

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційних радіотехнологій та технічного захисту інформації
(повна назва)

Кафедра медіаінженерії та інформаційних радіоелектронних систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)
(позначення документа)

Розробка мультимедійного симулятора торгівлі з синхронізацією з біржовими даними.
(тема)

Виконав:
студент 2 курсу, групи СТМм-22-1
Ковальов Т.С.
(прізвище, ініціали)

Спеціальність 171 Електроніка
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи, технології і комп'ютерні засоби мультимедіа
(повна назва освітньої програми)

Керівник доц. Роман ЦЕХМІСТРО
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Володимир КАРТАШОВ
(прізвище, ініціали)

2023 р

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій та технічного захисту інформації
Кафедра Медіаінженерії та інформаційних радіоелектронних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 171 Електроніка
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма "Системи, технології і комп'ютерні засоби мультимедіа"

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Ковальову Тимурі Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка мультимедійного симулятора торгівлі з синхронізацією з біржовими даними.

затверджена наказом по університету від " 20 " 11 2023 р. № 1371СТ.

2. Термін подання студентом роботи 10.01.2024 р.

3. Вихідні дані до проекту (роботи) _____

1. Виконати аналітичний огляд основних методів симуляції торгівлі, детально розглянути їх переваги та обмеження.

2. Проаналізувати математичні аспекти кожного методу симуляції для повного розуміння їх роботи.

3. Розкрити теоретичні основи симуляції, звертаючи особливу увагу на аспекти побудови моделей та взаємодії ринкових умов.

4. Порівняти різні методи симуляції торгівлі та їхній вплив на аналіз фінансових стратегій.

5. Провести практичні експерименти, включаючи застосування обраних методів симуляції в реальних умовах ринку.

4. Перелік питань, що потрібно опрацювати в роботі
ВСТУП

1. АНАЛІЗ ЗАВДАННЯ

2. ТЕОРЕТИЧНІ ОСНОВИ

3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ

ВИСНОВКИ

ПЕРЕЛІК ДжЕРЕЛ ПОСИЛАННЯ

ДОДАТКИ


5. Перелік графічного матеріалу із зазначенням обов'язкових креслеників, схем, плакатів, комп'ютерних ілюстрацій:


1. Інтерфейс симулятора торгівлі (1 аркуш А4); 2. Приклад проходження опитування (1 аркуш А4); 3. Постановка задачі (1 аркуш А4); 4. Вступ (1 аркуш А4); 5. Структурна схема аналізу потреб та вимог користувачів (1 аркуш А4); 6. Структурна схема реалізації симулятора торгівлі (1 аркуш А4); 7. Код JavaScript виведення результатів транзакцій (1 аркуш А4); 8. Код JavaScript виконання оновлення «портфеля» (1 аркуш А4); 9. Код JavaScript виконання функцій купівлі та продажу (1 аркуш А4); 10. Код JavaScript для синхронізації з біржовими даними (1 аркуш А4); 11. Видяг кнопки інтерфейсу персоналізованого опитувальника (1 аркуш А4); 12. Структурний код побудови логіки та результатів опитування (1 аркуш А4); 13. Висновки (1 аркуш А4).

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Огляд програмного забезпечення	10.9-14.9.23	
2	Огляд розробки додатків	15.9-21.9.23	
3	Виділення основних моментів для простого та легкого процесу створення додатків	22.9-15.10.23	
4	Формування основних етапів розробки	26.10-1.11.23	
5	Перевірка керівником проекту	02.11-6.11.23	
6	Перевірка нормоконтролем	07.12-12.23	
7	Перевірка зав. кафедрою, рецензування	15.12-10.01.24	

Дата видачі завдання 20.11.2023

Студент  Тимур КОВАЛЬОВ
(підпис)

Керівник роботи  доц. Роман ЦЕХМІСТРО
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 52 сторінок, 33 рисунка, 14 джерел.

РОЗРОБКА МУЛЬТИМЕДІЙНОГО СИМУЛЯТОРА ТОРГІВЛІ З СИНХРОНІЗАЦІЄЮ З БІРЖОВИМИ ДАНИМИ

Об'єкт дослідження – методи створення мультимедійного симулятора торгівлі фінансовими активами з синхронізацією з біржовими даними.

Мета роботи – дослідити різні методи розробки мультимедійного симулятора торгівлі, зосереджуючись на можливостях створення інноваційного навчального інструмента для трейдерів. Робота передбачає вивчення теоретичних основ створення симулятора, проведення математичних аналізів та експериментів для визначення його можливостей та обмежень, а також розробку функціональності симулятора для синхронізації з біржовими даними. Результатом роботи буде створення мультимедійного симулятора торгівлі, який дозволить користувачам проводити повноцінну торговельну діяльність в реальному часі.

У роботі проведено аналіз технічних вимог до симулятора та вивчено потреби користувачів в такому інструменті. Досліджено технічні аспекти симуляції торгівлі та методи синхронізації з біржовими даними. Також розглянуто технічні основи підбору індивідуального стилю торгівлі для користувачів симулятора.

ABSTRACT

Explanatory note to the qualification work: 52 pages, 33 figures, 14 sources.

DEVELOPMENT OF A MULTIMEDIA TRADING SIMULATOR WITH SYNCHRONIZATION WITH EXCHANGE DATA

The object of research is methods of creating a multimedia simulator for trading financial assets with synchronization with exchange data.

The purpose of the study is to investigate various methods of developing a multimedia trading simulator, focusing on the possibilities of creating an innovative educational tool for traders. The work involves studying the theoretical foundations of creating a simulator, conducting mathematical analyses and experiments to determine its capabilities and limitations, and developing the functionality of the simulator to synchronize with stock exchange data. The result of the work will be the creation of a multimedia trading simulator that will allow users to conduct full-fledged trading activities in real time.

The paper analyzes the technical requirements for the simulator and studies the needs of users for such a tool. The technical aspects of trading simulation and methods of synchronization with exchange data are investigated. The technical basis for selecting an individual trading style for simulator users is also considered.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
1 АНАЛІЗ ЗАВДАННЯ.....	10
1.1 Визначення технічних вимог до симулятора.....	10
1.2 Аналіз потреб користувачів і можливостей розробки.....	11
1.3 Аналіз завдання	13
1.4 Графічне відображення ринкових тенденцій	14
2 ТЕОРЕТИЧНІ ОСНОВИ	16
2.1 Технічні аспекти симуляції торгівлі.....	16
2.2 Методи синхронізації з біржовими даними.....	17
2.3 Технічні основи підбору індивідуального стилю торгівлі.....	20
3 РОЗРОБКА ТА ВПРОВАДЖЕННЯ	25
3.1 Реалізація мультимедійного симулятора торгівлі	25
3.2 Синхронізація з біржовими даними та технічні рішення.....	33
3.3 Розробка персоналізованого опитувальника	36
ВИСНОВКИ.....	41
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	42
ДОДАТКИ.....	44
ДОДАТОК А. Графічний матеріал.....	45
ДОДАТОК Б. Відомість кваліфікаційної роботи.....	47

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API - Інтерфейс програмування застосунків (Application Programming Interface).

CSS - Каскадні таблиці стилів (Cascading Style Sheets).

HTML - Мова розмітки гіпертексту (Hypertext Markup Language).

JS - JavaScript - мова програмування.

SCSS - Розширений синтаксис каскадних таблиць стилів (Sass Cascading Style Sheets).

UI - Інтерфейс користувача (User Interface).

UX - Взаємодія користувача (User Experience).

ROI - Прибуток від інвестицій (Return on Investment).

БД - База даних.

GUI - Графічний інтерфейс користувача (Graphical User Interface).

Портфель — набір реальних або фінансових інвестицій. Сукупність цінних паперів різного виду.

МПС - Мультиагентні системи (Multi-Agent Systems).

ООП - Об'єктно-орієнтоване програмування.

Фронтенд - Клієнтська частина веб-додатка.

Бекенд - Серверна частина веб-додатка.

Трейдинг - це фінансова діяльність, що включає стратегічну купівлю та продаж активів з метою вигідного використання коливань вартості ринку.

