

INVESTIQ

BY TEAM 02-->
TECHNO STACK



AGENDA

Team Member Roles and Responsibilities

Improvements from Professor Feedback

Project Description

Team Working Agreement

Personas

MVP

Technologies

Algorithms

Diagrams

Product Backlog

Sprint 2 Backlog

Metrics

Retrospective

Sprint 3 Planning

Project Demo - Sprint 2

GitHub Link

Live Application Demo

TEAM MEMBERS



Ruchitha Reddy Kuthuru -
Product Manager and Full
Stack developer



Mahidhar Reddy Kandula -
Machine learning Engineer

TEAM MEMBERS



Afrida Mehanaz Shaik-
Machine Learning Engineer



Sharan Jagini - Backend
Engineer and Tester

TEAM MEMBERS



Nadeem Hussain Shaik -
Frontend Developer



Shriniidhi Daheechi - Frontend
Developer

PROBLEM STATEMENT

Investors often find it challenging to make informed decisions due to overwhelming market information, constant fluctuations, and a lack of personalized guidance. Many existing tools provide data but do not offer meaningful insights tailored to individual needs. This app aims to simplify investment decisions by analyzing market trends, tracking relevant news, and offering timely recommendations based on personal financial goals and life events. Combining real-time updates with user-specific insights helps investors stay informed and make smarter financial choices with confidence.

PROJECT DESCRIPTION

Project Name:	<u>InvestIQ</u>
Team:	Techno Stack
Project Description:	<p>For investors who struggle with analyzing market trends and making informed financial decisions, the <u>InvestIQ</u> AI application is a smart investment assistant that analyzes stock market trends, processes real-time financial news, and provides personalized investment insights and risk alerts. Unlike traditional investment platforms that only offer raw data and generic analytics, our application leverages AI-driven time series analysis and natural language processing to deliver real-time, context-aware financial recommendations tailored to users' personal financial goals and life events.</p>
Benefit Outcomes:	<ul style="list-style-type: none">• Better investment decisions with AI-driven insights and alerts• Personalized financial recommendations based on user behavior and market trends• Real-time market sentiment analysis for timely and informed trading• Reduced research time by consolidating key financial insights into a single platform
<u>Github</u> Link:	https://github.com/htmw/2025S-Techno-Stack/wiki

TEAM WORKING AGREEMENT

Team Working Agreement

Team Name: Techno Stack

1. Communication & Meetings

- **Slack:** Our main channel for daily communication, quick questions, and status updates.
- **Virtual/In-Person Meetings:** We will conduct weekly meetings via Zoom or Google Meet. If in-person classes are held, we will transition to face-to-face meetings as needed. Team members are encouraged to raise doubts and support one another during these meetings.

2. Project Management

- **Jira:** We will use Jira to track tasks, set deadlines, and monitor progress. It's essential that all links in our documentation and code repositories are verified and kept up-to-date.
- **Timely Submissions:** Every team member must complete their assigned tasks on time to ensure the project stays on track schedule.

3. Documentation

- All documentation must include working links to code repositories, APIs, and other relevant resources. This ensures every team member can access the latest project updates and resources.

4. Roles & Responsibilities

- **Ruchitha Reddy Kuthuru (PM & Full Stack Developer):** Oversees the project, coordinates meetings, and contributes to frontend and backend development.
- **Mahidhar Reddy Kandula (ML Engineer):** Develops and integrates AI models for sentiment analysis and trend forecasting.
- **Afrida Mehanaz Shaik (ML Engineer):** Collaborates on developing and optimizing AI models and data processing workflows.
- **Sharan Jagini (Backend Developer & Tester):** Manages backend development, handles API integrations, and conducts thorough testing to ensure quality and functionality.
- **Nadeem Hussain Shaik (Frontend Developer):** Designs and develops a responsive user interface focusing on usability.
- **Shrinidhi Daheechi (Frontend Developer):** Supports frontend development and collaborates on UI/UX design improvements.

5. Collaboration & Support

- Team members are expected to help each other by promptly addressing doubts and challenges as they arise.
- Regular updates on Jira and other project management tools are required to inform everyone of progress.

- Ensuring that all documentation, code, and integrations (including working links) are thoroughly tested and maintained is a shared responsibility.

Agreed by:

- Ruchitha Reddy Kuthuru
- Mahidhar Reddy Kandula
- Nadeem Hussain Shaik
- Sharan Jagini
- Afrida Mehanaz Shaik
- Shrinidhi Daheechi

PERSONA

AARAV SHARMA

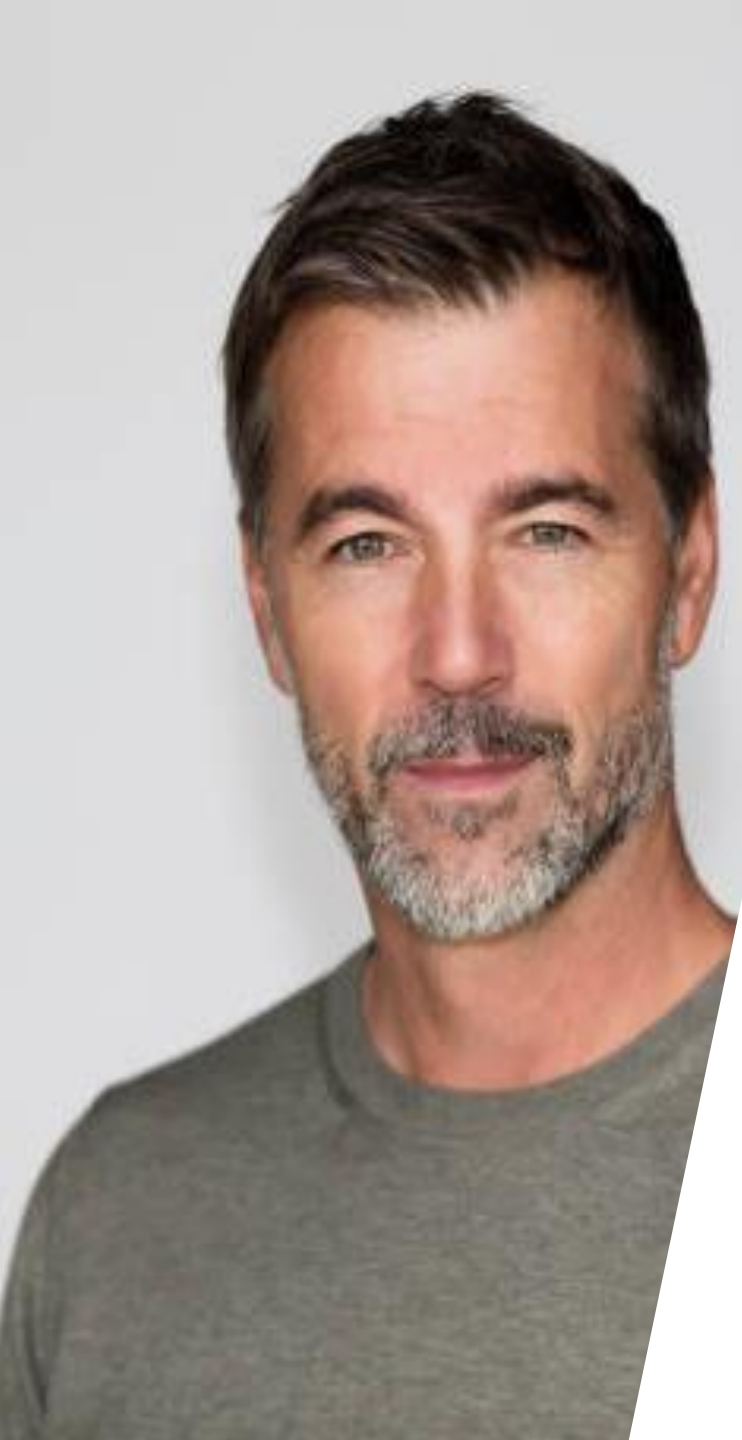
- Age: 28
- Occupation: Software Engineer
- Investment Experience: Intermediate
- Goals: Simplify his investment decision-making process by utilizing clear, actionable insights from real-time news.
- Pain Points: Feels overwhelmed by the vast amount of market data and frustrated by platforms that fail to provide timely news analysis.
- Description: Aarav values technology and data-driven insights; he seeks a tool that aggregates real-time news, enabling him to quickly grasp market shifts without the distraction of raw stock data.
- Challenges:
 - Struggles with filtering relevant financial information from the overwhelming amount of market data.
 - Finds it time-consuming to track news manually and correlate it with market trends.



PERSONA

THOMAS SHELBY

- Age: 40
- Occupation: Corporate Manager
- Investment Experience: Beginner to Intermediate
- Goals: Achieve steady portfolio growth by staying informed about key market developments through timely news updates.
- Pain Points: Finds traditional investment platforms cluttered with irrelevant data and lacking a focused approach to current financial news.
- Description: With a busy schedule, Thomas needs a user-friendly platform that provides curated real-time news and actionable insights, enabling him to make confident, long-term financial decisions without continuously monitoring fluctuating stock figures.
- Challenges:
 - Has limited time to stay updated on financial news and market changes.
 - Prefers risk-averse investments but struggles with identifying safe and promising opportunities.



PERSONA

SANJANA REDDY

- Age: 35
- Occupation: Financial Analyst
- Investment Experience: Advanced
- Goals: Optimize her investment strategy with comprehensive market analysis based on the latest financial news.
- Pain Points: Frustrated with generic investment tools that do not integrate timely news, leaving her without the nuanced insights she needs.
- Description: Sanjana is analytical and detail-oriented, searching for a sophisticated platform emphasizing real-time news analysis to provide deep insights and tailored recommendations rather than just raw stock data.
- Challenges:
 - Struggles with quickly assessing the sentiment of large volumes of financial news.
 - It requires advanced analytics but finds that many tools lack deep AI-driven insights.



MVP

- user registration and profile setup (to capture financial goals, risk tolerance, and preferences).
- Historical market trend analysis (using delayed or historical stock data from free APIs like Yahoo Finance or Alpha Vantage, which often provide historical data for free).
- Personalized investment recommendations (based on historical trends, user goals, and risk tolerance, using a simple rule-based algorithm instead of real-time data).
- A dashboard to view recommendations, historical trends, and market news.

TECHNOLOGIES - BACKEND

- Flask: A lightweight Python framework ideal for building RESTful APIs, easily integrating with machine learning models, and simplifying backend logic.
- Node.js: Offers non-blocking, event-driven architecture, making it great for handling real-time operations and concurrent tasks.



