**1 ВСТУП**

* 1. Огляд продукту

Розроблюваний програмний продукт є високонавантаженою аналітичною системою, яка забезпечує збір, обробку та візуалізацію комерційних та фінансових даних у режимі реального часу. Система орієнтована на ринок електронної комерції та дозволяє відстежувати зміни в цінах товарів і криптовалют, аналізувати категорії продукції, а також оперативно інформувати користувачів про оновлення.

* 1. Мета

Метою розробки є створення ефективного інструменту для отримання актуальної інформації з різних джерел електронної торгівлі, об’єднання цих даних у єдину структуру та надання користувачам можливості переглядати аналітичну інформацію без затримок, що особливо важливо в умовах високої динаміки цифрового ринку.

* 1. Межі

Розроблювана система призначена для збору, агрегації та візуалізації аналітичних даних з відкритих API маркетплейсів і криптовалютних сервісів у режимі реального часу. Основна увага зосереджена на демонстрації можливостей обробки великого обсягу інформації з мінімальною затримкою, формуванні агрегованих звітів та їх відображенні у веб-інтерфейсі. Система не охоплює підтримку мобільних платформ, роботу з особистими фінансовими даними користувачів або повноцінну поведінкову аналітику з використанням машинного навчання. Крім того, у межах поточного проєкту не передбачено інтеграцію з платіжними шлюзами чи реалізацію складних механізмів керування правами доступу. Пріоритетом є швидкодія, масштабованість та стабільність роботи при високому навантаженні на серверну частину системи.

* 1. Означення та абревіатури

Таблиця 1 - Означення

|  |  |
| --- | --- |
| **Термін** | **Означення** |
| **API** | Інтерфейс прикладного програмування, набір правил, які дозволяють взаємодіяти між різними програмними компонентами. |
| **ASP.NET Core** | Кросплатформенний фреймворк для розробки веб-додатків та API на основі .NET. |
| **SignalR** | Бібліотека від Microsoft для реалізації двосторонньої комунікації між клієнтом і сервером у режимі реального часу. |
| **REST API** | Архітектурний стиль побудови API, який використовує стандартні HTTP методи (GET, POST, PUT, DELETE) для взаємодії з ресурсами. |
| **MongoDB** | Документо-орієнтована NoSQL база даних, що дозволяє зберігати напівструктуровані дані у форматі BSON. |
| **Entity Framework** | ORM (Object-Relational Mapping) бібліотека для .NET, яка дозволяє працювати з базами даних через C# класи замість SQL-запитів. |
| **WebSocket** | Протокол для двостороннього обміну даними між браузером і сервером з мінімальною затримкою. |
| **RxJS** | Бібліотека JavaScript для реактивного програмування з використанням обсервабельних потоків. |

**2 ЗАГАЛЬНИЙ ОПИС**

2.1 Перспективи продукту

Програмна система аналітики даних електронної комерції у реальному часі має широкі перспективи розвитку в умовах цифрової трансформації бізнесу та стрімкого зростання обсягів онлайн-торгівлі. У сучасному e-commerce критично важливим є здатність оперативно реагувати на зміни попиту, поведінку споживачів, коливання цін та конкурентні дії. Продукт орієнтований на забезпечення миттєвої обробки великих потоків даних із різних джерел — маркетплейсів, криптовалютних бірж, відкритих бібліотек та інших сервісів. Завдяки використанню технологій ASP.NET Core, SignalR, MongoDB, MS SQL Server, Angular і Chart.js, система поєднує масштабовану серверну архітектуру з реактивним клієнтським інтерфейсом, забезпечуючи візуалізацію та трансляцію статистичних даних у реальному часі. У майбутньому платформа може бути розширена за рахунок підключення нових джерел, впровадження алгоритмів прогнозування на основі машинного навчання, розширення аналітичних можливостей через глибшу сегментацію даних та інтеграцію з ERP/CRM-системами для корпоративного використання. Система може бути використана як внутрішній інструмент для рітейлерів, аналітичних агентств, маркетологів, або як основа для SaaS-продукту з відкритим доступом до агрегованої аналітики через підписку.

