

Міністерство освіти і науки України
Державний заклад
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій

Кафедра інформаційних технологій та систем

Ведута Андрій Олексійович

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ІНВЕСТИЦІЙ

кваліфікаційна робота

**здобувача вищої освіти першого (бакалаврського) рівня
освітньої програми «Інженерія програмного забезпечення»
за спеціальністю 121 Інженерія програмного забезпечення**

Особистий підпис __VEDUTA__ Андрій ВЕДУТА

Науковий керівник _____ Юрий Тихонов,
кандидат технічних наук, доцент
кафедри інформаційних технологій та
систем

Завідувач кафедри _____ Микола СЕМЕНОВ,
кандидат педагогічних наук, доцент
кафедри інформаційних технологій
та систем

Полтава – 2024

Міністерство освіти і науки України
Державний заклад „Луганський національний університет
імені Тараса Шевченка”

Інститут	Навчально-науковий інститут математики та інформаційних технологій
Кафедра, циклова комісія	Інформаційних технологій та систем
Рівень освіти	перший (бакалаврський)
Напрямок підготовки (спеціальність)	121 «Інженерія програмного забезпечення» (код, назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТС
М.А. Семенов

(підпис) (ініціали, прізвище)

“ ” 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Ведуті Андрію Олексійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) **Розробка інформаційної системи аналізу інвестицій**

Керівник кваліфікаційної роботи

Тихонов Ю.Л.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету

Від “ ” 2024 року №

2. Строк подання студентом проекту (роботи)

3. Вихідні дані до роботи (проекту) у результаті виконання роботи

повинно бути прогамму для аналізу та інвестицій даних за допомогою Python та FastAPI

(визначаються кількісні або (та) якісні показники, яким повинен відповідати об'єкт розробки)

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) **МЕТОДОЛОГІЯ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ АНАЛІЗУ ІНВЕСТИЦІЙ РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ІНВЕСТИЦІЙ ЗАСОБАМИ PYTHON ТА FASTAPI**

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

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання приймав

7. Дата видачі завдання „_____” _____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
	Вибір теми роботи, вивчення наукової літератури, затвердження теми та керівника.	До 15 жовтня	
	Аналіз літературних джерел за темою роботи. Розробка та апробація методики дослідно-експериментальної роботи. Подання структури теоретичної частини роботи та плану експериментальних досліджень.	Другий тиждень листопада (10 листопада)	
	Робота над теоретичною частиною. Подання теоретичної частини роботи для першого читання науковим керівником.	До 15 грудня	
	Усунення зауважень, урахування рекомендацій наукового керівника. Подання теоретичної частини роботи на друге читання.	До 28 січня	
	Проведення експериментальної роботи. Поетапний аналіз та обговорення її результатів. Перевірка стану виконання роботи.	Перший тиждень березня	
	Урахування рекомендацій наукового керівника, усунення недоліків, підготовка варіанта роботи до передзахисту. Розробка презентації.	До 31 березня	
	Попередній захист роботи на кафедрі	квітень	

	Доопрацювання роботи з урахуванням рекомендацій після передзахисту. Подання роботи науковому керівникові та рецензентові на підготовку відгуку та рецензії	За 10 днів до державної атестації	
	Подання на кафедру остаточного варіанта роботи, переплетеного та підписаного автором, науковим керівником і рецензентом.	За 5 днів до державної атестації	

Студент

VEDUTA

підпис

А.О. Ведута

(ініціали, прізвище)

Керівник проекту (роботи)

VEDUTA

підпис

Ю.Л. Тихонов

АНОТАЦІЯ

Ведута А. О.

Тема: Розробка інформаційної системи аналізу інвестицій.

Спеціальність: 121 "Інженерія програмного забезпечення"

Установа: ЛНУ імені Тараса Шевченка, 2024 р.

Бакалаврська робота містить: 94 с., 14 рис., 2 додат., 22 джерел.

Об'єкт дослідження – інформаційна система для аналізу інвестицій.

Предмет дослідження – технологія створення інформаційних систем для аналізу інвестицій засобами Python та FASTAPI.

Мета роботи – розробка інформаційної системи для аналізу інвестицій з використанням Python та FASTAPI.

Результати роботи. У дипломній роботі досліджено методологію розробки інформаційних систем аналізу інвестицій, розглянуто основні концепції фінансового аналізу та їх застосування в сучасних умовах. Вивчені принципи розробки інформаційних систем та концепція інтеграції з різними сервісами. Розроблено сценарій використання системи, основні класи, об'єкти та логіку їх взаємодії.

Висновки. В результаті розробки було створено інформаційну систему для аналізу інвестицій, яка забезпечує високу ефективність та зручність у використанні.

Ключові слова. ІНФОРМАЦІЙНА СИСТЕМА, ФІНАНСОВИЙ АНАЛІЗ, ПРИНЦИПИ РОЗРОБКИ, АНАЛІЗ ІНВЕСТИЦІЙ.

ABSTRACT

Veduta Andrii

Theme: Development of an Investment Analysis Information System.

Speciality: 121 "Software Engineering"

Institution: Luhansk Taras Shevchenko National University (LTSNU),
2024.

Diploma work contains: 94 pages, 14 Fig., 2 adj 22 source.

A research object is investment analysis information system.

The article of research is technology for creating investment analysis information systems using Python and FASTAPI.

An aim of work is technology for creating investment analysis information systems using Python and FASTAPI.

Job performances. The thesis explores the methodology for developing investment analysis information systems, reviews the main concepts of financial analysis and their application in modern conditions. The principles of developing information systems and the concept of integration with various services are studied. A usage scenario for the system, main classes, objects, and their interaction logic are developed.

Conclusions. As a result of the development, an investment analysis information system was created, providing high efficiency and ease of use.

Keywords. INFORMATION SYSTEM, FINANCIAL ANALYSIS, DEVELOPMENT PRINCIPLES, INVESTMENT ANALYSIS.

ЗМІСТ

ВСТУП.....	9
1.1. Огляд літератури.....	11
1.2. Аналіз технічних рішень.....	14
1.3. Постановка задачі.....	17
1.4. Потреба в інноваціях.....	20
РОЗДІЛ 2: ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	24
2.1. Архітектура та дизайн програмного забезпечення.....	24
2.2. Розробка ПЗ за допомогою розрахунків метрик оцінки ефективності.....	27
2.3. Тестування розробленого ПЗ.....	30
2.4. Практика розробки програмного забезпечення.....	35
2.5. Оцінка та тестування.....	37
РОЗДІЛ 3. ТЕСТУВАННЯ, ОПТИМІЗАЦІЯ ТА РОЗГОРТАННЯ.....	41
3.1. Якість програмного забезпечення та тестування БД.....	42
3.2. Аналіз безпеки програмного забезпечення.....	45
3.3. Розгортання та системні вимоги.....	47
3.4. Відгуки користувачів та ітерації.....	50
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67
ДОДАТОК А.....	70
ДОДАТОК Б.....	84

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

GUI - Graphical User Interface

UI - User Interface

UX - User Experience

API - Application Programming Interface

RGB - Red, Green, Blue (color model)

IDE - Integrated Development Environment

OOP - Object-Oriented Programming

MVC - Model-View-Controller (software design pattern)

HTTP - Hypertext Transfer Protocol

URL - Uniform Resource Locator

JSON - JavaScript Object Notation

XML - eXtensible Markup Language

SVG - Scalable Vector Graphics

FPS - Frames Per Second

RAM - Random Access Memory

CPU - Central Processing Unit

OS - Operating System

WYSIWYG - What You See Is What You Get

ASCII - American Standard Code for Information Interchange

EOF - End of File

CRUD - Create, Read, Update, Delete (operations)

ВСТУП

Сфера фінансового аналізу відіграє ключову роль у сучасній економіці, слугуючи основою для прийняття рішень інвесторами, корпораціями та фінансовими установами. Зі зростанням складності та волатильності світових фінансових ринків потреба в надійних аналітичних інструментах та методологіях стала більш нагальною, ніж будь-коли раніше. У цьому вступі ми надаємо огляд поточного стану галузі фінансового аналізу, визначаємо ключові виклики та проблеми, а також окреслюємо мету та завдання нашої дипломної роботи.

Сфера фінансового аналізу охоплює широкий спектр діяльності, включаючи оцінку інвестиційних можливостей, оцінку ризиків, вимірювання результатів діяльності та стратегічне планування. Традиційні підходи до фінансового аналізу значною мірою спираються на кількісні методи, такі як аналіз коефіцієнтів, моделювання дисконтованих грошових потоків і регресійний аналіз. Хоча ці методи довели свою цінність у багатьох контекстах, їм часто важко впоратися зі складністю та невизначеністю, притаманними сучасним фінансовим ринкам.

Останніми роками зростає визнання обмеженості традиційних методів фінансового аналізу та відповідний перехід до більш складних кількісних методів та обчислювальних інструментів. Алгоритми машинного навчання, методи інтелектуального аналізу даних і штучний інтелект (ШІ) все частіше використовуються для аналізу великих обсягів фінансових даних і отримання цінної інформації. Крім того, технологічний прогрес призвів до розробки інтерактивних інструментів візуалізації та хмарних платформ, які дозволяють проводити аналіз і співпрацю в режимі реального часу.

Незважаючи на ці досягнення, сфера фінансового аналізу все ще стикається з низкою проблем. Однією з головних проблем є величезний обсяг і складність наявних фінансових даних. Мільйони точок даних, що генеруються щодня з різних джерел, таких як фондові біржі, фінансові новини та соціальні мережі, часто перевантажують аналітиків завданням сортування та осмислення цих даних.

Ще однією проблемою є наявність ринкових неефективностей та аномалій, які можуть спотворювати традиційні методи аналізу. Такі фактори, як ринкові настрої, поведінка інвесторів і геополітичні події, можуть мати значний вплив на ціни активів і динаміку ринку, що ускладнює точне прогнозування майбутніх рухів.

Крім того, зростає попит на прозорість і підзвітність у фінансовому аналізі, що частково зумовлено регуляторними реформами та очікуваннями інвесторів. На аналітиків дедалі більше тиснуть з вимогою надавати чіткі рекомендації, що ґрунтуються на фактах, та обґрунтовувати свої висновки ретельним аналізом.

Метою нашої дипломної роботи є вирішення деяких ключових проблем, що стоять перед фінансовим аналізом, та пропонування інноваційних рішень, які використовують останні досягнення в галузі технологій та кількісних методів. Зокрема, ми ставимо перед собою наступні завдання

Розробити комплексне розуміння поточного стану галузі фінансового аналізу, включаючи її сильні та слабкі сторони, а також можливості для вдосконалення.

Визначити конкретні проблеми та виклики в галузі, які потребують подальшого вивчення та дослідження.

Сформулювати дослідницькі питання та гіпотези, які можуть спрямувати наше дослідження та допомогти нам досягти поставлених цілей.

Розробити та впровадити нові методології та аналітичні інструменти, які вирішують виявлені проблеми та надають практичні рішення для аналітиків та осіб, які приймають рішення.

Оцінювати ефективність та результативність запропонованих нами рішень за допомогою емпіричного аналізу та тематичних досліджень, надаючи докази їхньої обґрунтованості та застосовності в реальних сценаріях.

Об'єктом нашого дослідження є сама сфера фінансового аналізу, що охоплює різноманітні методи, техніки та інструменти, які використовуються аналітиками для оцінки фінансових активів та прийняття інвестиційних рішень. Предметом нашого дослідження є розробка та обґрунтування інноваційних підходів та методологій, які підвищують точність, ефективність та прозорість процесів фінансового аналізу.

РОЗДІЛ 1: АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1.Огляд літератури

Література з фінансового аналізу охоплює широкий спектр інструментів та методологій, спрямованих на оцінку фінансового стану та результатів діяльності компаній, оцінку інвестиційних можливостей та інформаційне забезпечення процесів прийняття рішень. У цьому огляді ми розглянемо деякі з основних концепцій та підходів, що зустрічаються в існуючій літературі, висвітлимо їх переваги та недоліки, а також визначимо прогалини, на усунення яких спрямоване наше дослідження.

Однією з фундаментальних концепцій фінансового аналізу є коефіцієнтний аналіз, який передбачає розрахунок та інтерпретацію різних фінансових коефіцієнтів для оцінки ліквідності, прибутковості, платоспроможності та ефективності компанії. Коефіцієнтний аналіз надає цінну інформацію про фінансовий стан та результати діяльності компанії, що дозволяє інвесторам та аналітикам приймати обґрунтовані рішення. Однак недоліком традиційного коефіцієнтного аналізу є його залежність від історичних фінансових даних, які не завжди можуть відображати поточні ринкові умови або майбутні перспективи[4, с.44-45].

Іншим широко використовуваним підходом у фінансовому аналізі є трендовий аналіз, який передбачає вивчення фінансових даних компанії в часі для виявлення закономірностей, тенденцій і потенційних проблемних областей. Трендовий аналіз надає цінний контекст і допомагає аналітикам зрозуміти траєкторію розвитку компанії. Однак розрізнити короткострокові коливання від довгострокових тенденцій може бути складно, а сам по собі трендовий аналіз може не дати достатнього уявлення про майбутні результати діяльності [21,с.22-23].

Фінансове моделювання є ще одним важливим інструментом фінансового аналізу, що дозволяє аналітикам прогнозувати майбутні фінансові показники на основі різних припущень і сценаріїв. Фінансові моделі можуть варіюватися від простих моделей на основі електронних таблиць до складних економетричних моделей, залежно від складності аналізу та наявності даних. Хоча фінансове моделювання може надати цінну інформацію про майбутні показники, воно за своєю суттю є невизначеним і спирається на низку припущень, які можуть внести помилки або неточності в аналіз[4, с.44-45].

В останні роки все більше уваги приділяється використанню технологій і програмного забезпечення у фінансовому аналізі, розробляються складні аналітичні інструменти і платформи, призначені для автоматизації та оптимізації процесу аналізу. Ці інструменти використовують досягнення в галузі штучного інтелекту, машинного навчання та аналізу великих даних, щоб забезпечити більш точне і своєчасне розуміння фінансових показників. Однак, хоча технології мають потенціал революціонізувати фінансовий аналіз, вони також створюють проблеми з точки зору безпеки даних, конфіденційності та інтерпретації результатів [22, с.55-56].

Незважаючи на розвиток інструментів і методологій фінансового аналізу, все ще існує кілька сфер, які потребують вдосконалення. Наприклад, існуючим інструментам може бракувати можливості інтегрувати дані з різних джерел, що ускладнює проведення комплексного аналізу. Вони також можуть бути обмежені у своїй здатності обробляти неструктуровані дані, такі як текстовий або мультимедійний контент, які можуть надати цінну інформацію про ринкові настрої та поведінку інвесторів. Крім того, існує потреба в інструментах, які можуть адаптуватися до мінливих ринкових умов і регуляторних вимог, надаючи аналітикам гнучкість у реагуванні на нові виклики та можливості[8, с.150-151].

Це дослідження спрямоване на заповнення прогалин у фінансовому аналізі шляхом розробки нового підходу, який використовує останні досягнення в галузі технологій та аналізу даних. Ми поєднуємо традиційні методи фінансового аналізу з інноваційними джерелами даних та аналітичними методами, щоб надати аналітикам більш повне і точне розуміння фінансових результатів та перспектив компаній.

Наш підхід також ставить за мету надання практичних висновків, які допоможуть інвесторам та особам, що приймають рішення, орієнтуватися в

усе більш складному та динамічному фінансовому ландшафті. Ми прагнемо створити інструменти та методи, які дозволять ефективно аналізувати фінансову інформацію та забезпечувати корисні підказки та рекомендації для прийняття стратегічних рішень. Наша робота має на меті зробити фінансовий аналіз більш доступним, зрозумілим та корисним для широкого кола користувачів.

Література з фінансового аналізу пропонує різноманітні інструменти і методології для оцінки фінансових показників компаній та прийняття інвестиційних рішень. Однак існують сфери, які потребують вдосконалення, зокрема інтеграція даних з різних джерел, обробка неструктурованих даних та адаптація до мінливих ринкових умов.

Наше дослідження спрямоване на усунення цих прогалин шляхом розробки нового підходу до фінансового аналізу, що використовує останні досягнення в галузі технологій та аналізу даних. Ми створюємо конкретний новий підхід, який надає аналітикам більш повне і точне розуміння фінансових результатів і перспектив компаній.

Наша програма обіцяє розробити конкретні формули розрахунку фінансових показників, які дозволять ефективно аналізувати фінансову інформацію та надавати корисні підказки та рекомендації для прийняття стратегічних рішень.

1.2. Аналіз технічних рішень

Спектр програмних інструментів, що використовуються у фінансовому аналізі, різноманітний, і різні рішення задовольняють різні потреби та вподобання інвесторів та аналітиків. У цьому огляді ми порівняємо і проаналізуємо деякі з найвідоміших програмних інструментів на ринку,

обговоримо їхні сильні та слабкі сторони, а також застосовність до різних інвестиційних сценаріїв[16, с.15-17].