AJAX - Асинхронний JavaScript і XML.

JSON - Об'єктна нотація JavaScript (JavaScript Object Notation).

SQL - Мова структурованих запитів (Structured Query Language).

DOM (Об'єктна Модель Документу) - це подання веб-сторінки у вигляді дерева об'єктів для забезпечення зручної взаємодії та змін веб-елементів через програму.

ВСТУП

У сучасному світі ринку фінансових активів та інвестицій, доступ до якого стає все більше демократичним і широким завдяки розвитку інтернет-технологій, розробка і впровадження інноваційних навчальних інструментів стає актуальним завданням. Мультимедійні симулятори торгівлі в реальному часі стають важливими інструментами для освоєння навичок торгівлі та розуміння фінансових ринків.

Особливості симулятора полягають у тому, що він надає можливість практикувати торгівлю на реальних ринках з використанням віртуальних коштів. Симулятор дозволяє користувачам торгувати, ніби на біржі, і аналізувати ринок в режимі реального часу. Крім того, в ньому існує можливість створення графіків і аналізу ринку, а також ведення докладної історії за кожною угодою.

Однією з особливостей симулятора є його доступність: він доступний у веб-версії і може використовуватися на якому завгодно пристрої, будь то комп'ютер чи мобільний телефон. Також важливою є можливість персоналізації стилю торгівлі завдяки використанню опросника, що допомагає користувачам знайти оптимальний підхід до торгівлі.

Однією з ключових переваг цього симулятора є його безкоштовність та можливість торгувати в реальному режимі з необмеженими віртуальними коштами. В порівнянні з платними аналогами, симулятор надає доступ до всіх необхідних функцій без витрат на оплату.

Важливим аспектом є також детальна історія торгівлі, що фіксує кожну угоду користувача та надає можливість аналізу та вдосконалення їхньої стратегії.

Ця дипломна робота присвячена докладному розгляду розробки симулятора та його технічним аспектам, теоретичним основам та порівнянню з існуючими рішеннями на ринку. У практичній частині буде розглянуто реалізацію симулятора та можливості його використання.

1 АНАЛІЗ ЗАВДАННЯ

1.1 Визначення технічних вимог до симулятора

Для успішного розвитку навчання з інвестицій та трейдингу необхідно створити навчальний симулятор, який би надавав студентам можливість отримати практичні навички у сфері фінансових операцій та приймати обґрунтовані фінансові рішення. Для цього першим кроком є визначення технічних вимог до цього симулятора.

Функціональність симулятора - забезпечення можливості віртуально проводити інвестиційні операції на різних фінансових ринках, включаючи акційні, валютні та товарні ринки. Симулятор має надавати можливість аналізу фінансової інформації, визначення ризиків та прийняття обґрунтованих інвестиційних рішень.

Інтерактивність - симулятор повинен бути інтерактивним та забезпечувати можливість взаємодії студентів з фінансовими інструментами та ринками. Можливість відстежувати динаміку ринків у реальному часі та віртуально управляти портфелем активів є ключовою складовою.

Модульність - симулятор повинен бути модульним, щоб забезпечити можливість додавання нових функцій та ринків у майбутньому. Модульність також дозволить вдосконалювати та розширювати функціональність симулятора з плином часу.

Безпека та конфіденційність - з огляду на обробку фінансової інформації, симулятор повинен бути забезпечений високим рівнем безпеки та конфіденційності даних. Заходи забезпечення інформаційної безпеки повинні бути обов'язковими.

Можливість моніторингу та оцінки- симулятор має надавати можливість моніторингу та оцінки результатів торгівлі, щоб студенти могли аналізувати свої рішення та вдосконалювати свої навички.

Можливість інтеграції - симулятор повинен бути легко інтегрованим з існуючими навчальними платформами та системами управління навчанням для зручного користування студентами та викладачами.

Визначення технічних вимог до симулятора є кроком у напрямку створення ефективного та інноваційного засобу навчання, який допоможе студентам розвивати свої навички в сфері інвестицій та трейдингу.

1.2 Аналіз потреб користувачів і можливостей розробки

Для більш детального аналізу потреб користувачів і можливостей розробки, необхідно провести наступні дослідження та збір інформації:

- вивчення специфіки робочого процесу користувачів. Необхідно детально розглянути, яким чином користувачі взаємодіють з сучасними рішеннями та які саме функції і можливості вони очікують від нового симулятора;

- оцінка потреб користувачів щодо рівня реалістичності та функціональності симулятора. Це допоможе визначити, наскільки важливо реалізувати різні аспекти симуляції, такі як фізичні властивості та інші функціональні можливості;

- аналіз ринку конкурентів і існуючих рішень. Необхідно ретельно дослідити сучасний ринок симуляторів, визначити їхні переваги та недоліки, а також можливості для вдосконалення;

- вивчення технічних обмежень та можливостей розробки. Необхідно оцінити доступні технічні ресурси та можливості для розробки симулятора, включаючи обладнання, програмне забезпечення та інші фактори.

Цей аналіз допоможе зрозуміти, які конкретні функції та можливості повинен мати симулятор, щоб задовольнити потреби користувачів та визначити, які технічні ресурси потрібні для розробки.

Поруч з аналізом специфіки робочого процесу та потреб користувачів, також зосереджуємо увагу на оцінці технічних можливостей для розробки симулятора. Важливо визначити, які технічні ресурси доступні для створення програмного продукту та які технології можуть бути використані.

Цей аналіз дозволить нам підібрати оптимальні технічні рішення та зрозуміти, які обмеження можуть вплинути на реалізацію деяких функцій симулятора. Також важливо враховувати можливості майбутнього розвитку та підтримки програмного продукту.

Також ретельно аналізуємо ринок конкурентів та існуючих рішень, щоб зрозуміти переваги та недоліки існуючих симуляторів та знайти можливості для покращення та вдосконалення нашого продукту.

Для більш детального аналізу потреб користувачів і можливостей розробки, провели наступні дослідження та зібрали важливу інформацію:

1) Вивчення специфіки робочого процесу користувачів: В ході аналізу детально розглянули, яким чином користувачі взаємодіють з сучасними рішеннями та які функції та можливості вони очікують від нового симулятора. Цей аналіз дозволив нам легше розуміти робочі процеси користувачів;

2) Оцінка потреб користувачів щодо рівня реалістичності та функціональності симулятора: провели опитування користувачів, які вже використовують схожі симулятори, для визначення того, наскільки важливо для них реалізувати різні аспекти симуляції, включаючи фізичні властивості та інші функціональні можливості;

3) Аналіз ринку конкурентів і існуючих рішень: ретельно дослідили сучасний ринок симуляторів, визначивши переваги та недоліки

конкуруючих продуктів. Цей аналіз надає можливість виявити можливості для покращення та вдосконалення нашого симулятора;

4) Вивчення технічних обмежень та можливостей розробки: оцінили доступні технічні ресурси та можливості для розробки симулятора, включаючи обладнання, програмне забезпечення та інші фактори. Цей аналіз допомагає нам визначити, які технічні рішення можуть бути використані та які обмеження вплинуть на реалізацію функцій симулятора.

Ця інформація стане основою для подальшого розроблення симулятора, спрямованого на задоволення потреб користувачів та максимально ефективного використання технічних ресурсів.

1.3 Аналіз завдання

Визначення обсягу проекту: обсяг завдання чітко визначений і включає в себе реалізацію мультимедійного симулятора торгівлі, що передбачає розробку інтерфейсу для торгівлі акціями та обліку фінансових операцій. Крім того, проект включає синхронізацію симулятора з реальними біржовими даними та розробку персоналізованого опитувальника для користувачів.

Аналіз термінів та ресурсів: проведений аналіз термінів виконання проекту та визначені необхідні ресурси, включаючи написання коду, дизайн і серверні ресурси.

Визначення ключових завдань: основні завдання та цілі проекту включають в себе розробку інтерфейсу симулятора, реалізацію системи збору та обробки реальних біржових даних, інтеграцію з існуючими фінансовими інструментами, а також розробку інтерактивного опитувальника для користувачів.