TECHNOLOGIES - FRONTEND

- Next.js: Offers server-side rendering and static site generation, resulting in fast, SEO-friendly pages and a strong user experience interface.
- TypeScript: Improves code quality and maintainability by providing static type checking, which reduces runtime errors.

NEXT.js



TECHNOLOGIES - DATABASE

- PostgreSQL is a powerful, open-source relational database known for its reliability, scalability, and support for complex queries, ensuring secure and efficient data storage.



TECHNOLOGIES - APIS

Financial News API & Polygon API

- These APIs supply real-time financial news and market data, enabling the application to deliver timely and actionable insights without relying on raw stock data.



polygon.io

TECHNOLOGIES – MACHINE LEARNING

PyTorch is selected for its dynamic computation graph and user-friendly interface, making it ideal for developing and training deep learning models used in sentiment analysis and trending forecasting.



TECHNOLOGIES – HOSTING & DEPLOYMENT

- Vercel: Perfect for quickly deploying frontend applications with high performance and scalability.
- DigitalOcean: Offers dependable and scalable hosting for backend services and the entire application deployment.



TECHNOLOGIES – DEVELOPMENT TOOLS

- Visual Studio Code: A versatile code editor featuring extensive extensions and integrated debugging that enhances the developer experience productivity.
- Postman: Essential for testing APIs, confirming our endpoints function as expected during development.
- Git: To enable version control, enhance collaborative development, and improve code management.
- Docker: Used for containerization, ensuring consistency across development, testing, and production environments.
- Slack: Facilitates effective team communication and project management.



The background of the slide features a collage of financial data visualizations. On the left, there's a candlestick chart with a blue shaded area. To its right is a line chart with multiple data series. Below these, a bar chart with blue and orange bars is visible. The entire background is overlaid with a grid and some faint, illegible text, suggesting a complex financial analysis environment.

ALGORITHMS - NLP

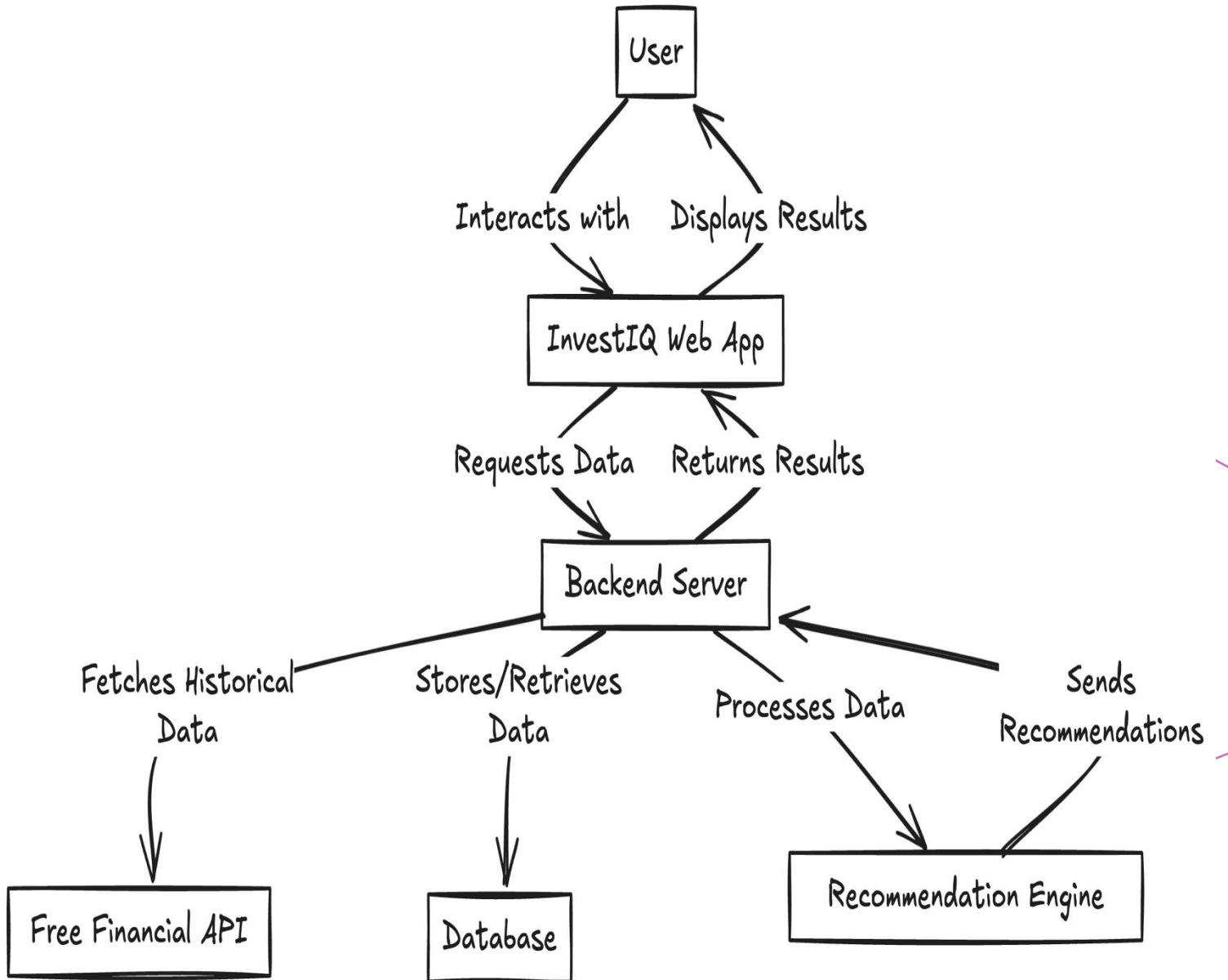
Our NLP module harnesses cutting-edge technology. The phi model for natural language processing is specifically designed to analyze real-time financial news. This module extracts sentiment and identifies key topics from the latest updates, enabling the application to interpret market sentiment accurately. By processing large volumes of textual data, the phi model provides actionable insights that inform users about market trends and emerging financial narratives, transforming raw news into clear, meaningful investment opportunities guidance.

The background of the slide features a collage of financial data visualizations. On the left, there's a candlestick chart with a blue shaded area. To its right is a line chart with multiple data series. Below these, a bar chart with blue and orange bars is visible. The overall theme is financial analysis and data visualization.

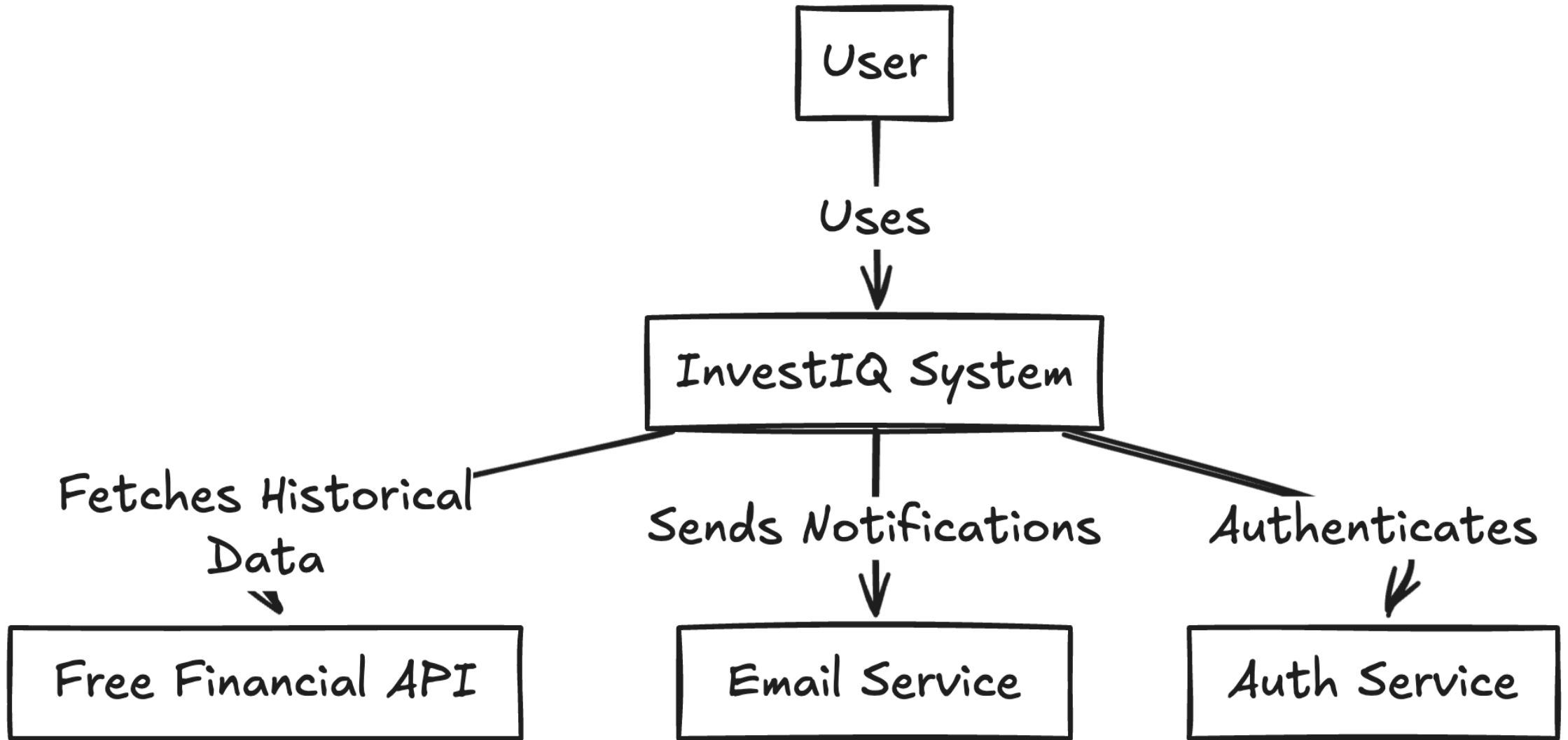
ALGORITHMS – TIME SERIES

- Our Time Series Analysis module examines historical stock market data using advanced deep learning techniques like LSTM and Temporal Fusion Transformers to capture underlying trends and seasonal patterns. This analysis provides essential context that complements the real-time insights derived from our NLP module, ensuring users have a strong understanding of market dynamics. Together, these two modules generate a strong synergy that enables investors to make informed decisions based on current news sentiment and historical market performance.

