, CI/CD пайплайнів та розгортання додатків.

Для тестування можна використовувати Xunit, який є фреймворком для написання автоматизованих тестів, та Moq – бібліотеку для створення моків при юніт-тестуванні.

Приклад набору інструментів та середовищ для роботи може виглядати наступним чином. Спочатку потрібно завантажити .NET SDK з офіційного сайту та встановити Visual Studio з підтримкою ASP.NET Core. Потім, за допомогою командного рядка, можна створити новий проект командою ``dotnet new mvc -n Cryptocurrencies-info``. Наступним кроком є налаштування бази даних з Entity Framework Core, додавши пакет Entity Framework Core командою ``dotnet add package Microsoft.EntityFrameworkCore.SqlServer``, а також налаштування контексту бази даних та міграцій. Після цього можна розпочати розробку та запуск проекту за допомогою Visual Studio для написання коду та налагодження, а також виконати команду ``dotnet run`` для запуску проекту.

Цей базовий набір інструментів і кроків допоможе ефективно розпочати роботу з ASP.NET Core і розробляти сучасні веб-додатки.

Для роботи з Entity Framework Core потрібен певний набір інструментів, програмного забезпечення та знань, які забезпечують ефективну розробку, тестування та взаємодію з базами даних. По-перше, важливо мати операційну систему. Entity Framework Core, як і ASP.NET Core, підтримує кросплатформенність. Можна розробляти та розгортати додатки на різних операційних системах, таких як Windows, macOS та Linux. Зазвичай використовують Windows 10 або новіші версії, останні версії macOS, або більшість дистрибутивів Linux, таких як Ubuntu, Debian, Fedora тощо.

Для роботи з командним рядком можна використовувати Terminal або Command Prompt для виконання команд .NET CLI. Також підійдуть PowerShell

(для Windows) або Bash (для Linux і macOS). Entity Framework Core підтримує роботу з різними базами даних. Нам знадобиться одна з наступних: SQL Server (Microsoft SQL Server для розробки з базами даних), PostgreSQL (відкрита реляційна база даних), SQLite (легка вбудована база даних для розробки та тестування) або MySQL (популярна реляційна база даних з відкритим кодом). У даному проєкті використовуватимемо MSSQL та PostgreSQL.

NuGet є основним пакетним менеджером для .NET і використовується для управління бібліотеками та інструментами. Для додавання Entity Framework Core у наш проєкт ми використовуватимемо NuGet пакети. Для роботи з Entity Framework Core необхідно додати відповідні бібліотеки у наш проєкт: Microsoft.EntityFrameworkCore (основна бібліотека для роботи з Entity Framework Core), Microsoft.EntityFrameworkCore.SqlServer (провайдер для роботи з SQL Server), Microsoft.EntityFrameworkCore.Sqlite (провайдер для роботи з SQLite), Microsoft.EntityFrameworkCore.Tools (інструменти для управління міграціями).

Створення та налаштування проєкту включає декілька кроків. Спочатку необхідно створити новий проєкт за допомогою командного рядка:

```
dotnet new mvc -n MyProject
cd MyProject
```

Потім додаємо необхідні пакети:

```
dotnet add package Microsoft.EntityFrameworkCore
dotnet add package Microsoft.EntityFrameworkCore.SqlServer
dotnet add package Microsoft.EntityFrameworkCore.Tools
```

Наступним кроком є налаштування контексту бази даних. Створюємо клас `DbContext` для представлення сеансу з базою даних:

```
public class ApplicationDbContext : DbContext
{
    public
    ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options) { }
```

```
public DbSet<MyEntity> MyEntities { get; set; }
}
```

Потім налаштуємо з'єднання з базою даних у `Startup.cs` або `Program.cs`:

```
services.AddDbContext<ApplicationDbContext>(options =>
options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));
```

Для створення та застосування міграцій виконуємо такі команди:

```
dotnet ef migrations add InitialCreate
dotnet ef database update
```

Для отримання докладної інформації слід звернутися до офіційної документації Entity Framework Core на сайті Microsoft. Також корисними будуть курси та туторіали на платформі Pluralsight, Udemy або Microsoft Learn. Для тестування можна використовувати xUnit, який є фреймворком для написання автоматизованих тестів, та Moq – бібліотеку для створення моків при юніт-тестуванні.