2.2 Функції продукту

Основні функції програмної системи:

* MF1: авторизація та реєстрація користувачів;
* MF2: перевірка електронної пошти за допомогою коду підтвердження;
* MF3: агрегація даних з маркетплейсів (DummyJSON, FakeStore тощо);
* MF4: отримання актуальної інформації про криптовалюти з CoinGecko;
* MF5: збереження історичних статистичних даних у базі MongoDB;
* MF6: трансляція оновлень у реальному часі за допомогою SignalR;
* MF7: візуалізація статистичних даних на клієнтській частині у вигляді діаграм і графіків;
* MF8: багатомовність інтерфейсу з підтримкою JSON-файлів перекладу;
* MF9: адміністрування користувачів і мов через захищену адмін-панель;
* MF10: REST API для взаємодії з клієнтськими додатками

2.3 Характеристики користувачів

Програмна система орієнтована на декілька категорій користувачів, кожна з яких має свій рівень доступу та набір функціональних можливостей.

Неаутентифікований користувач — відвідувач платформи, який ще не виконав вхід у систему. Такий користувач має обмежений доступ: він може переглядати загальнодоступну статистику, ознайомлюватися з графіками, але не має можливості змінювати налаштування чи взаємодіяти з персоналізованими функціями. Неаутентифіковані користувачі також можуть пройти реєстрацію або виконати вхід, щоб отримати розширений функціонал.

Аутентифікований користувач — зареєстрований користувач, який успішно пройшов процедуру входу в систему. Такий користувач має доступ до персоналізованих налаштувань, збереженої історії переглядів або участі в подіях, а також розширеної аналітики. В залежності від прав доступу, може брати участь у взаємодії з даними або налаштовувати параметри відображення.

Адміністратор системи — спеціальна роль з максимальними правами доступу, призначена для обслуговування й керування платформою. Адміністратор може керувати користувачами (архівування, перегляд списку), додавати та видаляти мовні пакети, змінювати вміст JSON-файлів перекладу, а також мати повний доступ до аналітичних даних і логів системи. Ця роль має захищений доступ, контрольований авторизацією на основі ролей.

Основні дії користувачів відображені на діаграмі варіантів використання  
(рис. 1).

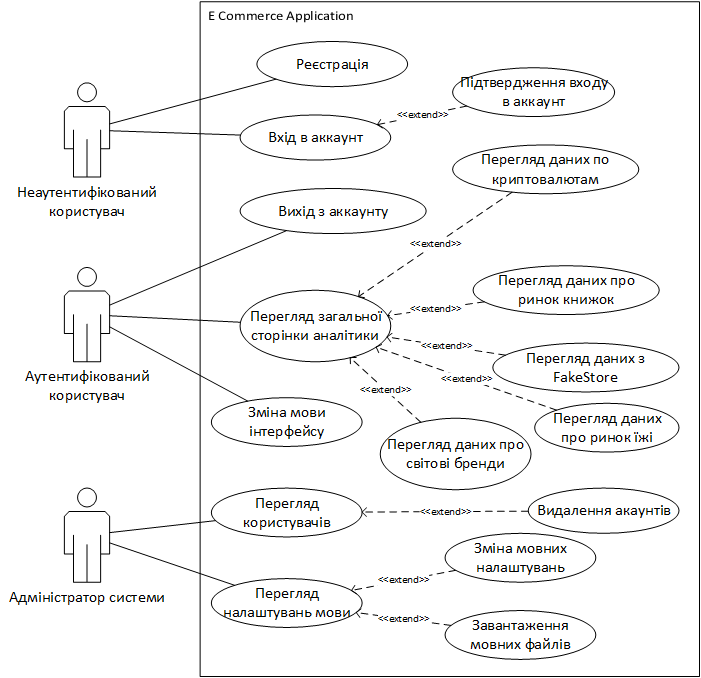


Рисунок 1 – Діаграма варіантів використання системи

2.4. Загальні обмеження

Визначені обмеження:

* **LI1**: рівень доступу до функцій залежить від ролі користувача (гостьовий, зареєстрований, адміністратор);
* **LI2**: система підтримує лише зовнішні джерела даних із фіксованим набором API (DummyJSON, FakeStore, CoinGecko тощо);
* **LI3**: клієнтський інтерфейс оптимізовано для сучасних браузерів і не гарантує коректну роботу в застарілих версіях;
* **LI4**: оновлення аналітичних даних відбувається з певною періодичністю, заданою на рівні фонового сервісу;
* **LI5**: передача даних у реальному часі забезпечується через SignalR, що вимагає стабільного WebSocket-з’єднання на клієнтській стороні.