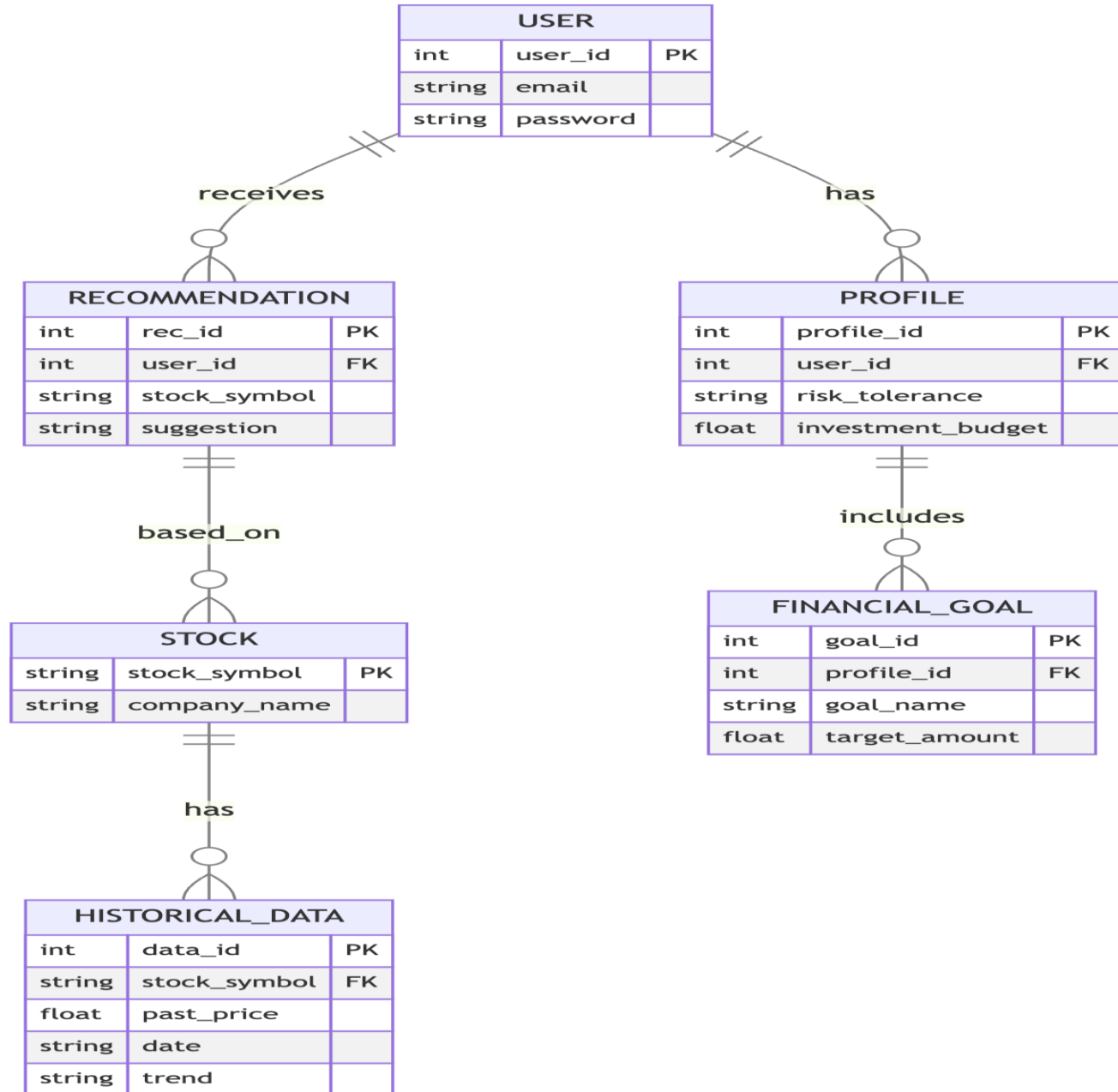
ARCHITECTURE DIAGRAM



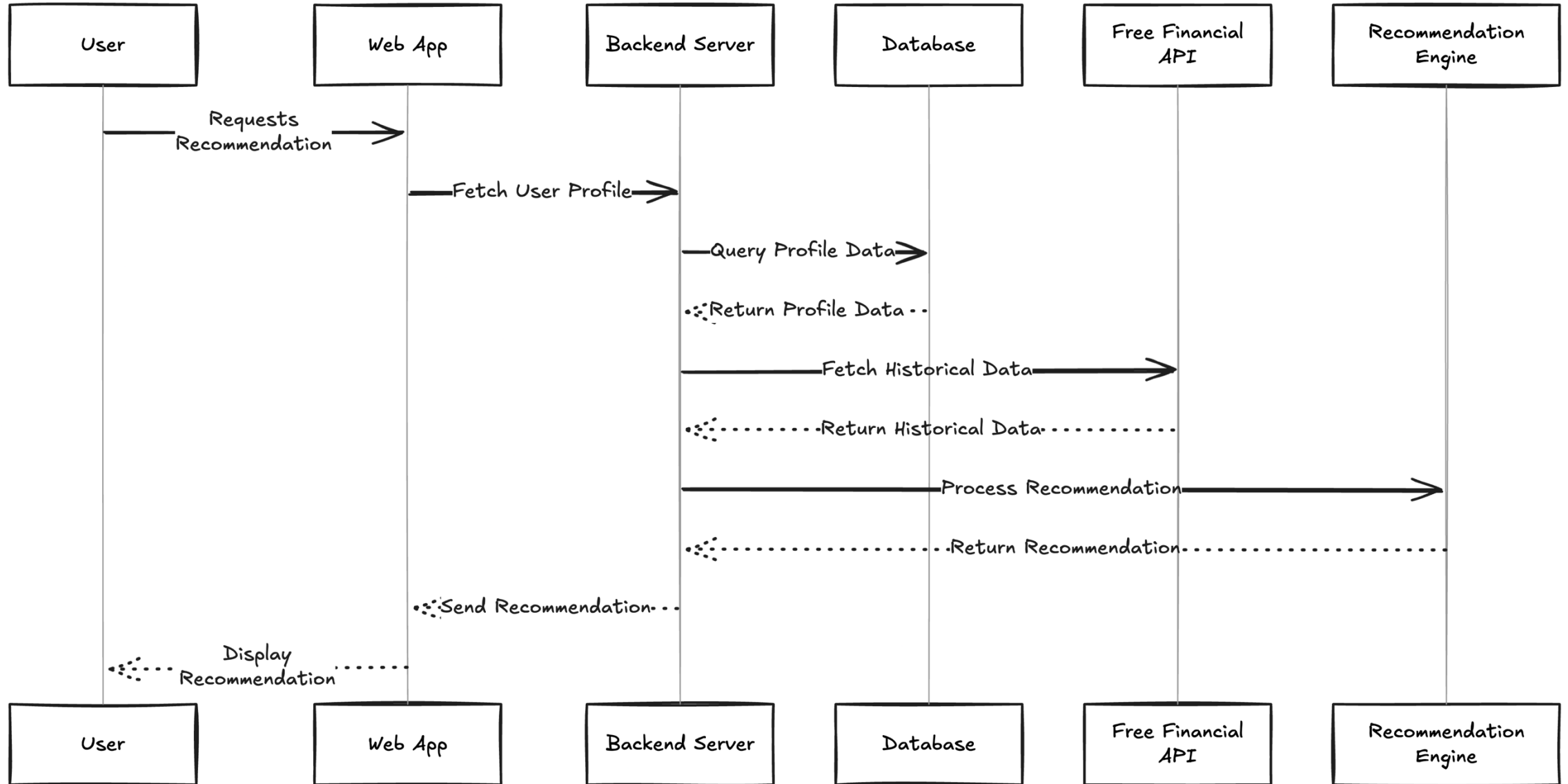
CONTEXT DIAGRAM



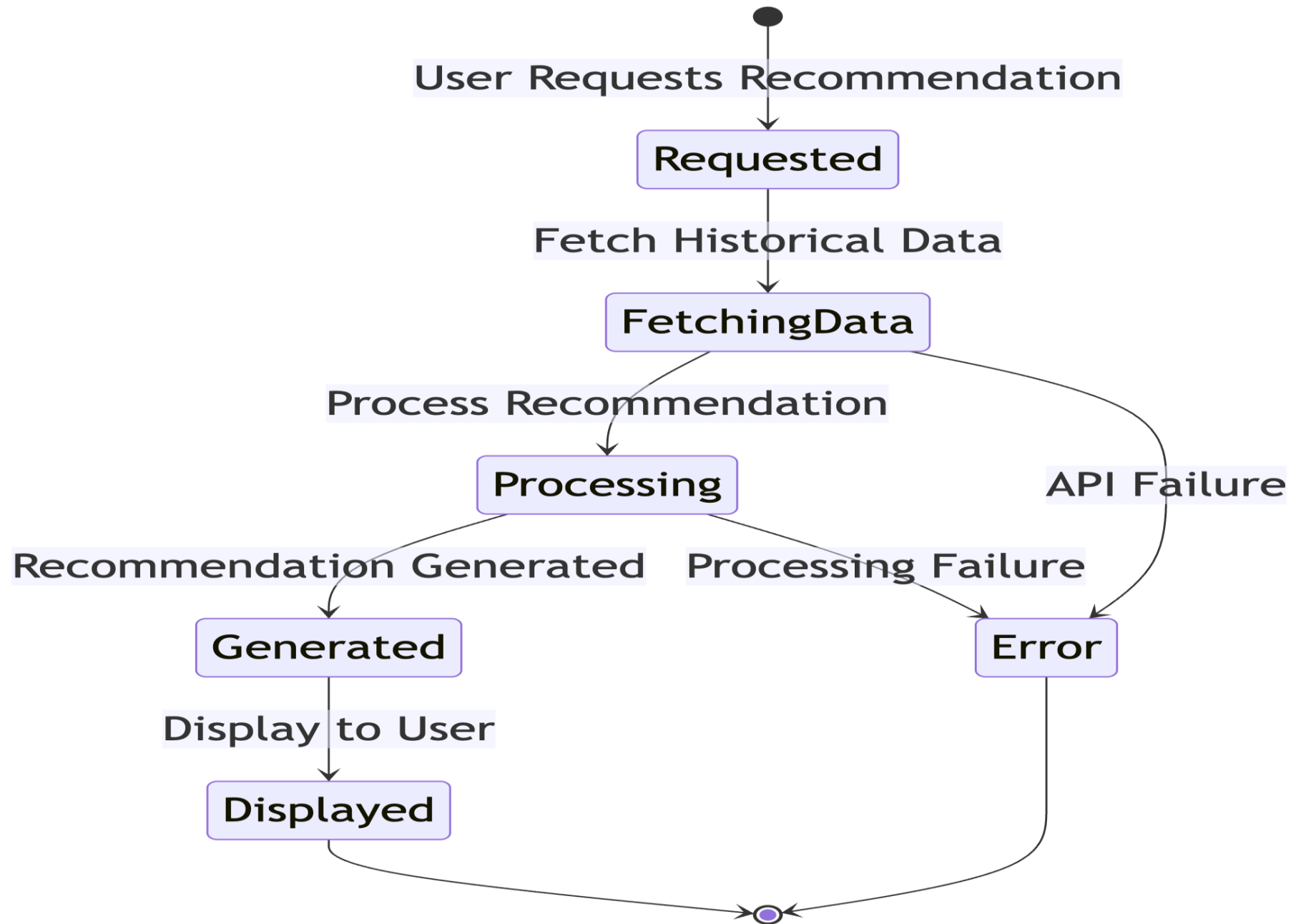
ER DIAGRAM (ENTITY- RELATIONSHIP DIAGRAM)