Отже, для роботи з Entity Framework Core необхідно мати відповідну операційну систему, інструменти розробки (IDE та .NET SDK), бази даних, пакетний менеджер NuGet, а також необхідні бібліотеки для роботи з базами даних. Процес включає створення нового проекту, додавання необхідних пакетів, налаштування контексту бази даних та виконання міграцій.

Для ефективної роботи з Microsoft SQL Server (MSSQL) нам потрібен певний набір інструментів, програмного забезпечення та знань, які допоможуть розробляти, адмініструвати та взаємодіяти з базами даних. По-перше, потрібна операційна система, на якій працюватиме SQL Server. MSSQL підтримує роботу на різних операційних системах: Windows (наприклад, Windows Server або Windows 10/11 для локального використання) та Linux (такі дистрибутиви, як Ubuntu, Red Hat Enterprise Linux (RHEL), CentOS, SLES, а також Docker контейнери).

Щоб розпочати роботу, нам необхідно завантажити та встановити останню версію Microsoft SQL Server з офіційного сайту Microsoft. Також знадобиться SQL Server Management Studio (SSMS) – інструмент для

управління SQL Server, який доступний для завантаження на офіційному сайті Microsoft.

Після встановлення необхідно налаштувати MSSQL. Це включає підключення до серверу за допомогою SSMS або іншого клієнта, створення баз даних за допомогою SSMS або SQL скриптів, а також налаштування користувачів і груп з відповідними правами доступу до баз даних та таблиць.

Для розробки з MSSQL потрібні інтегровані середовища розробки (IDE). Visual Studio є ідеальним для розробки .NET додатків з інтеграцією MSSQL, тоді як Visual Studio Code є легким редактором коду з розширенням для SQL Server.

Основними бібліотеками та драйверами, які нам знадобляться, є ADO.NET (набір класів .NET Framework, що забезпечує доступ до даних з баз даних), Entity Framework Core (ORM для роботи з базами даних у .NET додатках) та SQLClient (драйвер для підключення до MSSQL з .NET додатків).

Командний рядок також є важливим інструментом для управління MSSQL. SQLCMD – це інструмент командного рядка для виконання T-SQL команд та скриптів, а Bash/PowerShell використовуються для управління SQL Server на Linux або за допомогою Docker.

Для тестування та оптимізації ми можемо використовувати SQL Server Profiler (інструмент для профілювання запитів і моніторингу продуктивності) та Database Tuning Advisor (інструмент для аналізу та оптимізації продуктивності баз даних).

Для розгортання та управління базами даних нам стануть у пригоді Azure SQL Database (хмарний сервіс для розгортання та управління базами даних MSSQL) та контейнери Docker (для розгортання MSSQL у контейнерах).

Отже, для роботи з MSSQL потрібно мати відповідну операційну систему, інсталювати MSSQL сервер та SQL Server Management Studio, налаштувати сервер та бази даних, використовувати інструменти для розробки, такі як Visual Studio або Visual Studio Code, налаштувати бібліотеки та драйвери для доступу до баз даних, використовувати

командний рядок для управління сервером, а також використовувати інструменти для тестування, оптимізації та розгортання баз даних.

