INVESTIQ

BY TEAM 02 TECHNO STACK

AGENDA

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

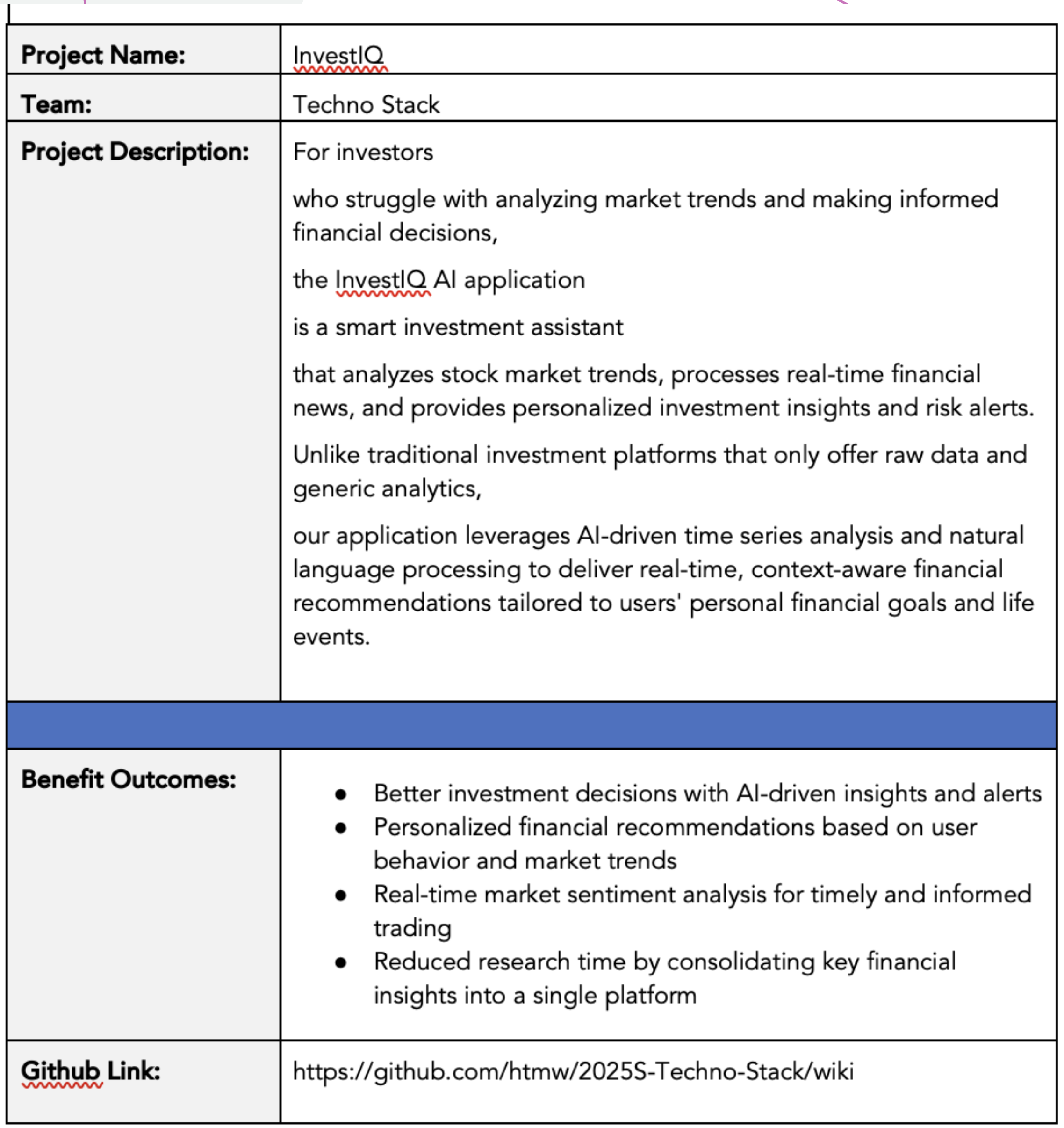
Nadeem Hussain Shaik – Frontend Developer

Shrinidhi Daheechi – Frontend Developer

PROBLEM STATEMENT

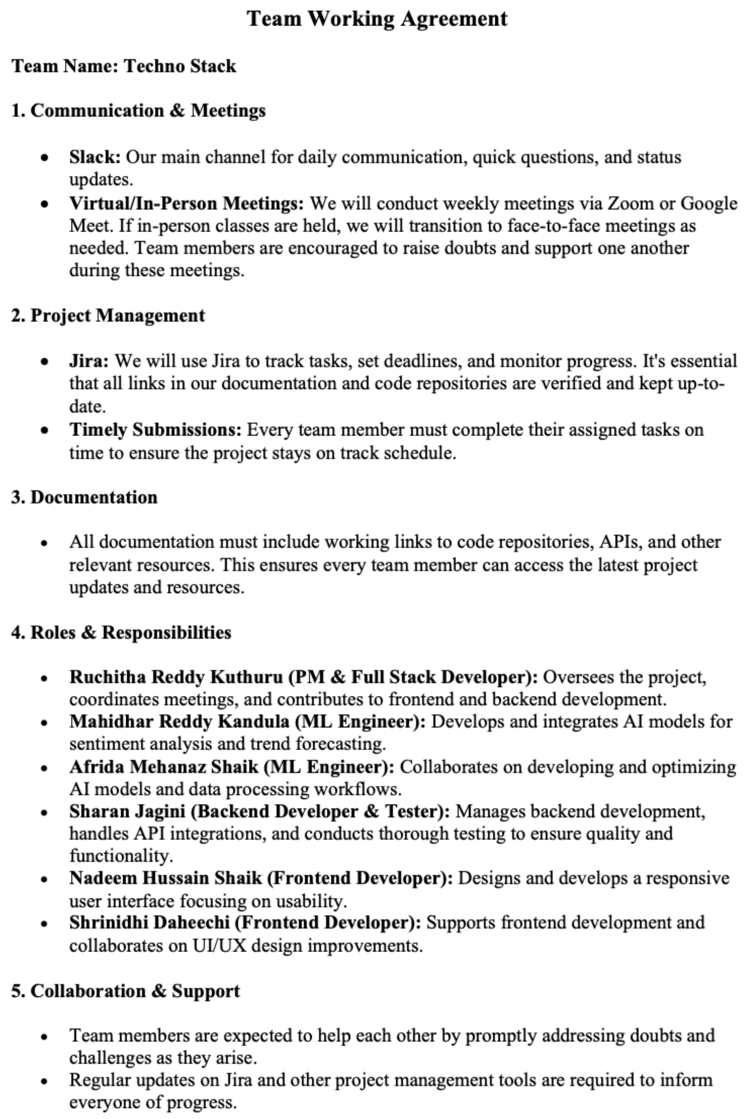
Investors often find it challenging to make informed decisions due to overwhelming market information, constant fluctuations, and a lack of personalized guidance. Many existing tools provide data but do not offer meaningful insights tailored to individual needs. This app aims to simplify investment decisions 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:	InvestIQ
Team:	Techno Stack
Project Description:	For investors who struggle with analyzing market trends and making informed financial decisions, the InvestIQ 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.
Benefit Outcomes:	• 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
Github 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

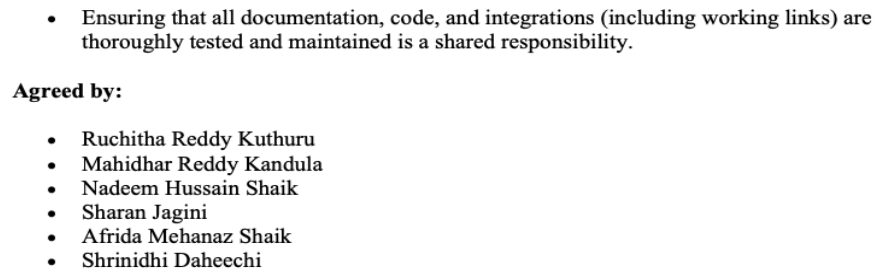
- All documentation must include working links to code repositories, APIs, and other relevant resources. This ensures every team member can access the latest project updates and resources.

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.

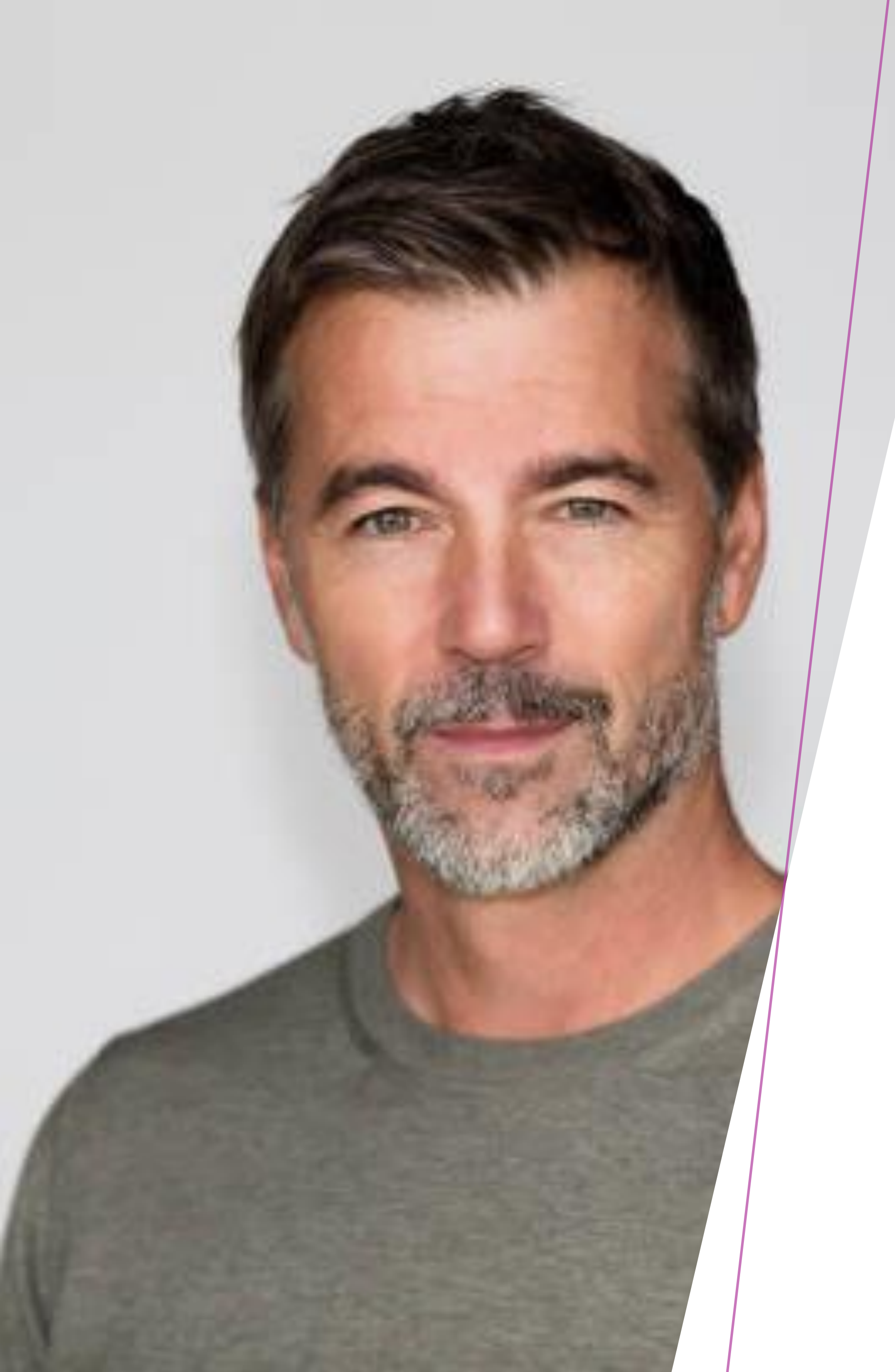
Agreed by:

- 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.

