Design and implementation of stock forecasting system with AI

Group member: Juncheng Zhao, Siyi Chen, Yiwei Li

Date: May.9 2025

Team member

Juncheng Zhao
jz1086@scarletmail.rutgers.edu

Siyi Chen
siyi.chen_tom@rutgers.edu

Yiwei Li
yl2190@scarletmail.rutgers.edu

Outline

Demo

System Architecture

Key Functionality

Technical Details

Contribution

Demo

System Architecture

System Architecture - Distributed Design

Frontend Architecture

Page Components

- Login/Registration Dashboard
- User Information Management Dashboard
- Stock Trading Dashboard
- Investment Analysis Dashboard
- AI Investment Advisory Dashboard

Backend Architecture

Server Configuration

- Express Server Setup
- Route Management
- Middleware Configuration

API Design

- RESTful Interfaces
- Data Validation
- Error Handling

System Architecture - Database Description

User table

Column name	Type	Description
email	VARCHAR(255)	PRIMARY KEY, Store users' email
password	VARCHAR(255) NOT NULL	Store user password
balance	DECIMAL(10, 2) DEFAULT 0.00	Store user balance in profile

System Architecture - Database Description

Stock transaction table

Column name	Type	Description
timestamp	TIMESTAMP DEFAULT CURRENT_TIMESTAMP PRIMARY KEY,	To store the timestamp of every transactions. Avoid primary key conflict
email	VARCHAR(255) NOT NULL,	Store the user's email of the transaction
stock_name	VARCHAR(255)	
number	INT	Store the number of stock user bought in this transaction
current_price	DECIMAL(10, 2) NOT NULL,	
is_sold	BOOLEAN DEFAULT FALSE	It will change to 0 if the user sold all of the stock

System Architecture - Function API interface

Function	API	Http method
User login	/api/login	post
User register	/api/register	post
Change password	/api/change-password	put
Deposit money	/api/deposit	post
Sell specific stock	/api/sell-stock	post
See stock in purchase	/api/active-stocks	get
Give advice	/ge	get
Get stock trend	/api/stock-trend	get

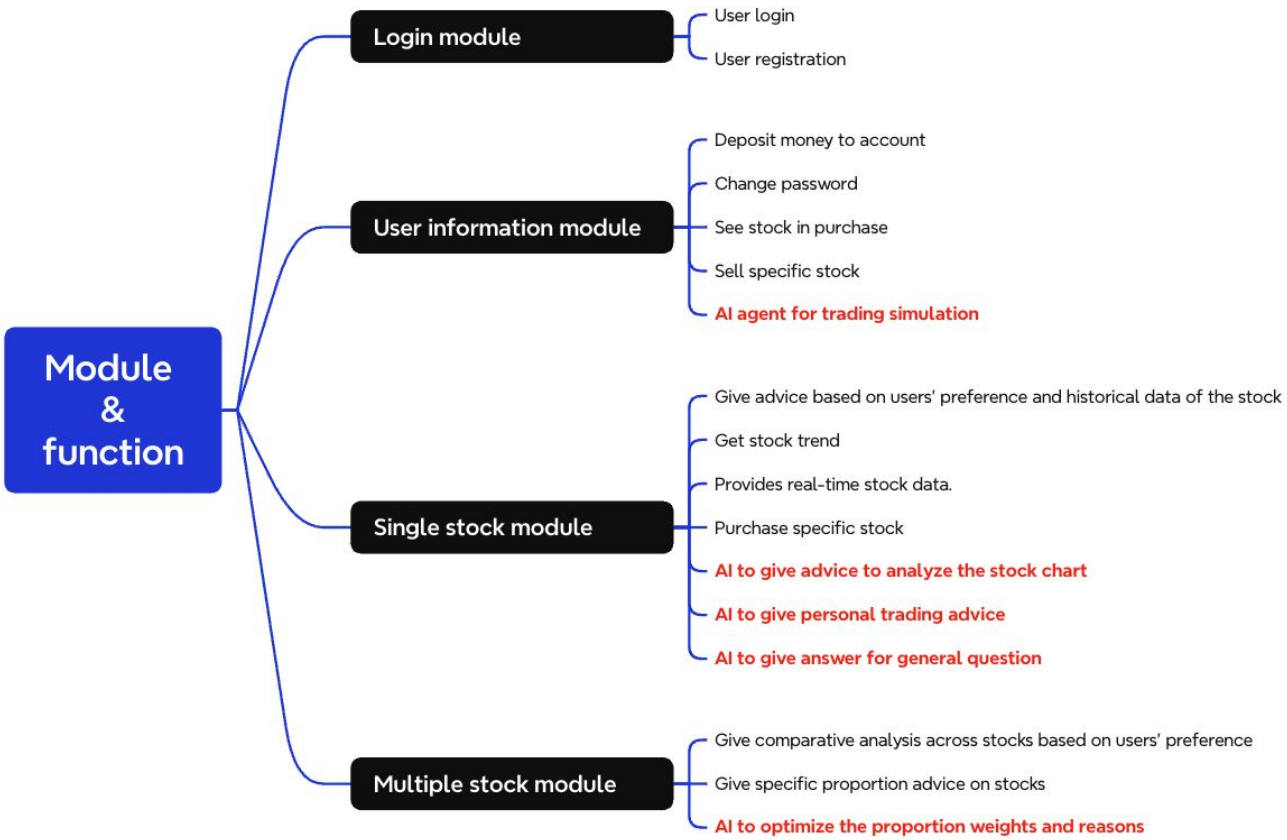
System Architecture - Function API interface