PostgreSQL - це потужна та гнучка система управління реляційними базами даних (СУБД), яка широко використовується в різних галузях. Її популярність обумовлена широким спектром функцій, високою продуктивністю та надійністю, а також відкритим кодом. Щоб розпочати роботу з PostgreSQL, вам знадобиться певне програмне та апаратне забезпечення, а також базове розуміння принципів роботи СУБД. У цій статті ми розглянемо основні вимоги до роботи з PostgreSQL, а також надамо деякі ресурси, які допоможуть вам розпочати роботу. PostgreSQL сумісна з широким спектром операційних систем, включаючи Linux, macOS та Windows. Найпопулярнішим вибором є Linux, адже він пропонує найкращу продуктивність, надійність та гнучкість для налаштування. Деякі популярні дистрибутиви Linux для PostgreSQL: Ubuntu, Debian, CentOS, Red Hat Enterprise Linux. Нам потрібно буде завантажити та встановити PostgreSQL на свій комп'ютер. Це можна зробити з офіційного сайту PostgreSQL: <https://www.postgresql.org/download/>. Для взаємодії з базою даних нам знадобиться клієнт. Існує безліч безкоштовних та платних клієнтів, як графічних, так і командних. Деякі популярні варіанти:

- pgAdmin: Графічний клієнт з широким спектром функцій для керування базами даних, користувачами, схемами, даними тощо.
- DBeaver: Універсальний інструмент для роботи з базами даних, що підтримує PostgreSQL, MySQL, Oracle, SQL Server та інші.
- psql: Командний рядок для прямої взаємодії з PostgreSQL за допомогою текстових команд SQL.

#### Додаткові міркування

- З'єднання з мережею: Якщо ми плануємо отримувати доступ до бази даних PostgreSQL з декількох комп'ютерів, вам знадобиться мережеве підключення.
- Безпека: Важливо вжити заходів для захисту бази даних PostgreSQL[13] від несанкціонованого доступу. Це може включати використання паролів, брандмауерів та шифрування.



- Резервне копіювання: Регулярно створювання резервних копій бази даних PostgreSQL[14], щоб уникнути втрати даних у разі збою системи або іншої аварії.

PostgreSQL - це потужна та гнучка СУБД, яка може використовуватися для вирішення широкого кола завдань. Дотримуючись вищеописаних рекомендацій, ви зможете встановити та налаштувати PostgreSQL[15] на своєму комп'ютері та розпочати роботу з цією популярною СУБД.

## 5.2 Використання Angular

Для роботи з Angular, популярним фреймворком для створення веб-додатків, нам знадобляться різноманітні інструменти, програмне забезпечення та знання. Нижче наведено основні компоненти та кроки, необхідні для роботи з Angular:

Angular є кросплатформенним фреймворком, тому ми можемо розробляти додатки на будь-якій сучасній операційній системі:

- Windows: Windows 10 або новіша версія.
- macOS: Сучасні версії macOS.
- Linux: Поширені дистрибутиви, такі як Ubuntu, Debian, Fedora тощо.
- Node.js: Angular потребує Node.js для роботи. Ми повинні завантажити та встановити останню стабільну версію Node.js з [офіційного сайту Node.js](<https://nodejs.org/>).

- npm (Node Package Manager): Разом з Node.js встановлюється npm, який використовується для управління пакетами та залежностями у нашому проекті.

- Angular CLI (Command Line Interface): Angular CLI є інструментом для створення, управління та розгортання Angular додатків. Ми можемо встановити його за допомогою npm:

```
npm install -g @angular/cli
```

Тепер нам потрібно обрати середу розробки:

- Visual Studio Code: Легке та потужне IDE з багатьма розширеннями для підтримки розробки Angular додатків. Ми можемо завантажити його з [офіційного сайту Visual Studio Code](<https://code.visualstudio.com/>).

- WebStorm: Потужне IDE від JetBrains з підтримкою Angular. Платне, але має безкоштовний пробний період.

- Sublime Text або інші редактори коду: Легкі редактори, які також можуть бути налаштовані для роботи з Angular за допомогою відповідних плагінів.

Створення нового проекту: Використовуючи Angular CLI, ми можемо створити новий проект:

```
ng new my-angular-app
cd my-angular-app
```

- Щоб запустити додаток у режимі розробки:

```
ng serve
```

Додаток буде доступний за адресою `http://localhost:4200`.