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.

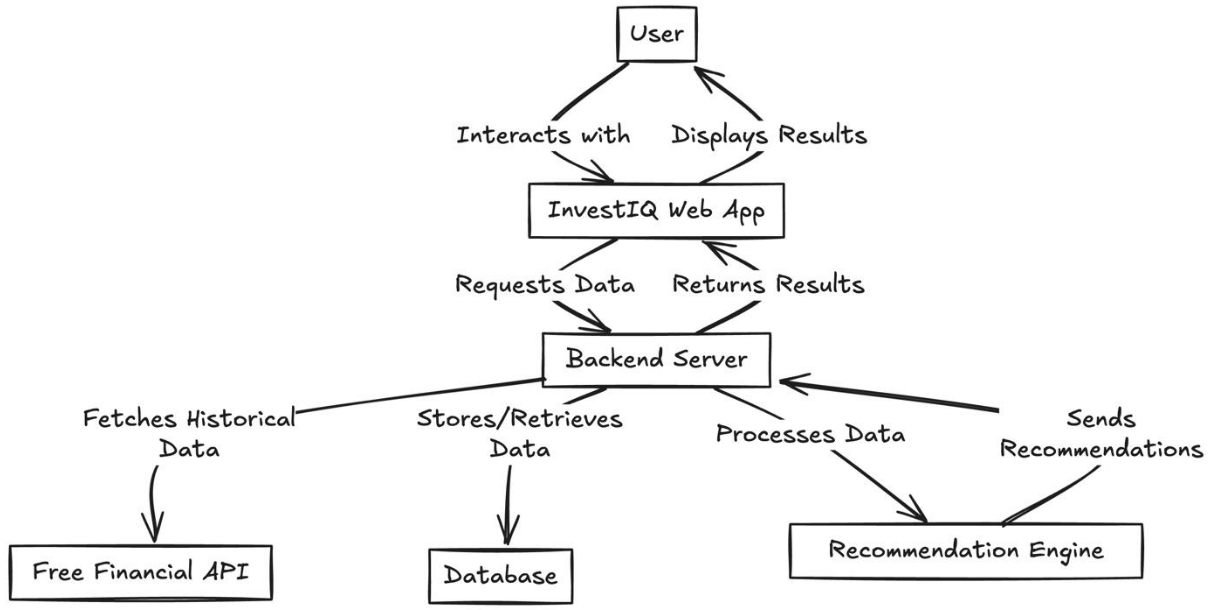
ALGORITHMS - NLP

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.

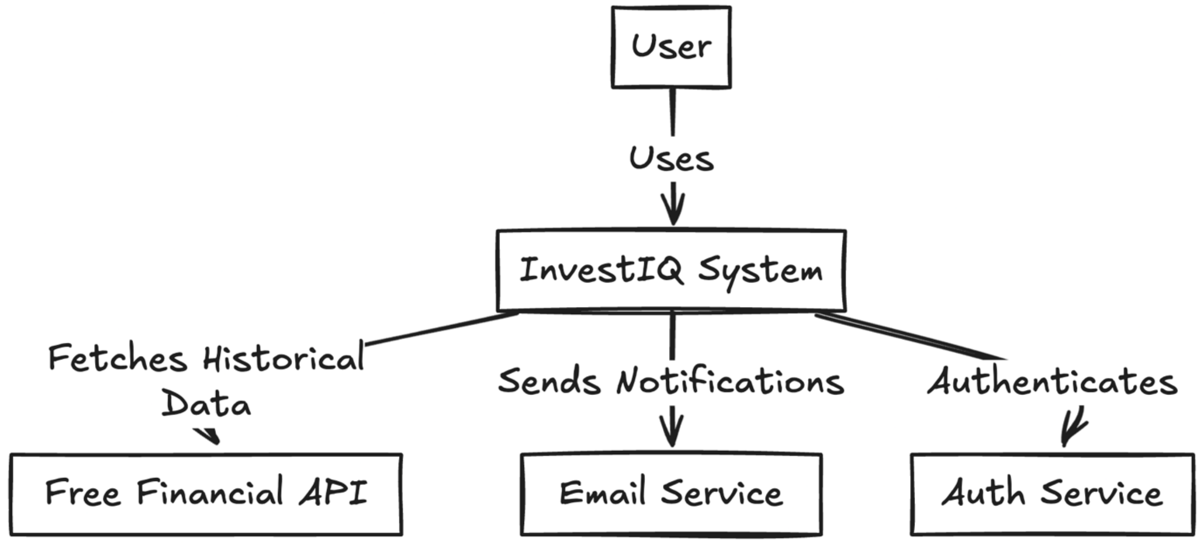
ALGORITHMS – TIME SERIES

- 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.