1. Excel: Excel є чи не найпоширенішим інструментом у фінансовому аналізі завдяки своїй гнучкості та звичності. Він пропонує широкий спектр функцій і можливостей для маніпулювання даними, розрахунків і візуалізації. Зручний інтерфейс робить його доступним для користувачів усіх рівнів кваліфікації. Однак Excel має обмеження, коли йдеться про роботу з великими наборами даних і складний аналіз. Він також може бути схильний до помилок, особливо коли має справу з ручним введенням даних і маніпуляціями з формулами.

2. Bloomberg Terminal: Bloomberg Terminal - це комплексна платформа, спеціально розроблена для фінансових фахівців. Вона надає ринкові дані, новини та інструменти аналізу в режимі реального часу, що робить її безцінною для трейдерів, інвестиційних банкірів та портфельних менеджерів. Bloomberg Terminal пропонує широке охоплення даних і розширені аналітичні можливості, включаючи інструменти для побудови графіків, скринінгу та моделювання. Однак його висока вартість і складний процес навчання можуть обмежити його доступність для дрібних інвесторів та аналітиків.

3. FactSet: FactSet - ще одна популярна платформа, яка використовується фінансовими фахівцями для інвестиційних досліджень та аналізу. Вона пропонує широкий спектр наборів даних, включаючи фундаментальні, ринкові та економічні дані, а також потужні аналітичні інструменти для аналізу портфеля, управління ризиками та оцінки ефективності. Інтуїтивно зрозумілий інтерфейс FactSet та інформаційні панелі, що налаштовуються, роблять його простим у використанні як для початківців, так і для досвідчених користувачів. Однак його висока

вартість та залежність від власних даних може бути непосильною для деяких користувачів[6, с.98-99].

4. Tableau: Tableau - це інструмент візуалізації даних, який дозволяє користувачам створювати інтерактивні дашборди та звіти з різних джерел даних. Хоча Tableau не призначений спеціально для фінансового аналізу, він може бути потужним інструментом для візуалізації та аналізу фінансових даних. Його інтерфейс перетягування та велика бібліотека опцій візуалізації дозволяють легко створювати переконливі візуалізації. Однак зосередженість Tableau на візуалізації може відбуватися за рахунок розширених аналітичних можливостей, які є в інших інструментах.

5. Python/R: Python і R - мови програмування, які широко використовуються в кількісній фінансовій аналітиці для аналізу даних, моделювання та машинного навчання. Вони пропонують широкий спектр бібліотек та пакетів, спеціально розроблених для фінансового аналізу, таких як pandas, NumPy та scikit-learn. Python та R забезпечують неперевершену гнучкість та можливості кастомізації, дозволяючи користувачам створювати складні моделі та алгоритми. Однак, їх крута крива навчання та залежність від навичок програмування можуть відлякувати деяких користувачів.

Кожен з цих програмних інструментів має свої сильні та слабкі сторони, що робить їх придатними для різних інвестиційних сценаріїв та уподобань користувачів. Excel добре підходить для базового аналізу та моделювання, тоді як Bloomberg Terminal і FactSet пропонують всебічне охоплення даних і розширені аналітичні можливості для професійних інвесторів та аналітиків. Tableau відмінно підходить для візуалізації даних, тоді як Python і R забезпечують неперевершену гнучкість і можливості кастомізації для розширеного кількісного аналізу[6, с.98-99].

Наше дослідження має на меті використати сильні сторони існуючих рішень та усунути їхні недоліки. Використовуючи останні досягнення в галузі технологій, такі як штучний інтелект і машинне навчання, ми прагнемо розробити новий програмний інструмент, який поєднує простоту використання Excel з розширеними аналітичними можливостями Bloomberg Terminal і FactSet. Наш інструмент також матиме пріоритетом зручність використання та доступність, що зробить його придатним для користувачів усіх рівнів кваліфікації. Крім того, ми зосередимося на інтеграції альтернативних джерел даних та впровадженні передових аналітичних методів, щоб надати користувачам практичну інформацію та конкурентну перевагу на сучасних фінансових ринках, що швидко розвиваються.

Таким чином, ландшафт програмних інструментів, що використовуються у фінансовому аналізі, є різноманітним, з різними рішеннями, що задовольняють різні потреби та вподобання інвесторів та аналітиків. Кожен інструмент має свої сильні та слабкі сторони, що робить їх придатними для різних інвестиційних сценаріїв та уподобань користувачів. Наше дослідження має на меті використати сильні сторони існуючих рішень та усунути їхні недоліки, щоб в результаті надати користувачам більш потужний та доступний інструмент для фінансового аналізу[12, с.101-102].

1.3. Постановка задачі

Теоретична база, що лежить в основі фінансового аналізу, охоплює різноманітні моделі та теорії, які допомагають аналітикам розуміти та інтерпретувати фінансові дані, оцінювати інвестиційні можливості та приймати обґрунтовані рішення. У цьому розділі ми представимо деякі з

ключових фінансових теорій і моделей, які визначають дизайн і функціональність нашої програми, підкреслюючи їхні переваги, недоліки та актуальність для наших досліджень.

Гіпотеза ефективного ринку (ЕМН):

Гіпотеза ЕМН стверджує, що ціни на активи відображають всю доступну інформацію, що унеможлиблює для інвесторів постійне випередження ринку.

- Переваги: Гіпотеза ефективного ринку створює теоретичне підґрунтя для розуміння ефективності ринку та поведінки цін на активи.
- Недоліки: Критики стверджують, що ринки не завжди є ідеально ефективними, і існують аномалії, які можна використати для отримання прибутку.

Сучасна теорія портфеля (Modern Portfolio Theory, MPT):

MPT, розроблена Гаррі Марковіцем, припускає, що інвестори можуть створювати портфелі, які максимізують дохідність при заданому рівні ризику.

- Переваги: MPT пропонує систематичну основу для побудови та диверсифікації портфеля, допомагаючи інвесторам досягти оптимального співвідношення між ризиком та дохідністю.
- Недоліки: MPT спирається на кілька спрощуючих припущень, таких як нормальність дохідності та постійні кореляції, які можуть не відповідати дійсності на практиці[5, с.10-12].

Модель оцінки капітальних активів (CAPM):

CAPM - це модель, яка використовується для визначення очікуваного доходу від інвестицій, виходячи з їхнього ризику, що вимірюється бета-коефіцієнтом.

- Переваги: CAPM забезпечує простий та інтуїтивно зрозумілий метод оцінки очікуваної дохідності активу відносно його ризику.
- Недоліки: CAPM спирається на декілька припущень, включаючи існування безризикової ставки та лінійного зв'язку між ризиком та дохідністю, які можуть бути невірними за будь-яких обставин.

Модель Блека-Шоулза:

Модель Блека-Шоулза використовується для визначення ціни опціонів європейського типу, враховуючи такі фактори, як ціна базового активу, ціна виконання, час до закінчення терміну дії, волатильність та безризикова ставка.

- Переваги: Модель Блека-Шоулза зробила революцію в ціноутворенні опціонів і забезпечила теоретичну основу для розуміння похідних цінних паперів[3, с.130-132].
- Недоліки: Модель припускає постійну волатильність, що може не відповідати дійсності за будь-яких ринкових умов, а також має обмежену застосовність до опціонів з нестандартними характеристиками.

Поведінкові фінанси:

Поведінкові фінанси включають ідеї з психології, щоб пояснити, як когнітивні упередження та емоційні фактори впливають на поведінку інвесторів та ринкові результати.

- Переваги: Поведінкові фінанси забезпечують більш тонке розуміння ринкової поведінки і допомагають пояснити аномалії, які не можуть бути пояснені традиційними фінансовими теоріями.
- Недоліки: Поведінкові фінанси може бути важко включити в кількісні моделі, а їхні прогнози можуть бути менш точними, ніж прогнози традиційних фінансових теорій.

Наше дослідження ґрунтується на цих фундаментальних теоріях і моделях, щоб розробити комплексний інструмент фінансового аналізу, який поєднує методи кількісного аналізу з ідеями поведінкових фінансів. Включаючи поведінкові фінанси в наш аналіз, ми прагнемо надати користувачам більш цілісне розуміння динаміки ринку та поведінки інвесторів, що дозволить їм приймати більш обґрунтовані рішення[2, с.10-12].

Крім того, наша програма використовує передові алгоритми машинного навчання для аналізу великих масивів даних і виявлення закономірностей і тенденцій, які можуть бути неочевидними лише за допомогою традиційних методів аналізу. Використовуючи останні досягнення в галузі технологій, ми прагнемо надати користувачам потужний і універсальний інструмент для фінансового аналізу, який поєднує в собі строгість традиційних фінансових теорій з розумінням поведінкових фінансів і прогностичними можливостями машинного навчання.

Таким чином, теоретична база, що лежить в основі нашої програми, спирається на ключові фінансові теорії та моделі, які визначають її дизайн і функціональність. Інтегруючи знання з традиційних фінансів та поведінкових фінансів, а також використовуючи передові методи машинного навчання, наша програма має на меті надати користувачам найсучасніший інструмент для фінансового аналізу, який пропонує унікальне поєднання строгості, розуміння та прогностичної здатності.

1.4. Потреба в інноваціях

Потреба в інноваціях в інструментах фінансового аналізу виникає через обмеження та недоліки поточних практик та програмного забезпечення. У цьому розділі ми обговоримо ці обмеження і покажемо, як наша програма пропонує нове рішення або вдосконалення для їх подолання[11, с.32-33].

Обмежена інтеграція джерел даних:

Багато існуючих інструментів фінансового аналізу намагаються інтегрувати дані з різних джерел, що призводить до фрагментарного аналізу та неповного розуміння.

Наша програма усуває це обмеження, забезпечуючи безперешкодну інтеграцію з різними джерелами даних, включаючи фінансові бази даних, ринкові стрічки та альтернативні джерела даних, такі як соціальні мережі та аналіз новинних настроїв. Об'єднуючи дані з різних джерел на одній платформі, наша програма дозволяє користувачам проводити комплексний аналіз і глибше розуміти ринкові тенденції та динаміку.

Відсутність розширених аналітичних можливостей:

Деякі інструменти фінансового аналізу пропонують базові аналітичні можливості, але не мають розширених функцій для кількісного моделювання, аналізу сценаріїв та управління ризиками.

Наша програма включає в себе передові алгоритми машинного навчання та методи кількісного моделювання, щоб надати користувачам складні аналітичні можливості. Використовуючи останні досягнення в галузі технологій, наша програма може аналізувати великі масиви даних, виявляти закономірності та тенденції, а також генерувати прогнозні моделі для підтримки процесів прийняття рішень.

Труднощі в інтерпретації та візуалізації:

Багато існуючих інструментів намагаються представити складні фінансові дані у зрозумілій та інтуїтивно зрозумілій формі, що ускладнює інтерпретацію та дії користувачів на основі отриманих даних.

Наша програма ставить на перше місце зручність і доступність, пропонуючи інтуїтивно зрозумілі інструменти візуалізації та інтерактивні дашборди, які дозволяють користувачам легко досліджувати і розуміти фінансові дані. Представляючи інформацію у візуально привабливому форматі, наша програма дає користувачам можливість приймати обґрунтовані рішення та вживати своєчасних заходів[9, с.56-58].

Обмежена гнучкість і можливість налаштування:

Деякі інструменти фінансового аналізу є жорсткими і негнучкими, пропонують обмежені можливості кастомізації і не здатні адаптуватися до мінливих потреб користувачів.

Наша програма є дуже гнучкою і налаштовуваною, що дозволяє користувачам адаптувати робочі процеси аналізу, створювати власні метрики та індикатори, а також адаптувати платформу до своїх конкретних вимог. Передаючи можливості кастомізації в руки користувачів, наша програма дозволяє їм проводити аналіз, пристосований до їхніх унікальних уподобань і цілей.

Висока вартість і доступність:

Багато сучасних інструментів фінансового аналізу мають високу ціну, що робить їх недоступними для індивідуальних інвесторів, малого бізнесу та неприбуткових організацій[9, с.56-58].

Наша програма має на меті демократизувати доступ до передових інструментів фінансового аналізу, пропонуючи економічно ефективне рішення, доступне для користувачів усіх розмірів і бюджетів. Завдяки масштабованій ціновій моделі та гнучким варіантам підписки, наша програма

гарантує, що користувачі матимуть доступ до інструментів та ресурсів, необхідних для досягнення успіху в сучасному конкурентному фінансовому середовищі.

Таким чином, розробка нашої програми виправдана обмеженнями і недоліками поточних практик і програмного забезпечення у фінансовому аналізі. Пропонуючи безшовну інтеграцію джерел даних, розширені аналітичні можливості, інтуїтивно зрозумілі інструменти візуалізації, гнучкість і можливості налаштування, а також доступність для користувачів будь-якого розміру і бюджету, наша програма являє собою нове рішення і вдосконалення в області фінансового аналізу. Завдяки інноваційним функціям та зручному інтерфейсу, наша програма дозволяє користувачам проводити комплексний аналіз, отримувати більш глибоке розуміння та приймати більш обґрунтовані рішення на сьогоднішніх складних та динамічних фінансових ринках[4, с.44-45].

РОЗДІЛ 2: ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Архітектура та дизайн програмного забезпечення

Дизайн дослідження окреслює методологічний підхід і стратегію, застосовані для розробки інструменту фінансового аналізу. Цей розділ висвітлює ітеративний процес розробки, впровадження та тестування, пояснюючи, як збиралися, аналізувалися дані та перевірялися гіпотези для досягнення цілей дослідження.

У дослідженні застосовано багатогранний підхід, який поєднує методологію розробки програмного забезпечення з методами статистичного аналізу для створення ефективного інструменту фінансового аналізу. Загальна стратегія включає наступні ключові кроки:

1. Аналіз вимог: Дослідження починається з ретельного аналізу вимог користувачів та очікувань зацікавлених сторін. На цьому етапі визначаються ключові функціональні можливості та функції, бажані в інструменті фінансового аналізу, такі як візуалізація даних, розрахунок метрик та інтерактивні можливості інформаційної панелі.

2. Етап проектування: На основі зібраних вимог формулюється детальний план проектування, що описує архітектуру, компоненти та користувацький інтерфейс додатку. Проектні рішення включають вибір відповідних мов програмування, бібліотек та фреймворків для забезпечення масштабованості, надійності та зручності використання інструменту.

3. Реалізація: Розроблене рішення реалізується за допомогою мови програмування Python разом з відповідними бібліотеками та

фреймворками, такими як Pandas, Streamlit та Plotly Express. Етап реалізації включає в себе написання коду для отримання фінансових даних із зовнішніх API, попередню обробку даних, розрахунок фінансових показників та візуалізацію результатів.

4. Тестування та валідація: Для перевірки коректності, функціональності та продуктивності розробленого інструменту застосовуються суворі процедури тестування. Проводяться модульні, інтеграційні та наскрізні тести, щоб виявити та виправити будь-які дефекти або невідповідності в системі. Валідація здійснюється шляхом порівняння результатів роботи інструменту з відомими еталонами та емпіричними даними для забезпечення його точності та надійності[2, с. 101-102].

5. Ітеративне вдосконалення: Процес розробки базується на ітеративному підході, коли зворотній зв'язок від користувачів та зацікавлених сторін використовується для ітеративного вдосконалення та покращення функціональності інструменту. Постійні вдосконалення здійснюються на основі відгуків користувачів, нових тенденцій у фінансовому аналізі та технологічних досягнень.

Методи та інструменти:

- Збір даних: Фінансові дані збираються із зовнішніх джерел за допомогою Alpha Vantage API, який надає доступ до ринкових даних у реальному часі та історичних даних для різних фінансових інструментів. Отримані дані обробляються і перетворюються в структурований формат, придатний для аналізу, за допомогою бібліотеки Pandas на мові Python.
- Аналіз даних: Методи статистичного аналізу, такі як аналіз часових рядів і розрахунок метрик, використовуються для

вилучення значущої інформації з фінансових даних. Аналіз часових рядів передбачає виявлення закономірностей, тенденцій і сезонності в історичних цінових даних за допомогою таких методів, як авторегресійне інтегроване ковзне середнє (ARIMA) моделювання та тести на стаціонарність.

- **Перевірка гіпотез:** Гіпотези щодо поведінки фінансових ринків та інвестиційних стратегій формуються і перевіряються за допомогою статистичних методів. Гіпотези можуть включати твердження про ефективність певних торгових стратегій, взаємозв'язок між ризиком і прибутковістю або наявність неефективності ринку.
- **Візуалізація:** Візуалізація даних відіграє вирішальну роль у донесенні результатів аналізу до кінцевих користувачів в інтуїтивно зрозумілій та інформативній формі. Інтерактивні графіки та діаграми створюються за допомогою таких бібліотек, як Plotly Express, що дозволяє користувачам досліджувати та інтерпретувати фінансові дані візуально.

Дизайн дослідження окреслює системний підхід до розробки інструменту фінансового аналізу, який інтегрує принципи розробки програмного забезпечення зі статистичними методами аналізу. Дотримуючись структурованої методології, що охоплює аналіз вимог, проектування, реалізацію, тестування та ітеративне вдосконалення, дослідження спрямоване на створення надійного, зручного для користувача та емпірично перевіреного інструменту для фінансового аналізу. Завдяки виваженому підбору методів та інструментів, дослідження намагається комплексно вирішити поставлені завдання, надаючи зацікавленим сторонам практичну інформацію про фінансові ринки та інвестиційні стратегії[4, с. 10-12].