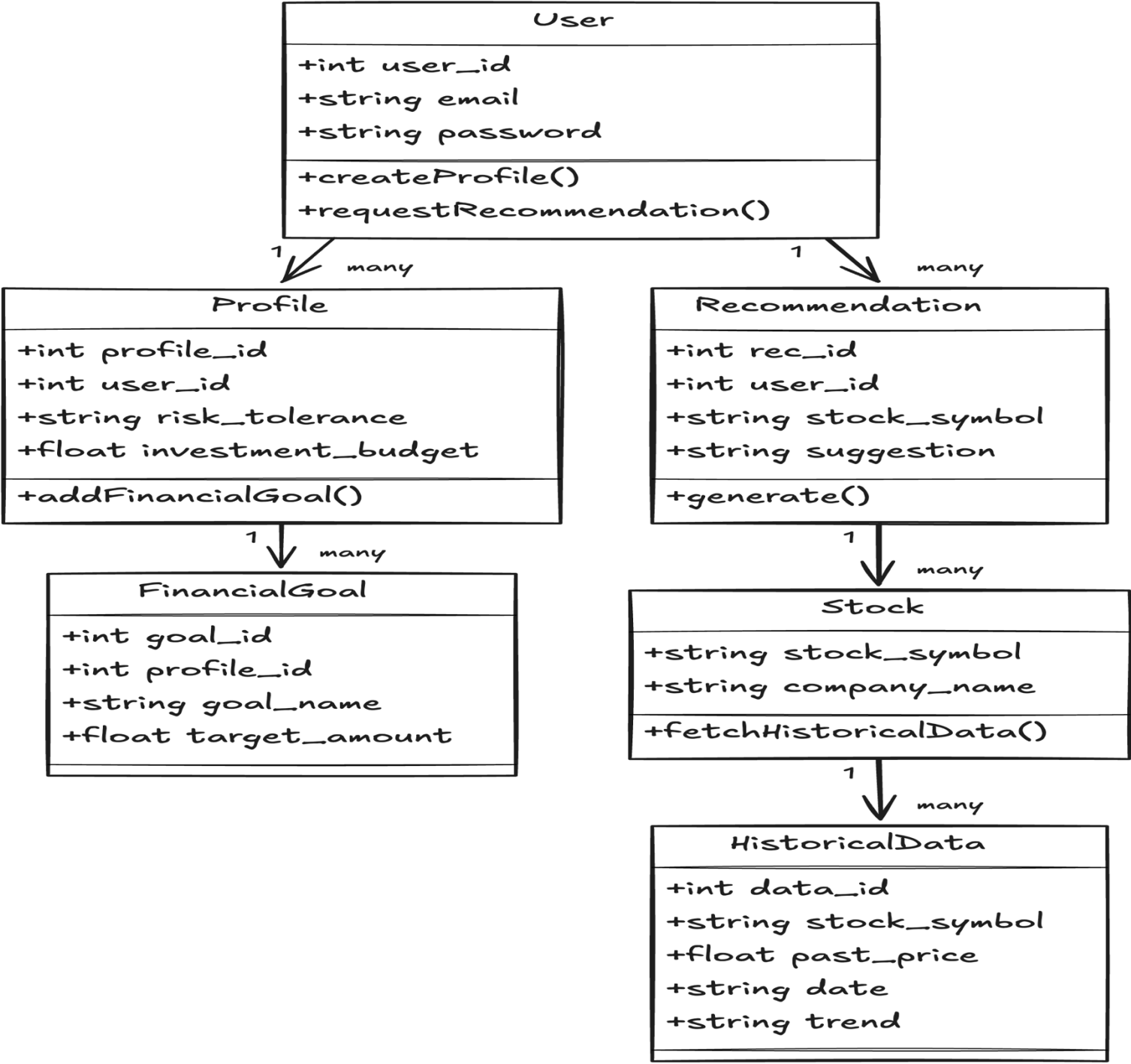
SEQUENCE DIAGRAM



STATE DIAGRAM



CLASS DIAGRAM



SPRINT 1 RECAP

- The frontend part is done during sprint 1 recap.

PRODUCT BACKLOG

Story ID	Feature	User Story	Acceptance Criteria	Story Points
US1	Basic Dashboard UI	Aarav Sharma wants to see a basic dashboard layout, so that he can understand the structure of the app.	<ol style="list-style-type: none">1. Dashboard has sections for recommendations, trends, and news (placeholders).2. Layout is clean with a header, sidebar, and main content area.3. UI is responsive for desktop and mobile.	5
US2	Profile Setup UI	Aarav Sharma wants to input his risk tolerance and budget in a form, so that he can simulate setting up his profile.	<ol style="list-style-type: none">1. Form includes dropdown for risk tolerance (Low, Medium, High) and input for budget.2. Form has a "Save" button that shows a confirmation message (no backend saving yet).3. Error message shown if budget is invalid (e.g., negative).	3
US3	Recommendation Placeholder UI	Thomas Shelby wants to see a placeholder for recommendations, so that he can visualize where his investment suggestions will appear.	<ol style="list-style-type: none">1. Dashboard has a section titled "Recommendations".2. Section shows 3 mock recommendations (e.g., "Stock: AAPL, Suggestion: Buy").3. Each recommendation has a button labeled "View Details" (non-functional for now).	3
TS1	Front-End Framework Setup	As a developer needs to set up a front-end framework (e.g., React), so that the team can build the UI efficiently.	<ol style="list-style-type: none">1. Project is initialized with React (or similar framework).2. Basic routing is set up for dashboard and profile pages.3. CSS framework (e.g., Tailwind or Bootstrap) is integrated for styling.	5

PRODUCT BACKLOG

Story ID	Feature	User Story	Acceptance Criteria	Story Points
US4	User Registration	Aarav Sharma wants to register with his email and password, So that he can access the InvestIQ platform.	<ol style="list-style-type: none">1. Aarav can enter email and password on a registration form.2. System validates email format and ensures password is at least 8 characters.3. Upon successful registration, Aarav is redirected to login page.4. Error messages shown if email is in use or inputs are invalid.	3
US5	User Login	Aarav Sharma wants to log in with his email and password, So that he can access his dashboard.	<ol style="list-style-type: none">1. Aarav can enter email and password on a login form.2. System validates credentials and logs Aarav in if correct.3. Aarav is redirected to the dashboard upon successful login.4. Error message shown if credentials are incorrect.	2
US6	Save Profile Data	Aarav Sharma wants to save his profile data, So that the system can use it for recommendations.	<ol style="list-style-type: none">1. Profile form (built in Sprint 1) now saves data to the backend.2. Aarav can edit and update his profile.3. Confirmation message shown after saving.	3
TS2	Backend API Integration	As a developer needs to integrate a free API (e.g., Yahoo Finance) to fetch historical stock data, So that the team can use it for recommendations.	<ol style="list-style-type: none">1. Backend fetches historical stock data for 5 sample stocks (e.g., AAPL, GOOGL).2. Data includes 30 days of historical prices and trends.3. Data is cached in the database.4. Error handling for API failures	5

PRODUCT BACKLOG

Story ID	Feature	User Story	Acceptance Criteria	Story Points
TS3	Database Schema Setup	Sanjana Reddy needs to create a database schema for users, profiles, and historical data, so that the team can store and retrieve data.	<ol style="list-style-type: none">1. Schema includes tables for users, profiles, and HistoricalData.2. Database uses a free solution (e.g., SQLite).3. Basic CRUD operations implemented for user and profile data.	3
US7	Historical Trends Display	Thomas Shelby wants to see historical trends for recommended stocks, so that he can understand the basis for recommendations.	<ol style="list-style-type: none">1. Dashboard displays a line chart showing 30-day price trend for each stock.2. Chart uses historical data from the database.3. Chart includes labels for dates and prices.	5
US8	Market News Display	Thomas Shelby wants to see recent market news, so that he can stay informed.	<ol style="list-style-type: none">1. Dashboard includes a section with market news headlines.2. News fetched from a free API (e.g., NewsAPI) and displayed as a list.3. Each news item includes title, source, and link to the article.	5
US9	Personalized Recommendations	Thomas Shelby wants to see personalized investment recommendations, so that he can make informed decisions.	<ol style="list-style-type: none">1. Recommendations replace the placeholder UI from Sprint 1.2. Recommendations are based on historical trends and user profile (e.g., risk tolerance).3. If no recommendations are available, a message like "No recommendations available at this time."	5

PRODUCT BACKLOG

Story ID	Feature	User Story	Acceptance Criteria	Story Points
TS4	Recommendation Logic	As a developer needs to implement a rule-based recommendation system, So that users receive personalized suggestions.	<ol style="list-style-type: none">1. Algorithm uses historical trends (e.g., 30-day moving average) and user risk tolerance.2. Example rule: Recommend stocks with upward trends for low-risk users.3. Logic is tested with sample profiles and data.	5
TS5	News API Integration	As a developer needs to integrate a free news API (e.g., NewsAPI), So that the team can display market news.	<ol style="list-style-type: none">1. Backend fetches news articles related to finance.2. News data is cached for 24 hours.3. Error handling for API failures.	3