Оцінка ризиків та можливостей: проведений аналіз можливих ризиків, включаючи зміни у фінансовому законодавстві, технічні проблеми з синхронізацією даних та конкуренцію на ринку симуляторів. Одночасно визначені можливості, такі як партнерство з фінансовими компаніями та розширення функціоналу симулятора.

Визначення критеріїв успіху: визначені конкретні критерії успіху, такі як позитивний фідбек від користувачів, вчасне впровадження проекту та зростання популярності серед трейдерів.

1.4 Графічне відображення ринкових тенденцій

Важливою частиною успішного трейдингу є розуміння та аналіз ринкових тенденцій. Симулятор надає користувачам можливість відстежувати динаміку цін, обсягу угод та інших ключових параметрів за допомогою графіків. Тут ми розглянемо два типи графіків (рис. 1.1), які надають користувачам інструменти для глибшого аналізу ринкових рухів.



Рисунок 1.1 - Графік цін на криптовалюту

Графік цін на криптовалюту надає візуальне представлення динаміки змін цін певного активу протягом часу (рис. 1.2). Користувачі можуть вибирати різні таймфрейми (наприклад, 1 година, 1 день, 1 тиждень) для отримання різних перспектив на ринкові рухи.

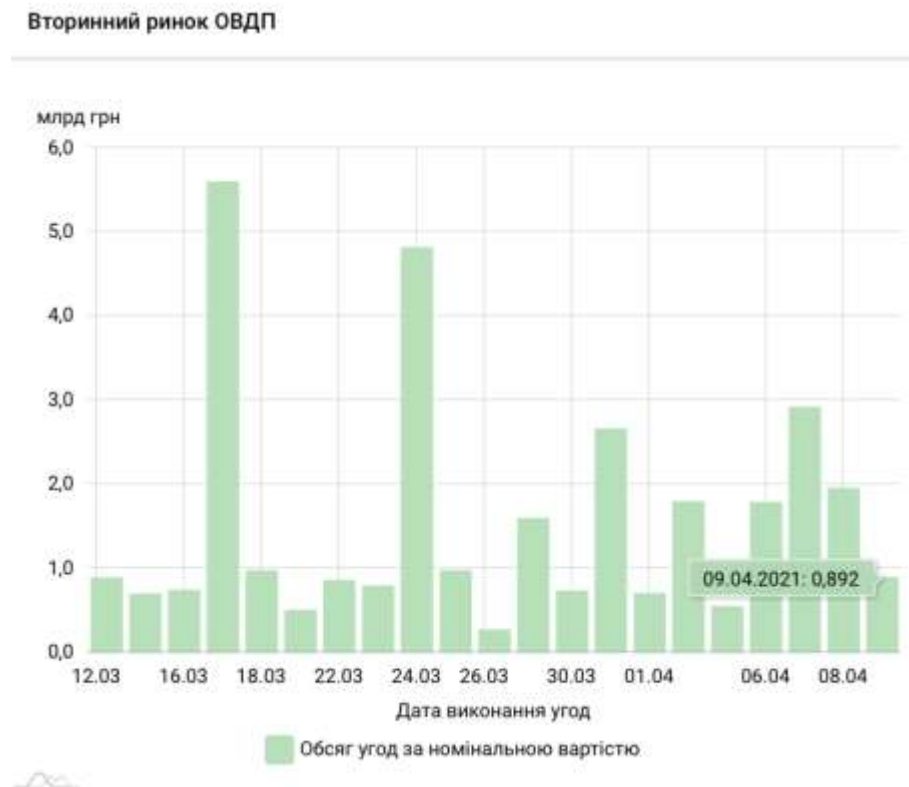


Рисунок 1.2 - Графік обсягу угод на ринку

Графік обсягу угод надає інформацію про кількість активу, яке було куплено чи продано протягом певного періоду. Цей показник може бути корисним для визначення сили або слабкості поточного тренду.

В розділі "Аналіз завдання" детально розглянуто вимоги та цілі, пов'язані із створенням інтерактивного симулятора для інвестицій та трейдингу. Висновок доповнює цей аналіз, зазначаючи, що графіки є важливою частиною інструментарію симулятора. Вони відіграють ключову роль у сприйнятті ринкових тенденцій трейдерами та

допомагають їм приймати обґрунтовані рішення. Ця взаємодія з графіками є необхідною для навчання та вдосконалення навичок трейдингу.

2 ТЕОРЕТИЧНІ ОСНОВИ

2.1 Технічні аспекти симуляції торгівлі

Симуляція торгівлі є складним та багатогранним завданням, яке вимагає вивчення та розуміння різних технічних аспектів. У цьому підрозділі розглянемо ключові аспекти, які впливають на технічну реалізацію симулятора торгівлі.

Один із важливих аспектів симуляції торгівлі - це обробка реальних біржових даних. Для отримання актуальних даних про ціни активів, обсяги торгів та іншу інформацію, плануємо використовувати API бірж та фінансових даних. Це дозволить нам отримувати найсвіжішу інформацію, яка необхідна для точної симуляції торгівлі.

Торгівельний інтерфейс симулятора грає ключову роль у взаємодії користувача з програмою. Планується використовувати сучасні технології для розробки інтуїтивно зрозумілого і зручного інтерфейсу. Він має надавати користувачам змогу виконувати торгові операції, аналізувати ринки та взаємодіяти з іншими елементами симулятора.

Основною частиною симуляції торгівлі є реалізація алгоритмів, які визначають поведінку симулятора та імітують ринкову динаміку[1]. плануємо використовувати різні алгоритми для моделювання торговельних операцій та розрахунку результатів. Це включає в себе алгоритми обробки заявок, стратегії торгівлі та аналізу ринкових умов.

Для забезпечення користувачів актуальною інформацією та інструментами для аналізу ринків, плануємо інтегрувати симулятор з

існуючими фінансовими інструментами. Це дозволить користувачам отримувати доступ до різноманітних аналітичних засобів та ресурсів для прийняття обґрунтованих торгових рішень.

Розробка персоналізованого опитувальника є важливою частиною симулятора, є можливість збирання відгуків користувачів та аналізу їхньої активності. Планується розробити персоналізований опитувальник, який дозволить користувачам надавати свої коментарі та відгуки щодо роботи симулятора [2].

Важливим етапом у симуляції торгівлі є аналіз та обробка біржових даних. У планах використовувати різні джерела для отримання інформації про ціни активів, обсяги торгів та інші параметри. Окрім API бірж, розглядаємо можливість використання технічних індикаторів для отримання додаткових даних та покращення якості симуляції.

Торгівельний інтерфейс визначає зручність та ефективність взаємодії користувача з програмою [3].

Інтеграція з існуючими фінансовими інструментами отримає новий вимір завдяки вдосконаленню API та підтримці різноманітних фінансових платформ. Це розширить можливості користувачів у вивченні та аналізі ринків.

На завершення, розглядаємо можливість розширення персоналізованого опитувальника, додавши до нього додаткові елементи для кращого розуміння потреб користувачів і вибору оптимального стилю торгівлі. Цей апгрейд буде спрямований на надання користувачам індивідуальних рекомендацій та повного пояснення їхнього вибору.

Ці доповнення враховують сучасні тенденції та технології, що сприятиме створенню більш інноваційного та ефективного симулятора торгівлі.

2.2 Методи синхронізації з біржовими даними

Реалізація ефективної синхронізації з біржовими даними є ключовим етапом у розробці симулятора торгівлі. Правильно налагоджена синхронізація дозволить отримувати та обробляти актуальну інформацію для забезпечення реалістичності та точності торгівельного процесу.

Методи отримання ринкових даних здійснюються з використанням відкритих API (рис. 2.1) бірж для отримання реальних даних є однією з ключових стратегій синхронізації. Детально налаштовані запити до біржових серверів дозволяють отримувати актуальні ціни, обсяги та інші важливі параметри для використання в симуляторі.