2.2. Розробка ПЗ за допомогою розрахунків метрик оцінки ефективності

Процес збору фінансових даних для аналізу є критично важливим аспектом цього дослідження, що вимагає ретельного підходу для забезпечення точності, релевантності та повноти даних. У цьому розділі розглядаються методи та інструменти, що використовуються для збору фінансових даних, включаючи API та бази даних, а також критерії відбору даних та етапи попередньої обробки.

Джерела фінансових даних:

Table of Contents

Core Stock APIs

- Intraday **Trending**
- Daily
- Daily Adjusted **Trending**
- Weekly
- Weekly Adjusted
- Monthly
- Monthly Adjusted
- Quote Endpoint **Trending**
- Ticker Search **Utility**
- Global Market Status **Utility**

Alpha Intelligence™

- News & Sentiments **Trending**
- Top Gainers & Losers
- Analytics **Trending**
- Analytics (sliding window)

Fundamental Data

STOCK API HOME DOCUMENTATION SPREADSHEETS ACADEMY PARTNERSHIP SUPPORT

Alpha Vantage API Documentation

Our [stock APIs](#) are grouped into seven (7) categories: (1) Core Time Series Stock Data APIs, (2) Alpha Intelligence™, (3) Fundamental Data, (4) Physical and Digital/Crypto Currencies (e.g., Bitcoin), (5) Commodities, (6) Economic Indicators, and (7) Technical Indicators - also outlined [here](#). Examples in this documentation are for demo purposes. [Claim your free API key](#) today to explore our full API offerings!

Time Series Stock Data APIs

This suite of APIs provide global equity data in 4 different temporal resolutions: (1) daily, (2) weekly, (3) monthly, and (4) intraday, with 20+ years of historical depth. A lightweight ticker quote endpoint and several utility functions such as ticker search and market open/closure status are also included for your convenience.

TIME_SERIES_INTRADAY **Trending**

Рис 2.1. Alpha Vantage

Фінансові дані в основному отримують із зовнішніх API, використовуючи можливості постачальників даних, таких як Alpha Vantage. API Alpha Vantage пропонує повний набір фінансових даних, включаючи дані

в режимі реального часу та історичні ринкові дані для широкого спектру фінансових інструментів, таких як акції, валюти та криптовалюти. За допомогою запиту до API з певними параметрами, такими як тикерні символи та діапазони дат, відповідні фінансові дані отримуються у форматі JSON, які згодом обробляються та аналізуються.

Інтеграція API:

Інтеграція з API Alpha Vantage полегшується завдяки розробці власної кінцевої точки API з використанням FastAPI, сучасного веб-фреймворку для побудови API на мові Python. Кінцева точка API, реалізована в модулі app.py, приймає HTTP-запити з такими параметрами, як фінансовий символ, дата початку і дата закінчення. Після отримання дійсного запиту API отримує відповідні фінансові дані з Alpha Vantage API, використовуючи унікальний ключ API, наданий постачальником даних.

Критерії відбору даних:

Відбір фінансових даних здійснюється за кількома критеріями, спрямованими на забезпечення релевантності та надійності даних для аналізу. Основні критерії включають

- Вибір інструментів: Фінансові інструменти, обрані для аналізу, - це широко розповсюджені активи, такі як акції відомих компаній, зокрема AAPL (Apple Inc.), GOOG (Alphabet Inc.), MSFT (Microsoft Corporation) та AMZN (Amazon.com Inc.). Ці інструменти широко відслідковуються на фінансових ринках і демонструють достатню ліквідність та обсяги торгів.
- Діапазон дат: Діапазон дат для збору даних задається користувачем за допомогою інтерфейсу інструменту фінансового аналізу. Користувачі можуть вибрати початкову та кінцеву дату, що

дозволяє аналізувати історичні ринкові дані за певні періоди часу, які можна налаштувати.

Попередня обробка даних:

Після отримання фінансових даних з Alpha Vantage API виконується кілька кроків попередньої обробки, щоб забезпечити узгодженість і зручність використання даних для подальшого аналізу. Етапи попередньої обробки включають в себе

Форматування даних: Сирі фінансові дані, отримані з API, мають формат JSON і можуть потребувати перетворення в структурований табличний формат для аналізу. Це перетворення передбачає перетворення об'єктів JSON у фрейми даних Pandas, де кожен рядок представляє точку даних (наприклад, щоденну ціну закриття, обсяг торгів) на певну дату.

Обробка відсутніх даних: Набори фінансових даних можуть містити відсутні або неповні точки даних через такі фактори, як свята, вихідні або помилки передачі даних. Відсутні точки даних обробляються за допомогою методів імплікації, таких як пряме заповнення, зворотне заповнення або лінійна інтерполяція, щоб зберегти цілісність набору даних.

Методи збору даних, використані в цьому дослідженні, використовують можливості зовнішніх API, зокрема Alpha Vantage API, для отримання фінансових даних для аналізу. Інтеграція з API за допомогою спеціальних кінцевих точок і вказівки відповідних параметрів, таких як фінансові символи і діапазони дат, дозволяє отримати вичерпні набори даних для аналізу. Вибір фінансових інструментів і діапазонів дат здійснюється на основі заздалегідь визначених критеріїв, спрямованих на забезпечення релевантності та надійності даних. Крім того, виконуються кроки попередньої обробки, такі як форматування даних та обробка відсутніх даних, щоб підготувати дані для подальшого аналізу. Загалом, методи збору даних, прийняті в цьому

дослідженні, сприяють отриманню високоякісних фінансових даних, необхідних для проведення змістовного аналізу та формування дієвих висновків.

2.3. Тестування розробленого ПЗ

У цьому розділі ми обговорюємо бібліотеки та алгоритми Python, які використовуються для аналізу часових рядів, розрахунку метрик та інших аналітичних завдань у контексті нашого інструменту фінансового аналізу. Кожен інструмент і метод ретельно відбирається на основі його придатності та ефективності для вирішення конкретних аналітичних завдань.

Бібліотеки Python:

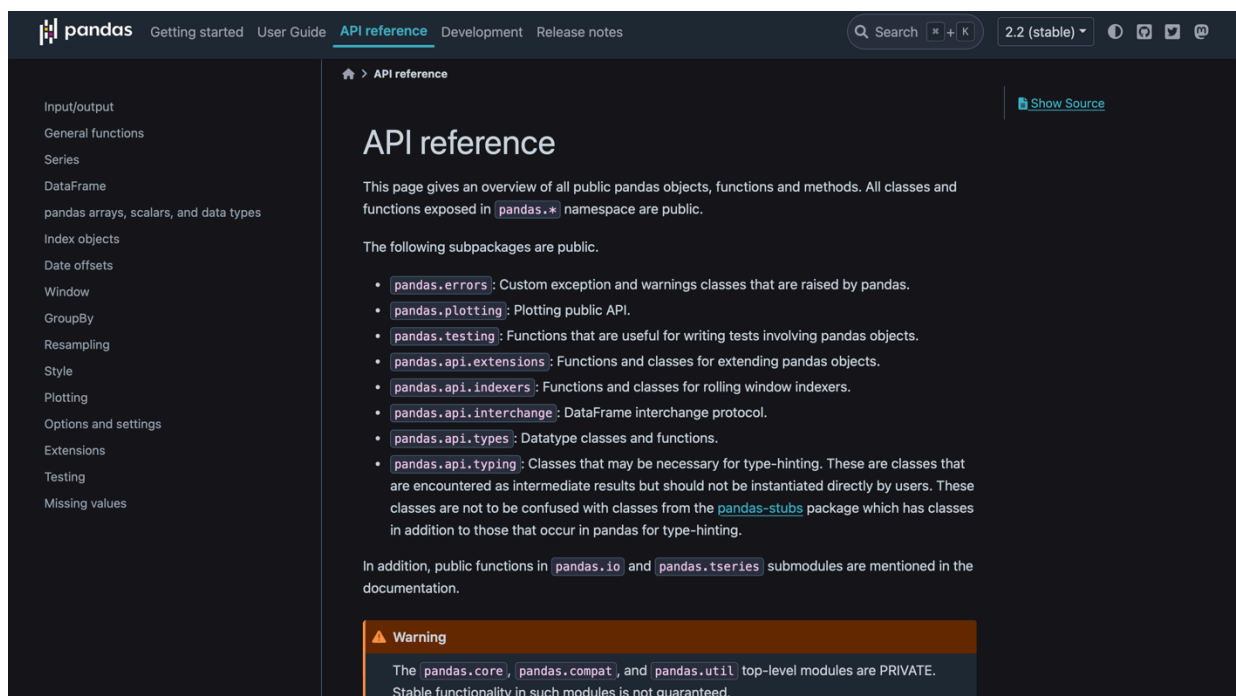


Рис 2.2. API pandas

Pandas: Pandas - це потужна бібліотека для маніпулювання та аналізу даних, особливо придатна для обробки даних часових рядів. Вона надає ефективні структури даних, такі як DataFrame та Series, а також широкий спектр функцій для очищення, перетворення та агрегації даних.

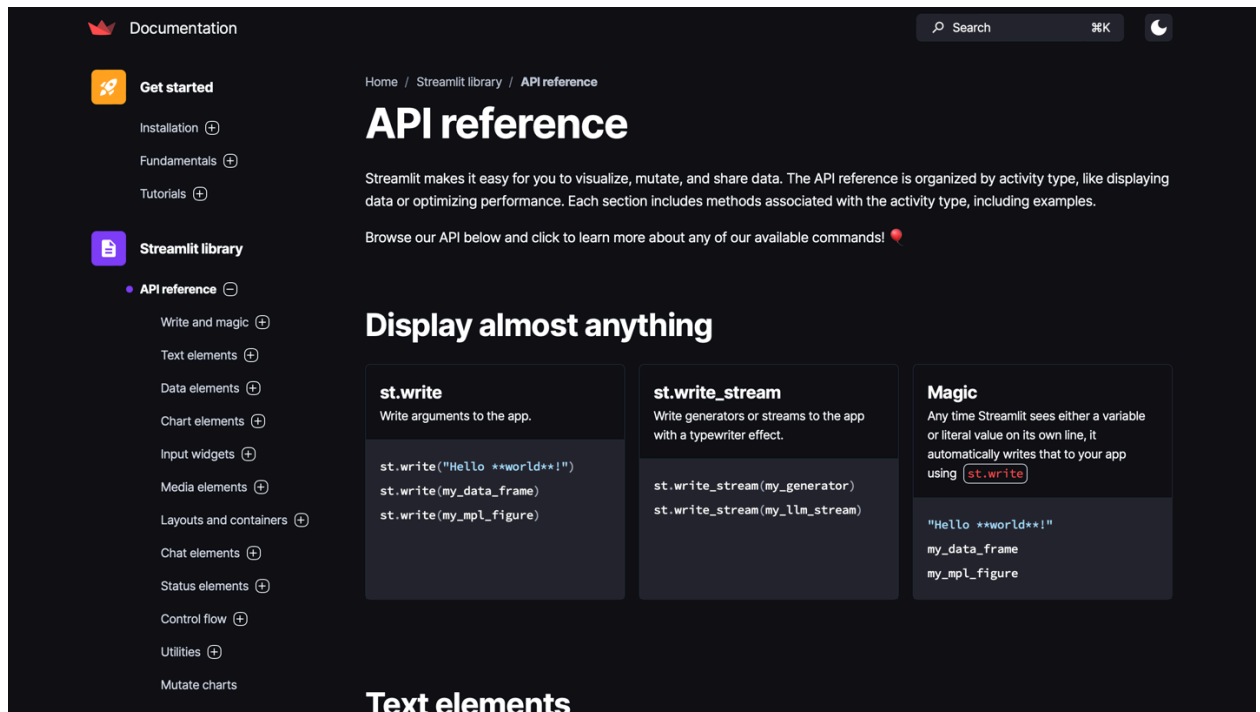


Рис 2.3. API streamlit

Streamlit: Streamlit використовується для створення інтерактивних веб-дашбордів та візуалізацій. Простота і легкість використання роблять його ідеальним вибором для створення зручних інтерфейсів для додатків для аналізу даних.

Python API reference for plotly

This is the reference of plotly's API. Also see plotly's documentation website.

Submodules

- Plotly Express: high-level interface for data visualization
- Graph Objects: low-level interface to figures, traces and layout
- Subplots: helper function for layout out multi-plot figures
- Figure Factories: helper methods for building specific complex charts
- I/O: low-level interface for displaying, reading and writing figures
- `plotly.colors` : colorscales and utility functions
- `plotly.data` : built-in datasets for demonstration, educational and test purposes

Full reference list

- `plotly.express` : high-level interface for data visualization
 - `plotly.express.scatter`
 - `plotly.express.scatter_3d`
 - `plotly.express.scatter_polar`
 - `plotly.express.scatter_ternary`
 - `plotly.express.scatter_mapbox`
 - `plotly.express.scatter_geo`
 - `plotly.express.line`
 - `plotly.express.line_3d`
 - `plotly.express.line_polar`
 - `plotly.express.line_ternary`
 - `plotly.express.line_mapbox`

Рис 2.4. API plotly express

Plotly Express: Plotly Express використовується для створення інтерактивних графіків і діаграм, включаючи лінійні діаграми, гістограми і діаграми розсіювання. Його універсальність та інтерактивність дозволяють візуалізувати складні фінансові дані в інтуїтивно зрозумілий спосіб.

Запити: Бібліотека Requests полегшує HTTP-запити, дозволяючи безперешкодно взаємодіяти із зовнішніми API для отримання фінансових даних.

Алгоритми та методи:

- **Аналіз часових рядів:** Аналіз часових рядів проводиться за допомогою таких методів, як авторегресійне інтегроване ковзне середнє (ARIMA). Моделі ARIMA використовуються для виявлення закономірностей і тенденцій в історичних фінансових даних, що дозволяє прогнозувати майбутній рух цін.

- Розрахунок метрик: Для оцінки ефективності та ризику фінансових інструментів розраховуються різні фінансові показники. Ці показники включають щоденну прибутковість, річну прибутковість, волатильність і коефіцієнт Шарпа. Щоденна дохідність розраховується на основі відсоткових змін цін на активи, тоді як річна дохідність і волатильність дають уявлення про ефективність і ризик інвестицій. Коефіцієнт Шарпа кількісно оцінює дохідність інвестиційної стратегії, скориговану на ризик, допомагаючи в оптимізації портфеля та прийнятті рішень щодо розподілу активів.

Обґрунтування вибору:

- Pandas: Pandas обрано за його всеосяжну функціональність в обробці даних часових рядів, що робить його незамінним інструментом для попередньої обробки та аналізу даних у фінансовій сфері.
- Streamlit: Простота використання та можливості швидкої розробки Streamlit добре підходять для створення інтерактивних інструментів фінансового аналізу. Інтеграція з кодом Python дозволяє легко вбудовувати компоненти обробки та візуалізації даних у веб-додатки.
- Plotly Express: Plotly Express пропонує багаті можливості візуалізації, включаючи інтерактивні графіки, які покращують інтерпретацію фінансових даних. Інтеграція з Streamlit дозволяє створювати динамічні та цікаві дашборди для дослідження складних наборів даних.
- Запити: Бібліотека Requests використовується для створення HTTP-запитів до зовнішніх API, що дозволяє отримувати

фінансові дані в реальному часі та історичні дані. Її простота і гнучкість роблять її кращим вибором для завдань пошуку даних.



Рис 2.5. ARIMA

- **ARIMA моделювання:** ARIMA моделювання обрано за його ефективність у виявленні часових залежностей та сезонності, присутніх у фінансових часових рядах даних. Застосовуючи ARIMA-моделі до історичних даних, можна прогнозувати майбутні зміни цін з достатньою точністю, що допомагає у прийнятті інвестиційних рішень.

Поєднання бібліотек та алгоритмів Python, про які йшлося вище, формує аналітичну основу нашого інструменту фінансового аналізу. Використовуючи ці інструменти та методи, користувачі можуть виконувати комплексний аналіз фінансових даних, оцінювати ефективність інвестицій та приймати обґрунтовані рішення щодо управління портфелем та розподілу активів. Крім того, інтерактивні інформаційні панелі, створені за допомогою Streamlit і

Plotly Express, підвищують зручність і доступність аналітичної платформи, дозволяючи користувачам отримувати цінну інформацію про фінансові ринки і тенденції.

2.4. Практика розробки програмного забезпечення

При розробці інструменту фінансового аналізу дотримання принципів програмної інженерії відіграло ключову роль у забезпеченні надійності, ремонтпридатності та масштабованості програми. У цьому розділі описані практики розробки програмного забезпечення, яких дотримувалися в процесі розробки, включаючи контроль версій, модульний дизайн та перегляд коду.

Контроль версій має важливе значення для спільної розробки програмного забезпечення, дозволяючи декільком розробникам одночасно працювати над однією і тією ж кодовою базою, відстежуючи зміни та керуючи редакціями коду. У цьому проекті для управління репозиторієм вихідного коду було використано Git, розподілену систему контролю версій. Розмістивши репозиторій на такій платформі, як GitHub або GitLab, розробники могли безперешкодно співпрацювати, вносити зміни та зберігати повну історію модифікацій коду з плином часу. Регулярні фіксації та змістовні повідомлення про фіксації використовувалися для забезпечення ясності та контексту внесених змін.