Function	API	Http method
Give comparative analysis across stocks based on users' preference	/api/portfolio-recommendation	get
Give specific proportion advice on stocks	/api/multiplestock-analysis	get
AI provide personal advice	/api/ai-personal-advice	post
AI optimize portfolio weight	/api/ai-predict	post
AI stock Q&A	/api/stock-qa	post
AI chart analyze	/api/analyze-chart	post

Key Functionality

Single stock prediction
Multiple stock prediction
User Information
AI agent for stock forecasting

Key functionality - Module overview



Key Functionality - Login logistics

Register: first time into the stock analysis system

Username: user defines

Password: user defines

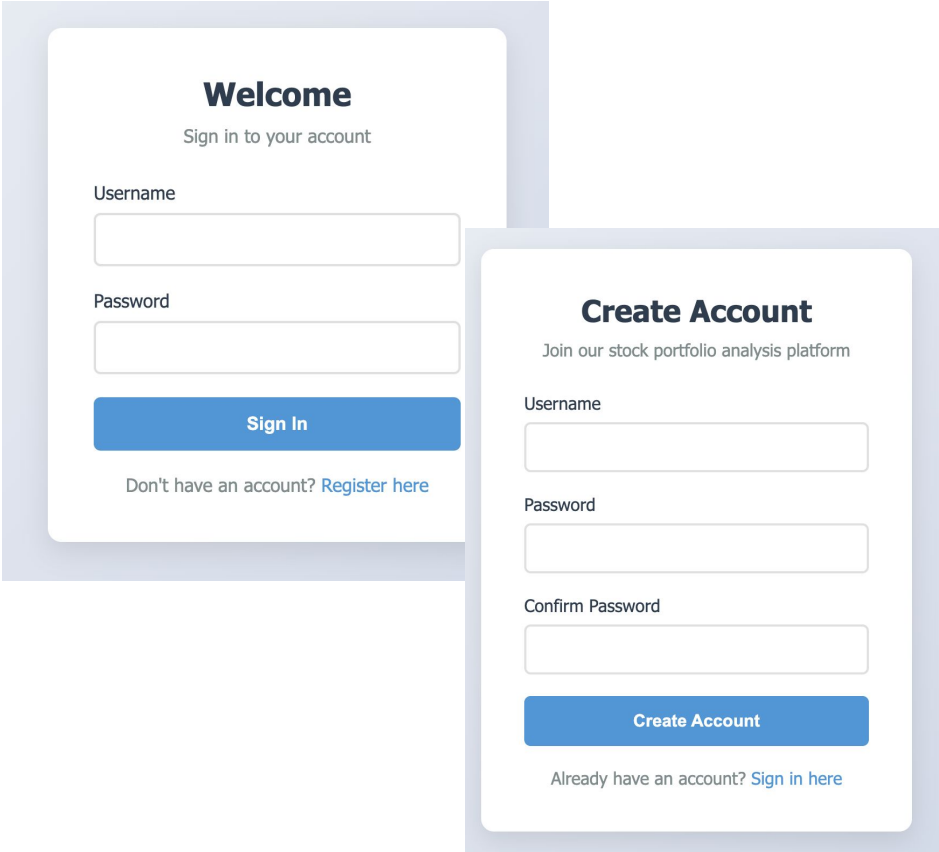
Login:after registration, register information are stored in the database