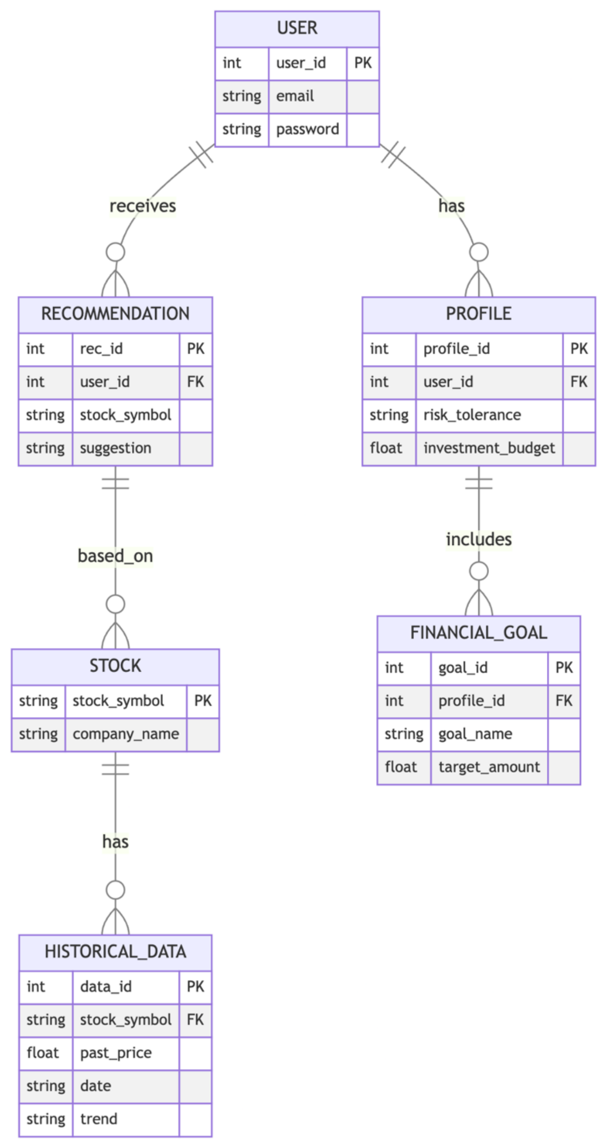
ARCHITECTURE DIAGRAM



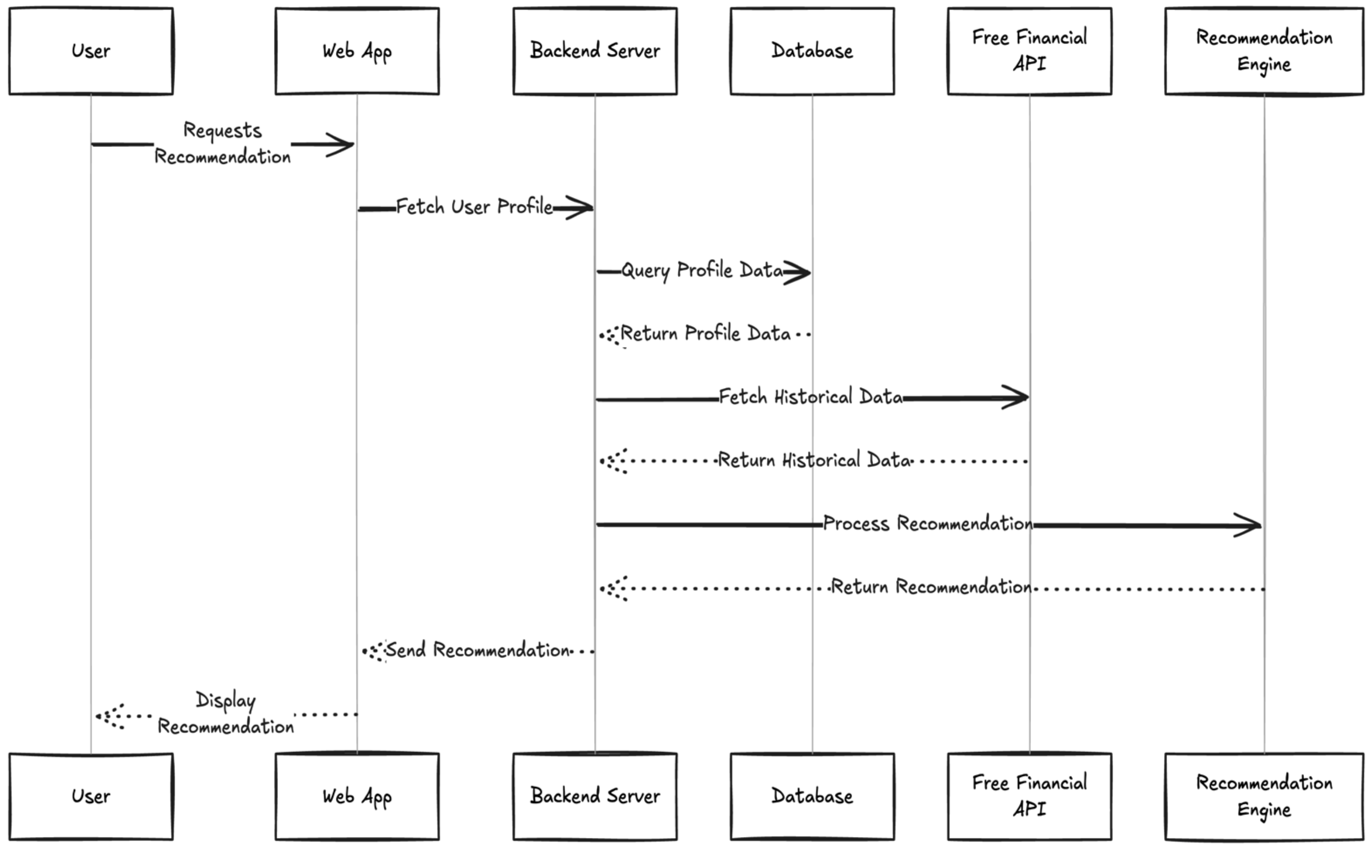
CONTEXT DIAGRAM



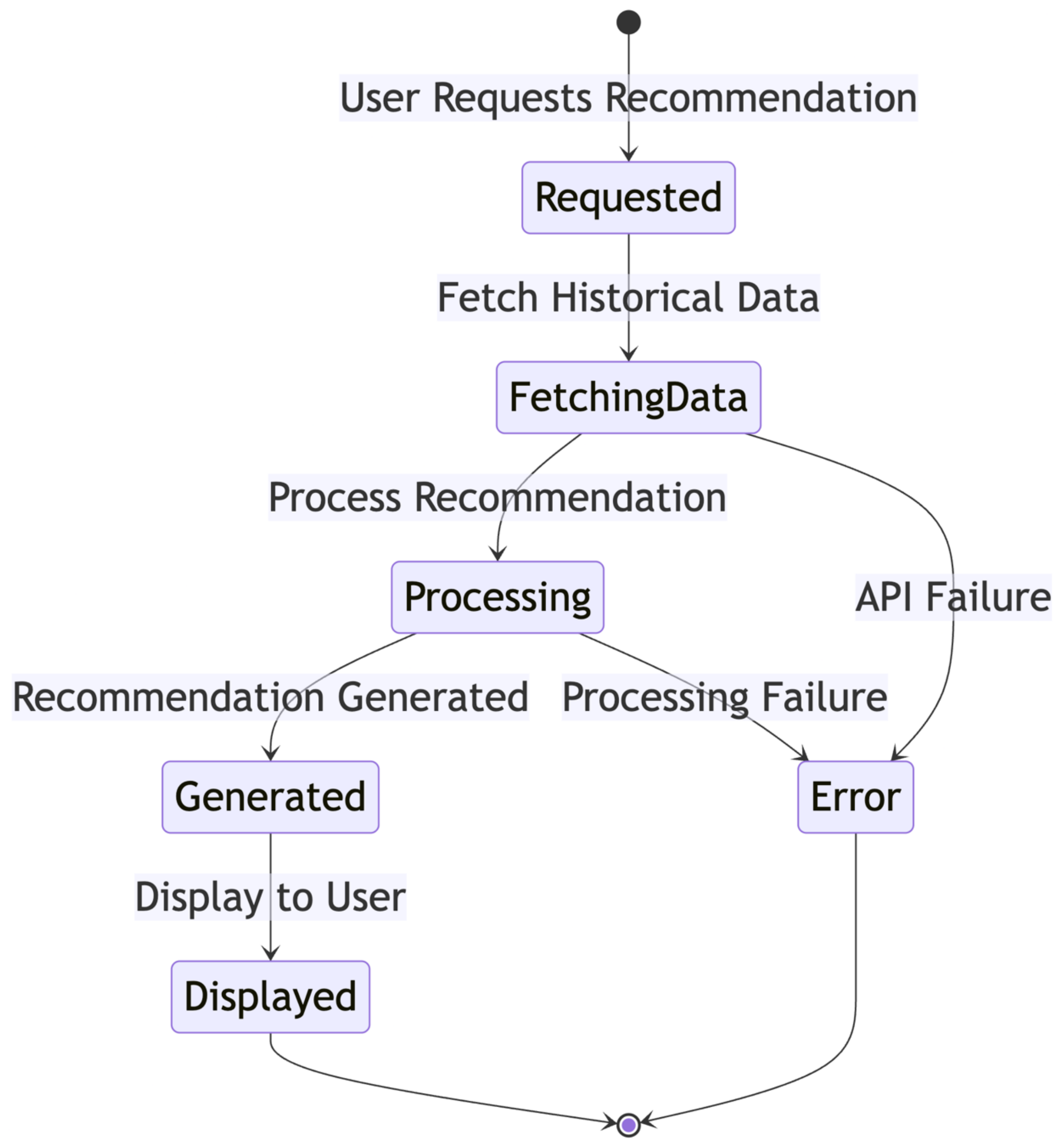
ER DIAGRAM (ENTITY-RELATIONSHIP DIAGRAM)



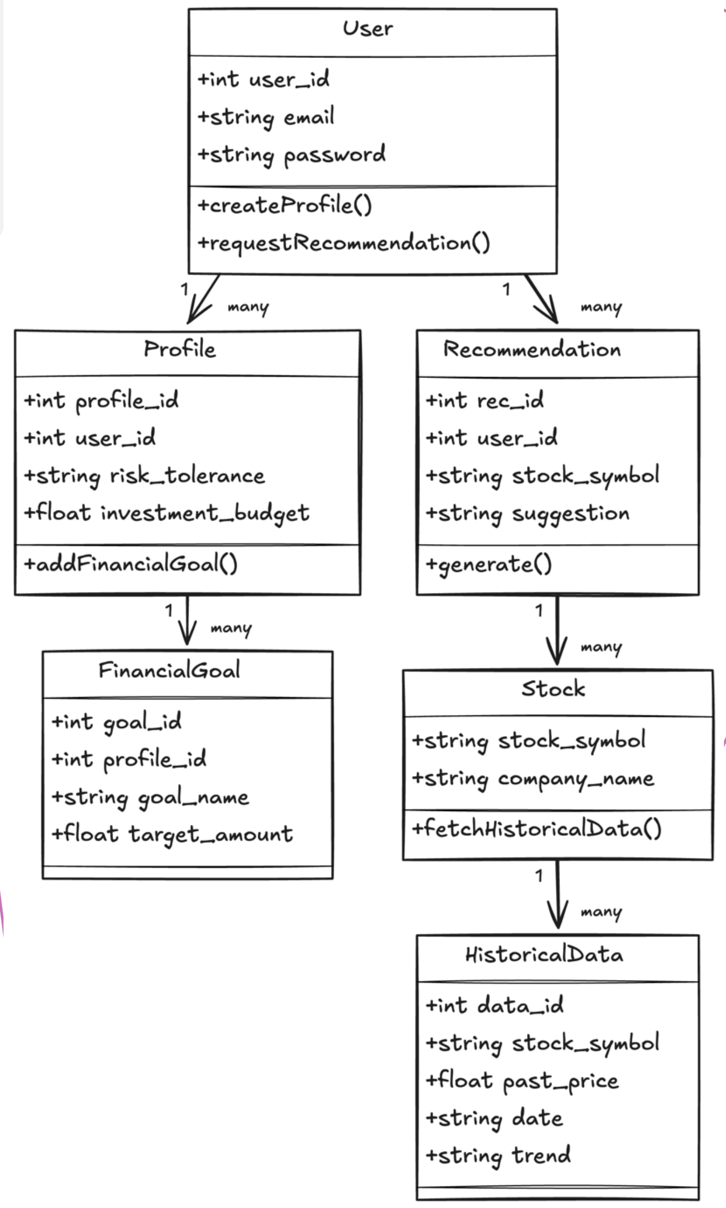
SEQUENCE DIAGRAM



STATE DIAGRAM



CLASS DIAGRAM



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.	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.	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.	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.	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.	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

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.	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.	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

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.	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.	1. Backend fetches news articles related to finance. 2. News data is cached for 24 hours. 3. Error handling for API failures.	3

SPRINT 1 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.	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.	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.	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.	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

Total Sprint 1. 16

TEST CASES SPRINT 1

Test Case ID	Story ID	Feature	Test Description	Test Steps	Expected Result	Actual Result	Status
TC1.1	US1	Basic Dashboard UI	Verify that the dashboard displays sections for recommendations, trends, and news.	Launch the InvestIQ application, navigate to the dashboard page, and observe if sections are present.	All sections are visible and correctly labeled.	Sections for recommendations, trends, and news are displayed as expected.	✅ Pass
TC1.2	US1	Basic Dashboard UI	Ensure that the dashboard has a header, sidebar, and main content area.	Open the dashboard and verify the presence of the header, sidebar, and main content.	The layout includes a header, sidebar, and content area.	All UI elements are properly structured.	✅ Pass
TC1.3	US1	Basic Dashboard UI	Validate UI responsiveness on different devices.	Open the app on desktop, resize the window, then open on a mobile device and verify layout adjustments.	Dashboard adjusts properly for both desktop and mobile screens.	UI responds correctly on different screen sizes.	✅ Pass
TC2.1	US2	Profile Setup UI	Check if the profile form includes a dropdown for risk tolerance and an input for budget.	Navigate to the profile setup page and check for the presence of the risk tolerance dropdown and budget input field.	The form contains all necessary fields.	Dropdown and input field are present.	✅ Pass

TEST CASES SPRINT 1

Test Case ID	Story ID	Feature	Test Description	Test Steps	Expected Result	Actual Result	Status
TC2.2	US2	Profile Setup UI	Validate the "Save" button functionality.	Enter data in the profile form, click the "Save" button, and observe the confirmation message.	A confirmation message appears after saving.	The confirmation message is displayed as expected.	✅ Pass
TC2.3	US2	Profile Setup UI	Ensure error handling for invalid budget input.	Enter a negative or non-numeric budget, click "Save," and check if an error message appears.	An error message should be displayed.	Proper error message is shown when input is invalid.	✅ Pass
TC3.1	US3	Recommendation Placeholder UI	Verify the presence of the "Recommendations" section on the dashboard.	Open the dashboard and look for the "Recommendations" section.	The section is visible.	"Recommendations" section appears in the correct location.	✅ Pass
TC3.2	US3	Recommendation Placeholder UI	Ensure that three mock recommendations are displayed.	Open the dashboard and check if three stock recommendations are visible.	Three placeholder recommendations should be displayed.	Three mock recommendations appear as expected.	✅ Pass