- Angular Core: Основні модулі для роботи з Angular.

- RxJS (Reactive Extensions for JavaScript): Бібліотека для роботи з асинхронним програмуванням за допомогою Observables.

- Angular Material або інші UI-бібліотеки: Для додавання стильних компонентів інтерфейсу.

- Karma: Тестовий раннер для запуску юніт-тестів.

- Jasmine: Фреймворк для написання тестів.

- Protractor: Інструмент для автоматизованого енд-ту-енд тестування Angular додатків.

Документація Angular є найкращим джерелом для навчання та довідки. Доступна на [angular.io](https://angular.io/docs).

- Хостинг: Ми можемо розгорнути Angular додатки на різних платформах, таких як Firebase, Netlify, Vercel, GitHub Pages або будь-який інший веб-сервер.

- Будування додатку: Щоб зібрати додаток для продакшн-середовища:

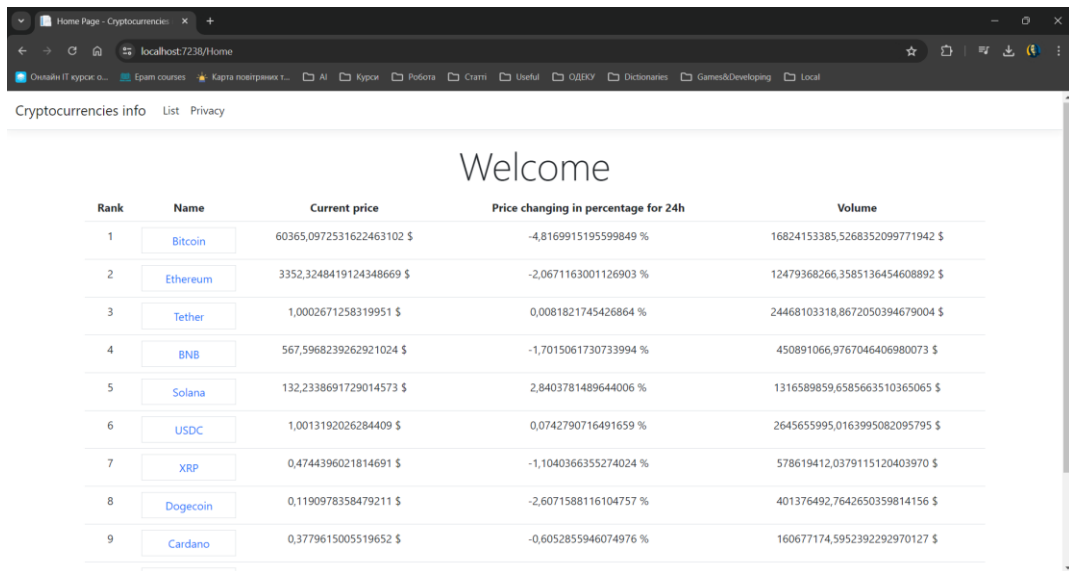
```
ng build --prod
```

Для роботи з Angular нам потрібні сучасна операційна система, встановлені Node.js та npm, Angular CLI для управління проектами, інтегроване середовище розробки, таке як Visual Studio Code, основні бібліотеки та модулі Angular, інструменти для тестування, офіційна

документація та навчальні ресурси, а також платформи для деплоюменту додатків. Цей набір інструментів та знань дозволить ефективно розробляти, тестувати та розгортати сучасні веб-додатки з використанням Angular.