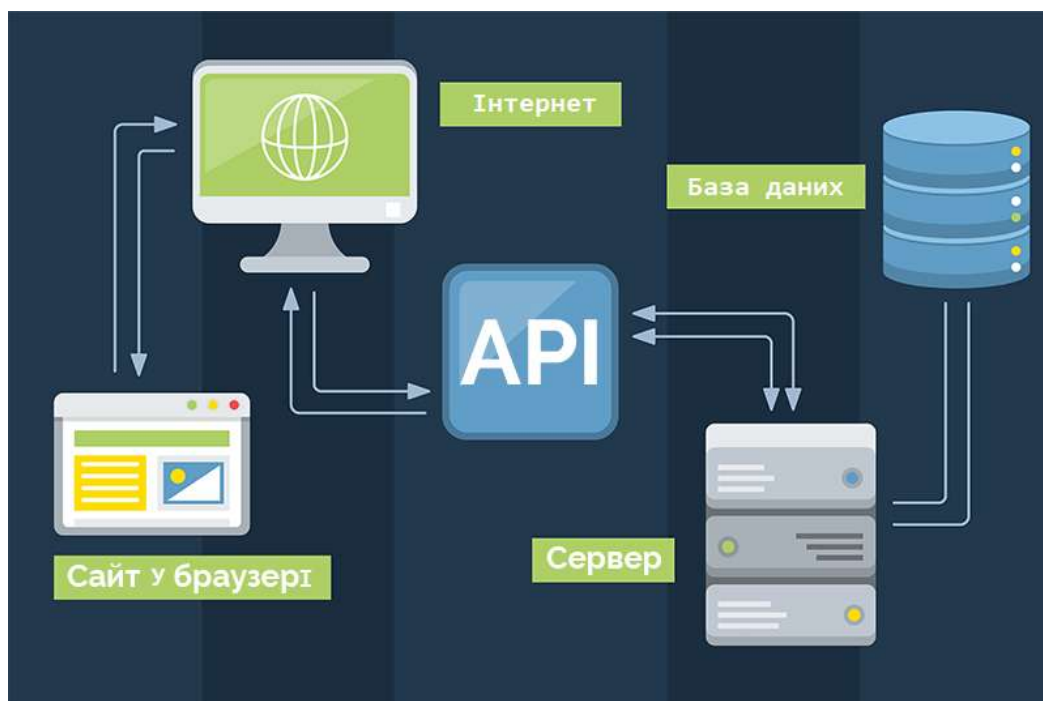


Рисунок 2.1 - Графічне представлення системи API

Відновлення в разі втрати з'єднання:

- механізми автоматичного відновлення даних у випадку втрати з'єднання чи технічних проблем забезпечать неперервну роботу симулятора. Це важливо для уникнення втрати важливої інформації та забезпечення стабільної роботи.

Система кешування даних (рис. 2.2) дозволяє зменшити завантаження на сервери бірж та прискорити обробку даних в симуляторі. Ефективне кешування дозволить швидше відгукувати на запити користувачів [4].



Рисунок 2.2 - Графічне представлення системи кешування даних

Оптимізація мережевого трафіку складається з методів оптимізації передачі даних. Це допоможе зменшити витрати мережевого трафіку, забезпечуючи економічне використання ресурсів та високу швидкість синхронізації.

Захист від даних з боку серверів бірж: Розробка механізмів захисту даних від потенційної атаки або несанкціонованого доступу гарантує конфіденційність та цілісність отриманих ринкових інформацій.

Контроль якості даних: Введення системи контролю якості даних дозволить виявляти та коригувати можливі помилки у вихідних даних, що забезпечить точність результатів та реалістичність симуляції.

Мінімізація затримок у синхронізації: Використання оптимізованих алгоритмів передачі даних та мережевих протоколів дозволить зменшити затримки у процесі синхронізації. Це важливо для забезпечення оперативності та плавності роботи симулятора.

Система синхронізації повинна бути гнучкою та адаптивною до можливих змін у специфікаціях біржових API. Це дозволить швидко впроваджувати зміни та підтримувати актуальність симуляції.

Розробка безпечних механізмів взаємодії з API бірж дозволить уникнути можливих загроз безпеці та незаконного доступу до фінансових даних.

Вдосконалення синхронізації з біржовими даними включатиме кілька ключових напрямків оптимізації. Одним із аспектів є використання алгоритмів компресії та кешування для зменшення обсягу передаваних даних та підвищення ефективності використання ресурсів. Це допоможе підтримувати стабільну роботу симулятора навіть при обмежених ресурсах та швидко реагувати на зміни на ринку.

Додатково, важливим елементом є впровадження механізмів автоматичного визначення та усунення конфліктів при синхронізації. Це гарантує консистентність даних та попереджає суперечки у відтворенні ринкової динаміки, забезпечуючи точність та достовірність інформації.

Для підвищення користувацького досвіду, планується введення графічного інтерфейсу, що відображатиме статус синхронізації. Це дозволить користувачам візуально спостерігати за процесом взаємодії з біржовими даними та забезпечить зручний моніторинг [5].

Ці методи синхронізації спрямовані на досягнення оптимальної ефективності та точності симуляції торгівлі, забезпечуючи високий рівень реалізму у віртуальному торговельному середовищі.

2.3 Технічні основи підбору індивідуального стилю торгівлі

Розробка технічних рішень для ефективного індивідуального підбору стилю торгівлі є необхідною для забезпечення гнучкості та персоналізації симулятора торгівлі. Зазначені технічні аспекти включають:

- глибокий аналіз торговельних стратегій;
- використання аналітичних інструментів для детального аналізу різноманітних торговельних стратегій, враховуючи їхню історію та результативність у різних ринкових умовах;
- розробка системи, що дозволяє користувачам налаштовувати параметри торговельних стратегій відповідно до їхніх власних вподобань та рівня прийняттого ризику;
- розробка інтерфейсу з інтуїтивно зрозумілими візуальними елементами, які відображають вплив різних стратегій на торговельні результати;
- забезпечення можливості користувачів надавати зворотний зв'язок стосовно ефективності та вдосконалення системи підбору стилю торгівлі.
- використання психометричних тестів та аналізу психологічних особливостей користувачів для підбору стратегій, що відповідають їхньому психологічному профілю;
- розробка алгоритмів для моніторингу емоційного стану користувача та адаптації стратегій торгівлі для зменшення впливу негативних емоцій на прийняття рішень;
- створення інтерактивних матеріалів, що пояснюють різні аспекти торговельних стратегій та допомагають користувачам зрозуміти їхню роботу;
- забезпечення можливості торгувати в реальних ринкових умовах, що найбільше відповідають стратегіям користувача (рис. 2.3);
- створення гнучкої системи, що дозволяє користувачам експериментувати з різними стратегіями без реального ризику втрат.

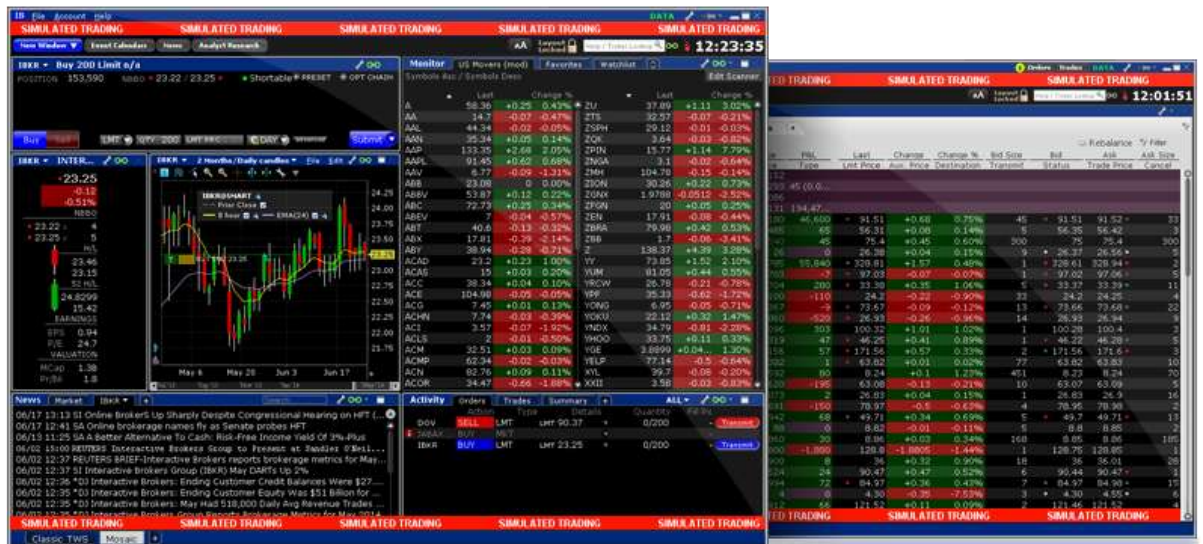


Рисунок 2.3 - Приклад інтерактивного навчального матеріалу з трейдингу.