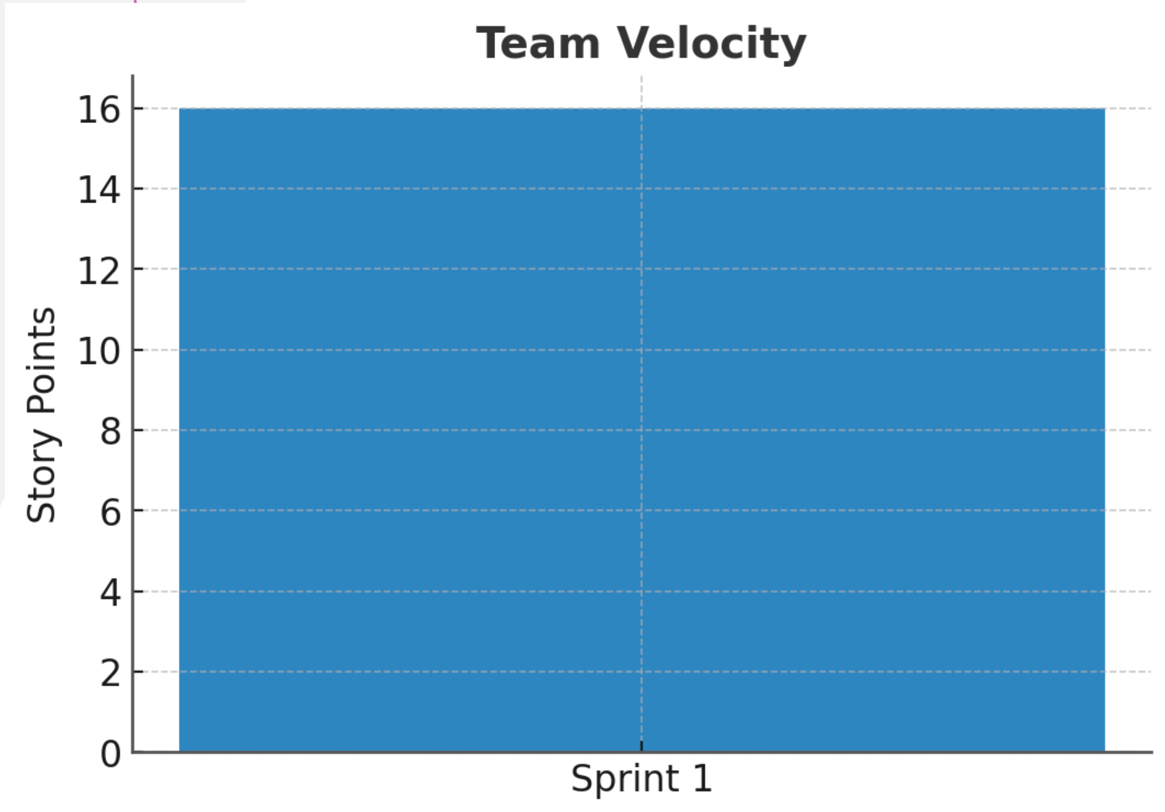
TEST CASES SPRINT 1

Test Case ID	Story ID	Feature	Test Description	Test Steps	Expected Result	Actual Result	Status
TC3.3	US3	Recommendation Placeholder UI	Validate that each recommendation has a "View Details" button.	Open the dashboard and check each recommendation for a "View Details" button.	All recommendations should have a button.	Each recommendation has a correctly labeled "View Details" button.	✅ Pass
TC4.1	TS1	Front-End Framework Setup	Ensure the project initializes successfully with React.	Clone the repo, run the project setup command, and start the application.	The project should compile and start without errors.	Project initialized and started successfully.	✅ Pass
TC4.2	TS1	Front-End Framework Setup	Verify basic routing between dashboard and profile pages.	Start the app, click on "Profile" from the dashboard, then click on "Dashboard" from the profile page.	The app should navigate between pages correctly.	Navigation between pages is smooth and functional.	✅ Pass
TC4.3	TS1	Front-End Framework Setup	Check if the CSS framework is applied correctly.	Inspect UI elements and ensure styling is consistent with the CSS framework used.	UI elements should reflect Tailwind/Bootstrap styles.	Styling is properly applied.	✅ Pass

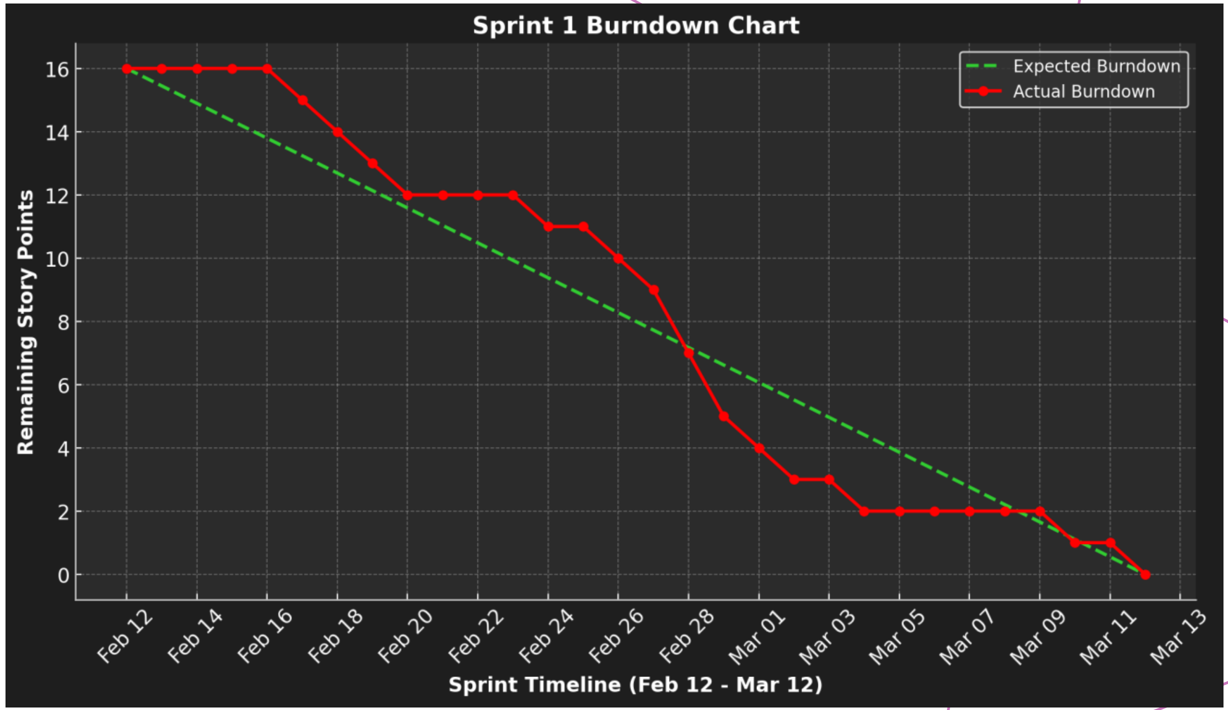
STORIES COMPLETED IN SPRINT 1

- **US1** – Basic Dashboard UI
- **US2** – Profile Setup UI
- **US3** – Recommendation Placeholder UI
- **TS1** – Front-End Framework Setup