### 5.3 Опис та тестування застосунку

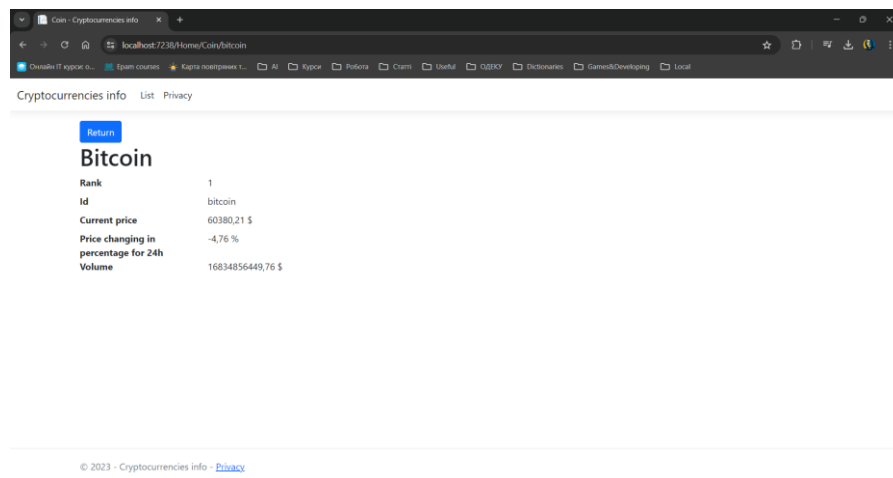
Значок застосунку зображений на рис. 5.13. Робота з застосунком починається з головної сторінки (рис. 5.14).



Rank	Name	Current price	Price changing in percentage for 24h	Volume
1	Bitcoin	60365,0972531622463102 \$	-4,8169915195599849 %	16824153385,5268352099771942 \$
2	Ethereum	3352,3248419124348669 \$	-2,0671163001126903 %	12479368266,3585136454608892 \$
3	Tether	1,0002671258319951 \$	0,0081821745426864 %	24468103318,8672050394679004 \$
4	BNB	567,5968239262921024 \$	-1,7015061730733994 %	450891066,9767046406980073 \$
5	Solana	132,2338691729014573 \$	2,8403781489644006 %	1316589859,6585663510365065 \$
6	USDC	1,0013192026284409 \$	0,0742790716491659 %	2645655995,0163995082095795 \$
7	XRP	0,4744396021814691 \$	-1,104036635274024 %	578619412,0379115120403970 \$
8	Dogecoin	0,1190978358479211 \$	-2,6071588116104757 %	401376492,7642650359814156 \$
9	Cardano	0,3779615005519652 \$	-0,6052855946074976 %	160677174,5952392292970127 \$

Рисунок 5.14 – Головна сторінка застосунку «Cryptocurrencies-info»

Користувач може обрати одну з валют та переглянути її інформацію (рис. 5.15).