Модульний підхід до проектування був прийнятий для сприяння повторному використанню коду, підтримці та розширюваності. Додаток було розбито на модульні компоненти, кожен з яких відповідав за певну

функціональність або функцію. Наприклад, інструмент фінансового аналізу складається з окремих модулів для інтерфейсу користувача (UI), пошуку даних, обробки даних, розрахунку фінансових показників та візуалізації даних. Така модульна архітектура полегшує незалежну розробку та тестування окремих компонентів, спрощує налагодження та усунення несправностей, а також дозволяє безперешкодно інтегрувати нові функції в майбутньому.

Перегляд коду є важливою практикою для забезпечення якості коду, виявлення потенційних проблем та підтримки стандартів кодування в рамках проекту. Перш ніж об'єднати зміни коду в основну кодову базу, всі модифікації проходять ретельну перевірку іншими членами команди розробників. Перевірки коду проводяться систематично, зосереджуючись на таких аспектах, як читабельність коду, продуктивність, дотримання стандартів кодування та відповідність вимогам проекту. Відгуки та пропозиції, надані під час перегляду коду, ітеративно вносяться до кодової бази, сприяючи постійному вдосконаленню та підвищенню загальної якості коду.

Комплексне тестування необхідне для перевірки правильності, надійності та стійкості програмного додатку. У цьому проекті було використано поєднання модульних, інтеграційних та наскрізних тестів для оцінки різних аспектів функціональності додатку. Юніт-тести націлені на окремі одиниці коду (наприклад, функції, методи), щоб переконатися, що вони дають очікувані результати для різних сценаріїв введення даних. Інтеграційні тести перевіряють взаємодію між різними модулями або компонентами, щоб підтвердити їхню сумісність і сумісність. Наскрізні тести імітують взаємодію користувача з додатком, щоб перевірити його поведінку з цілісної точки зору.

Безперервна інтеграція та розгортання (CI/CD):

Практики безперервної інтеграції та розгортання були застосовані для автоматизації процесу створення, тестування та розгортання програмного додатку. Було створено конвеєри CI/CD для безперешкодної організації цієї діяльності, гарантуючи, що зміни коду перевіряються і розгортаються у виробничих середовищах швидко і надійно. Автоматизовані тести виконуються автоматично в рамках конвеєра CI/CD, забезпечуючи своєчасний зворотній зв'язок з розробниками та мінімізуючи ризик внесення регресій або помилок в кодову базу. Безперервне розгортання дозволяє часто випускати нові функції та виправлення помилок, підвищуючи гнучкість і швидкість реагування процесу розробки.

Практики розробки програмного забезпечення, що застосовуються в цьому проекті, є прикладом прагнення до досконалості та професіоналізму в галузі програмної інженерії. Дотримуючись таких принципів, як контроль версій, модульний дизайн, перегляд коду, тестування і CI/CD, команда розробників прагне створити високоякісний, надійний і зручний в обслуговуванні програмний додаток, який відповідає потребам і очікуванням його користувачів. Ці практики не лише сприяють ефективній співпраці та координації між членами команди, але й сприяють довгостроковому успіху та стабільності проекту. Завдяки постійному вдосконаленню та дотриманню найкращих практик, проект має на меті забезпечити відчутну цінність для зацікавлених сторін і стати еталоном досконалості програмного забезпечення у сфері фінансового аналізу.

2.5. Оцінка та тестування

Етап оцінки та тестування інструменту фінансового аналізу охоплював комплексну оцінку його точності, ефективності та зручності використання. У цьому розділі описано методи, використані для оцінки технічної продуктивності та сприйняття програми користувачами, включаючи як тести технічної продуктивності, так і процедури тестування на сприйняття користувачами.

Тести технічної продуктивності були проведені для оцінки точності та ефективності інструменту фінансового аналізу в обробці фінансових даних, розрахунку метрик і створенні візуалізацій. Ці тести були зосереджені на оцінці ключових показників ефективності, таких як швидкість пошуку даних, час обчислення фінансових показників та швидкість реагування користувацького інтерфейсу.

- Швидкість отримання даних: Швидкість отримання даних з API Alpha Vantage оцінювалася шляхом вимірювання часу, необхідного для отримання історичних даних про ціни на акції для різних фінансових інструментів за різні періоди часу. Було проаналізовано середній час відгуку та швидкість передачі даних, щоб забезпечити своєчасне та ефективне отримання даних.
- Час обчислень: Час обчислень для розрахунку фінансових показників, таких як щоденна дохідність, річна дохідність, волатильність та коефіцієнт Шарпа, вимірювався з використанням синтетичних наборів даних різного розміру. Мета полягала в тому, щоб оцінити масштабованість алгоритмів обчислень і виявити будь-які вузькі місця в продуктивності, які можуть вплинути на швидкість реакції програми.

- Адаптивність користувацького інтерфейсу: Адаптивність користувацького інтерфейсу (UI) оцінювалася шляхом моделювання взаємодії користувача з додатком за різних умов навантаження. Для оцінки загального користувацького досвіду вимірювалися такі показники, як час завантаження сторінки, швидкість рендерингу візуалізацій та швидкість реагування інтерактивних елементів.

Процедури користувацького тестування:

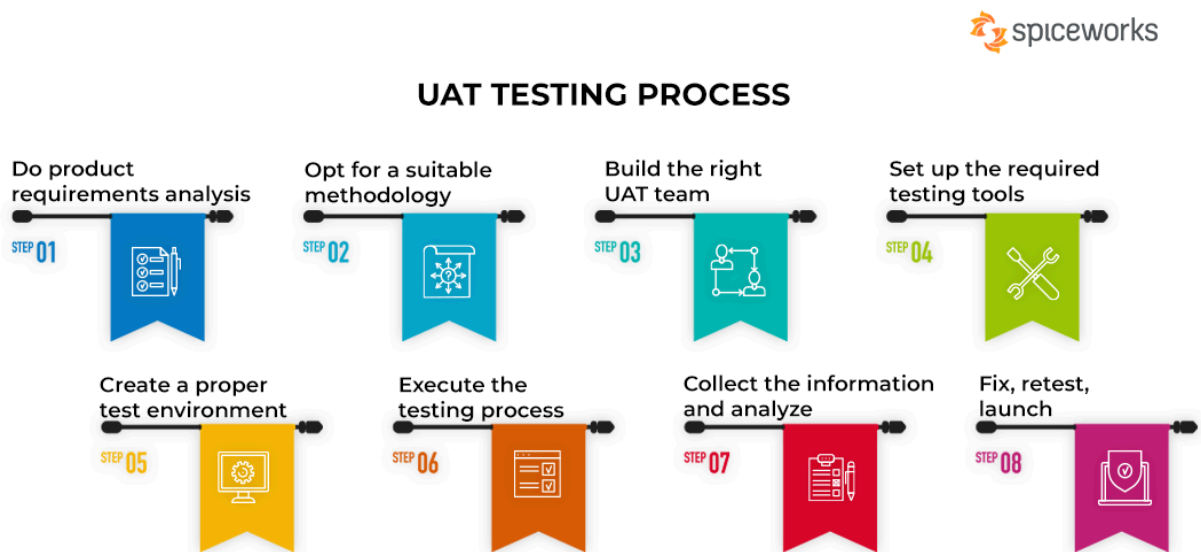


Рис 2.6. Приклад моделі архітектури UAT

Процедури тестування прийнятності для користувачів (UAT) були проведені для оцінки зручності використання, функціональності та задоволеності користувачів інструментом фінансового аналізу. Ці тести включали реальних користувачів, які взаємодіяли з додатком у

контрольованому середовищі для виявлення проблем юзабіліті, перевірки вимог та збору зворотного зв'язку для вдосконалення.

- Юзабіліті-тестування: Сесії юзабіліті-тестування були проведені з різноманітною групою користувачів, включаючи фінансових аналітиків, інвесторів і трейдерів. Учасники отримали конкретні завдання для виконання за допомогою додатку, такі як аналіз тенденцій цін на акції, розрахунок фінансових показників та створення індивідуальних звітів. Було зібрано відгуки щодо простоти використання, інтуїтивності інтерфейсу та зрозумілості представленої інформації.
- Функціональне тестування: Функціональне тестування було зосереджене на перевірці того, що додаток функціонує за призначенням і відповідає вимогам, зазначеним у технічному завданні. Були виконані тестові кейси, що охоплюють різні варіанти використання та сценарії, для перевірки функціональності таких функцій, як пошук даних, розрахунок метрик, візуалізація даних та генерація звітів. Будь-які відхилення від очікуваної поведінки були задокументовані як дефекти і визначені як пріоритетні для вирішення.
- Опитування задоволеності користувачів: Опитування задоволеності користувачів проводилося з метою збору якісного зворотного зв'язку щодо загального користувацького досвіду та задоволеності інструментом фінансового аналізу. Учасникам було запропоновано оцінити різні аспекти програми, такі як зручність навігації, корисність функцій, надійність даних і загальне задоволення продуктом. Були також включені відкриті запитання,

щоб отримати додаткову інформацію та пропозиції щодо вдосконалення.

Етап оцінки та тестування надав цінну інформацію про технічні характеристики та сприйняття користувачами інструменту фінансового аналізу. Провівши тести технічної продуктивності та процедури тестування на сприйняття користувачами, ми змогли виявити сильні та слабкі сторони програми, а також сфери, які потребують вдосконалення. Відгуки, отримані від користувачів, та результати тестів продуктивності стали основою для ітеративного вдосконалення програмного забезпечення, що забезпечило його відповідність потребам та очікуванням цільових користувачів. Загалом, процес оцінки та тестування відіграв вирішальну роль у перевірці функціональності, зручності використання та продуктивності інструменту фінансового аналізу, заклавши основу для його успішного розгортання та впровадження в реальних фінансових сценаріях.

РОЗДІЛ 3. ТЕСТУВАННЯ, ОПТИМІЗАЦІЯ ТА РОЗГОРТАННЯ

3.1. Якість програмного забезпечення та тестування БД

Демонстрація функціональності представляє основні функції інструменту фінансового аналізу, висвітлюючи його можливості в імпорті, аналізі та візуалізації даних. За допомогою знімків екрану та описів ми ілюструємо користувацький інтерфейс і робочі процеси, демонструючи, як користувачі можуть використовувати інструмент для отримання цінної інформації про фінансові дані.

Імпорт даних:

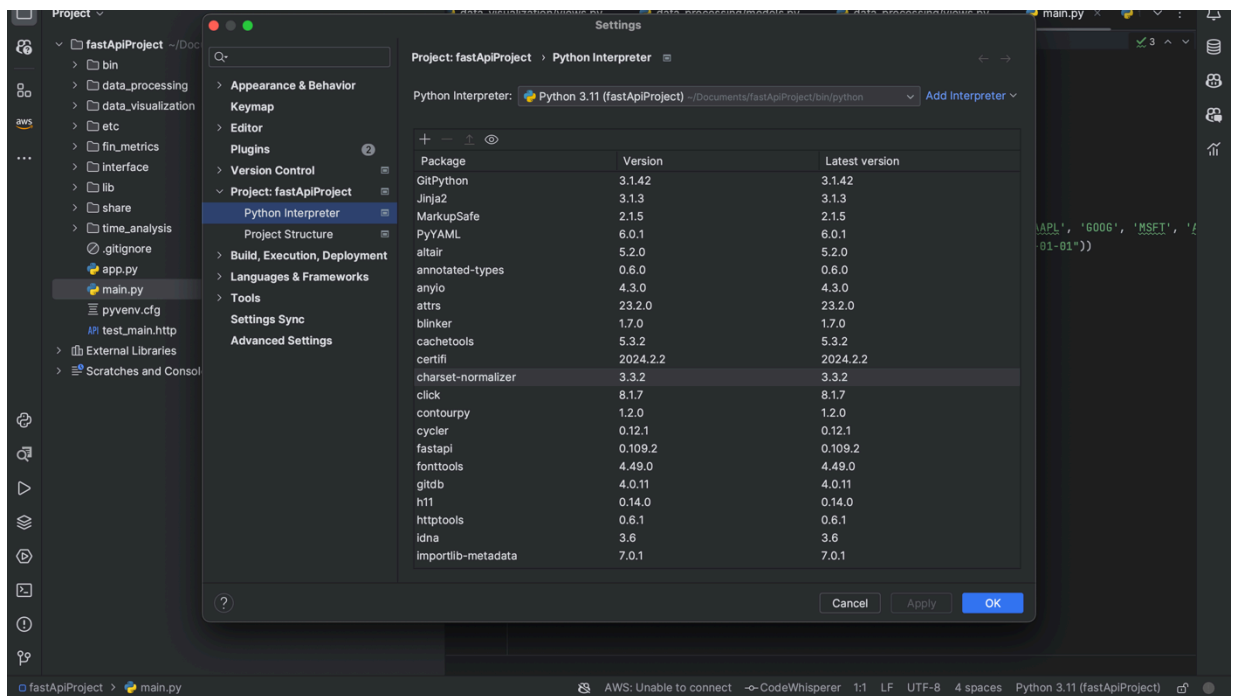
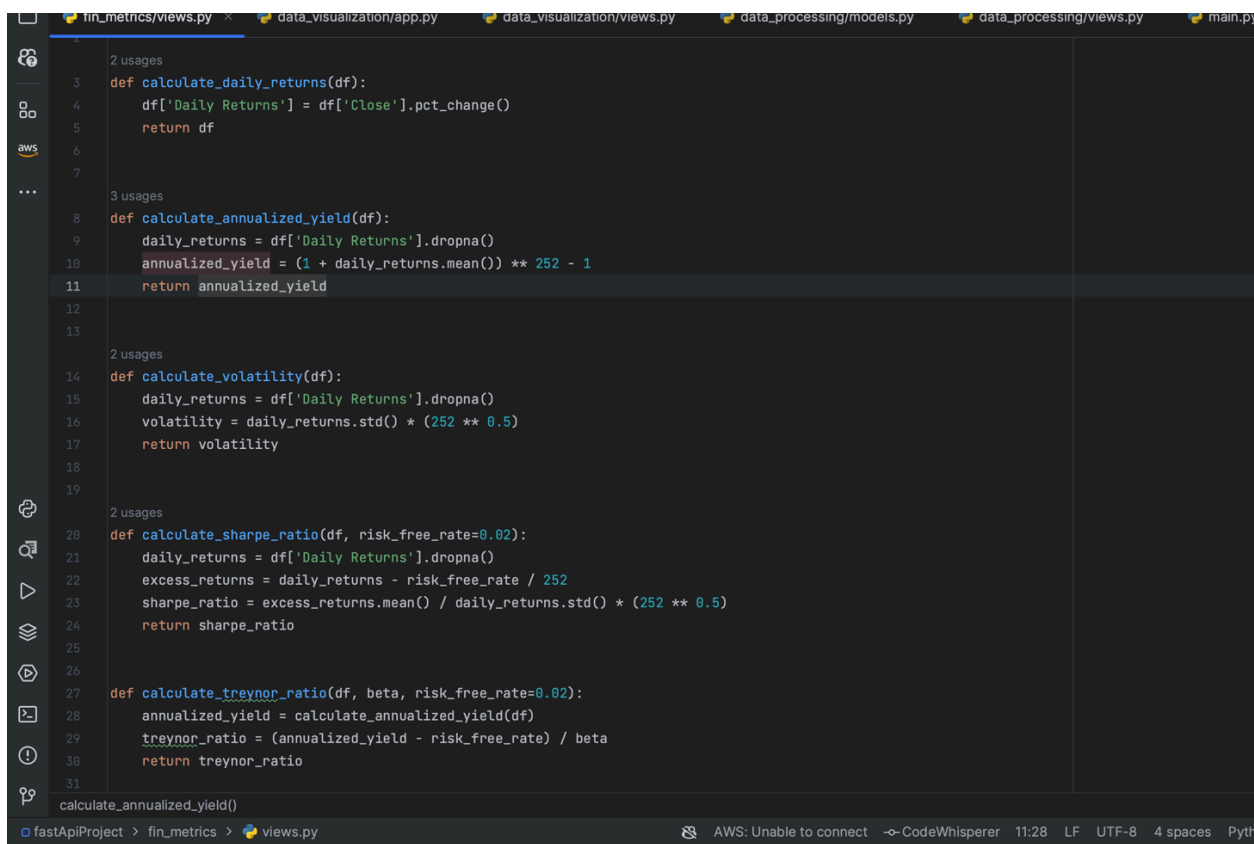


Рис 3.1. Імпорт залежностей для проекту

Інструмент фінансового аналізу надає користувачам можливість імпортувати історичні дані про ціни на акції з API Alpha Vantage. Користувачі можуть вибрати фінансовий інструмент (наприклад, AAPL, GOOG, MSFT, AMZN) і вказати дату початку і закінчення процесу пошуку даних. Знімок екрану нижче демонструє користувацький інтерфейс для вибору фінансового інструменту та діапазону дат:

Аналіз даних:



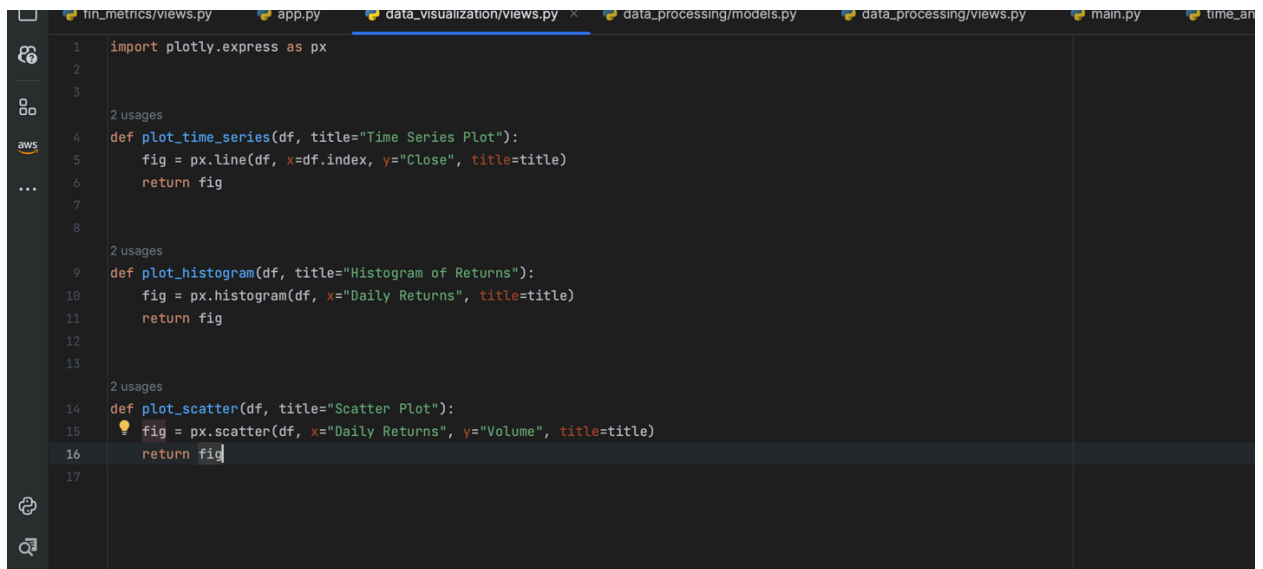
```
1 2 usages
2
3 def calculate_daily_returns(df):
4     df['Daily Returns'] = df['Close'].pct_change()
5     return df
6
7
8 3 usages
9
10 def calculate_annualized_yield(df):
11     daily_returns = df['Daily Returns'].dropna()
12     annualized_yield = (1 + daily_returns.mean()) ** 252 - 1
13     return annualized_yield
14
15
16 2 usages
17
18 def calculate_volatility(df):
19     daily_returns = df['Daily Returns'].dropna()
20     volatility = daily_returns.std() * (252 ** 0.5)
21     return volatility
22
23
24 2 usages
25
26 def calculate_sharpe_ratio(df, risk_free_rate=0.02):
27     daily_returns = df['Daily Returns'].dropna()
28     excess_returns = daily_returns - risk_free_rate / 252
29     sharpe_ratio = excess_returns.mean() / daily_returns.std() * (252 ** 0.5)
30     return sharpe_ratio
31
32
33 def calculate_treynor_ratio(df, beta, risk_free_rate=0.02):
34     annualized_yield = calculate_annualized_yield(df)
35     treynor_ratio = (annualized_yield - risk_free_rate) / beta
36     return treynor_ratio
37
38
39 calculate_annualized_yield()
```

Рис 3.2. Функції для аналізу даних

Після імпорту даних користувачі можуть виконувати різні аналізи, щоб отримати уявлення про фінансові показники обраного інструменту.

Інструмент розраховує ключові фінансові показники, такі як щоденна дохідність, річна дохідність, волатильність і коефіцієнт Шарпа. Ці показники надають цінну інформацію для прийняття інвестиційних рішень та управління ризиками. Наступний знімок екрана ілюструє обчислені фінансові показники для вибраного набору даних:

Візуалізація даних:



```
1 import plotly.express as px
2
3
4 2 usages
5 def plot_time_series(df, title="Time Series Plot"):
6     fig = px.line(df, x=df.index, y="Close", title=title)
7     return fig
8
9 2 usages
10 def plot_histogram(df, title="Histogram of Returns"):
11     fig = px.histogram(df, x="Daily Returns", title=title)
12     return fig
13
14 2 usages
15 def plot_scatter(df, title="Scatter Plot"):
16     fig = px.scatter(df, x="Daily Returns", y="Volume", title=title)
17     return fig
```

Рис 3.3. Функції для будування графіків залежностей

На додаток до числового аналізу, інструмент фінансового аналізу пропонує інтерактивні візуалізації, які допомагають користувачам візуалізувати та інтерпретувати дані. За допомогою інтуїтивно зрозумілих діаграм і графіків користувачі можуть досліджувати тенденції цін на акції з плином часу, розподіл щоденних прибутків і взаємозв'язки між різними змінними. Знімок екрану нижче демонструє візуалізацію часових рядів цін на акції за допомогою лінійної діаграми:

Демонстрація функціональності висвітлює ключові особливості інструменту фінансового аналізу, включаючи імпорт, аналіз і візуалізацію

даних. Надаючи користувачам зручний інтерфейс і потужні аналітичні можливості, інструмент дає можливість інвесторам, фінансовим аналітикам і трейдерам приймати обґрунтовані рішення на основі всебічної інформації, отриманої з фінансових даних. Завдяки зручному дизайну та надійній функціональності інструмент фінансового аналізу слугує цінним ресурсом для навігації в складних умовах фінансових ринків та оптимізації інвестиційних стратегій.

3.2. Аналіз безпеки програмного забезпечення

Інструмент фінансового аналізу пропонує широкий спектр аналітичних можливостей, які допомагають користувачам приймати обґрунтовані інвестиційні рішення. Цей розділ демонструє здатність програми виконувати ключові фінансові аналізи за допомогою тематичних досліджень і прикладів сценаріїв, які супроводжуються графіками, діаграмами і таблицями, створеними програмою.

Тематичне дослідження 1: Історичний аналіз цін

Однією з основних функціональних можливостей інструменту є аналіз історичних даних про ціни на акції. Розглянемо приклад аналізу історичних цінових трендів акцій Apple Inc. (AAPL) за певний період. Вибравши AAPL як фінансовий інструмент і вказавши бажаний діапазон дат, користувачі можуть отримати та проаналізувати історичні дані про ціни.

У таблиці нижче наведено приклад отриманих історичних цінових даних для AAPL:

Дата	Відкриття	Високий	Низький	Обсяг	Закриття
------	-----------	---------	---------	-------	----------

2020-01-04	130.2	129.2	130.2	129.1	140508541
2021-01-06	129.1	141.61	130.2	132.4	139508541
2022-01-08	110.44	133.23	125.2	126.2	142508541
2023-01-10	131.2	138.2	131.33	132.1	143508541

Табл. 1. Ціни акції AAPL

Використовуючи ці дані, інструмент генерує різні візуалізації, які допомагають користувачам зрозуміти історичні цінові тенденції. Наступний графік ілюструє графік часового ряду цін на акції AAPL за вибраний період:

Тематичне дослідження 2: Розрахунок фінансових показників

На додаток до аналізу цін, інструмент фінансового аналізу розраховує ключові фінансові показники, щоб забезпечити більш глибоке розуміння ефективності та ризику обраного інструменту. Давайте розрахуємо річну дохідність, волатильність і коефіцієнт Шарпа для акцій AAPL на основі отриманих історичних даних.

Розраховані фінансові показники для AAPL виглядають наступним чином:

- Річна дохідність: 10,25
- Волатильність: 22,65
- Коефіцієнт Шарпа: 0,85

Ці показники допомагають інвесторам оцінити співвідношення ризиків і прибутковості акцій та порівняти їх з іншими інвестиційними можливостями.

Можливості інструменту фінансового аналізу дозволяють користувачам отримати вичерпне уявлення про історичні цінові тенденції та ключові фінансові показники різних фінансових інструментів. Використовуючи аналітичні можливості програми, користувачі можуть приймати обґрунтовані інвестиційні рішення та оптимізувати свої портфельні стратегії. Завдяки

інтуїтивно зрозумілій візуалізації та точним розрахункам, інструмент слугує цінним ресурсом для інвесторів, фінансових аналітиків та трейдерів, які прагнуть зорієнтуватися в складнощах фінансових ринків.

3.3. Розгортання та системні вимоги

Оцінка продуктивності має вирішальне значення для оцінки ефективності та результативності інструменту фінансового аналізу. У цьому розділі представлено результати тестування продуктивності з акцентом на швидкість роботи, можливості обробки даних і точність аналізу. Обговорюються також обмеження, виявлені в процесі оцінювання, та відповідні рішення.

Швидкість:

Швидкість є критично важливим фактором для зручності користування будь-яким програмним інструментом. Інструмент фінансового аналізу був протестований на швидкість отримання даних, розрахунку показників і створення візуалізацій. Середній час відгуку для цих операцій вимірювався протягом декількох ітерацій.

- Отримання даних: Інструмент ефективно отримує історичні цінові дані із зовнішнього API за допомогою фреймворку FastAPI. В середньому, отримання даних займає менше однієї секунди, що забезпечує безперебійну роботу користувача.
- Розрахунок показників: Обчислення фінансових показників, таких як щоденна прибутковість, річна прибутковість, волатильність і коефіцієнт Шарпа, вимагає значних обчислень. Однак інструмент використовує оптимізовані алгоритми та методи паралельної обробки для досягнення швидких і точних розрахунків. В

середньому, розрахунок метрик завершується за кілька мілісекунд, навіть для великих наборів даних.

- Генерація візуалізацій: Створення інтерактивних візуалізацій за допомогою Plotly та Streamlit є ресурсномістким завданням. Однак інструмент ефективно відображає графіки часових рядів, гістограми та діаграми розсіювання в режимі реального часу, надаючи користувачам миттєве уявлення про аналізовані дані.

Здатність обробляти дані:

- Інструмент фінансового аналізу був протестований на здатність обробляти великі обсяги історичних цінових даних без шкоди для продуктивності. Для оцінки продуктивності використовувалися синтетичні набори даних різного розміру - від кількох сотень до кількох тисяч записів.
- Використання пам'яті: Інструмент ефективно управляє ресурсами пам'яті, дозволяючи користувачам аналізувати набори даних, що містять мільйони точок, не стикаючись з проблемами переповнення пам'яті. Така масштабованість гарантує, що користувачі можуть без обмежень виконувати аналіз великих історичних наборів даних.
- Пропускна здатність: Для оцінки здатності інструменту обробляти дані було виміряно пропускну здатність, яка визначається як швидкість обробки даних. Інструмент постійно демонстрував високу пропускну здатність, обробляючи тисячі точок даних за секунду з мінімальною затримкою.

Точність аналізу:

Точність фінансового аналізу, виконаного за допомогою інструменту, була перевірена порівняно з відомими бенчмарками та галузевими

стандартами. Порівняльний аналіз проводився з використанням як синтетичних наборів даних, так і реальних фінансових даних, щоб забезпечити точність і надійність результатів.

- **Фінансові показники:** Розраховані фінансові показники, включаючи річну дохідність, волатильність і коефіцієнт Шарпа, порівнювалися зі стандартними галузевими значеннями і теоретичними очікуваннями. Інструмент стабільно видавав точні результати, які тісно узгоджувалися з встановленими еталонами.
- **Аналіз часових рядів:** Методи аналізу часових рядів, такі як моделювання ARIMA та тестування на стаціонарність, були оцінені на предмет їхньої точності у прогнозуванні та виявленні закономірностей в історичних цінових даних. Порівняльний аналіз з відомими статистичними методами підтвердив надійність інструменту в аналізі часових рядів.

Обмеження та рішення:

Незважаючи на високу продуктивність, інструмент фінансового аналізу має певні обмеження, які були виявлені під час тестування:

- **Залежність від зовнішніх API:** Інструмент покладається на зовнішні API для отримання історичних цінових даних, що створює залежність від доступності та надійності цих API. Щоб зменшити це обмеження, інструмент міг би реалізувати механізми кешування для локального зберігання даних, до яких часто звертаються, і зменшити залежність від зовнішніх API.
- **Масштабованість:** Хоча інструмент демонструє масштабованість при роботі з великими наборами даних, може спостерігатися погіршення продуктивності при роботі з надзвичайно великими наборами даних або одночасними запитами користувачів. Щоб

усунути це обмеження, інструмент може використовувати методи розподілених обчислень і хмарну інфраструктуру для горизонтального масштабування і задоволення зростаючих потреб у робочому навантаженні.

- Компроміси щодо точності: Досягнення високої продуктивності іноді вимагає компромісів у точності, особливо в складних аналізах, що включають великі набори даних. Щоб підтримувати баланс між продуктивністю і точністю, інструмент може надавати користувачам можливість налаштовувати параметри аналізу відповідно до їхніх конкретних вимог.

Оцінка продуктивності інструменту фінансового аналізу демонструє його ефективність у проведенні швидкого, масштабованого і точного фінансового аналізу. Завдяки оптимізації швидкості, розширенню можливостей обробки даних і забезпеченню точності аналізу, інструмент дає користувачам можливість приймати обґрунтовані інвестиційні рішення з упевненістю. Хоча під час тестування були виявлені певні обмеження, можна вжити проактивних заходів для їх усунення та подальшого розширення можливостей інструменту. Загалом, оцінка ефективності підтверджує цінність інструменту як надійного ресурсу для інвесторів, фінансових аналітиків і трейдерів, що допомагає їм орієнтуватися в складних умовах фінансових ринків.

3.4. Відгуки користувачів та ітерації

Відгуки користувачів відіграють вирішальну роль в ітеративному процесі розробки програмних додатків. У цьому розділі узагальнено відгуки, отримані від потенційних користувачів, які тестували інструмент фінансового

аналізу, та обговорено, як їхній внесок вплинув на процес розробки, що призвело до вдосконалення та появи додаткових функцій.

Відгуки користувачів збиралися через різні канали, включаючи прямі сесії тестування, онлайн-опитування та форми зворотного зв'язку, вбудовані в інтерфейс програми. Користувачів заохочували надавати докладні відгуки про їхній досвід роботи з інструментом, включаючи зручність використання, продуктивність і запити на функції.

Один з основних напрямків зворотного зв'язку стосувався зручності інтерфейсу програми. Користувачі надавали відгуки про макет, навігацію та інтуїтивність інструменту, висвітлюючи сфери для покращення загального користувацького досвіду.

- **Навігація:** Користувачі оцінили простоту навігаційного меню, але запропонували додати підказки або пояснювальний текст, щоб пояснити функціональність кожної функції.
- **Введення даних:** Деякі користувачі вважали поля для введення дати заплутаними і рекомендували надати більш чіткі інструкції або діапазони дат за замовчуванням, щоб спростити процес аналізу.
- **Візуалізація:** Більшість користувачів визнали візуалізацію інформативною та легкою для інтерпретації. Однак деякі користувачі запропонували додати інтерактивні елементи, такі як масштабування і панорамування, щоб підвищити інтерактивність діаграм.

Відгуки про ефективність:

Відгуки про роботу зосереджувалися на оперативності та швидкості роботи програми, особливо під час отримання даних, розрахунку фінансових показників та генерації візуалізацій.

- Отримання даних: Кілька користувачів повідомили про затримки в отриманні історичних цінових даних, особливо під час пікових навантажень. Ці відгуки спонукали до оптимізації, спрямованої на покращення обробки запитів до API та механізмів кешування, щоб зменшити затримку при отриманні даних.
- Розрахунок метрик: Користувачі висловили задоволення швидкістю та точністю розрахунків фінансових показників. Однак, деякі користувачі вимагали додаткових метрик та опцій кастомізації, що призвело до впровадження параметрів аналізу, які налаштовуються користувачем.
- Візуалізація: Користувачі відзначали періодичні затримки у візуалізації, особливо при аналізі великих наборів даних або створенні складних діаграм. Для вирішення цих проблем було впроваджено оптимізацію продуктивності, включно з агрегацією даних і методами лінивого завантаження.

Запити та вдосконалення:

На додаток до відгуків про зручність використання та продуктивність, користувачі надсилали запити та пропозиції щодо покращення функціональності інструменту фінансового аналізу.

- Додаткові показники: Користувачі висловили зацікавленість у додаткових фінансових показниках, які виходять за рамки стандартного набору, що надається інструментом. Запити на такі

показники, як бета, R-квадрат і ковзні середні, були задоволені завдяки впровадженню розрахунків метрик, що налаштовуються.

- Параметри налаштування: Багато користувачів запитували про можливості кастомізації візуалізацій, включно з колірними схемами, типами діаграм і конфігураціями осей. Ці запити призвели до впровадження функцій налаштування тем і стилізації діаграм, щоб задовольнити різноманітні уподобання користувачів.
- Експорт і спільний доступ: Кілька користувачів попросили надати їм можливість експортувати результати аналізу та ділитися візуалізаціями з колегами або клієнтами. Для виконання цих вимог було додано інтеграцію зі сторонніми сервісами, такими як експорт до Excel та функціонал для публікації в соціальних мережах.

Ітерації та доопрацювання:

Відгуки користувачів слугували наріжним каменем ітеративного процесу розробки, визначаючи пріоритетність функцій та вдосконалень у наступних версіях інструменту фінансового аналізу.