TEAM VELOCITY



BURNDOWN CHART



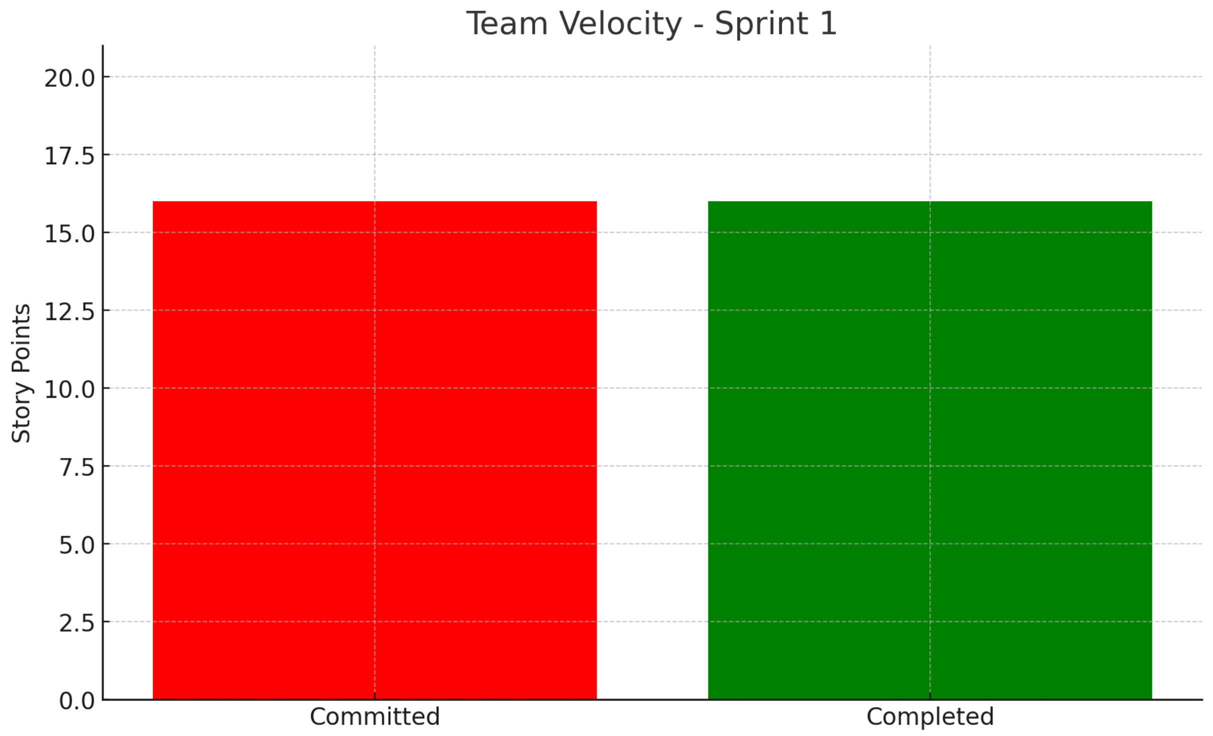
COMPLETED/COMMITTED RATIO - SPRINT 1

Committed Story Points: 16

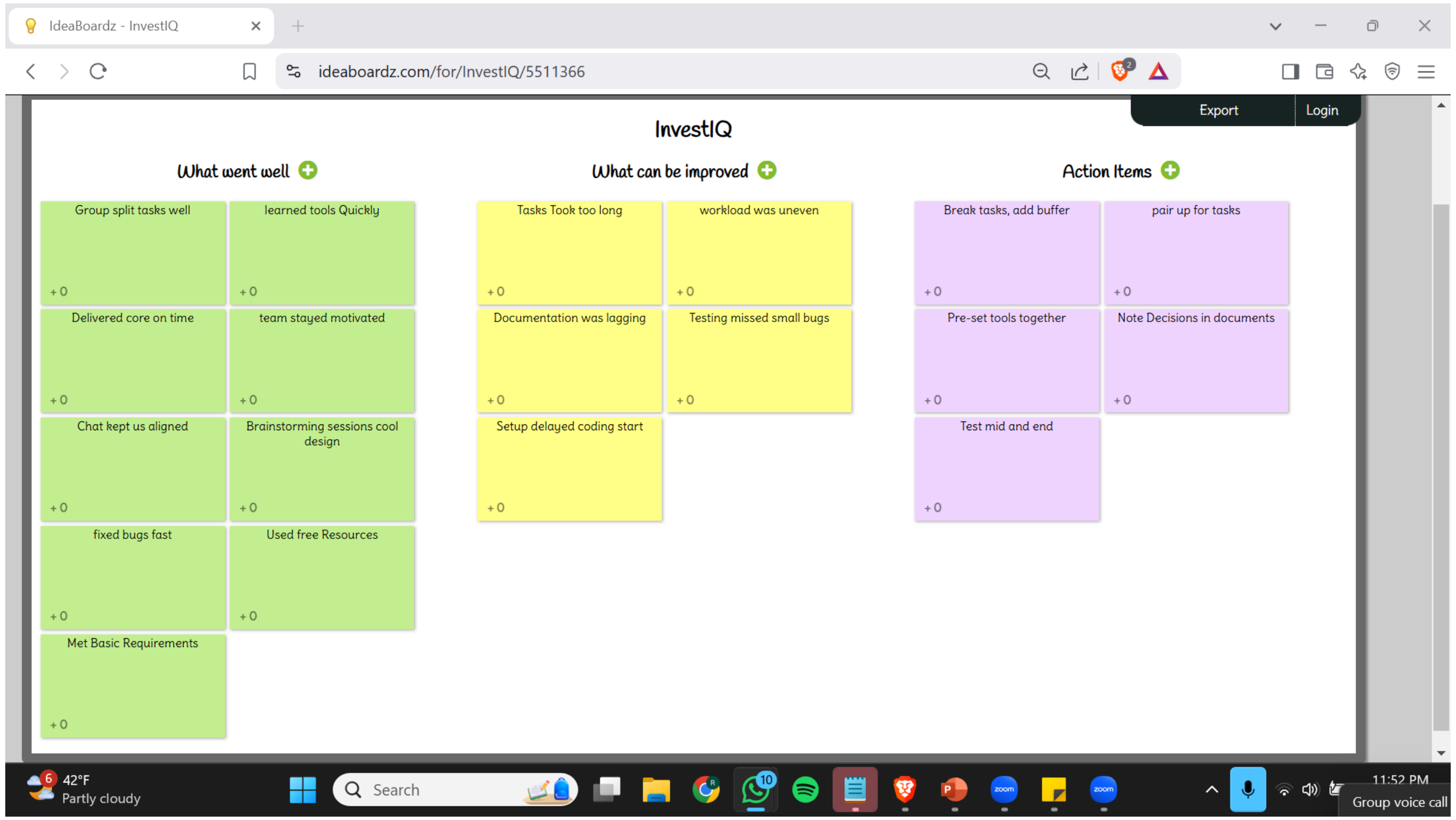
Completed Story Points: 16

Completed/Committed Ratio: (16/16) × 100 = 100%

COMPLETED/COMMITTED RATIO - SPRINT 1



RETROSPECTIVE



InvestIQ

What went well		What can be improved		Action Items	
Group split tasks well +0	learned tools Quickly +0	Tasks Took too long +0	workload was uneven +0	Break tasks, add buffer +0	pair up for tasks +0
Delivered core on time +0	team stayed motivated +0	Documentation was lagging +0	Testing missed small bugs +0	Pre-set tools together +0	Note Decisions in documents +0
Chat kept us aligned +0	Brainstorming sessions cool design +0	Setup delayed coding start +0		Test mid and end +0	
fixed bugs fast +0	Used free Resources +0				
Met Basic Requirements +0					

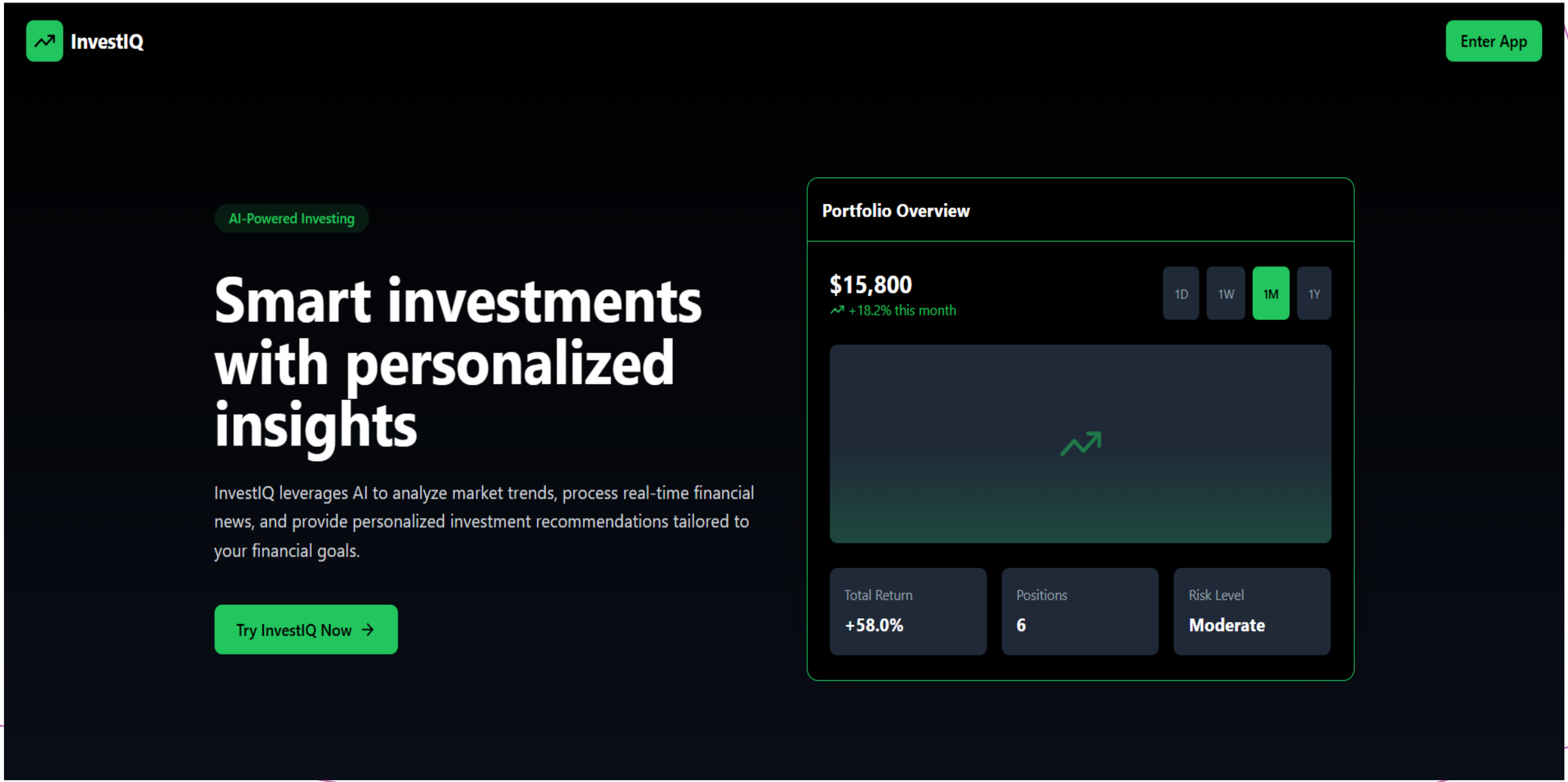
SPRINT 2 PLANNING

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 PLANNING

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.	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

SCREENSHOTS



InvestIQ

Enter App

AI-Powered Investing

Smart investments with personalized insights

InvestIQ leverages AI to analyze market trends, process real-time financial news, and provide personalized investment recommendations tailored to your financial goals.

Try InvestIQ Now →

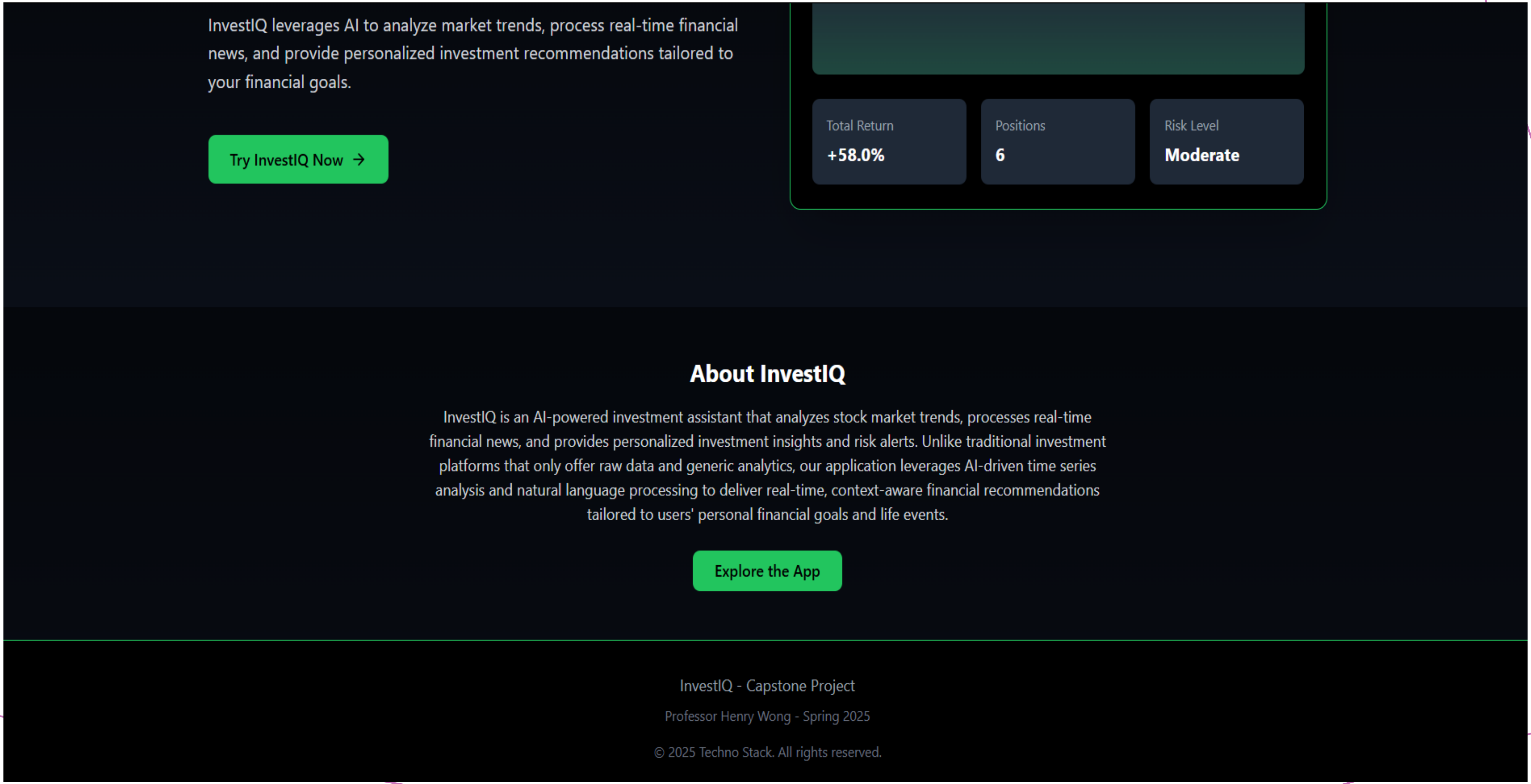
Portfolio Overview

$15,800
+18.2% this month

1D 1W 1M 1Y

Total Return	Positions	Risk Level
+58.0%	6	Moderate

SCREENSHOTS


InvestIQ leverages AI to analyze market trends, process real-time financial news, and provide personalized investment recommendations tailored to your financial goals.