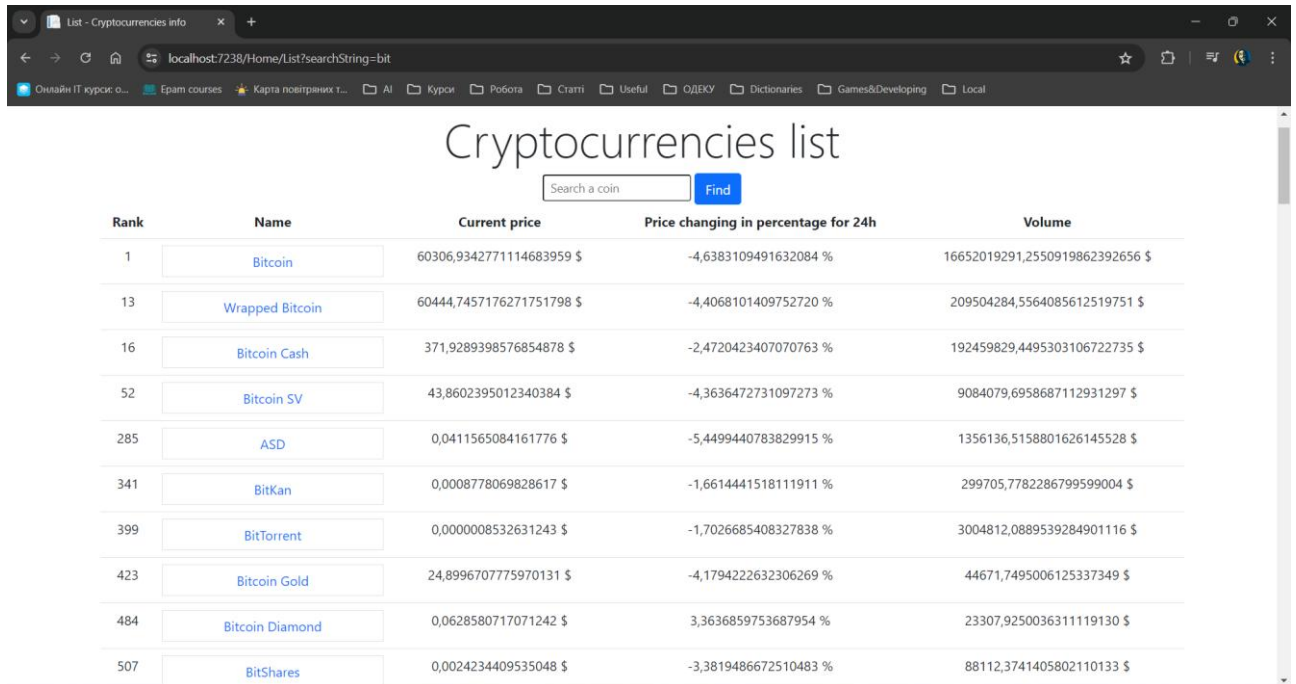


Rank	1
Id	bitcoin
Current price	60380,21 \$
Price changing in percentage for 24h	-4,76 %
Volume	16834856449,76 \$

© 2023 - Cryptocurrencies info - [Privacy](#)

Рисунок 5.15 – Сторінка валюти "Bitcoin"

З головної сторінки користувач також може перейти до перегляду популярніших валют (рис. 5.16)



The screenshot shows a web browser window with the address bar displaying 'localhost:7238/Home/List?searchString=bit'. The page title is 'Cryptocurrencies list'. Below the title is a search bar with the placeholder text 'Search a coin' and a 'Find' button. The main content is a table with the following data:

Rank	Name	Current price	Price changing in percentage for 24h	Volume
1	Bitcoin	60306,9342771114683959 \$	-4,6383109491632084 %	16652019291,2550919862392656 \$
13	Wrapped Bitcoin	60444,7457176271751798 \$	-4,4068101409752720 %	209504284,5564085612519751 \$
16	Bitcoin Cash	371,9289398576854878 \$	-2,4720423407070763 %	192459829,4495303106722735 \$
52	Bitcoin SV	43,8602395012340384 \$	-4,3636472731097273 %	9084079,6958687112931297 \$
285	ASD	0,0411565084161776 \$	-5,4499440783829915 %	1356136,5158801626145528 \$
341	BitKan	0,0008778069828617 \$	-1,6614441518111911 %	299705,7782286799599004 \$
399	BitTorrent	0,0000008532631243 \$	-1,7026685408327838 %	3004812,0889539284901116 \$
423	Bitcoin Gold	24,8996707775970131 \$	-4,1794222632306269 %	44671,7495006125337349 \$
484	Bitcoin Diamond	0,0628580717071242 \$	3,3636859753687954 %	23307,925003631119130 \$
507	BitShares	0,0024234409535048 \$	-3,3819486672510483 %	88112,3741405802110133 \$

Рисунок 5.16 – Сторінка валют

На сторінці пошуку є можливість шукати валюти за ключовими символами. При наявності збігів додаток надає їх.