Ці технічні рішення спрямовані на забезпечення індивідуалізації навчального процесу та підвищення ефективності навчання торгівлі, дозволяючи кожному користувачеві знайти та розвинути свій власний стиль торгівлі [6].

Ключовим елементом технічної підтримки індивідуального підбору стилю торгівлі є розробка інтелектуальної системи аналізу та рекомендацій.

Додатковою технічною складовою є впровадження інструментів візуалізації (рис. 2.4). Графіки, діаграми та інші візуальні засоби допоможуть користувачам аналізувати результати своїх торговельних рішень.

Ще однією технічною ініціативою є створення адаптивного інтерфейсу, який забезпечить зручність користування сервісом на різних платформах та пристроях. Це стане ключовою складовою для забезпечення доступності та зручності використання для широкого кола користувачів [7].

Для кращого взаємодії із системою, планується в майбутньому використання механізмів голосового керування та інші інноваційні рішення. Це створить нові можливості для користувачів і забезпечить більш ефективну комунікацію із симулятором.



Рисунок 2.4 - Візуалізація результатів торгівлі в симуляторі.

Технічний підхід до індивідуального підбору стилю торгівлі спрямований на створення адаптивного та інтелектуального сервісу, який враховує потреби та особливості кожного користувача. Це створить ефективні умови для розвитку та успішної торгівлі кожним учасником системи.

Забезпечення реалістичних умов торгівлі є ще однією важливою частиною технічного підходу [8].

Також враховується необхідність створення гнучкої системи, яка дозволить користувачам експериментувати з різними стратегіями без

реального ризику втрат. Це створить сприятливі умови для вдосконалення та самовдосконалення кожного трейдера.

В розділі "Теоретичні основи" було розглянуто ряд ключових аспектів, пов'язаних з роботою інтерфейсу програмування додатків (API) в контексті симулятора для інвестицій та трейдингу. Важливо відзначити, що API використовується для забезпечення синхронізації з біржовими даними, отримання реальних ринкових показників та реалізації real-time оновлень.

API стає критично важливим елементом для отримання актуальної інформації з фінансових ринків та ефективного управління даними для користувачів симулятора. Висновок цього розділу може вирізняти значення правильної реалізації та оптимізації використання API для забезпечення ефективності та точності симуляції фінансових операцій.

3 РОЗРОБКА ТА ВПРОВАДЖЕННЯ

3.1 Реалізація мультимедійного симулятора торгівлі

Для реалізації мультимедійного симулятора торгівлі було використано сучасні технічні засоби та програмні рішення. Даний симулятор надає користувачам можливість віртуальної торгівлі криптовалютами (рис. 3.1) в реальному часі та взаємодії з іншими елементами [9].

```

function () {
  // Stock Data
  var companies = [
    {
      name: 'Bitcoin',
      symbol: 'BTC',
      price: '$',
      shares: '$'
    },
    {
      name: 'Ethereum',
      symbol: 'ETH',
      price: '$',
      shares: '$'
    },
    {
      name: 'Bitcoin coin',
      symbol: 'BTC',
      price: '$',
      shares: '$'
    },
    {
      name: 'Doge',
      symbol: 'DOGE',
      price: '$',
      shares: '$'
    },
    {
      name: 'Polkadot',
      symbol: 'DOT',
      price: '$',
      shares: '$'
    }
  ];
  var totalProfit = 0;

  // Profit and Loss
  var profitLoss = 0;

  // transaction history
  var transactionHistory = [];

  // Limits
  var buyLimit = 0;
  var sellLimit = 0;

  // function to add a record to the trade history
  function addTradeHistory(date, coin, quantity, amountUSD, profit) {
    transactionHistory.push({
      date: date,
      coin: coin,
      quantity: quantity,
      amountUSD: amountUSD,
      profit: profit // Add profit to trade history
    });
  }

  // Update the trade history table after adding a record
  updateTradeHistory();

  // function to update the trade history on the page
  function updateTradeHistory() {
    var tbody = $('tbody');
    tbody.empty();

    for (var i = 0; i < transactionHistory.length; i++) {
      var trade = transactionHistory[i];
      var row = '<tr><td>' + trade.date + '</td><td>' + trade.coin + '</td><td>' + trade.quantity + '</td><td>' + trade.amountUSD + '</td><td>' + trade.profit + '</td></tr>';
      tbody.append(row);
    }
  }
}

```

Рисунок 3.1 – Програмний опис змінних для позначення торгівельних компаній, для відображення на сайті

Таблиці та інформація про інвестиційний “портфель”.

Симулятор включає в себе таблиці для відображення інформації про поточні акції, історію торгів та загальний баланс (рис. 3.2). Також присутній візуальний елемент, який відображає кількість грошей, витрачених на покупку та отриманих від продажів.

```

<div class="row">
  <div class="span10">
    <h3>Торгова: <span class="tooltip">!</span></h3>
    <table class="styled-table" id="myPortfolio">
      <thead>
        <tr>
          <td> Назва акції </td>
          <td> Символ </td>
          <td> Ціна </td>
          <td> Кількість якою ви володієте </td>
          <td> Купити </td>
          <td> Продати </td>
        </tr>
      </thead>
      <tbody>
      </tbody>
    </table>
  </div>
</div>

<div class="row">
  <div class="span10">
    <h3>Загальний баланс: <span id="netWorth">10000</span></h3>
    <h3>Ваша кількість грошей: <span id="cashFlow">10000</span></h3>
    <h3>Вартість криптовалюти: <span id="portfolio">8</span></h3>
    <h3>Загальний прибуток: <span id="profit">0.00$</span></h3>
  </div>
</div>

<div class="row">
  <div class="span10">
    <h3>Історія торгів:</h3>
    <table class="styled-table" id="myTradeHistory" style="margin: 0 auto;">
      <thead>
        <tr>
          <th>Дата</th>
          <th>Монета</th>
          <th>Кількість</th>
          <th>Ціна</th>
          <th>Всього в USD</th>
          <th>Прибуток</th>
        </tr>
      </thead>
      <tbody>
      </tbody>
    </table>
  </div>
</div>

```

Рисунок 3.2 - Код для виводу на сайт інформації про загальний баланс та загальний прибуток

Використання TradingView Widget – засіб для зручного відслідковування графіків, котировок та фінансових інструментів на ринку.

Для відображення графіків ринку та актуальних цін було використано TradingView Widget (рис. 3.3). Цей інтерактивний інструмент дозволяє користувачам спостерігати за динамікою цін на вибрані криптовалютні пари в режимі реального часу.

```
<!-- TradingView Widget BEGIN -->
<div class="tradingview-widget-container">
  <div id="tradingview_d902b"></div>

  <script type="text/javascript" src="https://s3.tradingview.com/tv.js"></script>
  <script type="text/javascript">
    new TradingView.widget(
    {
      "width": 950,
      "height": 510,
      "symbol": "BINANCE:BTCUSDT",
      "interval": "1",
      "timezone": "Etc/UTC",
      "theme": "light",
      "style": "1",
      "locale": "en",
      "toolbar_bg": "#f1f3f6",
      "enable_publishing": false,
      "hide_side_toolbar": false,
      "allow_symbol_change": true,
      "container_id": "tradingview_d902b"
    }
  );
  </script>
</div>
<!-- TradingView Widget END -->
```

Рисунок 3.3 - Код відображення інформації, яка надійшла з платформи технічного аналізу

Мова програмування JavaScript та процедура AJAX для взаємодії з Binance API.