2.4. Припущення та залежності

Визначені припущення:

* AS-1: зовнішні API будуть доступні та відповідатимуть специфікаціям, зазначеним на момент інтеграції;
* AS-2: користувачі вводять достовірну та актуальну інформацію при реєстрації та взаємодії з платформою;
* AS-3: серверна інфраструктура має достатню продуктивність для обробки запитів без значних затримок у пік навантаження.

Визначені залежності:

* DE-1: працездатність системи залежить від наявності стабільного з’єднання з базами даних MongoDB та Microsoft SQL Server;
* DE-2: успішна робота двосторонньої комунікації залежить від коректної роботи SignalR і підтримки WebSocket протоколу в браузері;
* DE-3: правильне функціонування клієнтської частини системи потребує попередньої ініціалізації Angular застосунку з дійсними маршрутами та сервісами.

**3 КОНКРЕТНІ ВИМОГИ**

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Інтерфейс користувача повинен бути інтуїтивно зрозумілим, адаптивним до різних типів пристроїв (настільні ПК, планшети, смартфони), забезпечувати зручну навігацію між розділами та оперативну взаємодію з аналітичними даними. Особливу увагу приділено візуалізації статистики, а також доступності функцій для різних ролей користувачів. Інтерфейс має підтримувати мультимовність та коректно відображати динамічно оновлювану інформацію в режимі реального часу.

Вимоги до інтерфейсу:

* UI адаптується до мобільних та десктопних екранів;
* дані виводяться у вигляді графіків і таблиць;
* користувач має доступ до оновлень у реальному часі;
* авторизовані користувачі бачать персоналізовану аналітику;
* інтерфейс доступний мінімум українською та англійською мовами;
* усі інтерактивні елементи мають чіткі підписи та інструкції;
* помилки вводів супроводжуються зрозумілими повідомленнями;
* кольорова схема відповідає принципам контрастності та доступності;
* панель адміністратора має окремий доступ і додаткові функції;
* пошук і фільтрація даних доступні у ключових розділах;

3.1.2 Апаратний інтерфейс

Апаратний інтерфейс визначає базові технічні засоби, необхідні для коректної роботи програмної системи аналітики даних електронної комерції у реальному часі як на стороні клієнта, так і на стороні сервера. Оскільки система реалізована як веб-застосунок, її апаратні вимоги є загалом невисокими, однак для забезпечення стабільної роботи потрібна відповідність певним умовам.

З боку користувача:

* пристрій з підтримкою сучасного браузера (наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge або Safari);
* клавіатура та миша (для ПК) або сенсорне керування (для мобільних пристроїв);
* активне підключення до Інтернету зі стабільною швидкістю передачі даних.

З боку серверної інфраструктури:

* сервер, здатний виконувати .NET-додатки та хостити API з використанням ASP.NET Core;
* встановлене середовище виконання для MongoDB та Microsoft SQL Server;
* щонайменше 8 ГБ оперативної памʼяті та достатній об’єм SSD для обробки запитів і зберігання аналітичних даних;
* резервне копіювання даних та базова система моніторингу навантаження для забезпечення відмовостійкості.

3.1.3 Програмний інтерфейс

Програмний інтерфейс описує взаємодію між компонентами системи, а також зовнішніми сервісами і платформами, необхідними для коректної роботи системи аналітики даних електронної комерції у реальному часі. В системі передбачено використання API для отримання та обробки даних із різних джерел, а також інтерфейси для комунікації між серверною та клієнтською частинами.

Основні вимоги до програмних інтерфейсів:

* система повинна підтримувати REST API для взаємодії клієнтської частини з сервером;
* використання SignalR для забезпечення двосторонньої комунікації у реальному часі між сервером і клієнтом;;
* серверна частина повинна інтегруватися з API зовнішніх джерел даних (CoinGecko, DummyJSON, FakeStore, OpenFood, OpenLibrary);
* використання Entity Framework Core для роботи з реляційною базою даних Microsoft SQL Server;
* використання драйвера MongoDB для зберігання та отримання історичних аналітичних даних;
* підтримка аутентифікації та авторизації через JWT токени відповідно до ролей користувачів;
* забезпечення безпечного обміну даними через HTTPS протокол.