SPRINT 2 BACKLOG

Story ID	Feature	User Story	Acceptance Criteria	Story Points
US4	User Registration	Aarav Sharma wants to register with his email and password, So that he can access the InvestIQ platform.	1. Aarav can enter email and password on a registration form. 2. System validates email format and ensures password is at least 8 characters. 3. Upon successful registration, Aarav is redirected to login page. 4. Error messages shown if email is in use or inputs are invalid.	3
US5	User Login	Aarav Sharma wants to log in with his email and password, So that he can access his dashboard.	1. Aarav can enter email and password on a login form. 2. System validates credentials and logs Aarav in if correct. 3. Aarav is redirected to the dashboard upon successful login. 4. Error message shown if credentials are incorrect.	2
US6	Save Profile Data	Aarav Sharma wants to save his profile data, So that the system can use it for recommendations.	1. Profile form (built in Sprint 1) now saves data to the backend. 2. Aarav can edit and update his profile. 3. Confirmation message shown after saving.	3
TS2	Backend API Integration	As a developer needs to integrate a free API (e.g., Yahoo Finance) to fetch historical stock data, So that the team can use it for recommendations.	1. Backend fetches historical stock data for 5 sample stocks (e.g., AAPL, GOOGL). 2. Data includes 30 days of historical prices and trends. 3. Data is cached in the database. 4. Error handling for API failures.	5

SPRINT 2 BACKLOG

<u>Story ID</u>	<u>Feature</u>	<u>User Story</u>	<u>Acceptance Criteria</u>	<u>Story Points</u>
TS3	Database Schema Setup	Sanjana Reddy needs to create a database schema for users, profiles, and historical data, so that the team can store and retrieve data.	<ol style="list-style-type: none">1. Schema includes tables for Users, Profiles, and HistoricalData.2. Database uses a free solution (e.g., SQLite).3. Basic CRUD operations implemented for user and profile data.	3

TEST CASES SPRINT 2

Test Case ID	Story ID	Feature	Test Description	Test Steps	Expected Result	Actual Result	Status
TC4.1	US4	User Registration	Register with valid credentials	1. Open registration form 2. Enter test@example.com 3. Enter password: Test@1234 4. Click Register	Redirected to login page after successful registration	✗ Registration failed - not redirected	Failed
TC4.2	US4	User Registration	Invalid email format	1. Open registration form 2. Enter email: invalid-email 3. Enter valid password 4. Click Register	Error message about invalid email format	✗ validation skipped - allowed invalid email input	Failed
TC4.3	US4	User Registration	Password too short	1. Open registration form 2. Enter valid email 3. Enter password: 123 4. Click Register	Error message about password length	✗ Error not triggered for short password	Failed

TEST CASES SPRINT 2

Test Case ID	Story ID	Feature	Test Description	Test Steps	Expected Result	Actual Result	Status
TC4.4	US4	User Registration	Email already in use	1. Open registration form 2. Enter already registered email 3. Enter valid password 4. Click Register	Error shown: email already in use	✗ Duplicate email allowed – no validation	Failed
TC5.1	US5	User Login	Login with valid credentials	1. Go to login page 2. Enter registered email and password 3. Click Login	Redirected to dashboard	✗ Login form unresponsive – dashboard not reached	Failed
TC5.2	US5	User Login	Invalid login	1. Go to login page 2. Enter wrong email or password 3. Click Login	Error message for invalid credentials	✗ No error shown – system allowed invalid login	Failed

TEST CASES SPRINT 2

<u>Test Case ID</u>	<u>Story ID</u>	<u>Feature</u>	<u>Test Description</u>	<u>Test Steps</u>	<u>Expected Result</u>	<u>Actual Result</u>	<u>Status</u>
TC6.1	US6	Save Profile Data	Save new profile	1. Open profile form 2. Fill out all fields (age, risk level, goals) 3. Click Save	Data saved and confirmation shown	Profile saved successfully	Passed
TC6.2	US6	Save Profile Data	update existing profile	1. Open profile 2. Modify any field 3. Click Save	Updated profile and confirmation message shown	Profile updated successfully	Passed
TC2.1	TS2	API Integration	Fetch 30-day data for 5 stocks	1. Trigger backend to fetch historical data 2. Select stocks like AAPL, GOOGL, etc. 3. Inspect returned data	30-day price/trend data fetched correctly	Data fetched as expected	Passed

TEST CASES SPRINT 2

<u>Test Case ID</u>	<u>Story ID</u>	<u>Feature</u>	<u>Test Description</u>	<u>Test Steps</u>	<u>Expected Result</u>	<u>Actual Result</u>	<u>Status</u>
TC2.2	TS2	API Integration	API failure handling	1. Simulate API failure (e.g., disconnect) 2. Trigger fetch again	Graceful error handling, app stays responsive	Error handled correctly, no crash	Passed
TC2.3	TS2	API Integration	Data caching	1. Fetch data for the first time 2. Fetch same data again 3. Monitor performance/logs	Second fetch should use cache, faster response time	Cached data used, response improved	Passed
TC3.1	TS3	DB Schema Setup	Check required tables	1. Inspect DB schema 2. Ensure Users, Profiles, and HistoricalData tables exist	All required tables present in schema	Tables found as expected	Passed

TEST CASES SPRINT 2

<u>Test Case ID</u>	<u>Story ID</u>	<u>Feature</u>	<u>Test Description</u>	<u>Test Steps</u>	<u>Expected Result</u>	<u>Actual Result</u>	<u>Status</u>
TC3.2	TS3	DB Schema Setup	CRUD on Users table	1. Create user 2. Read user 3. Update user 4. Delete user	All CRUD operations successful	All operations completed successfully	Passed
TC3.3	TS3	DB Schema Setup	CRUD on Profiles table + Auth check	1. Create profile linked to user 2. Read profile 3. Update profile 4. Delete profile 5. Validate user-profile auth relationship	All operations work, profile links to correct user	CRUD passed but ✗ Auth check failed - user-profile linkage not verified	Failed