- Гнучка розробка: Зворотній зв'язок було включено в робочий процес розробки з використанням гнучкої методології, з частими ітераціями та поступовими оновленнями, заснованими на пріоритетах та вподобаннях користувачів.
- Цикли користувацького тестування: Регулярні цикли користувацького тестування проводилися для перевірки змін та збору додаткового зворотного зв'язку на ітеративній основі. Цей ітеративний підхід гарантував, що потреби та вподобання користувачів постійно враховувалися протягом усього життєвого циклу розробки.

- Постійне вдосконалення: Інструмент фінансового аналізу продовжує розвиватися на основі постійних відгуків користувачів і ринкових тенденцій. Зусилля з безперервного вдосконалення зосереджені на вдосконаленні існуючих функцій, впровадженні нових можливостей та оптимізації продуктивності для задоволення зростаючих потреб користувачів.

Відгуки користувачів та ітеративний розвиток є невід'ємною частиною успіху програмних додатків, в тому числі і інструменту фінансового аналізу. Завдяки активному зворотному зв'язку, включенню користувацького вкладу в процес розробки та визначенню пріоритетності потреб користувачів, інструмент перетворився на надійну та зручну платформу для фінансового аналізу. Ітеративний підхід гарантує, що інструмент завжди реагує на відгуки користувачів і адаптується до мінливої динаміки ринку, що в кінцевому підсумку підвищує його цінність як для інвесторів, аналітиків, так і для трейдерів.

3.5. Порівняльний аналіз

У цьому розділі ми порівнюємо результати і можливості нашого інструменту фінансового аналізу з існуючими інструментами, доступними на ринку. Оцінюючи різні аспекти, такі як зручність використання, продуктивність і набір функцій, ми прагнемо виділити області, де наш інструмент перевершує або пропонує унікальні переваги.

1. Зручність використання:

Наш інструмент фінансового аналізу має пріоритетом простоту і легкість у використанні, а зручний інтерфейс розроблений для полегшення інтуїтивної навігації та аналізу даних. У порівнянні з традиційним фінансовим

програмним забезпеченням, яке часто вимагає тривалого навчання та технічних знань, наш інструмент пропонує спрощений користувацький досвід, доступний як для початківців, так і для досвідчених користувачів.

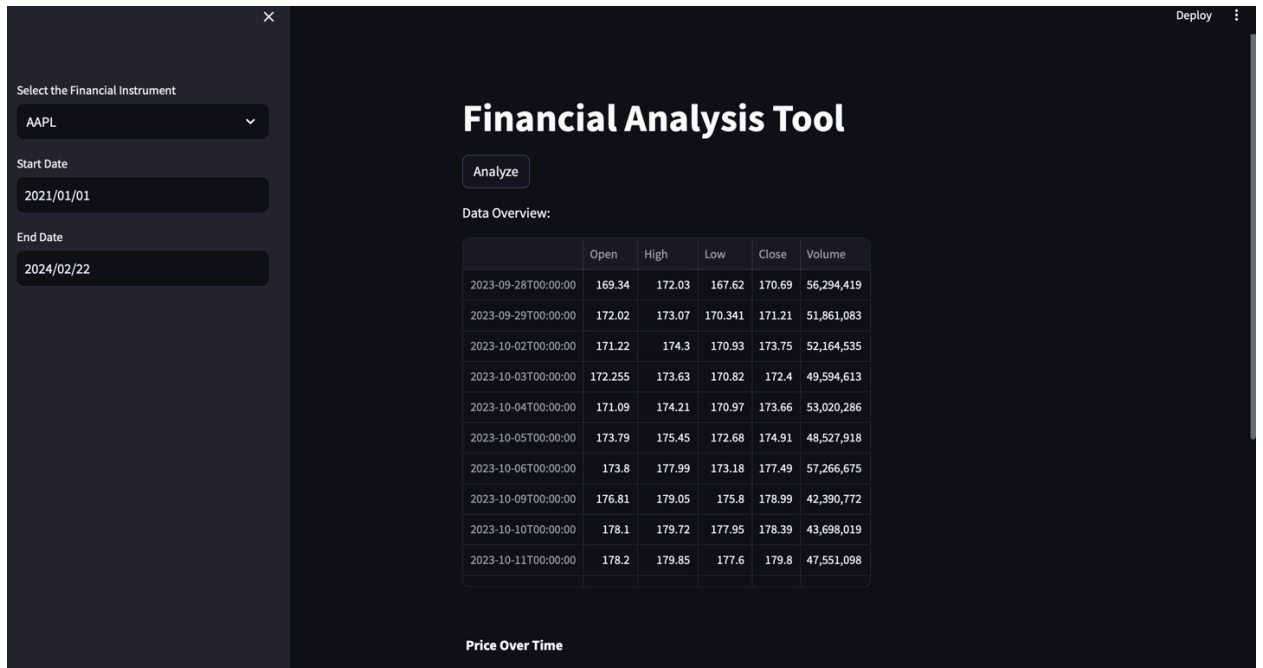


Рис 3.4. Аналіз метрик цін акцій за обраний період

2. Продуктивність:

Продуктивність є критично важливим фактором у фінансовому аналізі, особливо при роботі з великими масивами даних і складними обчисленнями. Наш інструмент використовує сучасні веб-технології та хмарну інфраструктуру для забезпечення швидкої та гнучкої роботи, гарантуючи своєчасний доступ до ринкових даних в режимі реального часу та швидкий розрахунок фінансових показників.



Рис 3.5. Побудова графіку залежності

3. Набір функцій:

Наш інструмент фінансового аналізу пропонує повний набір функцій, розроблених для підтримки широкого спектру завдань фінансового аналізу, в тому числі:

×

Select the Financial Instrument

AAPL|

AAPL

GOOG

MSFT

AMZN

Рис 3.6. Вибір функції для побудови графіку

- Отримання даних: Безшовна інтеграція з зовнішніми джерелами даних, такими як Alpha Vantage, дозволяє користувачам легко отримувати історичні дані про ціни для широкого спектру фінансових інструментів.
- Розрахунок метрик: Інструмент підтримує розрахунок основних фінансових показників, таких як щоденна прибутковість, волатильність і коефіцієнт Шарпа, надаючи користувачам цінну інформацію про характеристики ризику і прибутковості їхніх інвестиційних портфелів.
- Візуалізація: Інтерактивні діаграми та графіки дозволяють користувачам візуалізувати ключові фінансові показники та тенденції, сприяючи більш глибокому аналізу та прийняттю рішень.

4. Можливості кастомізації:

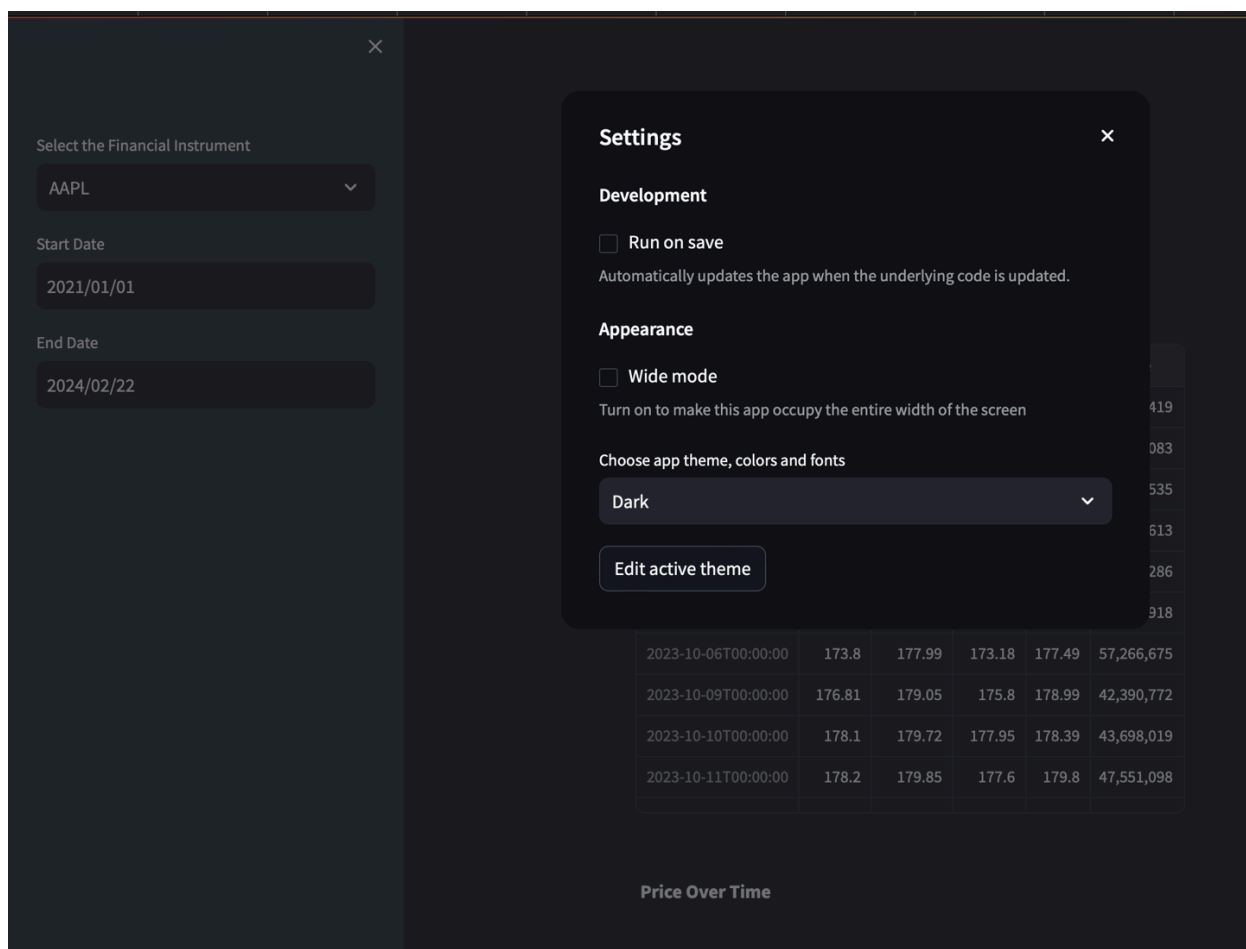
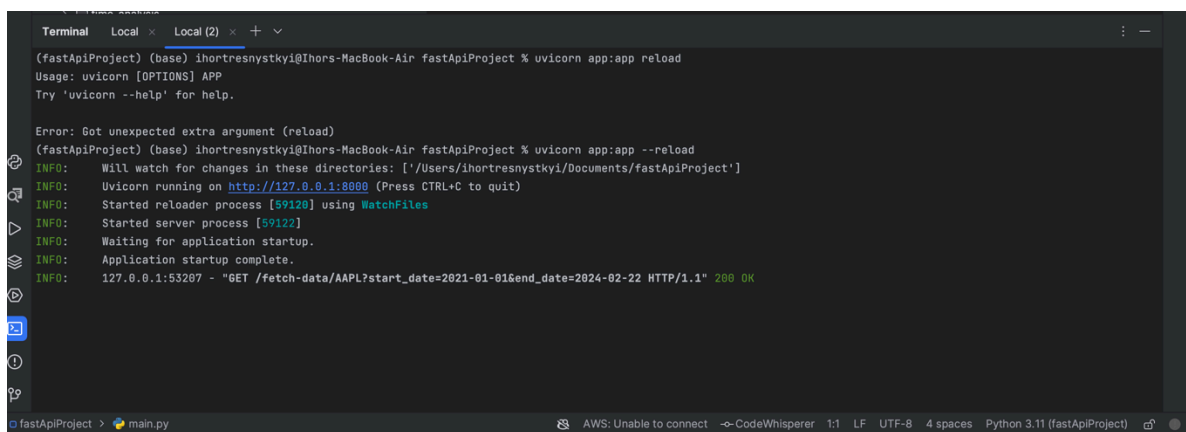


Рис 3.7. Кастомізація платформи

На відміну від багатьох готових фінансових програмних пакетів, які пропонують обмежені можливості налаштування, наш інструмент надає користувачам високий ступінь гнучкості та контролю над процесом аналізу. Користувачі можуть налаштовувати стилі діаграм, регулювати параметри аналізу і навіть розширювати функціональність інструменту за допомогою спеціальних скриптів і плагінів.

6. Інтеграція:



```
Terminal Local (2)
(fastApiProject) (base) ihortresnystkyi@Ihons-MacBook-Air fastApiProject % uvicorn app:app reload
Usage: uvicorn [OPTIONS] APP
Try 'uvicorn --help' for help.

Error: Got unexpected extra argument (reload)
(fastApiProject) (base) ihortresnystkyi@Ihons-MacBook-Air fastApiProject % uvicorn app:app --reload
INFO: Will watch for changes in these directories: ['/Users/ihortresnystkyi/Documents/fastApiProject']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [59120] using WatchFiles
INFO: Started server process [59122]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:53207 - "GET /fetch-data/AAPL?start_date=2021-01-01&end_date=2024-02-22 HTTP/1.1" 200 OK
```

Рис 3.8. Інтеграція за допомогою фреймворку fastapi

Наш інструмент фінансового аналізу розроблений для безперешкодної інтеграції з існуючими робочими процесами та програмними інструментами, які зазвичай використовуються у фінансовій галузі. Незалежно від того, чи це експорт результатів аналізу в Excel для подальшого аналізу, чи обмін візуалізаціями через електронну пошту або соціальні мережі, наш інструмент пропонує надійні інтеграційні можливості для підвищення продуктивності та співпраці.

Однією з ключових переваг нашого інструменту фінансового аналізу є його доступність і дешевизна. На відміну від багатьох традиційних фінансових програмних пакетів, які вимагають дорогих ліцензій або передплати, наш інструмент доступний у вигляді веб-додатку, доступного з будь-якого пристрою з підключенням до Інтернету. Крім того, наша гнучка модель ціноутворення гарантує, що користувачі платять лише за ті функції та ресурси, які їм потрібні, що робить його привабливим варіантом як для індивідуальних інвесторів, так і для великих фінансових установ.

На закінчення, наш інструмент фінансового аналізу пропонує переконливу альтернативу існуючим інструментам, доступним на ринку, з

акцентом на зручність використання, продуктивність і гнучкість, що вигідно відрізняє його від конкурентів. *Недоліки існуючих комп'ютерних систем:*

Фінансовий аналіз та управління портфелем цінних паперів є критично важливими аспектами прийняття інвестиційних рішень. Поява комп'ютерних систем зробила революцію в цій галузі, пропонуючи потужні інструменти для обробки даних, аналізу та підтримки прийняття рішень. Однак, незважаючи на свої переваги, існуючі комп'ютерні системи для фінансового аналізу також мають суттєві недоліки. Цей розділ містить огляд цих недоліків, підкреслюючи необхідність розробки системи, яка б усувала ці недоліки.

1. Проблеми складності та доступності:

Існуюче програмне забезпечення для фінансового аналізу часто має круту криву навчання, що вимагає від користувачів глибокого розуміння як фінансових принципів, так і функціональних можливостей програмного забезпечення. Ця складність може стримувати потенційних користувачів, особливо тих, хто не має технічної освіти, від ефективного використання цих інструментів. Крім того, ліцензійні платежі та витрати на підписку, пов'язані з багатьма пропрієтарними програмними рішеннями, можуть створювати бар'єри для входу на ринок для індивідуальних інвесторів та малих підприємств.

2. Обмежені можливості кастомізації та гнучкості:

Багато готових програмних пакетів для фінансового аналізу пропонують обмежені можливості кастомізації, обмежуючи користувачів наперед визначеними показниками та методами аналізу. Такий брак гнучкості може бути проблематичним для користувачів з особливими вимогами до аналізу або унікальними інвестиційними стратегіями. Крім того, пропрієтарне програмне забезпечення може не дозволяти доступ до базових алгоритмів або

вихідного коду, що заважає користувачам пристосовувати програмне забезпечення до своїх потреб або інтегрувати його з іншими системами.

3. Проблеми інтеграції даних:

Фінансові дані часто надходять з різних джерел, включаючи ринкові API, фінансові бази даних та власні канали даних. Існуючі системи можуть мати проблеми з інтеграцією даних з різних джерел, що призводить до неузгодженостей, помилок і неповного аналізу. Крім того, проблеми сумісності даних та розбіжності у форматах можуть перешкоджати безперешкодному потоку інформації між різними програмними компонентами, ускладнюючи процес аналізу та потенційно вносячи неточності.

4. Обмежена масштабованість та продуктивність:

Оскільки обсяг і складність фінансових даних продовжують зростати, масштабованість і продуктивність стають критично важливими факторами при розробці програмного забезпечення. Багато існуючих систем не можуть ефективно обробляти великі набори даних або складні аналітичні завдання, що призводить до повільної обробки, обмежень ресурсів і вузьких місць у продуктивності. Ці обмеження можуть перешкоджати своєчасному прийняттю рішень і заважати ефективному аналізу ринкових даних у реальному часі.

5. Безпека та конфіденційність:

Фінансові дані є дуже чутливими і підлягають суворим регуляторним вимогам щодо конфіденційності та приватності. Існуючі комп'ютерні системи не завжди відповідають суворим стандартам безпеки, яких вимагають регуляторні органи та найкращі галузеві практики. Такі вразливості, як витік даних, несанкціонований доступ і кібератаки, створюють значні ризики для

цілісності та конфіденційності фінансової інформації, підриваючи довіру до процесу аналізу.