3.1.4 Комунікаційний протокол

Комунікаційний протокол визначає правила та стандарти обміну даними між клієнтською частиною та сервером системи аналітики електронної комерції у реальному часі. У проекті обрано технології, що забезпечують надійну, швидку та ефективну передачу інформації з мінімальною затримкою, що є критичним для аналітичних систем із високими вимогами до оперативності.

Характеристики комунікаційного протоколу:

* використання протоколу http/https для запитів між клієнтом і сервером;
* застосування REST API для стандартних операцій обміну даними;
* впровадження signalr з webSocket для двонаправленої комунікації у реальному часі;
* забезпечення шифрування даних під час передачі через ssl/tls;
* підтримка повторного встановлення з’єднання у разі втрати зв’язку;
* оптимізація передачі даних шляхом використання стиснення payload там, де це можливо.

3.1.5 Обмеження пам’яті

Обмеження пам’яті визначають максимальні ресурси оперативної та постійної пам’яті, доступні для роботи системи аналітики електронної комерції. Враховуючи характер обробки великих обсягів даних у реальному часі, оптимальне використання пам’яті є критично важливим для забезпечення продуктивності та стабільності роботи системи.

Визначені обмеження:

* серверна частина повинна працювати в межах доступного обсягу оперативної пам’яті, який не перевищує 16 ГБ;
* обсяг дискового простору для зберігання даних залежить від масштабу навантаження і має бути не менше 100 ГБ;
* клієнтський додаток повинен коректно працювати на пристроях із мінімум 4 ГБ оперативної пам’яті;
* обробка великих масивів даних реалізується з використанням ефективних алгоритмів та потокової обробки, щоб мінімізувати споживання пам’яті;
* система передбачає механізми кешування, які дозволяють зменшити навантаження на оперативну пам’ять без втрати актуальності даних.

3.1.6 Операції

Система аналітики електронної комерції забезпечує комплекс операцій, спрямованих на збір, обробку та відображення даних у реальному часі. Основні операції включають періодичне отримання інформації з різних джерел, таких як маркетплейси та криптовалютні біржі, через інтегровані API. Отримані дані агрегуються та зберігаються у відповідних базах даних, що дає змогу формувати статистичні звіти та аналітичні графіки.

Для користувачів система надає функції авторизації та персоналізації, забезпечує доступ до актуальної інформації через динамічні інтерфейси, що оновлюються в реальному часі за допомогою технології SignalR. Адміністратори мають можливість управляти користувачами, контролювати джерела даних та налаштовувати параметри агрегації.

Операції виконуються в рамках встановлених часових інтервалів і підтримують масштабування навантаження, що забезпечує стабільність роботи навіть при великій кількості одночасних підключень. Крім того, система реалізує механізми обробки помилок та логування, що сприяють підтримці безперервності сервісу і оперативному виявленню проблем.

3.1.7 Функції продукту

3.1.7.1 Реєстрація

3.1.7.1.1 Вступ

Незареєстровані користувачі мають можливість створити обліковий запис у системі, що дає доступ до персоналізованих сервісів. Для реєстрації користувач переходить на сторінку реєстрації та заповнює необхідні поля.

3.1.7.1.2 Вхідні дані

Користувач вводить електронну пошту, пароль та підтвердження пароля.

3.1.7.1.3 Обробка

Після натискання кнопки "Зареєструватися" система перевіряє, чи введена електронна пошта унікальна серед вже існуючих записів у базі даних. Якщо електронна пошта унікальна, система оцінює відповідність пароля політиці безпеки (довжина, складність). При позитивній перевірці створюється новий обліковий запис, інформація зберігається у базі даних, а користувач отримує підтвердження. У випадку невідповідності даних реєстрація не відбувається.

3.1.7.1.4 Вихідні дані

Користувач бачить повідомлення про успішну реєстрацію і отримує можливість увійти до системи.

3.1.7.1.5 Обробка помилок

Якщо електронна пошта вже використовується, система виводить повідомлення про необхідність вибрати іншу пошту. Якщо пароль не відповідає вимогам, користувачу показується інформація про правила формування паролю.

3.1.7.2 Авторизація

3.1.7.2.1 Вступ

Зареєстровані користувачі мають можливість увійти в систему, щоб отримати доступ до персоналізованих функцій і даних.

3.1.7.2.2 Вхідні дані

Користувач вводить електронну пошту та пароль.

3.1.7.2.3 Обробка

Після введення даних система перевіряє відповідність логіну та паролю у базі даних. Якщо дані коректні, генерується токен доступу (JWT), який використовується для аутентифікації в подальших запитах. У разі успішної авторизації користувач отримує доступ до системи.