STORIES COMPLETED IN SPRINT 2

US6 - Save
Profile Data

TS2 - Backend
API Integration

TS3 - DB
Schema Setup

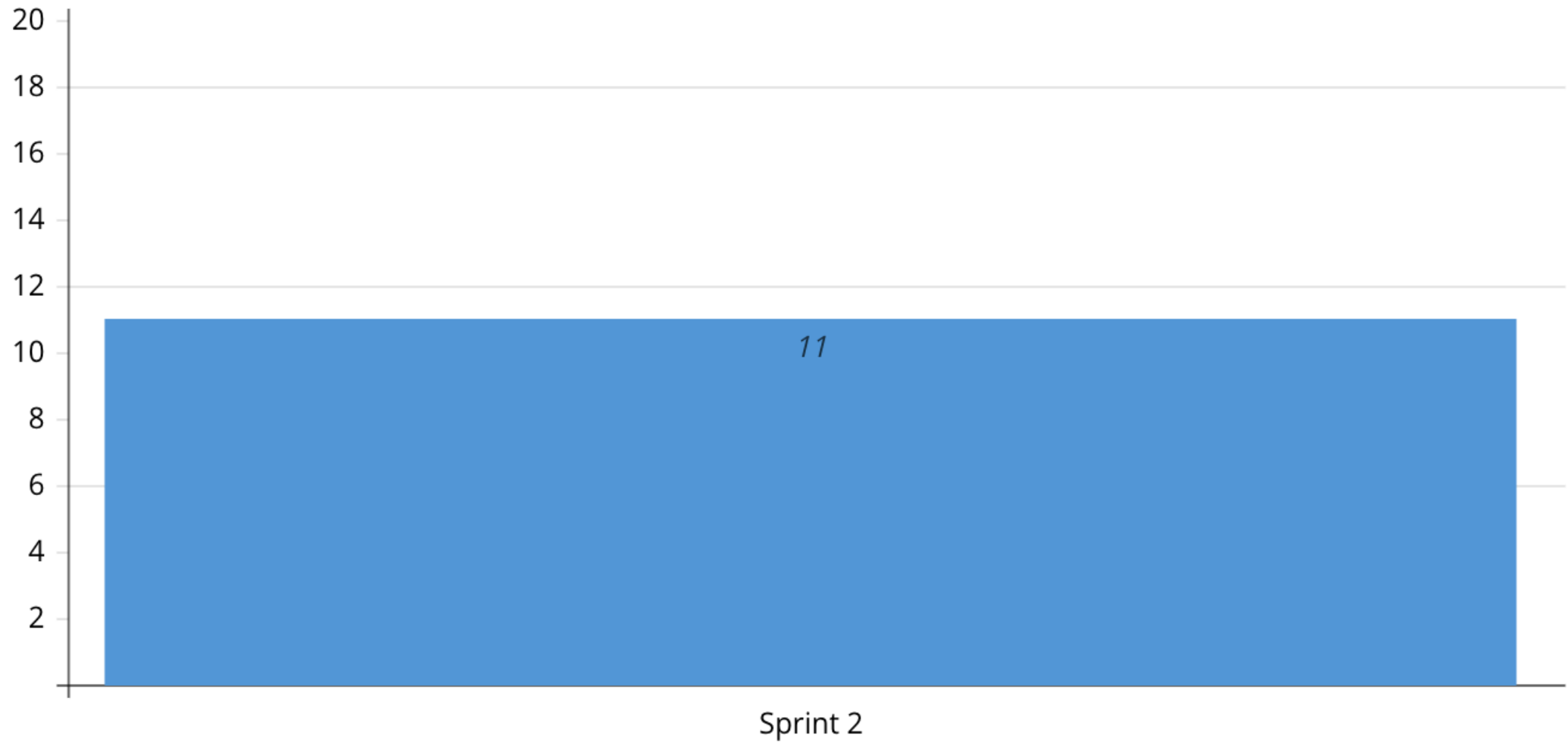
STORIES INCOMPLETED IN SPRINT 2

US4 - User Registration

US5 - User Login

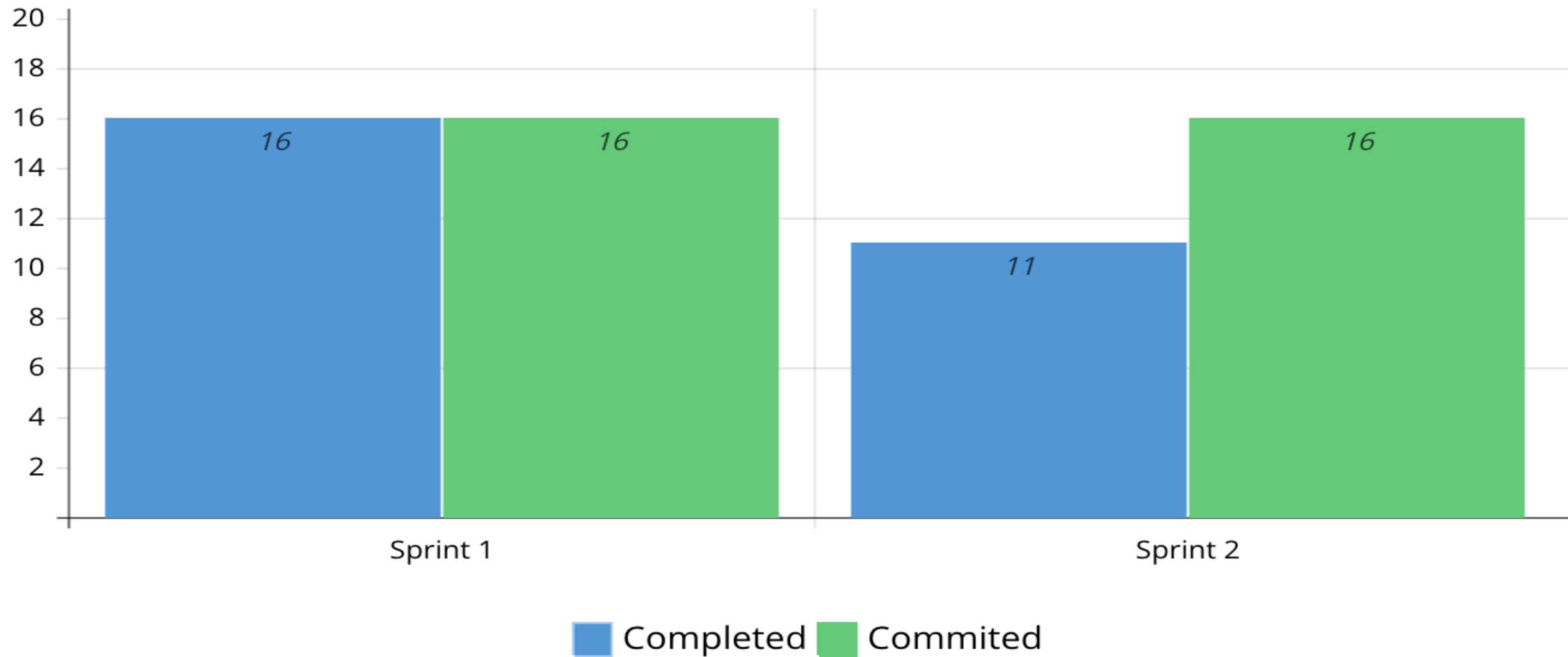
TEAM VELOCITY

Team Velocity



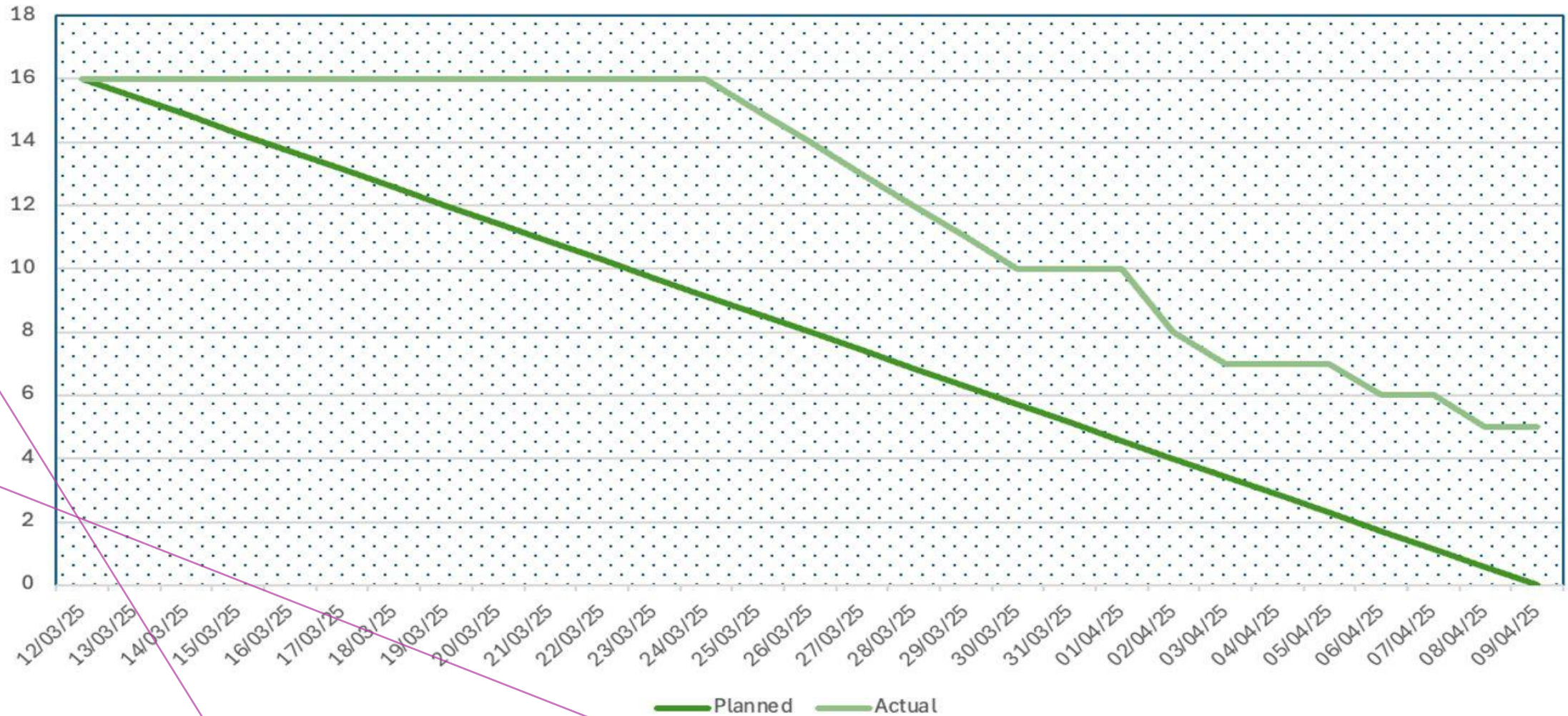
HISTORICAL TEAM VELOCITY

Historical Team Velocity



BURNDOWN CHART

Sprint 2 Burndown Chart



COMPLETED/COMMITTED RATIO - SPRINT 2



Committed Story
Points: 16



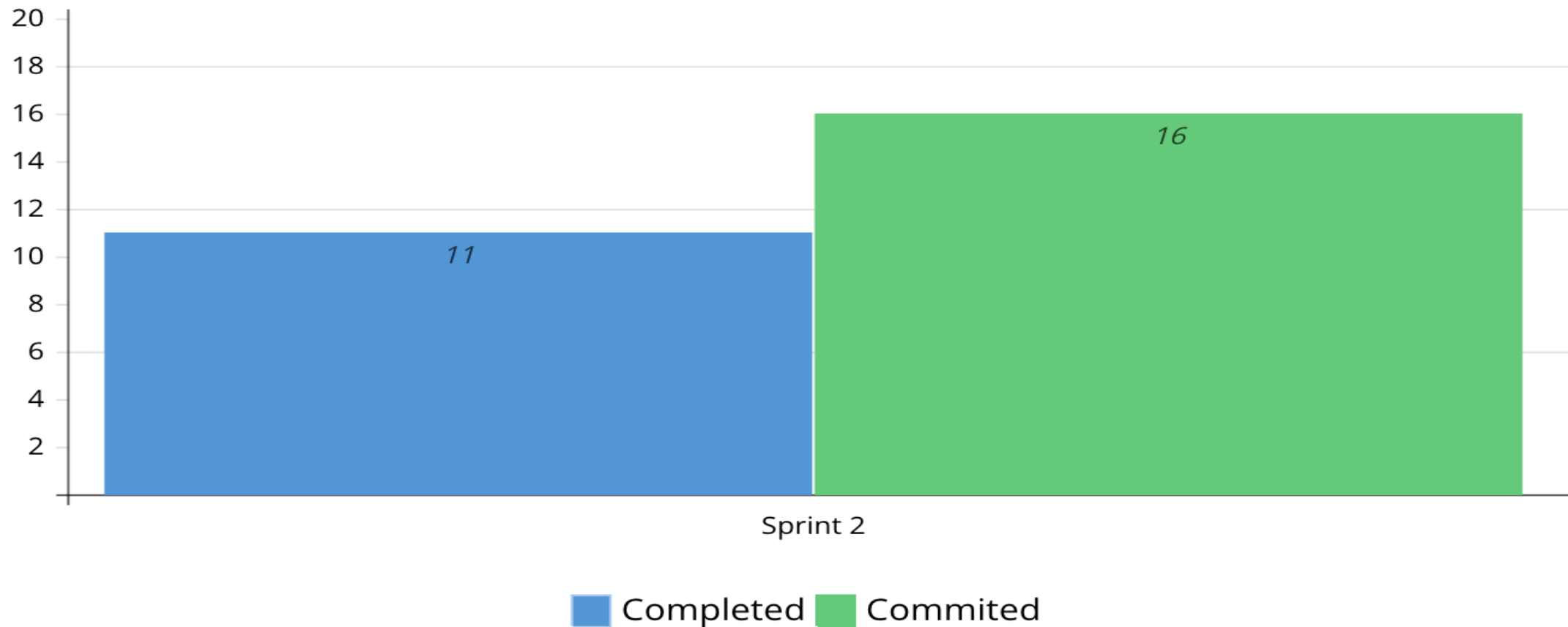
Completed Story
Points: 11



Completed/Committed Ratio:
 $(11/16) \times 100 =$
68.75%

COMPLETED/COMMITTED RATIO - SPRINT 2

Completed/Committed Ratio



RETROSPECTIVE

Welcome Ruchitha reddy k

My Boardz

Export

Logout

Techno Stack

What went well +

Team coordination + 0	Completed most of the tasks for sprint 2 + 0
Backend Api integration + 0	Implementation of testing model + 0
front end ui was good + 0	team collaboration + 0
Db schema setup + 0	