Total Return	Positions	Risk Level
+58.0%	6	Moderate

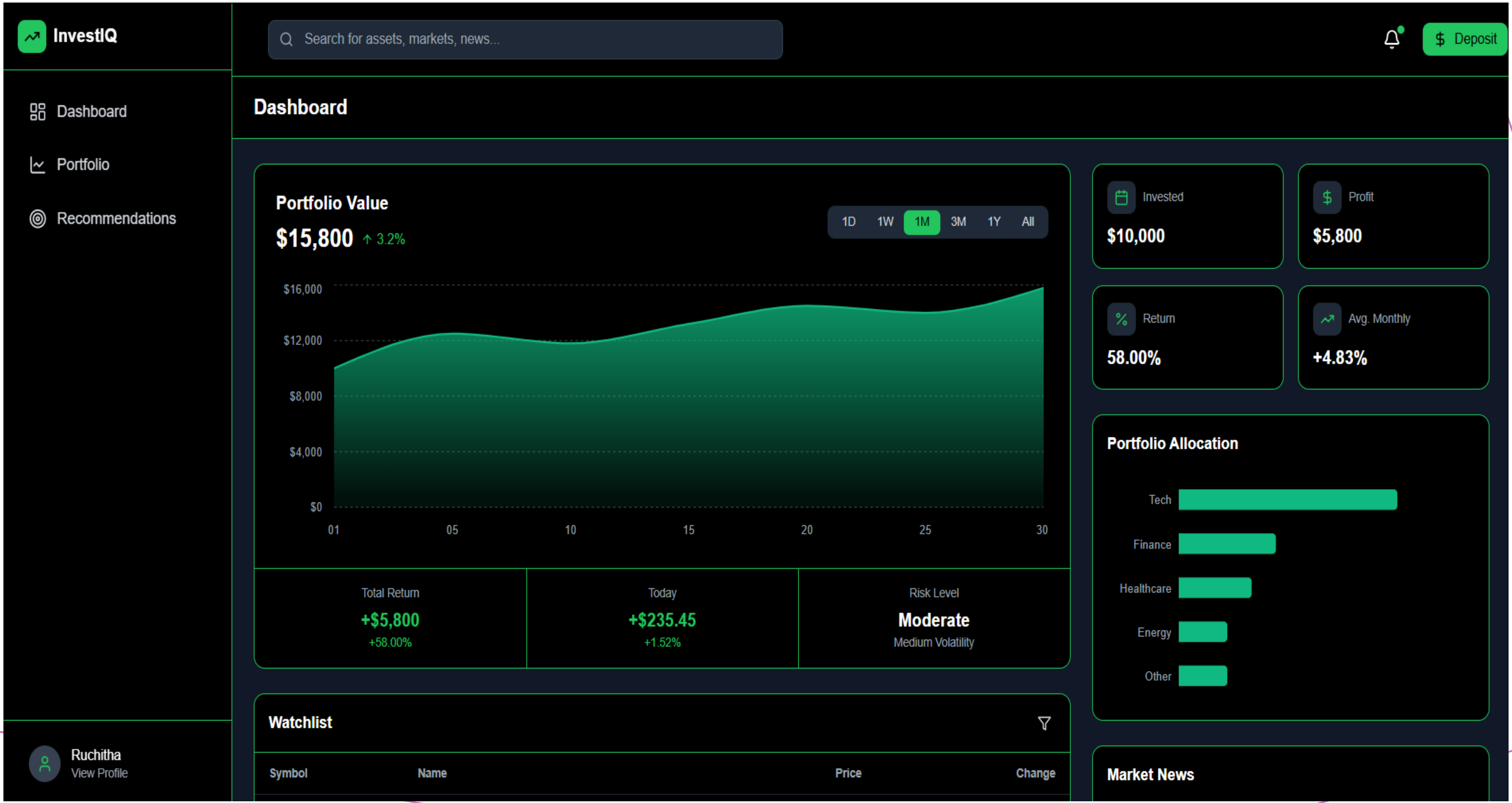
About InvestIQ

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.

InvestIQ - Capstone Project

Professor Henry Wong - Spring 2025

SCREENSHOTS



Dashboard

Portfolio Value

$15,800 ↑ 3.2%

Invested	Profit
$10,000	$5,800

Return	Avg. Monthly
58.00%	+4.83%

Total Return	Today	Risk Level
+$5,800	+$235.45	Moderate
+58.00%	+1.52%	Medium Volatility

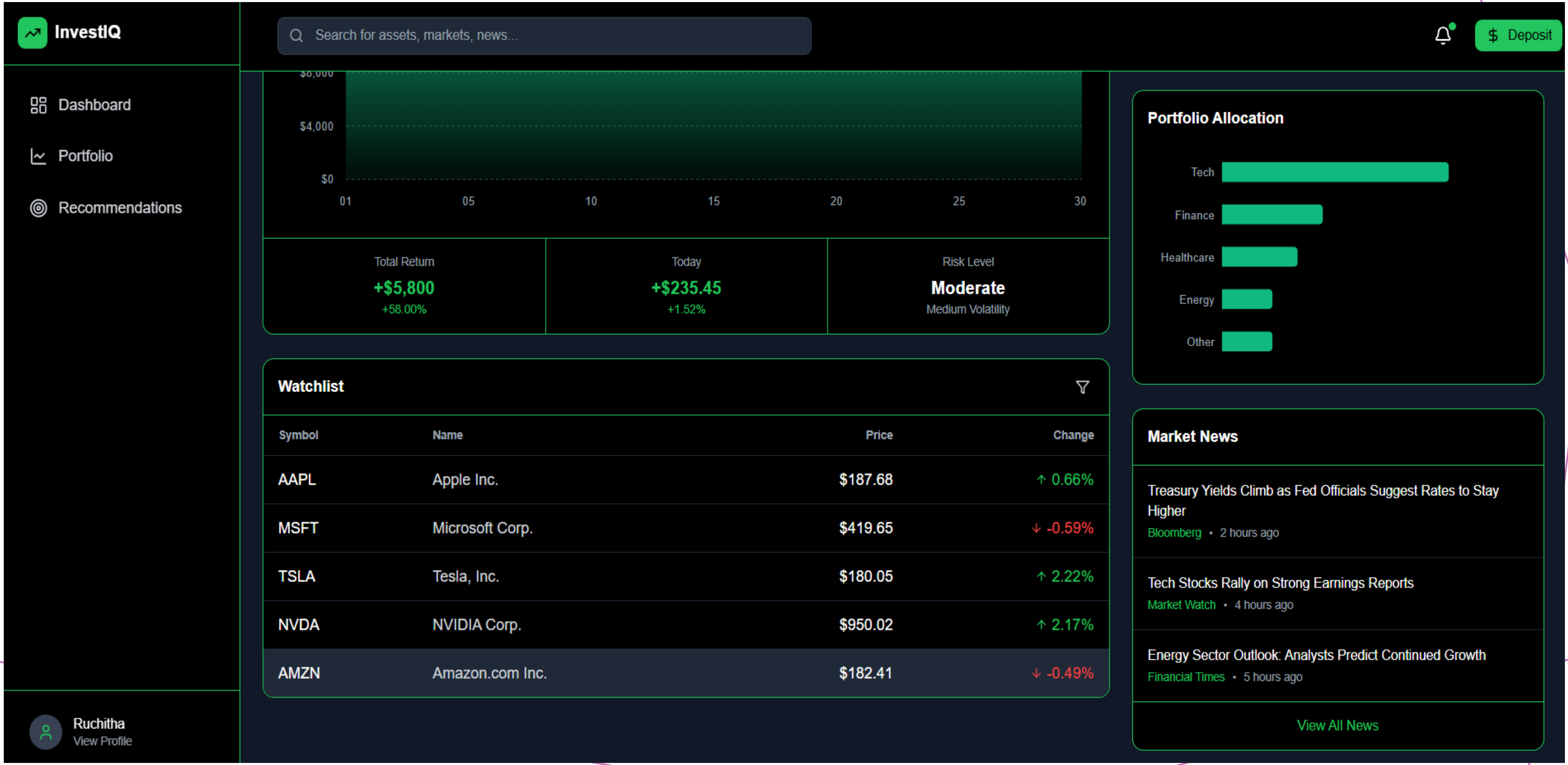
Portfolio Allocation

Watchlist

Symbol	Name	Price	Change

Market News

SCREENSHOTS



Total Return	Today	Risk Level
+$5,800	+$235.45	Moderate
+58.00%	+1.52%	Medium Volatility

Watchlist

Symbol	Name	Price	Change
AAPL	Apple Inc.	$187.68	↑ 0.66%
MSFT	Microsoft Corp.	$419.65	↓ -0.59%
TSLA	Tesla, Inc.	$180.05	↑ 2.22%
NVDA	NVIDIA Corp.	$950.02	↑ 2.17%
AMZN	Amazon.com Inc.	$182.41	↓ -0.49%

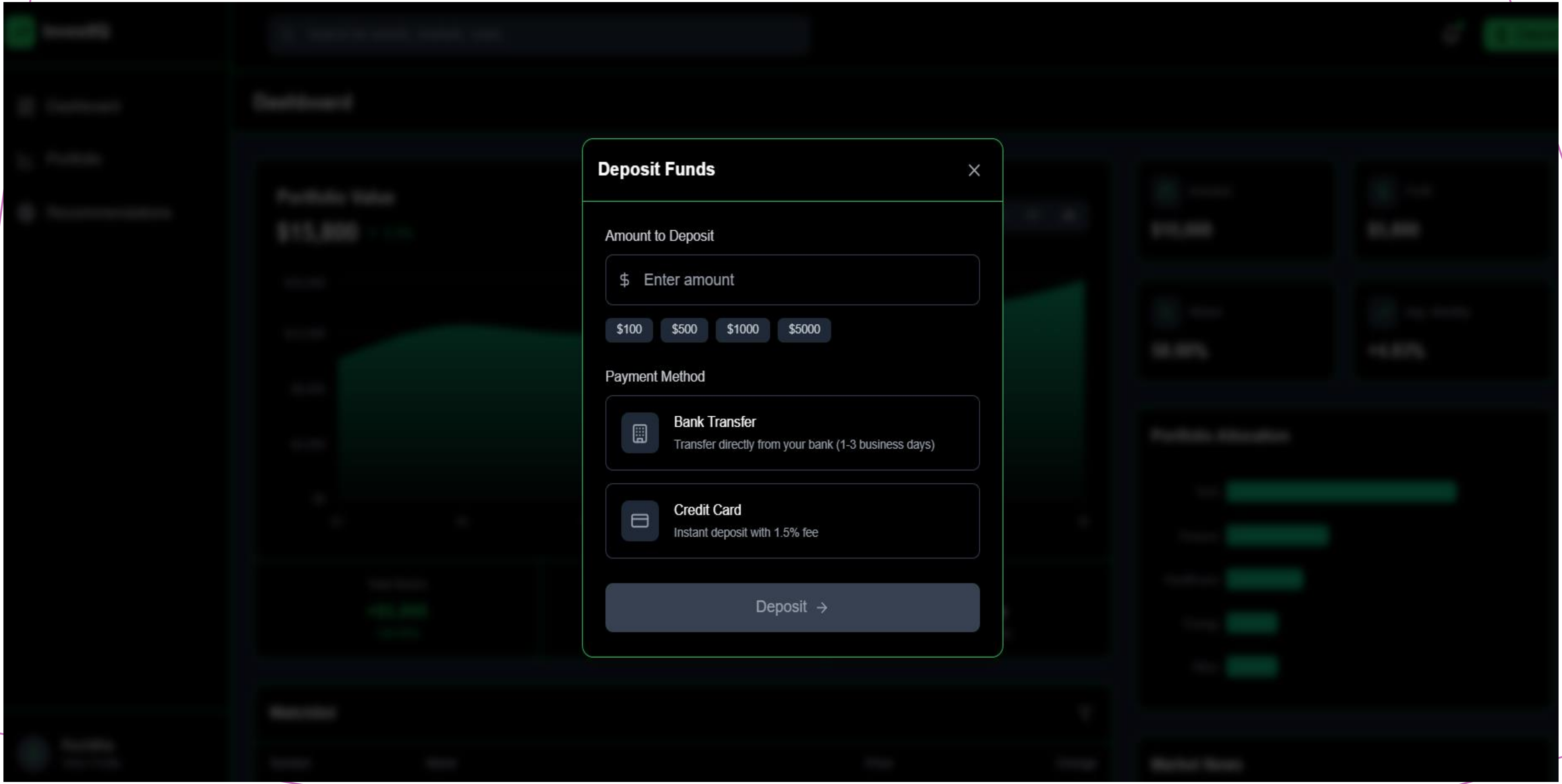
Market News

Treasury Yields Climb as Fed Officials Suggest Rates to Stay Higher
Bloomberg • 2 hours ago