3.1.7.2.4 Вихідні дані

Користувач отримує підтвердження успішного входу та токен доступу.

3.1.7.2.5 Обробка помилок

Якщо введені дані некоректні або користувач не зареєстрований, система виводить повідомлення про помилку і пропонує повторити спробу.

3.1.7.3 Агрегація даних з маркетплейсів

3.1.7.3.1 Вступ

Система автоматично збирає актуальні дані про товари, ціни та категорії з різних маркетплейсів за допомогою API, щоб забезпечити користувачів оновленою інформацією.

3.1.7.3.2 Вхідні дані

Запити до API маркетплейсів (DummyJSON, FakeStore) з параметрами отримання списку товарів, категорій та цін.

3.1.7.3.3 Обробка

Фонові сервіси періодично виконують запити до API маркетплейсів, отримують дані, уніфікують їх за структурою, обробляють (наприклад, фільтрація, нормалізація), а потім зберігають у центральній базі даних. Обробка забезпечує актуальність і цілісність даних.

3.1.7.3.4 Вихідні дані

Оновлена база даних, яка містить свіжу інформацію про товари, категорії та їхні ціни для подальшої аналітики та відображення користувачам.

3.1.7.3.5 Обробка помилок

У разі проблем із доступом до API або помилок відповіді, система реєструє помилки у логах і здійснює повторні спроби через задані інтервали.

3.1.7.4 Моніторинг цін криптовалют

3.1.7.4.1 Вступ

Система відслідковує зміни в цінах криптовалют, ринкову капіталізацію та динаміку за останні 24 години, забезпечуючи актуальні дані для користувачів.

3.1.7.4.2 Вхідні дані

Запити до API CoinGecko для отримання актуальних даних про криптовалюти.

3.1.7.4.3 Обробка

Фонові сервіси періодично роблять запити до CoinGecko API, отримують свіжі дані, аналізують і зберігають їх у базі даних. Ці дані потім використовуються для аналітики та візуалізації.

3.1.7.4.4 Вихідні дані

Оновлена інформація по цінах, ринковій капіталізації і динаміці криптовалют, доступна у системі.

3.1.7.4.5 Обробка помилок

У разі недоступності API або помилок при отриманні даних система реєструє проблему і намагається повторити запит через певний інтервал.

3.1.7.5 Обробка та збереження історичних даних

3.1.7.5.1 Вступ

Система забезпечує накопичення, обробку та зберігання історичних даних для подальшого аналізу та відображення трендів.

3.1.7.5.2 Вхідні дані

Агреговані дані з маркетплейсів та криптовалют, отримані у режимі реального часу.

3.1.7.5.3 Обробка

Отримані дані регулярно зберігаються у базі даних з часовими позначками. Відбувається їх нормалізація, перевірка на дублікати та структуризація для ефективного доступу. Проводиться обробка для формування звітів і аналітичних вибірок.

3.1.7.5.4 Вихідні дані

Історичні записи, доступні для запитів і візуалізації, що відображають зміни та тренди за визначені періоди.

3.1.7.5.5 Обробка помилок

У випадку помилок збереження даних система фіксує їх у логах і намагається повторити операцію зберігання. При некоректних вхідних даних відбувається ігнорування або корекція відповідно до правил.

3.1.7.6 Візуалізація аналітичних даних в реальному часі

3.1.7.6.1 Вступ

Користувачі отримують динамічні графіки та діаграми, які відображають оновлені аналітичні дані без необхідності оновлення сторінки.

3.1.7.6.2 Вхідні дані

Оновлені дані, що надходять через SignalR у вигляді статистики по товарах, категоріях і криптовалютах.

3.1.7.6.3 Обробка

Система обробляє потоки даних, трансформує їх у формат, зручний для візуалізації, та відправляє на клієнтську частину. Клієнтська частина за допомогою Chart.js відображає ці дані у вигляді інтерактивних графіків.

3.1.7.6.4 Вихідні дані

Інтерактивні графіки, що відображають актуальні зміни у статистиці та трендах, доступні користувачу в режимі реального часу.

3.1.7.6.5 Обробка помилок

Якщо оновлення даних не отримані, система виводить повідомлення про тимчасову відсутність інформації або залишає останні доступні дані.