What can be improved +

Availability + 0	knowledge sharing + 1
Group agreement on when to turn deliverables + 0	internal team communication + 0
make sure we are all in agreement about the best way to implement features + 0	git commits + 0

Action Items +

Active learning + 0	have more meaningful commits and branches + 0
deadline submissions in internal team deadline for individuals + 0	

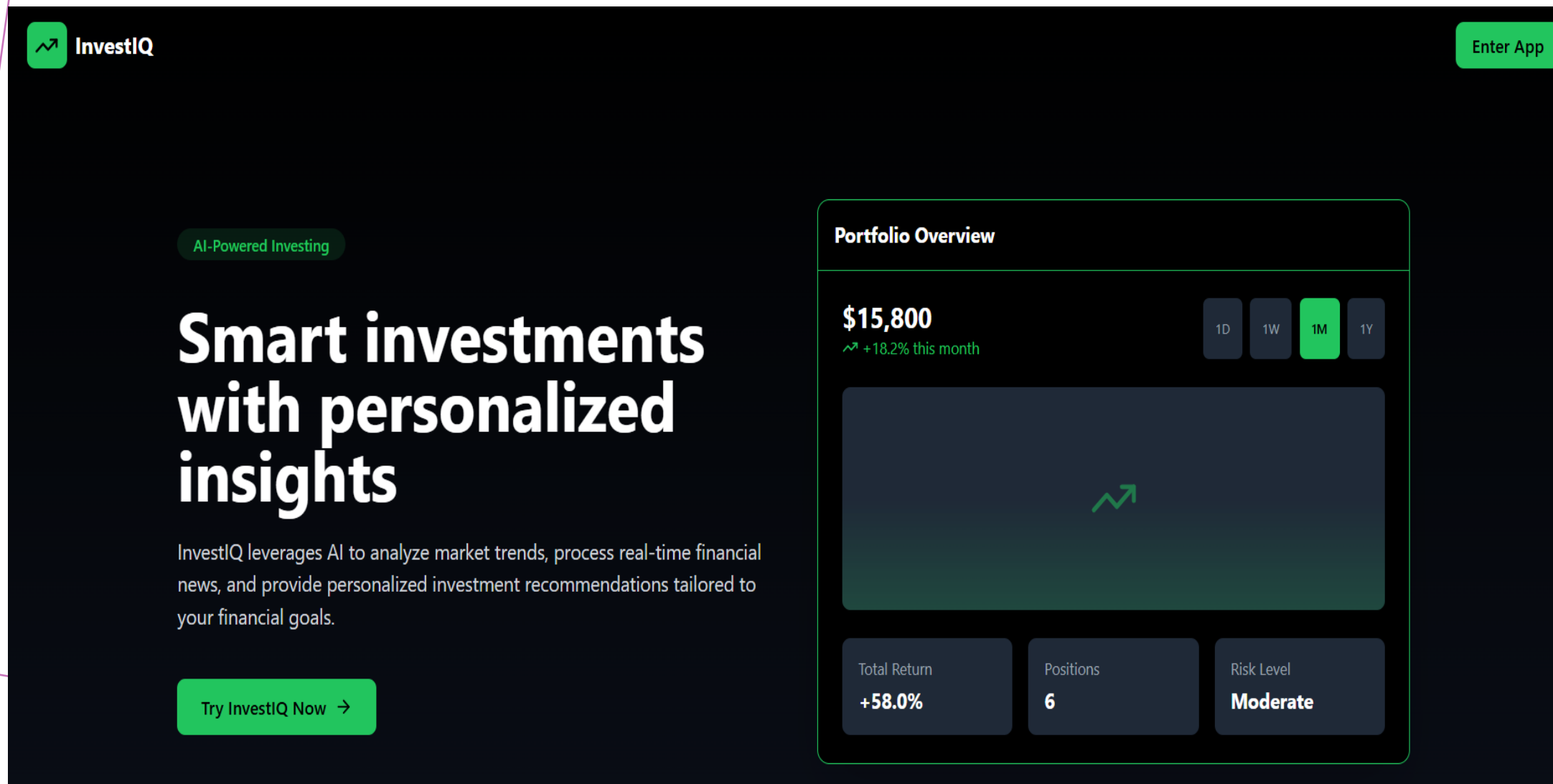
SPRINT 3 PLANNING

Story ID	Feature	User Story	Acceptance Criteria	Story Points
US7	Historical Trends Display	Thomas Shelby wants to see historical trends for recommended stocks, So that he can understand the basis for recommendations.	1. Dashboard displays a line chart showing 30-day price trend for each stock. 2. Chart uses historical data from the database. 3. Chart includes labels for dates and prices.	5
US8	Market News Display	Thomas Shelby wants to see recent market news, So that he can stay informed.	1. Dashboard includes a section with market news headlines. 2. News fetched from a free API (e.g., NewsAPI) and displayed as a list. 3. Each news item includes title, source, and link to the article.	5
US9	Personalized Recommendations	Thomas Shelby wants to see personalized investment recommendations, So that he can make informed decisions.	1. Recommendations replace the placeholder UI from Sprint 1. 2. Recommendations are based on historical trends and user profile (e.g., risk tolerance). 3. If no recommendations are available, a message like "No recommendations at this time" is shown.	5
US10	Recommendation Logic	As a developer needs to implement a rule-based recommendation system, So that	1. Algorithm uses historical trends (e.g., 30-day moving average) and user risk tolerance. 2. Example rule: Recommend stocks with upward trends for low-risk users.	5

SPRINT 3 PLANNING

Story ID	Feature	User Story	Acceptance Criteria	Story Points
TS5	News API Integration	As a developer needs to integrate a free news API (e.g., NewsAPI), So that the team can display market news.	1.Backend fetches news articles related to finance. 2.News data is cached for 24 hours. 3.Error handling for API failures.	3
US4	User Registration	Aarav Sharma wants to register with his email and password, So that he can access the InvestIQ platform.	1. Aarav can enter email and password on a registration form. 2. System validates email format and ensures password is at least 8 characters. 3. Upon successful registration, Aarav is redirected to login page. 4. Error messages shown if email is in use or inputs are invalid.	3
US5	User Login	Aarav Sharma wants to log in with his email and password, So that he can access his dashboard.	1. Aarav can enter email and password on a login form. 2. System validates credentials and logs Aarav in if correct. 3. Aarav is redirected to the dashboard upon successful login. 4. Error message shown if credentials are incorrect.	2

SCREENSHOTS



SCREENSHOTS

your financial goals.

Try InvestIQ Now →

Total Return
+58.0%

Positions
6

Risk Level
Moderate

About InvestIQ

InvestIQ is an AI-powered investment assistant that analyzes stock market trends, processes real-time financial news, and provides personalized investment insights and risk alerts. Unlike traditional investment platforms that only offer raw data and generic analytics, our application leverages AI-driven time series analysis and natural language processing to deliver real-time, context-aware financial recommendations tailored to users' personal financial goals and life events.

Explore the App

InvestIQ - Capstone Project

Professor Henry Wong - Spring 2025

© 2025 Techno Stack. All rights reserved.

SCREENSHOTS



SCREENSHOTS



SCREENSHOTS

Deposit Funds

×

Amount to Deposit

\$ Enter amount


\$100

\$500

\$1000


\$5000

Payment Method



Bank Transfer

Transfer directly from your bank (1-3 business days)

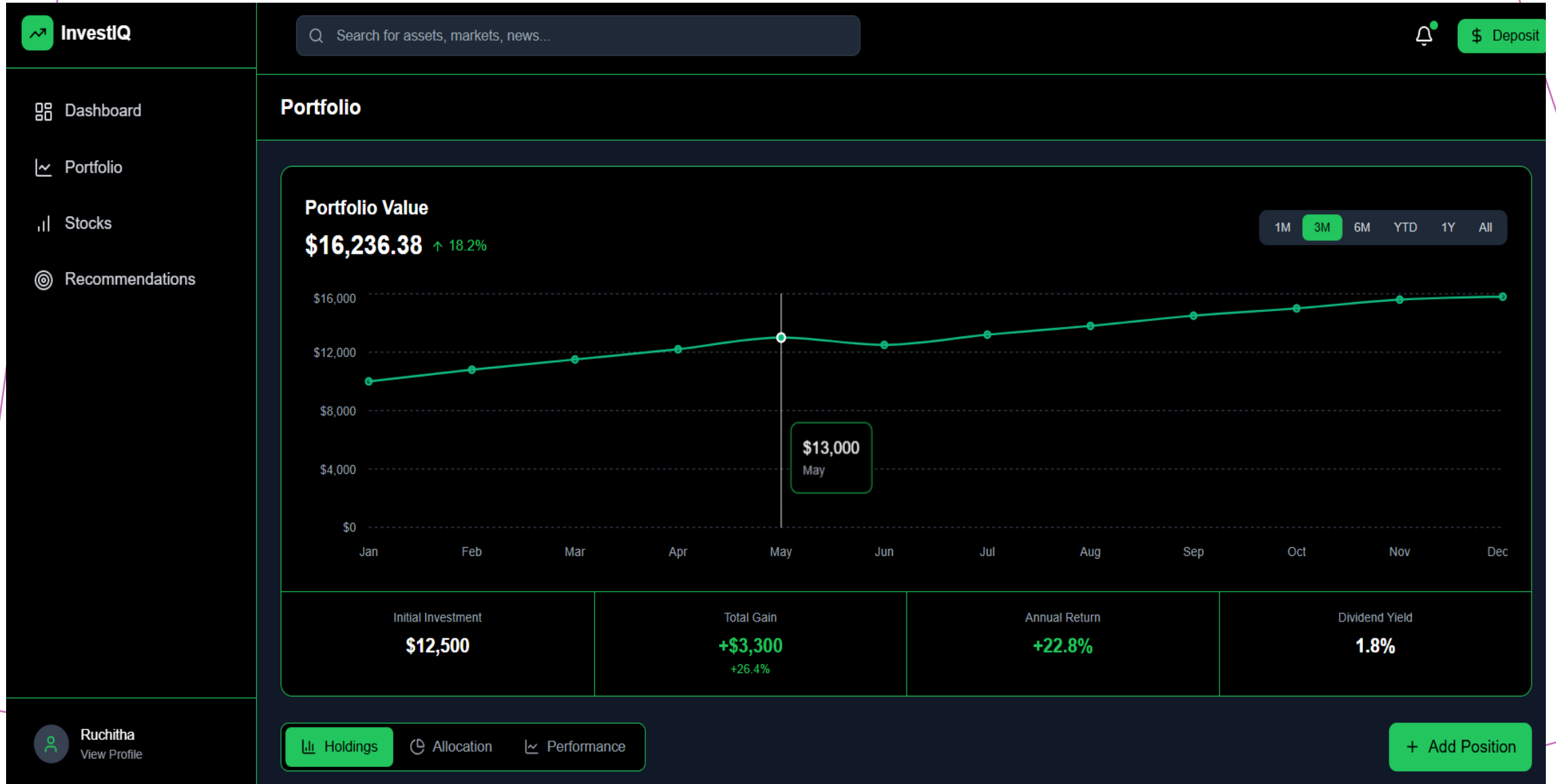


Credit Card

Instant deposit with 1.5% fee

Deposit →

SCREENSHOTS



SCREENSHOTS

InvestIQ

Dashboard

Portfolio

Stocks

Recommendations

Ruchitha

View Profile

Search for assets, markets, news...