Rank	Name	Current price	Price changing in percentage for 24h	Volume
1	Bitcoin	60380,9671318855962318 \$	-4,8169915195599849 %	16836426839,1295932237232733 \$
2	Ethereum	3352,2585654364477849 \$	-2,0671163001126903 %	12481459851,6594620045769427 \$
3	Tether	1,0003748081287492 \$	0,0081821745426864 %	24468721700,0863159182756321 \$
4	BNB	567,5909598908089290 \$	-1,7015061730733994 %	450976628,1366076380677285 \$
5	Solana	132,1504241299073174 \$	2,8403781489644006 %	1316728205,8566824667885817 \$
6	USDC	1,0013624568356256 \$	0,0742790716491659 %	2645919748,2937353285372411 \$
7	XRP	0,4745445415668316 \$	-1,1040366355274024 %	578624500,7511056846734099 \$
8	Dogecoin	0,1191207850419468 \$	-2,6071588116104757 %	401501153,4284587164629012 \$
9	Cardano	0,3780317388345054 \$	-0,6052855946074976 %	160665878,0902845867096622 \$

Рисунок 5.19 – Валюти, що збіглися

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра був створений веб-застосунок «Cryptocurrencies-info» для пошуку інформації про криптовалюти з використанням технологій ASP.NET Core, Entity Framework та Angular. Для цього був проведений аналіз предметної області та порівняльний аналіз існуючих програм-аналогів у пошуковнику Google. В якості програмного засобу розробки було обрано середовище Rider[9] з огляду на легкість та багатофункціональність.

На етапі проектування були створені UML-діаграми варіантів використання (прецедентів), активності і послідовності та розроблені макети інтерфейсу застосунку. З огляду на необхідність зберігання великої кількості даних для застосунку та постійне додавання даних було вирішено використовувати базу даних MSSQL або PostgreSQL, їх можна перемикати у налаштування сервера за потреби.

В роботі представлені можливості застосунку та поетапний приклад його використання. Було виконано тестування роботи застосунку. Проблем використання не виявлено.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Metanit: Посібник ASP.NET Core. URL:  
<https://metanit.com/sharp/aspnet6/> (дата звернення 14.06.2024)
2. Metanit: Посібник Angular. URL: <https://metanit.com/web/angular2/>  
(дата звернення 14.05.2024)
3. Microsoft документація: Офіційний посібник ASP.NET Core. URL:  
<https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0>  
(дата звернення 14.06.2024)
4. Angular документація: Офіційний посібник Angular. URL:  
<https://angular.dev/overview> (дата звернення 14.05.2024)
5. CoinGecko: Сайт криптовалют CoinGecko. URL:  
<https://www.coingecko.com/uk> (дата звернення 14.06.2024)
6. CoinCap: Сайт криптовалют CoinCap. URL: <https://coincap.io/> (дата  
звернення 14.05.2024)
7. Visual Studio: офіційна IDE від Microsoft. URL:  
<https://visualstudio.microsoft.com/> (дата звернення 14.05.2024)
8. Visual Studio Code: легкомістка IDE від Microsoft. URL:  
<https://code.visualstudio.com/> (дата звернення 14.06.2024)
9. Rider: кросплатформена IDE від JetBrains. URL:  
<https://www.jetbrains.com/rider/> (дата звернення 14.05.2024)
10. SonarLint: потужний лінтер. URL:  
<https://www.sonarsource.com/products/sonarlint/> (дата звернення  
14.06.2024)
11. OmniSharp: лінтер з відкритим кодом. URL:  
<https://www.omnisharp.net/> (дата звернення 14.05.2024)
12. Refactoring Guru: сайт патернів та рефакторингу. URL:  
<https://refactoring.guru/> (дата звернення 14.06.2024)
13. PostgreSQL: Документація PostgreSQL. URL:  
<https://www.postgresql.org/docs/> (дата звернення 14.05.2024)

14. PostgreSQL: Посібник для початківців PostgreSQL. URL: <https://www.postgresql.org/docs/current/tutorial.html> (дата звернення 14.06.2024)
15. Відеоуроки з PostgreSQL: <https://www.youtube.com/watch?v=qw--VYLpxG4> (дата звернення 14.05.2024)
16. Програмування С#, Ріхтер: <https://ptgmedia.pearsoncmg.com/images/9780735667457/samplepages/9780735667457.pdf> (дата звернення 14.06.2024)
17. Вихідний код програми: <https://github.com/MoonWalker/Cryptocurrencies-info>