Для отримання актуальних цін та оновлення інформації в реальному часі використовується JavaScript, що використовує AJAX-запит для взаємодії з Binance API (рис. 3.4). Це дозволяє симулятору отримувати та відображати найновішу інформацію з ринку [10].

```
// Function to get the current stock price using the Binance API
function getStockPrice(symbol) {
    var apiUrl = 'https://api.binance.com/api/v3/ticker/price?symbol=' + symbol + 'USDT';

    $.get(apiUrl, function(response) {
        var price = parseFloat(response.price);
        updateStockPrice(symbol, price);
    });
}
```

Рисунок 3.4 – Приклад використання мови програмування JavaScript для взаємодії з Binance API

Симулятор дозволяє користувачам взаємодіяти з базовим функціоналом, таким як купівля та продаж акцій. Кнопки "Купити" (рис 3.5) та "Продати" в таблиці акцій дозволяють виконувати відповідні операції. Кожна торгова операція реєструється в історії торгів для зручності відстеження.

```

if (id.indexOf('Buy') > 0) {
    symbol = id.substr(0, id.indexOf('Buy'));
    quantity = parseFloatLocale(prompt("Укажите количество акций, которое вы хотите купить:", '1'));
    if (!isNaN(quantity)) {
        var thisObj = 0;

        for (var key in companies) {
            var obj = companies[key];

            for (var prop in obj) {
                if (prop === 'symbol') {
                    if (obj[prop] === symbol) {
                        thisObj = key;
                    }
                }

                if (prop === 'price') {
                    if (key === thisObj) {
                        PriceForThisStock = parseFloat(obj[prop]);

                        if (cashflow > PriceForThisStock * quantity) {
                            var subtractPrice = true;
                            totalAmount = PriceForThisStock * quantity;
                            cashflow = cashflow - totalAmount - (totalAmount * 0.001);
                            calculateProfitLoss(symbol, quantity, PriceForThisStock, 'buy');
                        } else {
                            alert('У вас недостаточно средств для покупки указанного количества акций.');
```

Рисунок 3.5 – Код мови програмування JavaScript для обробки подій нажаття на кнопку «Купити»

Відображення історії торгів здійснюється у вигляді таблиці (рис. 3.6), яка містить інформацію про дату, акцію, кількість, ціну, загальну вартість та прибуток чи збиток для кожної торгової операції.

```

<div class="row">
  <div class="span10">
    <h3>Історія торгів:</h3>

    <table class="styled-table" id="myTradeHistory" style="margin: 0 auto;">
      <thead>
        <tr>
          <th>Дата</th>
          <th>Монета</th>
          <th>Кількість</th>
          <th>Ціна</th>
          <th>Всього в USD</th>
          <th>Прибуток</th>
        </tr>
      </thead>
      <tbody>
      </tbody>
    </table>
  </div>
</div>

```

```

// Function to update the trade history on the page
function updateTradeHistory() {
  var tableBody = $('#myTradeHistory tbody');
  tableBody.empty();

  for (var i = 0; i < transactionHistory.length; i++) {
    var trade = transactionHistory[i];
    var row = '<tr><td>' + trade.date + '</td><td>' + trade.coin + '</td><td>' + trade.quantity + '</td><td>' + trade.amountInUSD + '</td><td>' + trade.profit + '</td></tr>';
    tableBody.append(row);
  }
}

```

Рисунок 3.6 – Код побудови таблиці для відображення історії торгів

Для забезпечення актуальності інформації про ціни та стан портфеля дані оновлюються автоматично через певний інтервал часу. Це реалізовано за допомогою JavaScript і функції `setInterval` [11].

Симулятор автоматично оновлює ціни акцій у реальному часі та відображає їх на інтерфейсі. Також відбувається перерахунок загальної вартості портфеля (рис. 3.7) та здійснення операцій у випадку достатнього маржинального балансу.

Загальний баланс: 10003.512709\$

Ваша кількість грошей: 8247.531709\$

Вартість криптовалют: 1755.981\$

Загальний прибуток: 0.00\$

Історія торгів:

Дата	Монета	Кількість	Ціна	Всього в USD	Прибуток
18.11.2023, 17:18:51	ETH	1	\$1944.81	\$1944.81	
18.11.2023, 17:18:53	ETH	0.1	\$1944.81	\$194.48	

Рисунок 3.7 - Демонстрація Перерахунку вартості балансу

Симулятор веде облік усіх активів користувача, включаючи назву акції, символ, ціну, кількість та можливість купівлі або продажу (рис. 3.8). Це забезпечує повний контроль над фінансовим станом та історією операцій.

Назва акції	Символ	Ціна	Кількість якою ви володієте	Купити	Продати
Bitcoin	BTC	36610 \$ ▼	0	Buy	
Ethereum	ETH	1949.01 \$ ▼	0.9	Buy	Sell
Binance coin	BNB	244.1 \$ ▼	0	Buy	
Doge	DOGE	0.08026 \$ ▼	0	Buy	
Polkadot	DOT	5.214 \$ ▼	0	Buy	

Рисунок 3.8 - Таблиця для відображення активів та операцій

Симулятор автоматично розраховує та відображає загальний прибуток чи збиток користувача (рис. 3.9). Історія торгів дозволяє детально вивчати результати кожної операції.

```
<h3>Загальний прибуток: <span id="profit">0.00</span></h3>
```

```
// Function for transaction history
function addToTransactionHistory(symbol, shares, price, type, profit) {
  transactionHistory.push({
    symbol: symbol,
    shares: shares,
    price: price,
    type: type,
    profit: profit
  });
}

function calculateProfitLoss(symbol, shares, price, type) {
  var transactionAmount = shares * price;
  if (type === 'buy') {
    profitLoss -= transactionAmount;
  } else {
    for (var i = 0; i < companies.length; i++) {
      if (companies[i].symbol === symbol) {
        var sharesOwned = parseFloat(companies[i].shares);
        var profit = (shares * price) - (shares * companies[i].price);
        addToTransactionHistory(symbol, shares, price, 'sell', profit);

        // Обновляем общий прибуток после продажи
        profitLoss += profit;
        $('#profit').text(profitLoss.toFixed(2) + '$');

        // Обновляем портфель после продажи
        var portfolioValue = sharesOwned * price;
        $('#portfolio').text(portfolioValue.toFixed(2) + '$');

        break;
      }
    }
  }
}
```

Рисунок 3.9 – Код на мові Javascript, що до візуалізації показників прибутку та збитків

Реалізація мультимедійного симулятора торгівлі надає користувачам можливість не лише вивчати основи торгівлі, а й отримувати практичний досвід без реального фінансового ризику. Інтерактивний інтерфейс, автоматичне оновлення даних та зручна візуалізація історії торгів роблять симулятор ефективним засобом для вивчення інвестицій та трейдингу.

3.2 Синхронізація з біржовими даними та технічні рішення

У пункті 3.2 розглянуто синхронізацію з біржовими даними та впроваджені технічні рішення для забезпечення ефективної та надійної роботи симулятора трейдингу. Основний акцент зроблено на виборі технічних засобів, обробці даних та забезпеченні реалістичних real-time оновлень.

Для реалізації синхронізації з біржовими даними використано мову програмування JavaScript та бібліотеку jQuery. Це надає зручність в обміні даними з API біржі та забезпечує швидку взаємодію з клієнтською стороною.

Використано AJAX-запити для отримання та оновлення реальних даних про ціни, обсяги та інші ринкові параметри. Це забезпечує найсвіжішу інформацію для користувачів [12].

Створено механізм real-time оновлення даних в реальному часі, щоб користувачі могли спостерігати за ринковими подіями та приймати рішення на основі актуальної інформації. Використано таймер для періодичного оновлення цін активів.

Реалізовано систему обробки помилок та автоматичного відновлення синхронізації для забезпечення безперебійної роботи симулятора навіть при можливих виникненнях проблем.