Tech Stocks Rally on Strong Earnings Reports
Market Watch • 4 hours ago

Energy Sector Outlook: Analysts Predict Continued Growth
Financial Times • 5 hours ago

SCREENSHOTS



Deposit Funds

Amount to Deposit

$ Enter amount

$100 $500 $1000 $5000

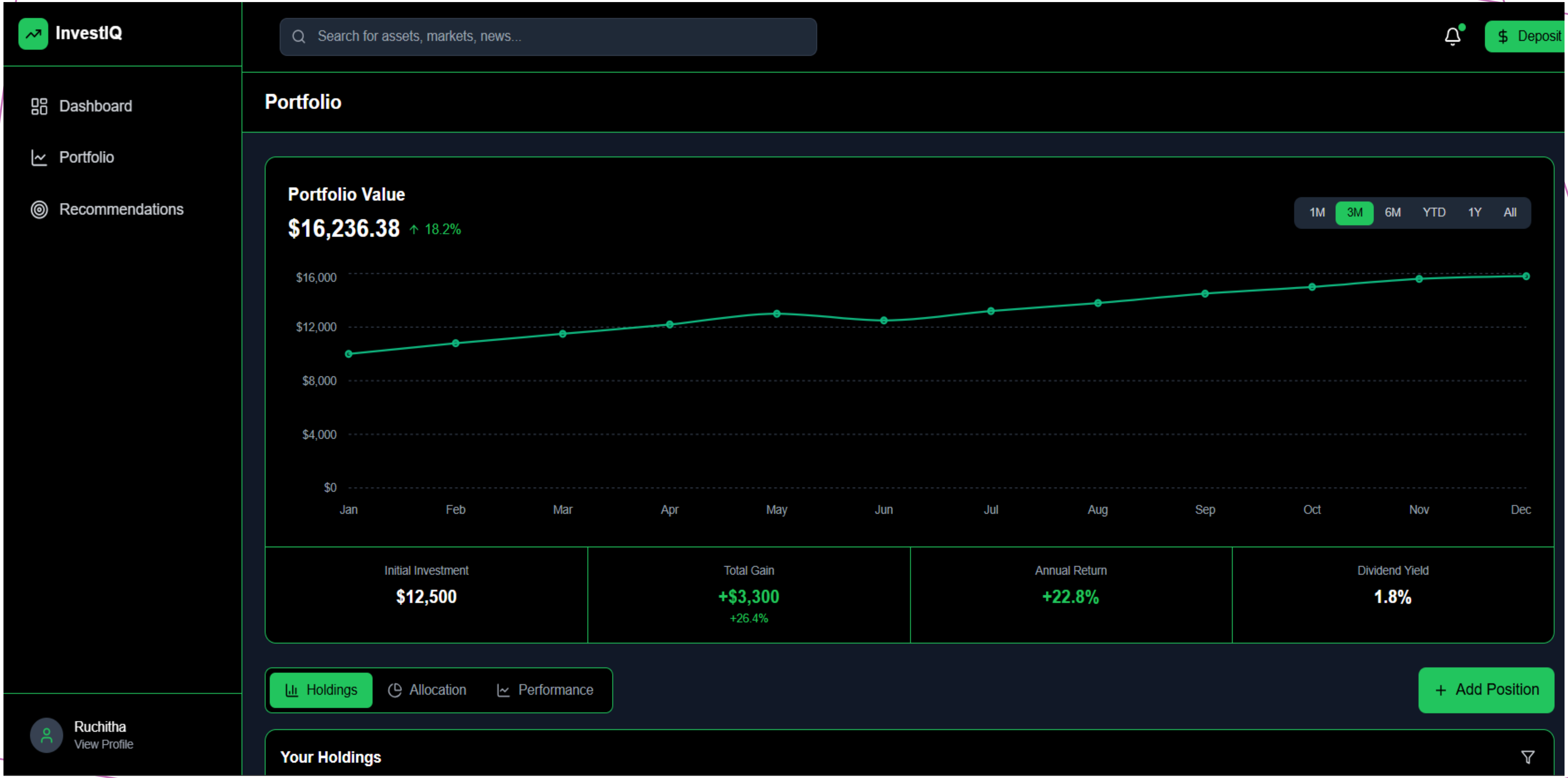
Payment Method

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

Credit Card
Instant deposit with 1.5% fee

Deposit →

SCREENSHOTS



InvestIQ

Search for assets, markets, news...

$ Deposit

Dashboard

Portfolio

Recommendations

Portfolio

Portfolio Value

$16,236.38 ↑ 18.2%

1M [✓ 3M] 6M YTD 1Y All

Initial Investment	Total Gain	Annual Return	Dividend Yield
$12,500	+$3,300 +26.4%	+22.8%	1.8%