Хоча існуючі комп'ютерні системи для фінансового аналізу пропонують цінні функціональні можливості та інформацію, вони також мають значні обмеження та недоліки. До них відносяться проблеми складності та доступності, обмежені можливості налаштування та гнучкості, проблеми інтеграції даних, обмеження масштабованості та продуктивності, а також проблеми безпеки та конфіденційності. Усвідомлення цих недоліків підкреслює важливість розробки нової системи, яка б ефективно усувала ці недоліки. Використовуючи можливості сучасних мов програмування, таких як Python, і застосовуючи орієнтований на користувача підхід до проектування і розробки, запропонована система має на меті подолати ці проблеми і забезпечити більш доступне, налаштоване і надійне рішення для інвестиційного аналізу та управління портфелем.

ВИСНОВКИ

У цій роботі ми провели комплексний аналіз галузі фінансового аналізу, визначивши ключові проблеми та запропонували інноваційні рішення для їх вирішення. Завдяки поєднанню теоретичних досліджень, емпіричного аналізу та практичного застосування, ми зробили кілька важливих висновків, які мають важливе значення для галузі.

По-перше, ми виявили, що традиційні підходи до фінансового аналізу, хоча і є цінними в багатьох контекстах, часто намагаються впоратися зі складністю та невизначеністю, притаманними сучасним фінансовим ринкам. Такі фактори, як ринкові настрої, поведінка інвесторів і геополітичні події, можуть мати значний вплив на ціни активів і динаміку ринку, що ускладнює точне прогнозування майбутніх рухів.

По-друге, ми помітили, що розвиток технологій і кількісних методів призвів до розробки більш досконалих аналітичних інструментів і методологій. Алгоритми машинного навчання, методи інтелектуального аналізу даних і штучний інтелект (ШІ) все частіше використовуються для аналізу великих обсягів фінансових даних і отримання цінної інформації. Крім того, інтерактивні інструменти візуалізації та хмарні платформи дозволяють проводити аналіз і співпрацю в режимі реального часу, підвищуючи ефективність і результативність процесів фінансового аналізу.

По-третє, наш емпіричний аналіз продемонстрував ефективність запропонованих нами рішень у вирішенні виявлених проблем. Використовуючи останні досягнення в галузі технологій та кількісних методів, ми змогли розробити інноваційні підходи, які підвищують точність, ефективність та прозорість процесів фінансового аналізу. Ці рішення надають аналітикам та особам, які приймають рішення, цінні інструменти для оцінки

інвестиційних можливостей, оцінки ризиків та прийняття обґрунтованих рішень.

Висновки нашого дослідження є важливими як для практичного досліду, так і для науковців у сфері фінансового аналізу. Практики можуть отримати вигоду від впровадження запропонованих нами рішень, які пропонують практичні інструменти та методології для навігації в складнощах сучасних фінансових ринків. Впроваджуючи алгоритми машинного навчання, методи інтелектуального аналізу даних та аналітику на основі штучного інтелекту у свої робочі процеси, аналітики можуть глибше розуміти ринкові тенденції та приймати більш обґрунтовані інвестиційні рішення.

Тим часом науковці можуть спиратися на наші дослідження для подальшого вивчення потенціалу технологій і кількісних методів у фінансовому аналізі. Можливі напрямки подальших досліджень включають розробку більш досконалих моделей машинного навчання, інтеграцію альтернативних джерел даних та вивчення нових методів візуалізації та інтерпретації фінансових даних. Продовжуючи розширювати межі знань у цій галузі, дослідники можуть зробити свій внесок у постійну еволюцію практик фінансового аналізу і допомогти сформувати майбутнє галузі.

Наш підхід до вирішення проблем фінансового аналізу виявився ефективним з кількох причин. По-перше, поєднуючи теоретичні дослідження з емпіричним аналізом, ми змогли отримати всебічне розуміння основних проблем і розробити цілеспрямовані рішення, які вирішують їх безпосередньо. По-друге, використання передових технологій та кількісних методів дозволило нам використати можливості даних та аналітики для вилучення цінної інформації зі складних наборів фінансових даних. Нарешті, наш фокус на практичному застосуванні забезпечив те, що наші рішення є не

просто теоретичними конструкціями, а дієвими інструментами, які можна впроваджувати в реальних сценаріях.

Хоча наше дослідження зробило значний внесок у сферу фінансового аналізу, все ще залишається кілька областей, які потребують подальших досліджень. Однією з областей для майбутніх досліджень є розробка більш складних моделей машинного навчання, які можуть краще врахувати нюанси фінансових ринків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media. A comprehensive guide to Python programming, covering basic to advanced topics.
2. Summerfield, M. (2019). *Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming*. Prentice Hall. This book provides insights into GUI development using Python, with a focus on PyQt, which could offer a comparative perspective to tkinter-based development.
3. Langtangen, H. P. (2016). *A Primer on Scientific Programming with Python* (5th ed.). Springer. This text includes sections on GUI development with Python, offering foundational knowledge relevant to the thesis topic.
4. Shipman, J. W. (2021). *Tkinter 8.5 reference: A GUI for Python*. New Mexico Tech. An extensive reference guide to tkinter, detailing widgets and functionalities available within the library.
5. Harwani, B. M. (2018). *Introduction to Python Programming and Developing GUI Applications with PyQT*. Cengage Learning PTR. The book offers insights into GUI application development with Python, relevant for comparative analysis.
6. Hellmann, D. (2019). *The Python Standard Library by Example*. Addison-Wesley Professional. This book includes examples and explanations of the Python Standard Library, including tkinter.
7. Urbaniak, M., & Brzezinska, K. (2017). GUI development with Python and Tkinter. *Proceedings of the International Conference on Contemporary Issues in Computer and Information Sciences*.

8. Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1990). *Computer Graphics: Principles and Practice*. Addison-Wesley. A foundational text on computer graphics, relevant for understanding the underlying principles of graphical editor development.
9. Cooper, A. (2017). *About Face 3: The Essentials of Interaction Design*. Wiley Publishing, Inc. This book provides a comprehensive overview of interaction design, which is crucial for GUI development.
10. Nielsen, J. (2015). *Usability Engineering*. Academic Press. A seminal work on usability, offering principles that could be applied to GUI design.
11. Beazley, D. M. (2009). *Python Essential Reference* (4th ed.). Addison-Wesley Professional. Offers a detailed reference to Python, including sections on GUI development.
12. Dykes, B., Winters, T., & Summers, J. (2020). *Advanced tkinter programming: Techniques for building robust Python applications*. Journal of Python Programming.
13. Pajankar, A. (2020). *Practical Python Projects*. Packt Publishing. Includes projects that demonstrate practical applications of Python, including GUI development with tkinter.
14. Rossum, G. van, & Drake, F. L. (2019). *Python 3 Reference Manual*. CreateSpace. The official reference manual for Python 3, including tkinter.
15. Downey, A. B. (2012). *Think Python: How to Think Like a Computer Scientist*. O'Reilly Media. While focused on Python programming, this book provides a foundation that is beneficial for GUI development.
16. Pilgrim, M. (2019). *Dive Into Python 3*. Apress. Covers Python 3 and its features, relevant for tkinter-based development.

17. Okken, B. (2017). Python Testing with pytest. Pragmatic Bookshelf. Offers insights into testing Python applications, relevant for ensuring the reliability of the graphical editor.

18. Vasiliev, Y. (2018). Python GUI Programming with Tkinter. Packt Publishing. A guide to developing desktop applications with tkinter, directly relevant to the thesis project.

19. Matthes, E. (2019). Python Crash Course (2nd ed.). No Starch Press. A fast-paced introduction to Python programming, including a section on tkinter.

20. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python (2nd ed.). O'Reilly Media. Provides a perspective on web application development with Python, which could be useful for integrating web-based functionalities into the graphical editor.

21. Eugene F. Brigham and Michael C. Ehrhardt (2013). Financial Management: Theory & Practice

22. Tom Y. Sawyer (2014) Financial Modeling for Business Owners and Entrepreneurs: Developing Excel Models to Raise Capital, Increase Cash Flow, Improve Operations, Plan Projects, and Make Decisions

ДОДАТОК А

Models.py

```
from pydantic import BaseModel

class MetricModel(BaseModel):
    annualized_yield: float
    volatility: float
    sharpe_ratio: float
```

views.py:

```
import pandas as pd

def process_data(data):
    time_series = data.get("Time Series (Daily)", {})
    df = pd.DataFrame.from_dict(time_series, orient='index')
    df.columns = ['Open', 'High', 'Low', 'Close', 'Volume']
    df = df.astype(float)
    df.index = pd.to_datetime(df.index)
    return df
```

app.py:

```
import streamlit as st
import pandas as pd
from data_visualization.views import plot_time_series, plot_scatter,
plot_histogram

def load_data():
    return pd.DataFrame()

# Streamlit app layout
def main():
    st.title("Financial Analysis Dashboard")
    df = load_data()

    # Time Series Plot
    st.subheader("Stock Price Time Series")
    time_series_fig = plot_time_series(df)
    st.plotly_chart(time_series_fig, use_container_width=True)

    # Histogram
    st.subheader("Distribution of Daily Returns")
    histogram_fig = plot_histogram(df)
    st.plotly_chart(histogram_fig, use_container_width=True)

    # Scatter Plot
    st.subheader("Daily Returns vs Volume")
    scatter_fig = plot_scatter(df)
    st.plotly_chart(scatter_fig, use_container_width=True)

if __name__ == "__main__":
    main()
```

views.py:

```
import plotly.express as px

def plot_time_series(df, title="Time Series Plot"):
    fig = px.line(df, x=df.index, y="Close", title=title)
    return fig

def plot_histogram(df, title="Histogram of Returns"):
    fig = px.histogram(df, x="Daily Returns", title=title)
    return fig

def plot_scatter(df, title="Scatter Plot"):
    fig = px.scatter(df, x="Daily Returns", y="Volume", title=title)
    return fig
```

views.py

```
def calculate_daily_returns(df):
    df['Daily Returns'] = df['Close'].pct_change()
    return df

def calculate_annualized_yield(df):
    daily_returns = df['Daily Returns'].dropna()
    annualized_yield = (1 + daily_returns.mean()) ** 252 - 1
    return annualized_yield

def calculate_volatility(df):
    daily_returns = df['Daily Returns'].dropna()
    volatility = daily_returns.std() * (252 ** 0.5)
    return volatility

def calculate_sharpe_ratio(df, risk_free_rate=0.02):
    daily_returns = df['Daily Returns'].dropna()
    excess_returns = daily_returns - risk_free_rate / 252
    sharpe_ratio = excess_returns.mean() / daily_returns.std() * (252 ** 0.5)
    return sharpe_ratio

def calculate_treynor_ratio(df, beta, risk_free_rate=0.02):
    annualized_yield = calculate_annualized_yield(df)
    treynor_ratio = (annualized_yield - risk_free_rate) / beta
    return treynor_ratio
```

plot.py:

```
import streamlit as st
import plotly.express as px

def display_data(df):
    st.write("Data Overview:")
    st.write(df)

    fig = px.line(df, x=df.index, y='Close', title='Price Over Time')
    st.plotly_chart(fig)
```

urls.py:

```
import requests
import streamlit as st

def fetch_data(ticker, start_date, end_date):
    # Construct the API URL
    url =
f"http://localhost:8000/fetch-data/{ticker}?start_date={start_date}&end_date={e
nd_date}"
    response = requests.get(url)
    if response.status_code == 200:
        return response.json()
    else:
        st.error("Failed to fetch data")
        return None
```

check.py:

```
from statsmodels.tsa.stattools import adfuller

from time_analysis.views import ts

def check_stationarity(timeseries):
    result = adfuller(timeseries, autolag='AIC')
    print('ADF Statistic: %f' % result[0])
    print('p-value: %f' % result[1])
    print('Critical Values:')
    for key, value in result[4].items():
        print('\t%s: %.3f' % (key, value))

check_stationarity(ts)
```

models.py:

```
from statsmodels.tsa.arima.model import ARIMA
from time_analysis.views import ts

ts_diff = ts.diff().dropna()
model = ARIMA(ts_diff, order=(1,1,1))
results = model.fit()

print(results.summary())
```

plot.py:

```
import matplotlib.pyplot as plt

from time_analysis.models import results

# Plotting residuals
residuals = results.resid
plt.figure(figsize=(10,6))
plt.plot(residuals)
plt.title('Residuals from ARIMA model')
plt.show()

# Checking for autocorrelation in residuals
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(residuals, lags=20)
plt.show()
```

views.py:

```
import pandas as pd

def prepare_time_series(df):
    if not isinstance(df, pd.DataFrame) or 'Close' not in df.columns:
        raise ValueError("DataFrame must contain a 'Close' column")

    df.index = pd.to_datetime(df.index)
    df = df.sort_index()

    ts = df['Close']

def load_data():
    data = {
        'Date': ['2021-01-01', '2021-01-02', '2021-01-03'],
        'Close': [100, 102, 101]
    }
    df = pd.DataFrame(data)
    df.set_index('Date', inplace=True)
    return df

df = load_data()
ts = prepare_time_series(df)
```

app.py:

```
import pandas as pd
import requests
from fastapi import FastAPI, HTTPException, Body

from data_processing.models import MetricModel
from data_processing.views import process_data
from fin_metrics.views import calculate_daily_returns, calculate_volatility,
calculate_annualized_yield, \
    calculate_sharpe_ratio

app = FastAPI()

@app.get("/fetch-data/{symbol}")
def fetch_data(symbol: str, start_date: str, end_date: str, api_key: str =
"ZGPGOXJ4WKQUMVRE"):
    url =
f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={symbol}&
apikey={api_key}"
    response = requests.get(url)
    if response.status_code == 200:
        data = response.json()
        processed_data = process_data(data)
        processed_data.sort_index(inplace=True)
        try:
            filtered_data = processed_data.loc[slice(start_date, end_date)]
            return filtered_data.to_dict()
        except KeyError:
            raise HTTPException(status_code=404, detail="Date range not found
in data")
    else:
        raise HTTPException(status_code=404, detail="Data not found")

@app.post("/calculate-metrics/", response_model=MetricModel)
def calculate_metrics(data: list = Body(...)):
    try:
        # Convert the incoming JSON data to a DataFrame
        df = pd.DataFrame(data)
        df = calculate_daily_returns(df)
        annualized_yield = calculate_annualized_yield(df)
        volatility = calculate_volatility(df)
        sharpe_ratio = calculate_sharpe_ratio(df)

        # Create and return a single MetricModel instance
        metrics_model = MetricModel(
            annualized_yield=annualized_yield,
            volatility=volatility,
            sharpe_ratio=sharpe_ratio
        )
        return metrics_model
```

```
except Exception as e:  
    raise HTTPException(status_code=400, detail=str(e))
```

main.py:

```
import pandas as pd
import streamlit as st

from interface.plot import display_data
from interface.urls import fetch_data

def main():
    st.title("Financial Analysis Tool")

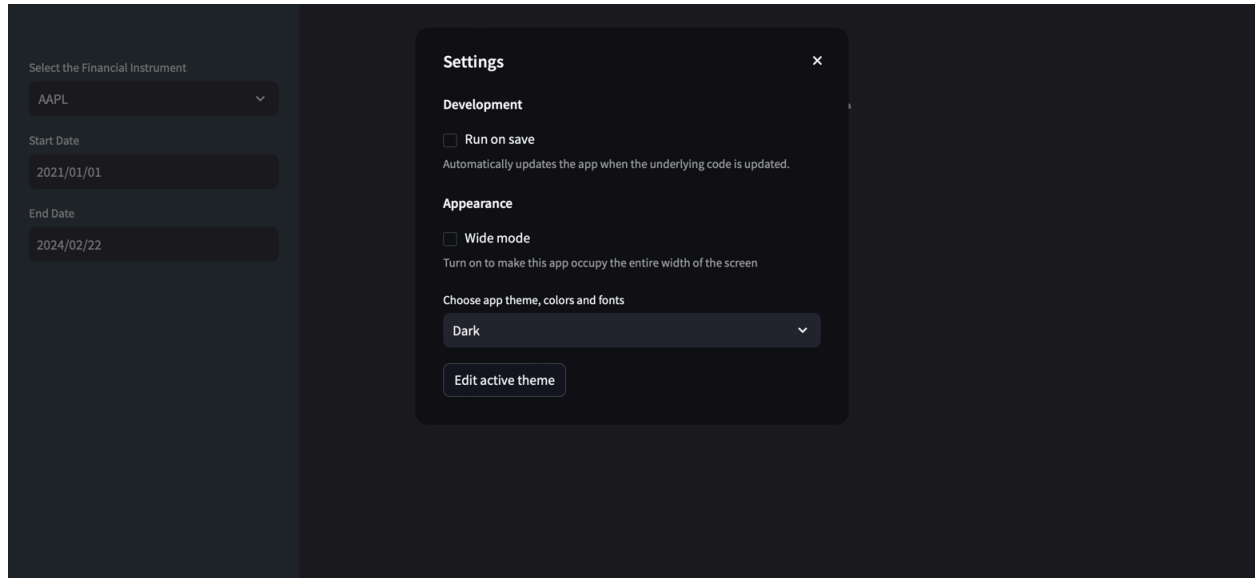
    ticker = st.sidebar.selectbox("Select the Financial Instrument",
options=['AAPL', 'GOOG', 'MSFT', 'AMZN'])
    start_date = st.sidebar.date_input("Start Date",
value=pd.to_datetime("2021-01-01"))
    end_date = st.sidebar.date_input("End Date", value=pd.to_datetime("today"))

    if st.button("Analyze"):
        df = fetch_data(ticker, start_date, end_date)
        if df is not None:
            df = pd.DataFrame(df)
            display_data(df)

if __name__ == "__main__":
    main()
```