Введено систему кешування даних для поліпшення швидкодії та зменшення навантаження на сервери біржі. Це дозволяє зберігати часто використовувані дані локально для швидкого доступу.

Впроваджено систему логування для відстеження операцій синхронізації та ідентифікації можливих проблем. Додано моніторинг для оперативної реакції на будь-які аномалії.

Застосовано графічний інтерфейс для відображення динаміки цін на криптовалюту, що створює візуальну репрезентацію ринкових тенденцій та дозволяє користувачам глибше аналізувати рухи на ринку.

На сторінці реалізовано таблицю, яка надає користувачам інформацію про їхні інвестиції та загальний стан портфеля. Це включає вартість кожного активу, кількість акцій, загальний баланс, доступну суму грошей для торгівлі та інші ключові показники.

Створено віддільну таблицю для відображення історії торгів, включаючи дату, назву акції, кількість, ціну, вартість у USD та прибуток. Це надає користувачам можливість відстежувати та аналізувати свою діяльність на ринку.

Розглянемо особливості побудови та функціоналу кнопки "Купити" та "Продати".

Мною додано функціонал для інтерактивного управління акціями. Кнопки "Купити" дозволяють користувачам придбати певну кількість акцій, а кнопки "Продати" дозволяють їх продати. Реалізована система обмежень та перевірок для запобігання некоректним торговельним операціям.

Цей функціонал дозволяє користувачам активно взаємодіяти з симулятором та отримувати практичний досвід управління інвестиціями та трейдингом. Додатково, структура інтерфейсу дозволяє зручно спостерігати за ринковою ситуацією та вчасно приймати обґрунтовані рішення.

Такий підхід до розробки симулятора сприяє покращенню навичок трейдера та глибшому розумінню ринкових тенденцій. Цей інструмент може використовуватися як для навчання новачків, так і для вдосконалення вмінь досвідчених трейдерів.

Я використовував JavaScript для взаємодії з Binance API та оновлення інформації в реальному часі. Актуальні котирування тільки так

можна витягнути, з біржі, де йдуть реальні торги, для криптовалюти я вибрав API Бінанс.

Для роботи з асинхронними запитами до сервера я використовував AJAX. Приклад оголошення запиту на сервер виглядає так:

```
var httpRequest = новий XMLHttpRequest();
httpRequest.open('GET',
'https://api.binance.com/api/v3/ticker/price?symbol=BTCUSDT', true);
httpRequest.onreadystatechange = function() {
    if (httpRequest.readyState == 4 && httpRequest.status == 200) {
        var response = JSON.parse(httpRequest.responseText);
        // Обробка відповіді сервера
    }
};
httpRequest.send();
```

Цей код надсилає GET-запит до Binance API для отримання актуальної ціни Bitcoin по відношенню до USDT.

У моєму коді використовується jQuery для полегшення маніпуляцій із DOM та виконання асинхронних запитів. Ось кілька моментів, де використовується jQuery:

Виконання асинхронних запитів із Binance API:

```
$.get(apiUrl, function(response) {
    var price = parseFloat(response.price);
    updateStockPrice(symbol, price);
});
```

Тут \$.get – це скорочена форма для виконання HTTP GET запиту. Вона зручна до виконання асинхронних запитів до API, у разі до Binance.

Маніпуляції з DOM:

```
$('#price' + symbol).html('<span style="color: ' + priceColor + '>' + price
+ ' ' + priceArrow + '</span>');
```

І тут:

```
var tableBody = $('#myTradeHistory tbody');
tableBody.empty();
```

Тут \$ - це селектор, який використовується для вибору елементів DOM за їх ідентифікатором (#) або класом (.).

Обробка подій кліка на кнопках:

```
$(document).on('click', ".my-link", function() {
    // Обробка подій при натисканні на кнопки
});
```

Це використовується для обробки подій кліку на елементах із класом .my-link. \$(document).on дозволяє обробляти події для елементів, які можуть бути додані динамічно у майбутньому.

Використання інтервалів для періодичного оновлення даних:

```
setInterval(function() {
    // Код, який виконується кожні 1000 мілісекунд (1 секунда)
}, 1000);
```

Тут setInterval використовується для періодичного виконання певного коду з інтервалом 1000 мілісекунд.

Формат JSON використовується для передачі даних між клієнтом та сервером. У моєму проекті, наприклад, відповідь від Binance API представлена у форматі JSON, який легко декодувати і використовувати JavaScript для оновлення інтерфейсу симулятора.

3.3 Розробка персоналізованого опитувальника визначення індивідуального стилю торгівлі

У рамках даного проекту був розроблений персоналізований опитувальник з метою визначення стилів торгівлі та інвестиційних

підходів користувачів. Проект базується на використанні технологій HTML, CSS та JavaScript для створення інтерактивного веб-додатку[13].

Опитувальник складається з декількох питань, що охоплюють різні аспекти інвестицій та трейдингу. Кожне питання (рис. 3.10) має кілька варіантів відповідей, з якими користувач може вибрати найбільш підходящий для себе.

```
var questions = [
  {question: "Яку суму ви готові інвестувати?", answers: [{text: "Менше 1 000$", weight: 1}, {text: "Від 1 000 до 10 000$", weight: 2}, {text: "Більше 10 000$", weight: 3}], weight: 1},
  {question: "Який рівень прибутковості ви хотіли б отримувати від інвестицій?", answers: [{text: "10%-15% на рік", weight: 3}, {text: "15%-20% на рік", weight: 2}, {text: "Більше 20% на рік", weight: 1}], weight: 2},
  {question: "Яку суму готові втратити, якщо ваші інвестиції підуть невдало?", answers: [{text: "Менш ніж 10% від початкової суми", weight: 1}, {text: "10-20% від початкової суми", weight: 2}, {text: "Більше 20% від початкової суми", weight: 3}], weight: 3},
  {question: "Як ви ставитеся до волатильності на ринку?", answers: [{text: "Готовий ризикувати для отримання більшого прибутку", weight: 1}, {text: "Не готовий ризикувати", weight: 2}], weight: 4},
  {question: "Як би ви охарактеризували свій рівень досвіду у торгівлі?", answers: [{text: "Новачок", weight: 3}, {text: "Досвідчений", weight: 2}, {text: "Професіонал", weight: 1}], weight: 5},
  {question: "Скільки часу ви готові приділяти торгівлі щодня?", answers: [{text: "Менше 1 години", weight: 3}, {text: "1-3 години", weight: 2}, {text: "Більше 3 години", weight: 1}], weight: 6},
  {question: "На якому ринку ви хотіли б торгувати?", answers: [{text: "Форекс", weight: 2}, {text: "Акції", weight: 3}, {text: "Криптовалюта", weight: 1}], weight: 7},
  {question: "Який рівень ліквідності ваших інвестицій важливий для вас?", answers: [{text: "Найменший, готов забути про вкладення", weight: 1}, {text: "Середній", weight: 2}, {text: "Високий", weight: 3}], weight: 8},
  {question: "Який інструмент аналізу ви найчастіше використовуєте?", answers: [{text: "Технічний аналіз", weight: 1}, {text: "Фундаментальний аналіз", weight: 2}, {text: "Інтуїція", weight: 3}], weight: 9},
  {question: "Яку частину свого капіталу ви готові ризикувати в одній угоді", answers: [{text: "Менше 1%", weight: 5}, {text: "1-5%", weight: 3}, {text: "Більше 5%", weight: 1}], weight: 10},
  {question: "Як ви оцінюєте свою психологічну стійкість?", answers: [{text: "Дуже високо", weight: 1}, {text: "Високо", weight: 2}, {text: "Середньо", weight: 3}, {text: "Низько", weight: 4}], weight: 11},
  {question: "Як ви зазвичай реагуєте на збиткову угоду?", answers: [{text: "Намагаюся зберегти позицію до рівня стоп-лосу", weight: 1}, {text: "Забиваю стоп-лос", weight: 2}, {text: "Продовжую торгувати", weight: 3}], weight: 12}
];
```

Рисунок 3.10 – Код змінних побудови стандартних питань для визначення особливості процедур інвестування

Під час вибору користувачем відповідей, застосовується логіка обчислення загального балу (рис. 3.11), який визначається вагою обраних відповідей. Після завершення опитування користувача виводиться рекомендація стосовно його стилю торгівлі чи інвестицій.


```

$("#nextButton").click(function() {
  var selectedAnswer = $('input[name=answer]:checked').val();
  if (selectedAnswer) {
    totalScore += parseInt(selectedAnswer);
    currentQuestion++;
    if (currentQuestion < questions.length) {
      showQuestion();
    } else {
      showResult();
      $(this).hide();
    }
  } else {
    alert("Выберите вариант ответа!");
  }
}

```

Рисунок 3.12 - Код для оновлення питань та відповідей

Дизайн опитувальника адаптований для різних розмірів екранів за допомогою медіа-запитів CSS. Це забезпечує оптимальний вигляд та функціональність як на комп'ютерах, так і на мобільних пристроях[14].