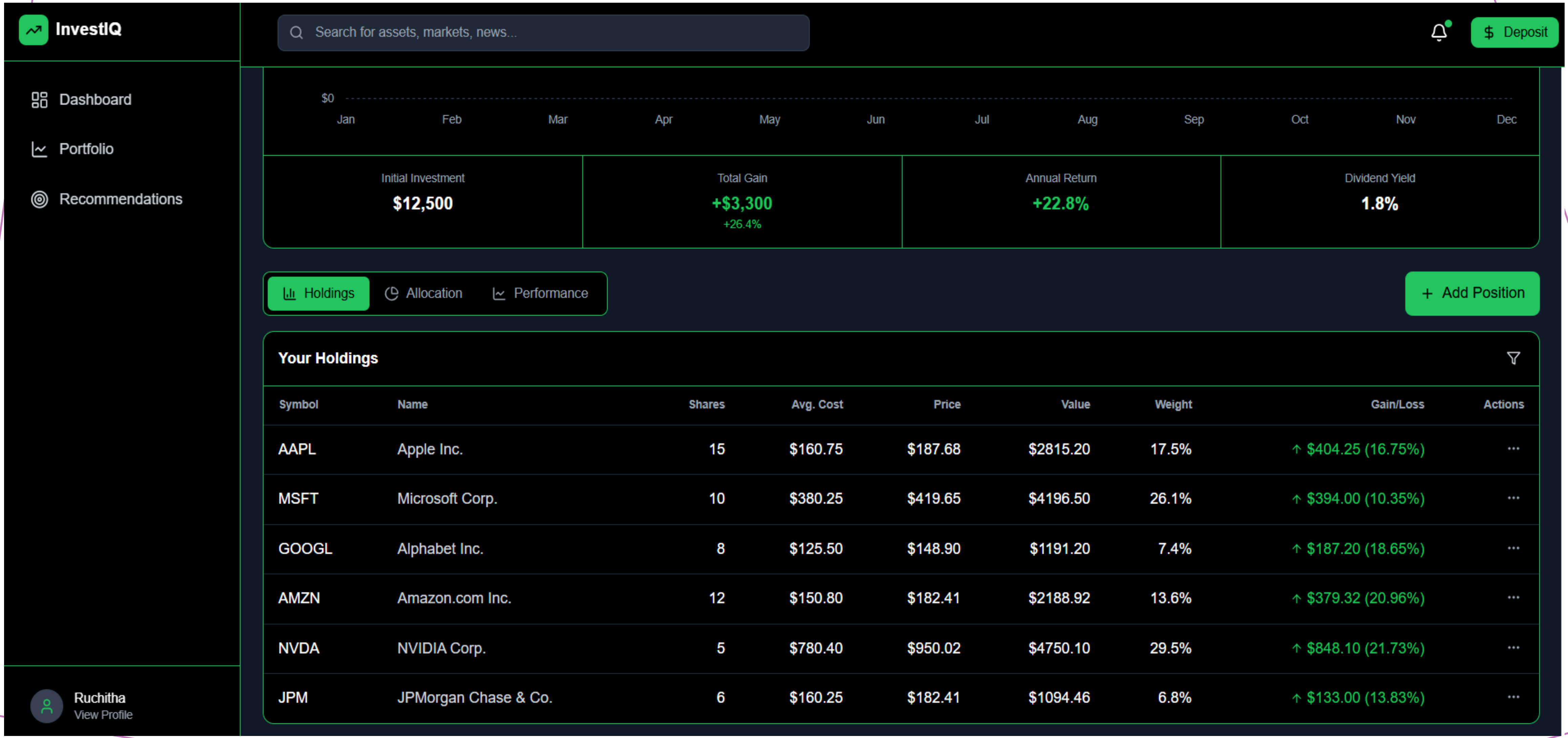
[✓ Holdings] Allocation Performance

+ Add Position

Your Holdings

Ruchitha
View Profile

SCREENSHOTS

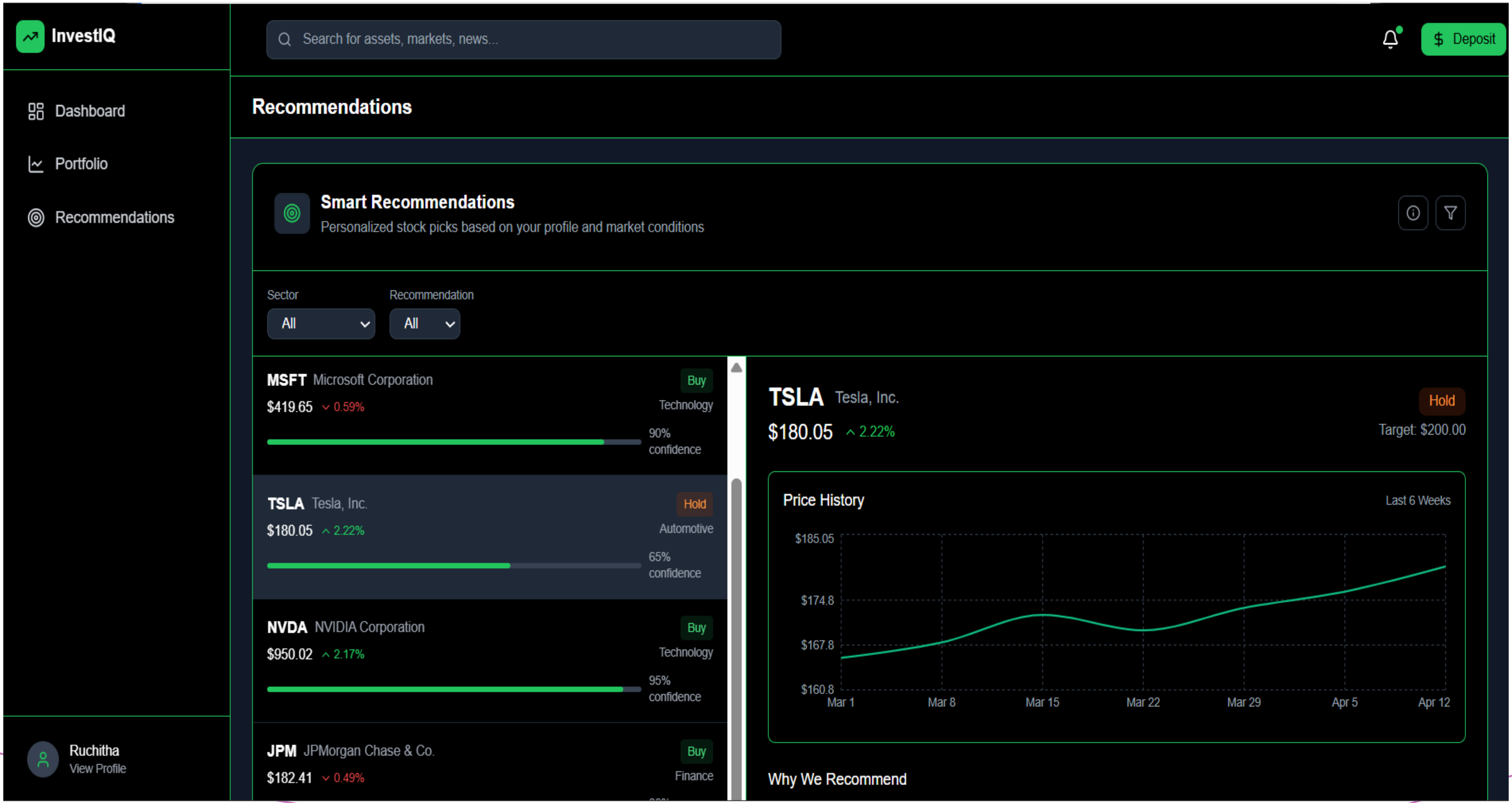


Initial Investment	Total Gain	Annual Return	Dividend Yield
$12,500	+$3,300 +26.4%	+22.8%	1.8%

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



InvestIQ

Search for assets, markets, news...

Deposit

Dashboard

Portfolio

Recommendations

Ruchitha
View Profile

Recommendations

Smart Recommendations
Personalized stock picks based on your profile and market conditions

Sector: All
Recommendation: All

MSFT Microsoft Corporation — Buy
$419.65 ⌄ 0.59% — Technology
90% confidence

TSLA Tesla, Inc. — Hold
$180.05 ^ 2.22% — Automotive
65% confidence

NVDA NVIDIA Corporation — Buy
$950.02 ^ 2.17% — Technology
95% confidence

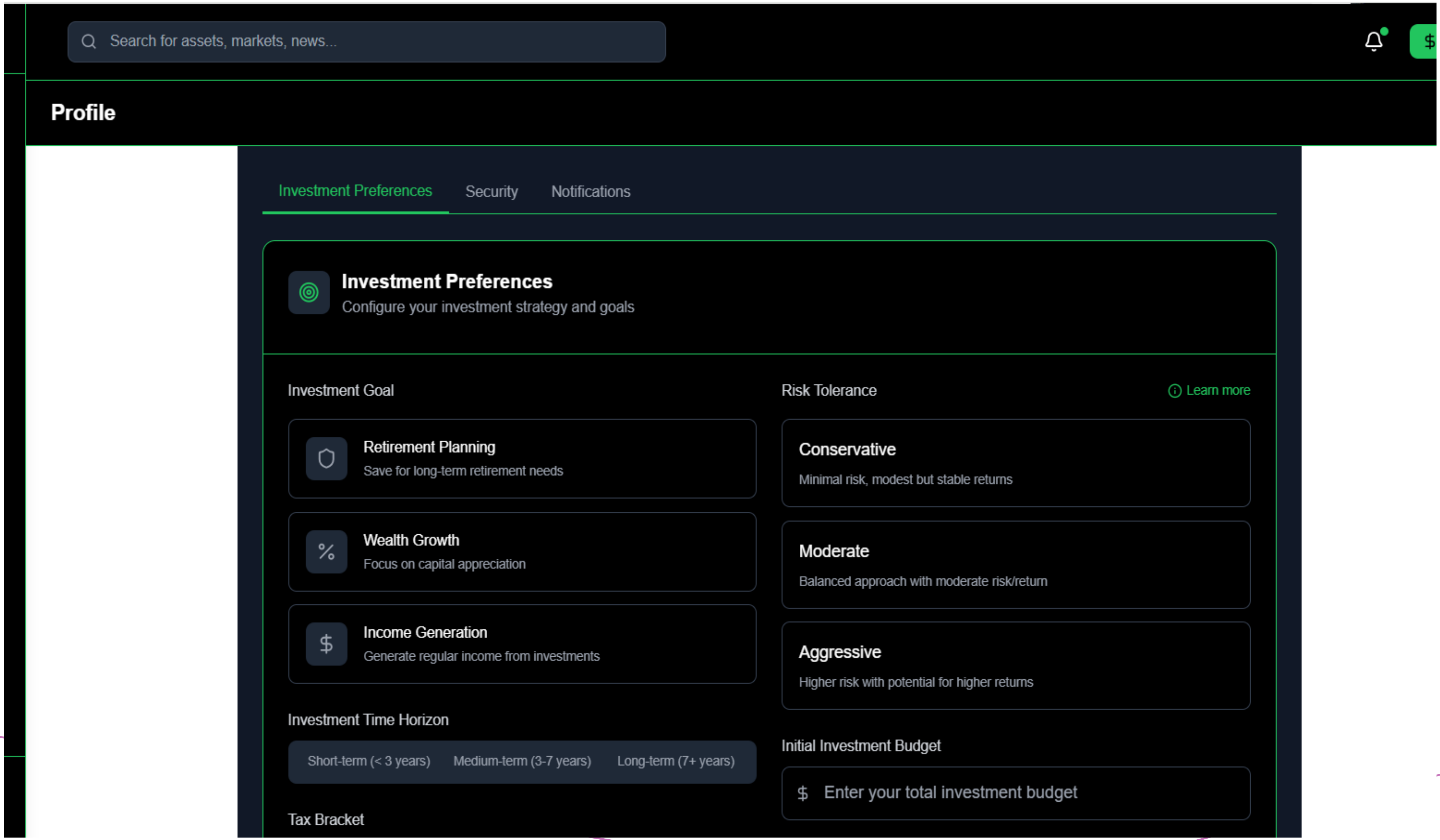
JPM JPMorgan Chase & Co. — Buy
$182.41 ⌄ 0.49% — Finance

TSLA Tesla, Inc. — Hold
$180.05 ^ 2.22%
Target: $200.00

Price History — Last 6 Weeks

Why We Recommend

SCREENSHOTS



Search for assets, markets, news...

Profile

Investment Preferences | Security | Notifications

Investment Preferences

Configure your investment strategy and goals

Investment Goal

Retirement Planning
Save for long-term retirement needs

Wealth Growth
Focus on capital appreciation

Income Generation
Generate regular income from investments

Investment Time Horizon

Short-term (< 3 years) | Medium-term (3-7 years) | Long-term (7+ years)

Tax Bracket

Risk Tolerance

Learn more

Conservative
Minimal risk, modest but stable returns

Moderate
Balanced approach with moderate risk/return

Aggressive
Higher risk with potential for higher returns

Initial Investment Budget

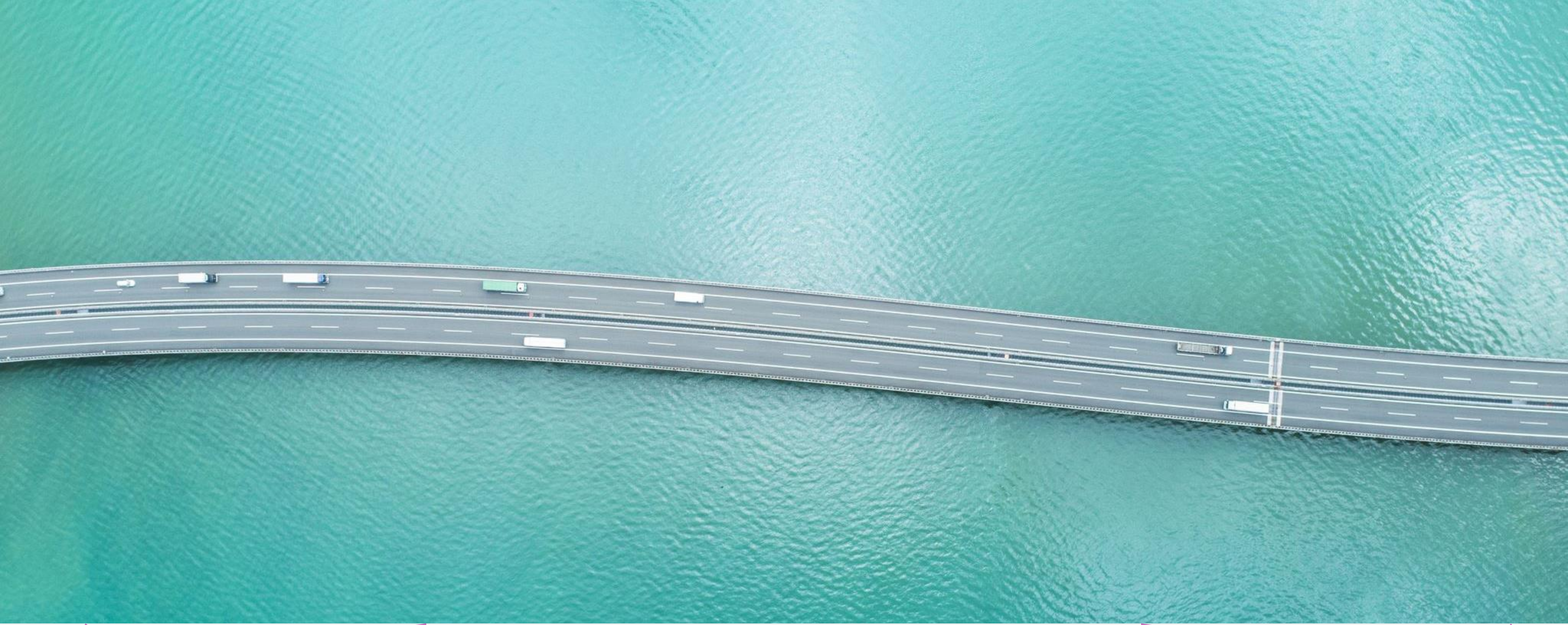
$ Enter your total investment budget

WIKI PAGE LINK

https://github.com/htmw/2025S-Techno-Stack

LIVE DEMO

https://2025-s-techno-stack.vercel.app/



THANK YOU