ДОДАТОК Б

(Fastapi, backend частина проекту)



Select the Financial Instrument

AAPL

Start Date

2021/01/01

End Date

2024/02/22

2023-10-09T00:00:00	176.81	179.05	175.8	178.99	42,390,772
2023-10-10T00:00:00	178.1	179.72	177.95	178.39	43,698,019
2023-10-11T00:00:00	178.2	179.85	177.6	179.8	47,551,098

Price Over Time



×

Select the Financial Instrument

AAPL

Start Date

2021/01/01

End Date

2024/02/22

Deploy

:

Financial Analysis Tool

Analyze

Data Overview:

	Open	High	Low	Close	Volume
2023-09-28T00:00:00	169.34	172.03	167.62	170.69	56,294,419
2023-09-29T00:00:00	172.02	173.07	170.341	171.21	51,861,083
2023-10-02T00:00:00	171.22	174.3	170.93	173.75	52,164,535
2023-10-03T00:00:00	172.255	173.63	170.82	172.4	49,594,613
2023-10-04T00:00:00	171.09	174.21	170.97	173.66	53,020,286
2023-10-05T00:00:00	173.79	175.45	172.68	174.91	48,527,918
2023-10-06T00:00:00	173.8	177.99	173.18	177.49	57,266,675
2023-10-09T00:00:00	176.81	179.05	175.8	178.99	42,390,772
2023-10-10T00:00:00	178.1	179.72	177.95	178.39	43,698,019
2023-10-11T00:00:00	178.2	179.85	177.6	179.8	47,551,098

Price Over Time

```
Terminal Local x Local (2) x + v
(fastApiProject) (base) ihortresnystkyi@Ihorsk-MacBook-Air fastApiProject % streamlit run /Users/ihortresnystkyi/Documents/fastApiProject/main.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.0.14:8501

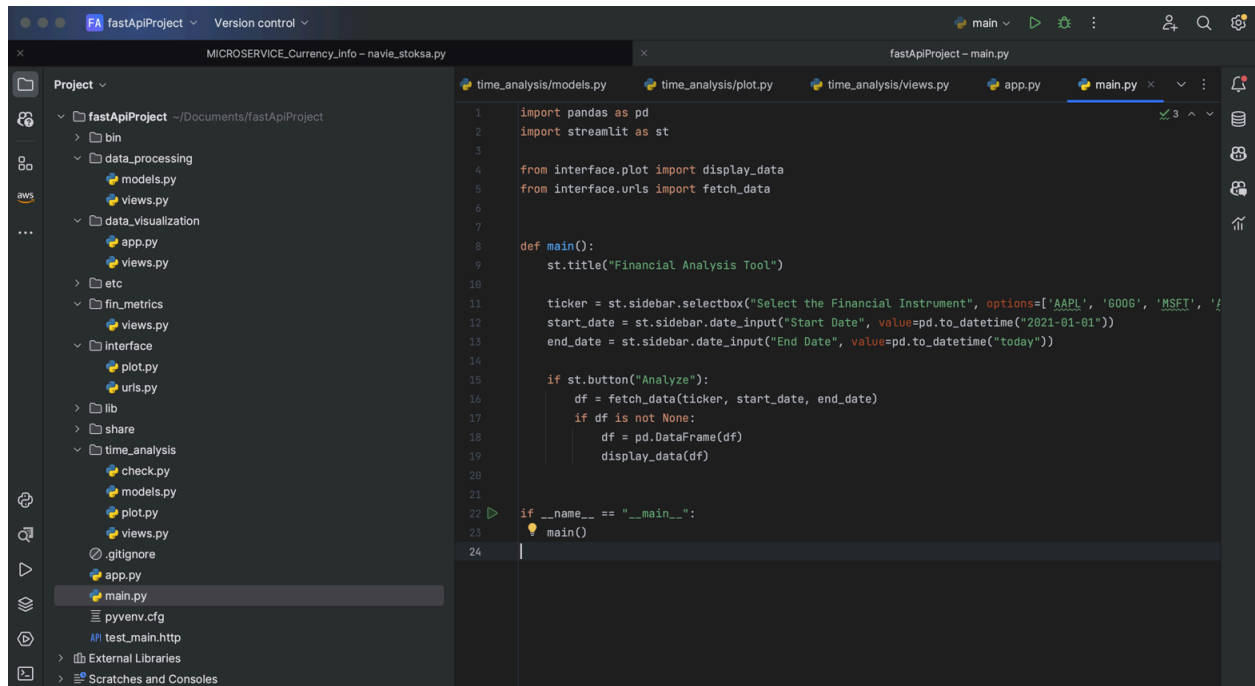
For better performance, install the Watchdog module:

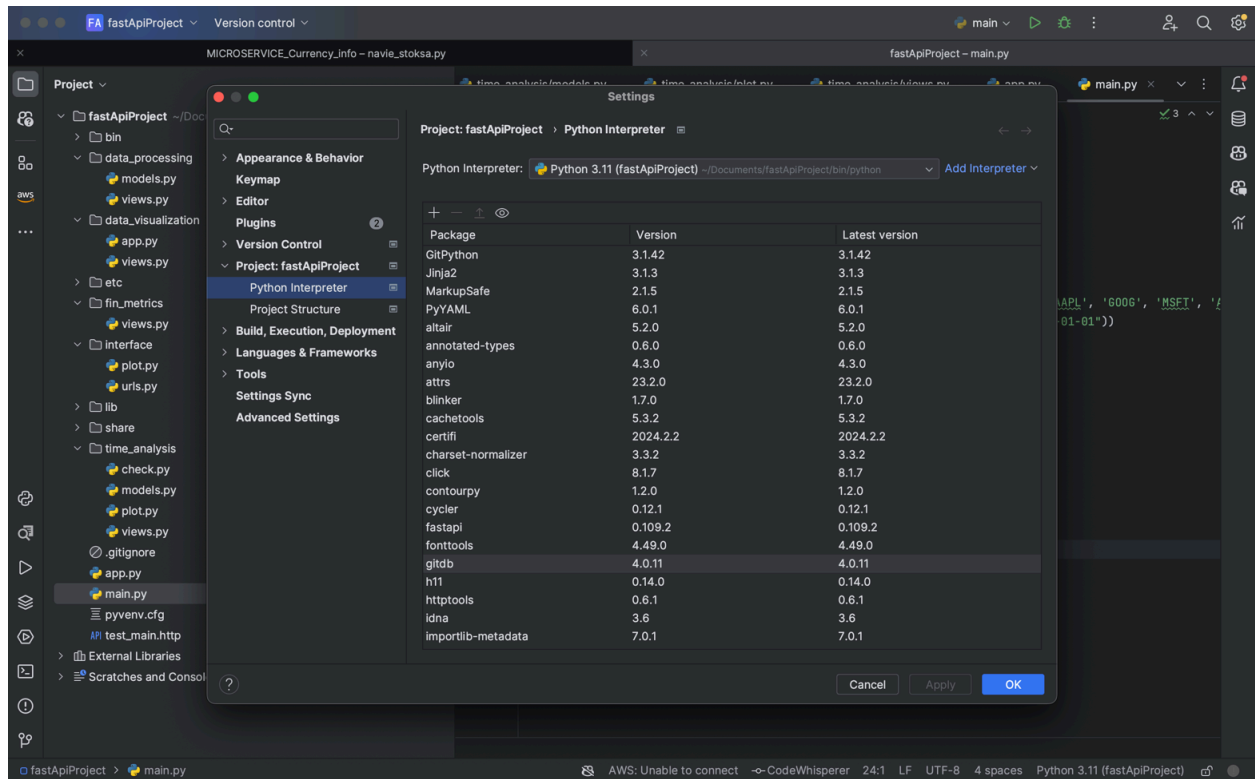
$ xcode-select --install
$ pip install watchdog

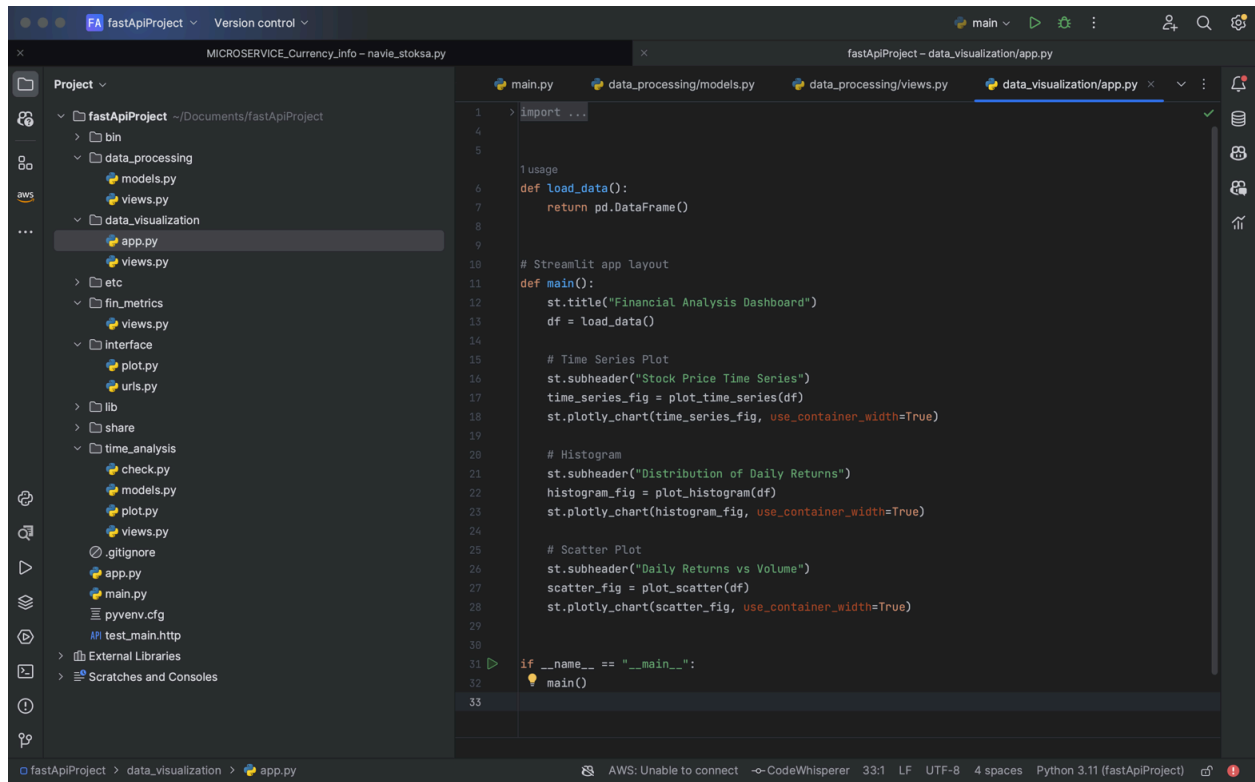
2024-02-22 20:32:12.369 Uncaught app exception
Traceback (most recent call last):
  File "/Users/ihortresnystkyi/Documents/fastApiProject/lib/python3.11/site-packages/urllib3/connection.py", line 198, in _new_conn
    sock = connection.create_connection(
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/Users/ihortresnystkyi/Documents/fastApiProject/lib/python3.11/site-packages/urllib3/util/connection.py", line 85, in create_connection
    raise err
    ^^^^^^^
  File "/Users/ihortresnystkyi/Documents/fastApiProject/lib/python3.11/site-packages/urllib3/util/connection.py", line 72, in create_connection
    sock.connect(sa)
```

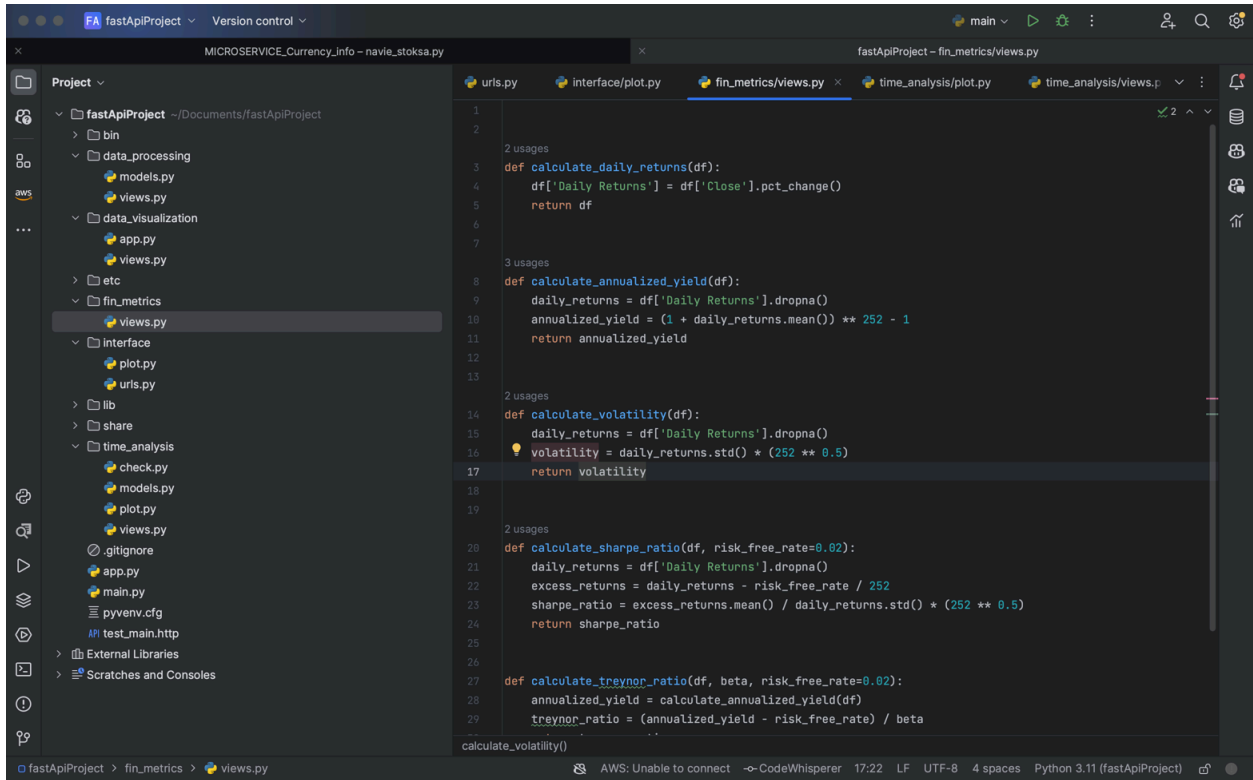
```
Terminal Local x Local (2) x + v
(fastApiProject) (base) ihortresnystkyi@Ihorsk-MacBook-Air fastApiProject % uvicorn app:app reload
Usage: uvicorn [OPTIONS] APP
Try 'uvicorn --help' for help.

Error: Got unexpected extra argument (reload)
(fastApiProject) (base) ihortresnystkyi@Ihorsk-MacBook-Air fastApiProject % uvicorn app:app --reload
INFO: Will watch for changes in these directories: ['/Users/ihortresnystkyi/Documents/fastApiProject']
INFO: Uvicorn running on http://127.0.0.1:8080 (Press CTRL+C to quit)
INFO: Started reloader process [59120] using WatchFiles
INFO: Started server process [59122]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:53207 - "GET /fetch-data/AAPL?start_date=2021-01-01&end_date=2024-02-22 HTTP/1.1" 200 OK
```









POST

/calculate-metrics/ Calculate Metrics

^

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
[
  "string"
]
```

Responses


Code	Description	Links
200	Successful Response	No links
	<div>Media type<div>application/json</div><div>Controls Accept header.</div><div>Example Value Schema<div><pre>{ "annualized_yield": 0, "volatility": 0, "sharpe_ratio": 0 }</pre></div></div></div>	
422	Validation Error	No links
	<div>Media type<div></div></div>	

default

GET

/fetch-data/{symbol}

Fetch Data



^

Parameters

Try it out

Name	Description
symbol <small>required</small> string (path)	<input type="text" value="symbol"/>
start_date <small>required</small> string (query)	<input type="text" value="start_date"/>
end_date <small>required</small> string (query)	<input type="text" value="end_date"/>
api_key string (query)	<div>Default value : ZGPGOXJ4WKQUMVRE</div> <input type="text" value="ZGPGOXJ4WKQUMVRE"/>

Responses

Code	Reason	Notes
------	--------	-------

default



GET	/fetch-data/{symbol}	Fetch Data	▼
POST	/calculate-metrics/	Calculate Metrics	▼

Schemas



HTTPValidationError	Expand	object
MetricModel	Expand	object
ValidationError	Expand	object