Опитувальник використовує бібліотеку jQuery для спрощення маніпуляцій з DOM та обробки подій. Це полегшує розробку та зберігає час.

Для збереження та обробки результатів опитування ви можете розглядати взаємодію з сервером за допомогою технологій, таких як AJAX або Fetch API. Це дозволить зберігати та аналізувати дані користувачів, враховуючи їхні відповіді та надавати більш індивідуалізовані рекомендації.

Важливо забезпечити захист від несанкціонованого доступу до результатів опитування та особистих даних користувачів. Застосування аутентифікації на рівні сервера дозволить забезпечити конфіденційність інформації.

Однією з важливих частин опитувальника є використання сучасних дизайнерських елементів, таких як кнопка. Дизайн кнопки використовує мову SCSS та вражає своєю красою та інтерактивністю. Кнопка має анімацію при наведенні та натисканні, що створює враження легкості та динамічності.

Кнопка створена з використанням каскадних таблиць стилів (CSS) та препроцесору SCSS. Вона має не тільки естетично приємний зовнішній вигляд, але і забезпечує інтерактивність для кращого залучення користувача. Її реалізація включає в себе використання градієнтів, анімацій та інших елементів, що підкреслюють сучасний та інноваційний характер проекту.

Цей дизайн кнопки (рис. 3.14) є не лише елементом інтерфейсу, але і складовою частиною стратегії взаємодії з користувачем, сприяючи зрозумінню та легкості використання опитувальника.



Рисунок 3.13 – Кнопка в статистиці



Рисунок 3.14 – Кнопка при взаємодії з користувачем

Усі ці аспекти допомагають у створенні ефективного та зручного інструменту для визначення стилів торгівлі та інвестування.

Висновки

У цьому дипломному проекті я намагався створити не просто навчальний інструмент, але щось, що може зацікавити та допомогти кожному, хто має бажання вивчити трейдинг. Моя мета була проста — зробити складні фінансові концепції доступними для всіх.

Розроблений симулятор торгівлі має на меті створити реалістичне середовище, де користувачі можуть навчатися без ризику втрат. Опитувальник дозволяє персоналізувати процес навчання, адаптуючи його до потреб кожного учасника.

Написання цього диплому було захоплюючим викликом, і я сподіваюся, що він стане корисним інструментом для всіх, хто хоче вивчити та освоїти світ трейдингу. Вірю, що навчання повинно бути захоплюючим, і я сподіваюся, що мій проект допоможе у цьому кожному, хто з ним зіткнеться.

Під час розробки цього проекту я стикнувся з безліччю викликів і вивчив багато нового. Від роботи з біржовими даними до створення інтерактивного інтерфейсу, кожен етап вимагав від мене зосередження та творчого підходу.

Важливо відзначити, що дипломна робота - це лише початок. Як студент, я розумію, що завдання, яке я поставив перед собою, набагато ширше за межі цього проекту. Трейдинг - це постійно змінюючийся світ, і намагатися зберегти актуальність та ефективність симулятору - завдання, над яким я буду працювати й надалі.

У цьому дипломному проекті я не лише розробив продукт, але й набув багато цінних навичок у сфері програмування, дизайну та аналізу фінансових ринків.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Роберт С. Мартін – Чистий код – Фабула - 2008. – 312с.
2. Ед Тіттел, Кріс Міннік - HTML5 і CSS3 для чайників – Діалектика-Вільямс - 2017. – 400с.
3. Девід Фланаган - JavaScript. The Definitive Guide» - 2011. – Весна - 982с.
4. Стів Круг - Не змушуйте мене думати. Веб'юзабіліті та здоровий глузд - Діалектика-Вільямс - 2016. – 370с.
5. С.М. Порошин, В.М. Карташов, В.В. Усик, Р.І. Цехмістро, І.С.Беліков. Технології створення складових мультимедійного контенту. Анімація та web-анімація. Навчальний посібник –Харків, НТУ.ХПІ, 2022.- 314с.
6. М.А. Омаров Основи технологій сучасної web-анімації. [Текст] Навчальний посібник /М.А. Омаров, В.М. Карташов, Р.І. Цехмістро, В.В. Усик – Харків ХНУРЕ- 2022р.- 214с
7. Карташов В.М., Корытцев И.В., Олейников В.Н., Зубков О.В., Шейко С.А., Бабкин С.И. ОПТИКО-ЭЛЕКТРОННЫЕ МЕТОДЫ ОБНАРУЖЕНИЯ ВОЗДУШНЫХ ОБЪЕКТОВ И ИЗМЕРЕНИЯ ИХ КООРДИНАТ// Радиотехника. (Харьков). — 2020. — Вып. 202. — С. 153-59. DOI:10.30837/rt.2020.3.202.16
8. Рябуха В.П., Карташов В.М. Методы обнаружения-распознавания радиолокационных, акустических, оптических и инфракрасных сигналов беспилотных летательных аппаратов / В.П. Рябуха, В.М. Карташов// Известия высших учебных заведений. Радиоэлектроника. — 2020. — Т. 63, № 11. — С. 1–35.
9. В.М. Карташов, В.Н. Олейников, В.И. Леонидов, канд. Техн. Наук, В.В. Воронин, А.И. Капуста, И.С. Селезнев, Е.В. Першин/

Комплексная обработка сигналов интегрированной системы наблюдения беспилотных летательных аппаратов с использованием целеуказания//Радиотехника. (Харьков). — 2020. — Вып. 203.

10. Карташов В.М., Харченко О.И., Чумаков В.И. Использование эффекта стохастического резонанса для анализа спектров акустического излучения малых беспилотных летательных аппаратов // Радиотехника. (Харьков). — 2019. — Вып. 197. — С. 100-106.

11. Карташов В.М., Сидоров Г.И., Толстих Є.Г., Шаповалов С.В. Акустичний вимірювач швидкості вітру в атмосферному прикордонному шарі// Радиотехника. (Харьков). — 2019. — Вып. 199. — С. 54-58.

12. Олейников В.Н., Зубков О.В, Карташов В.М., Кориццев И.В., Бабкин С.И., Шейко С.А, Селезнев И.С. Экспериментальная оценка эффективности алгоритмов пеленгования беспилотных летательных аппаратов по акустическому излучению// Радиотехника. (Харьков). — 2019. — Вып. 199. — С. 29-37.

13. Карташов В.М., Олейников В.Н., Колендовская М.М., Тимошенко Л.П., Капуста А.И., Рыбников Н.В. Комплексование изображений при обнаружении беспилотных летательных аппаратов // Радиотехника. (Харьков). — 2020. — Вып. 201. — С. -120-129.

14. Карташов В.М., Олейников В.Н., Воронин В.В., Рябуха В.П., Капуста А.И., Рыбников Н.В., Селезнев И.С. Методы комплексной обработки и интерпретации радиолокационных, акустических, оптических и инфракрасных сигналов беспилотных летательных аппаратов //Радиотехника. (Харьков). — 2020. — Вып. 202. — С. 173-182-
DOI:10.30837/rt.2020.3.202.19__