3.1.7.7 Зміна мови інтерфейсу

3.1.7.7.1 Вступ

Система підтримує зміну мови інтерфейсу для зручності користувачів з різних регіонів.

3.1.7.7.2 Вхідні дані

Вибір користувачем однієї з доступних мов інтерфейсу (наприклад, українська, англійська).

3.1.7.7.3 Обробка

Після вибору мови система завантажує відповідний мовний пакет, оновлює текстові елементи інтерфейсу, кнопки, повідомлення та меню, забезпечуючи повний переклад.

3.1.7.7.4 Вихідні дані

Інтерфейс користувача відображається обраною мовою без необхідності перезавантаження сторінки.

3.1.7.7.5 Обробка помилок

Якщо мовний пакет відсутній або пошкоджений, система повертається до мови за замовчуванням та повідомляє користувача про проблему.

3.1.7.8 Керування користувачами

3.1.7.8.1 Вступ

Адміністратори системи мають можливість управляти обліковими записами користувачів, включаючи редагування, блокування та видалення.

3.1.7.8.2 Вхідні дані

Ідентифікатор користувача, нові дані для оновлення або команда на зміну статусу (активний, заблокований тощо).

3.1.7.8.3 Обробка

Система приймає адміністративні запити на зміну даних користувача, перевіряє права доступу, проводить валідацію та оновлює відповідні записи в базі даних. Заборонені дії блокуються з відповідним повідомленням.

3.1.7.8.4 Вихідні дані

Підтвердження успішного оновлення, повідомлення про блокування або видалення користувача.

3.1.7.8.5 Обробка помилок

У випадку відсутності прав адміністратора або некоректних даних система виводить повідомлення про помилку та відхиляє запит.

3.1.7.9 Управління мовними ресурсами

3.1.7.9.1 Вступ

Адміністратори можуть додавати, редагувати або видаляти мовні пакети і переклади для підтримки багатомовності системи.

3.1.7.9.2 Вхідні дані

Файли перекладів або ключі з відповідними текстовими значеннями для різних мов інтерфейсу.

3.1.7.9.3 Обробка

Система приймає зміни мовних ресурсів, перевіряє їх коректність і зберігає у відповідному сховищі. Внесені зміни одразу доступні для застосування в інтерфейсі.

3.1.7.9.4 Вихідні дані

Підтвердження успішного оновлення мовних ресурсів і повідомлення про помилки, якщо вони виникають.

3.1.7.9.5 Обробка помилок

У разі некоректного формату файлів або конфліктів у ключах система інформує адміністратора про помилки і не застосовує зміни.

3.2. Властивості програмного продукту

Розроблювана система аналітики даних електронної комерції у реальному часі відзначається високою продуктивністю та масштабованістю, що дозволяє ефективно обробляти сотні тисяч подій за секунду без втрати якості або швидкості реагування. Використання сучасних технологій, таких як ASP.NET Core, SignalR і Angular, забезпечує стабільну та надійну роботу як серверної, так і клієнтської частини системи. Завдяки архітектурним рішенням із застосуванням фонового агрегаційного сервісу та реактивного підходу до обробки даних, платформа легко адаптується до зростаючих навантажень і дозволяє підтримувати безперервне оновлення аналітики в режимі реального часу.

Однією з ключових властивостей системи є гнучкість та зручність користування. Інтуїтивно зрозумілий інтерфейс, багатомовна підтримка та інтерактивна візуалізація даних сприяють покращенню користувацького досвіду. Користувачі мають змогу швидко отримувати актуальну аналітичну інформацію через графіки, діаграми та інші візуальні інструменти, що допомагають приймати оперативні бізнес-рішення. Крім того, система підтримує багаторівневу авторизацію та керування користувачами, що забезпечує безпеку і персоналізацію доступу відповідно до ролей.

Важливим аспектом є інтеграція з різними зовнішніми джерелами даних, такими як маркетплейси та криптовалютні біржі, що дозволяє створювати комплексний та об’єктивний огляд ринку в режимі реального часу. Завдяки регулярній агрегації, обробці і збереженню історичних даних, система забезпечує аналітику не лише по поточних, а й по минулих періодах, що значно підвищує якість прогнозування і стратегічного планування. Також реалізовані механізми швидкого оновлення мовних ресурсів і гнучкого управління, що дозволяє підтримувати актуальність інформації та зручність роботи для користувачів із різних регіонів.