Deposit

JanFebMarAprMayJunJulAugSepOctNovDec

Initial Investment

\$12,500

Total Gain

+\$3,300

+26.4%

Annual Return

+22.8%

Dividend Yield

1.8%

Holdings

Allocation

Performance

+ Add Position

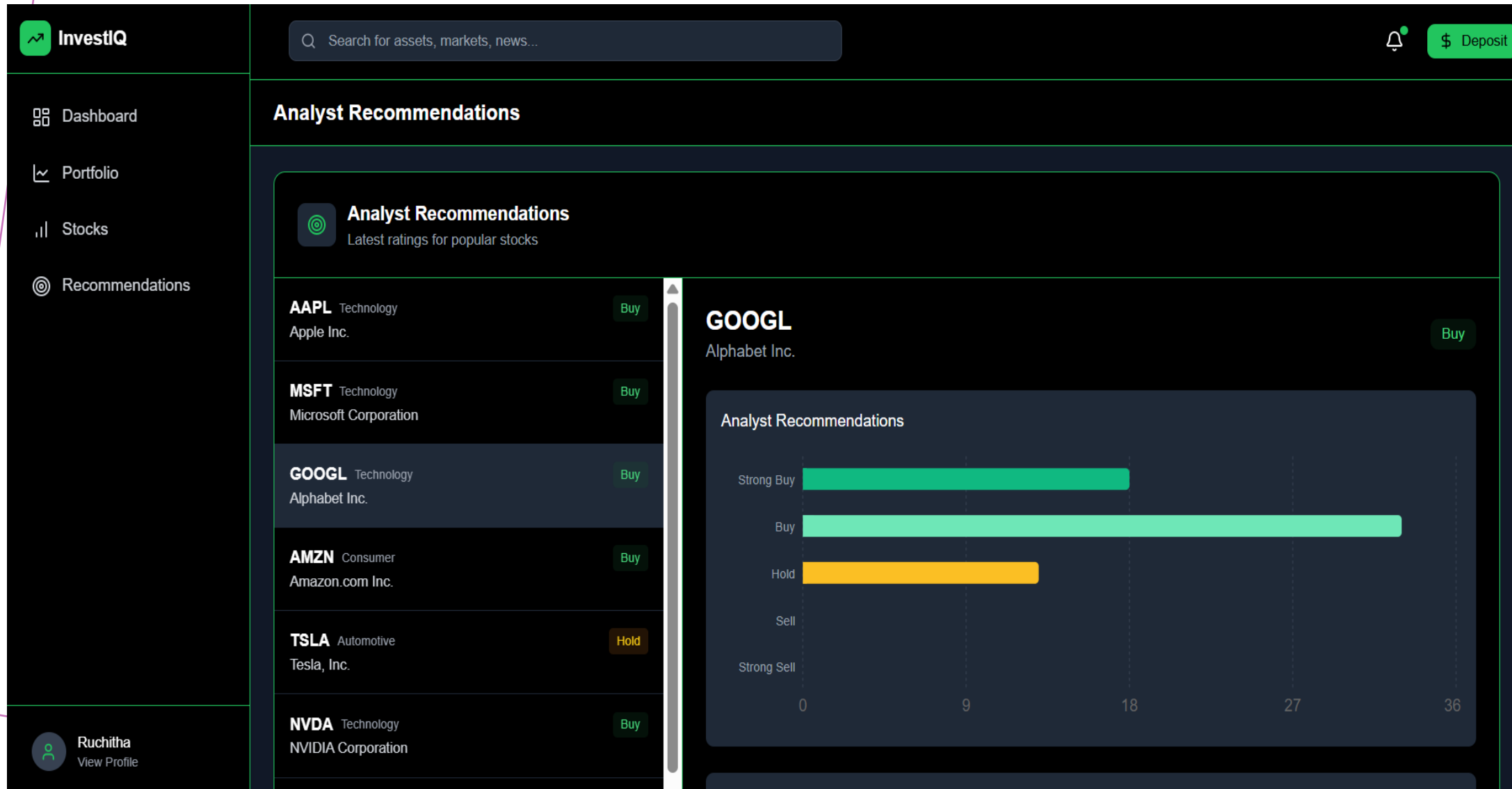
Your Holdings

Symbol	Name	Shares	Avg. Cost	Price	Value	Weight	Gain/Loss	Actions
AAPL	Apple Inc.	15	\$160.75	\$187.68	\$2815.20	17.5%	↑ \$404.25 (16.75%)	...
MSFT	Microsoft Corp.	10	\$380.25	\$419.65	\$4196.50	26.1%	↑ \$394.00 (10.35%)	...
GOOGL	Alphabet Inc.	8	\$125.50	\$148.90	\$1191.20	7.4%	↑ \$187.20 (18.65%)	...
AMZN	Amazon.com Inc.	12	\$150.80	\$182.41	\$2188.92	13.6%	↑ \$379.32 (20.96%)	...
NVDA	NVIDIA Corp.	5	\$780.40	\$950.02	\$4750.10	29.5%	↑ \$848.10 (21.73%)	...
JPM	JPMorgan Chase & Co.	6	\$160.25	\$182.41	\$1094.46	6.8%	↑ \$133.00 (13.83%)	...

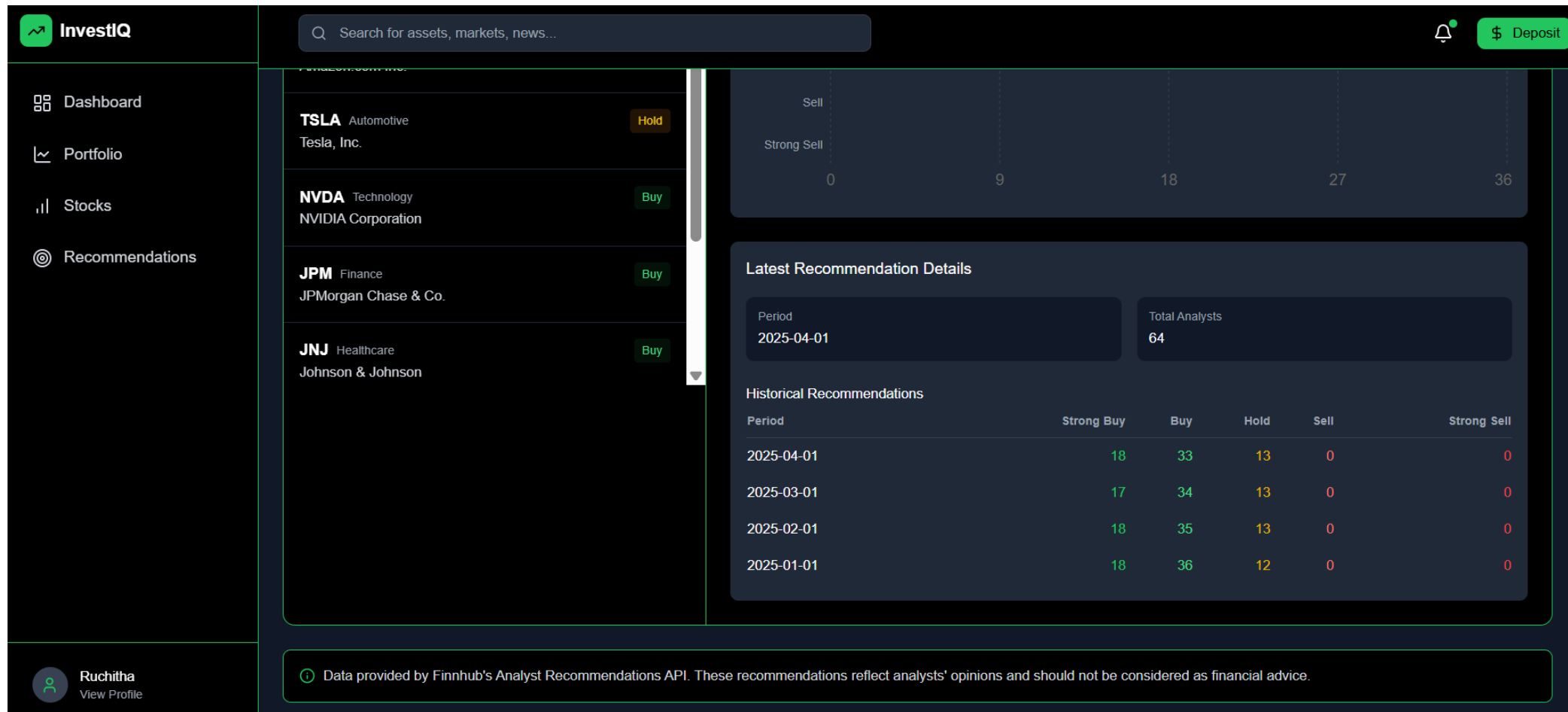
SCREENSHOTS

[illegible]

SCREENSHOTS



SCREENSHOTS



SCREENSHOTS

~/Documents/temp (0.068s)

```
curl "http://localhost:3000/api/recommendations?symbol=AAPL"
```

```
[{"buy":24,"hold":7,"period":"2025-03-01","sell":0,"strongBuy":13,"strongSell":0,"symbol":"AAPL"}, {"buy":17,"hold":13,"period":"2025-02-01","sell":5,"strongBuy":13,"strongSell":0,"symbol":"AAPL"}]
```

WIKI PAGE LINK

[https://github.com/htmhw/2025S-
Techno-Stack](https://github.com/htmhw/2025S-Techno-Stack)

LIVE DEMO





THANK YOU