3.3 Атрибути програмного продукту

3.3.1 Надійність

Система забезпечує стабільну роботу навіть при високих навантаженнях та великій кількості одночасних запитів. Завдяки використанню перевірених технологій і правильній архітектурі з поділом обробки на фонові сервіси та основний API, знижено ризик збоїв та втрати даних. Механізми обробки помилок і логування дозволяють оперативно виявляти та усувати несправності, підтримуючи безперервність роботи системи.

3.3.2 Доступність

Система розроблена як веб-застосунок, що забезпечує користувачам доступ з будь-яких сучасних браузерів та пристроїв із підключенням до Інтернету. Впроваджені механізми кешування, повторних спроб підключення та відновлення сесій гарантують, що користувачі можуть безперебійно працювати з платформою навіть за умов нестабільного мережевого з’єднання. Платформа підтримує багатомовний інтерфейс, що підвищує її доступність для широкої аудиторії.

3.3.3 Безпека

Система реалізує багаторівневі механізми захисту, включаючи автентифікацію та авторизацію користувачів, захист від несанкціонованого доступу та шифрування переданих даних. Впроваджуються сучасні стандарти безпеки, такі як HTTPS, JWT-токени та політики контролю доступу на рівні API. Регулярні оновлення та моніторинг системи дозволяють швидко реагувати на потенційні загрози і забезпечують конфіденційність інформації.

3.3.4 Супроводжуваність

Кодова база системи організована відповідно до кращих практик розробки, що полегшує її розширення, модифікацію та тестування. Використання популярних фреймворків та бібліотек із широкою спільнотою підтримки гарантує наявність необхідних ресурсів для технічної підтримки. Документування архітектури, API та процесів розробки дозволяє швидко інтегрувати нових розробників та забезпечує стабільність у подальшій експлуатації.

3.3.5. Переносимість

Програмна система реалізована з урахуванням кросплатформенності, що дозволяє запускати серверну частину на різних операційних системах, підтримуючи .NET Core. Клієнтська частина є веб-додатком, доступним у будь-якому сучасному браузері незалежно від пристрою користувача. Такий підхід дозволяє легко розгортати систему у різних середовищах та адаптувати під потреби бізнесу без значних витрат на доопрацювання.

3.3.6 Продуктивність

Система оптимізована для обробки великих обсягів даних з мінімальною затримкою, що досягається використанням реактивних технологій, таких як SignalR та RxJS, а також фонових сервісів для агрегації та обробки інформації. Архітектура підтримує масштабування як вертикальне, так і горизонтальне, що дозволяє ефективно розподіляти навантаження та забезпечувати швидкий відгук системи навіть при значній кількості користувачів і запитів.

3.4 Вимоги до бази даних

База даних відповідає за збереження, організацію та швидкий доступ до великого обсягу аналітичних даних, інформації про користувачів, товари, криптовалютні ціни та інші сутності. Вона має забезпечувати високу продуктивність та надійність, враховуючи особливості роботи з даними у реальному часі та необхідність масштабування.

Основні вимоги до бази даних включають:

* підтримка масштабованості — база даних повинна ефективно працювати при збільшенні обсягів даних та зростанні кількості одночасних запитів, підтримувати горизонтальне або вертикальне масштабування без втрати продуктивності;
* висока швидкість запису та читання — оскільки система працює з потоковими даними, важливо забезпечити швидке записування великої кількості подій та одночасний оперативний доступ до агрегованих результатів;
* надійність та цілісність даних — система повинна гарантувати збереження інформації без втрат і корупції, підтримувати механізми резервного копіювання та відновлення;
* підтримка складних запитів і аналітики — база даних має дозволяти виконувати складні агрегатні запити для формування статистики, аналітичних звітів та візуалізацій;
* безпека даних — реалізація контролю доступу на рівні таблиць і рядків, захист від несанкціонованого доступу, шифрування конфіденційної інформації;
* сумісність із застосовуваними технологіями — база даних має бути сумісною з Entity Framework Core для серверної частини, а також підтримувати інтеграцію з фоновими сервісами та API.

Для виконання цих вимог система використовує комбінацію реляційної бази даних для зберігання структурованих даних та NoSQL документо-орієнтованої бази для гнучкого зберігання історичних та потокових даних. Такий підхід дозволяє максимально ефективно обробляти різноманітні типи інформації, оптимізуючи продуктивність і масштабованість.