Username: user input

Password: user input

Error handling:

If user inputs a wrong password or nonexistent username. It will give alert

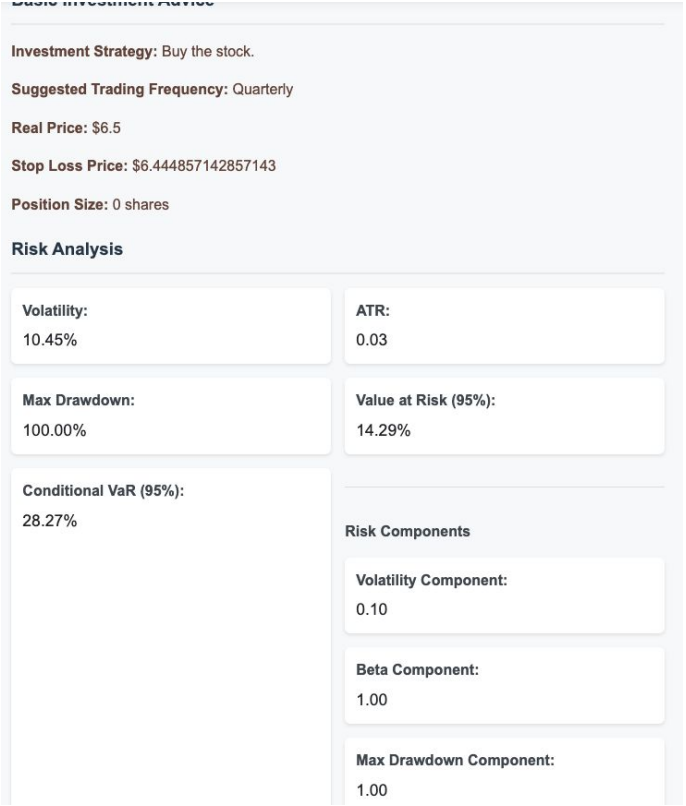


Single Stock Module - Single Stock Analysis

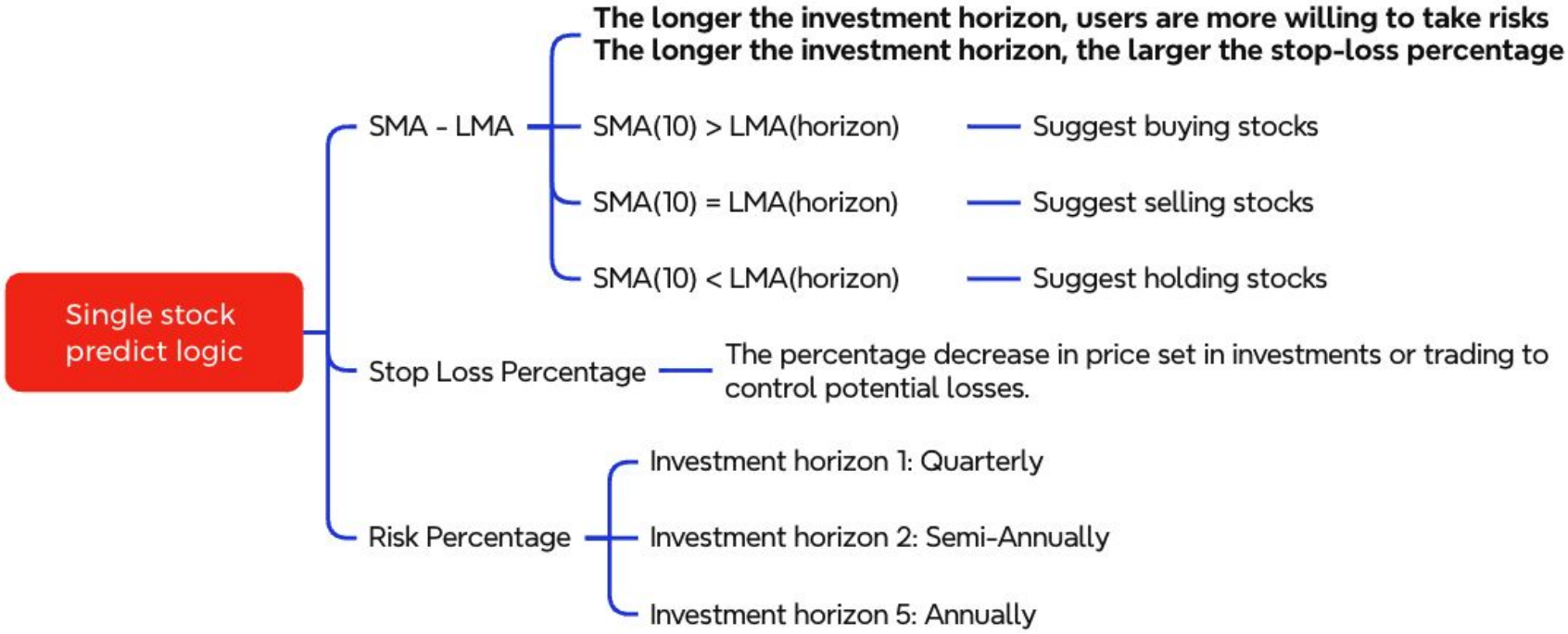
User input:

- **Stock ticker:** User chooses the ticker symbol
- **Initial capital:** User inputs the amount and it can get suggestions about the stock ticker
- **Real price:** User can view the real-time-price of the stock ticker

Trend chart: User can view the historical price of the stock ticker and can click on specific date to check the closing price



Single Stock Module - Single Stock Prediction Logic



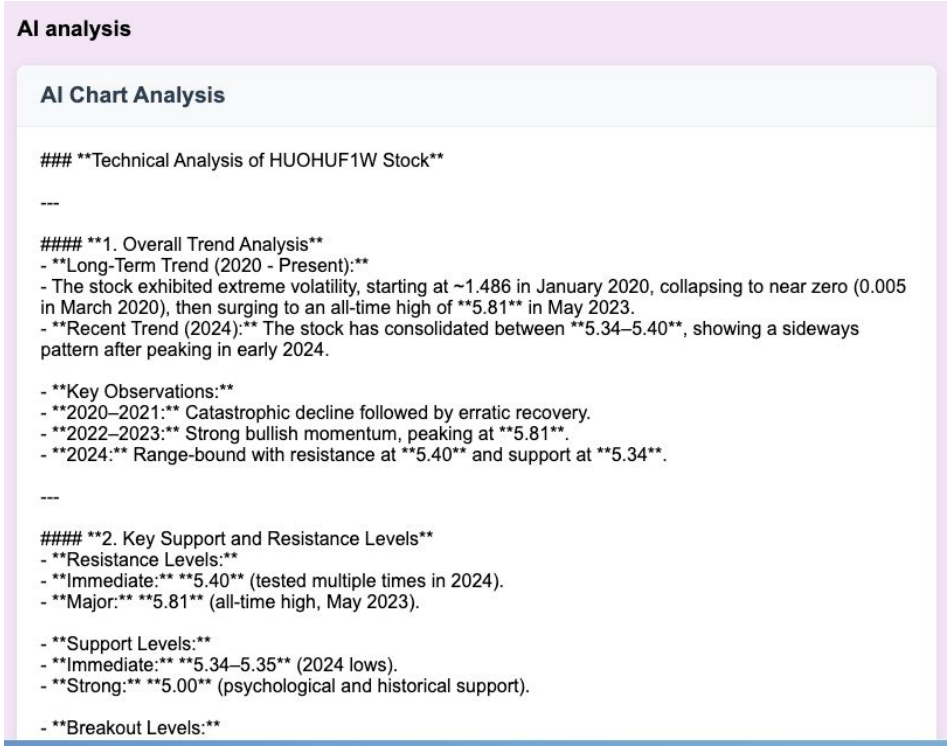
Single Stock Module - Single Stock AI analyze

Analysis Metrics

- Analysis metrics
- Strategy
- Frequency
- currentPrice
- maxDrawdown
- Var
- Cvar
- atr
- riskMetrics
- Volatility

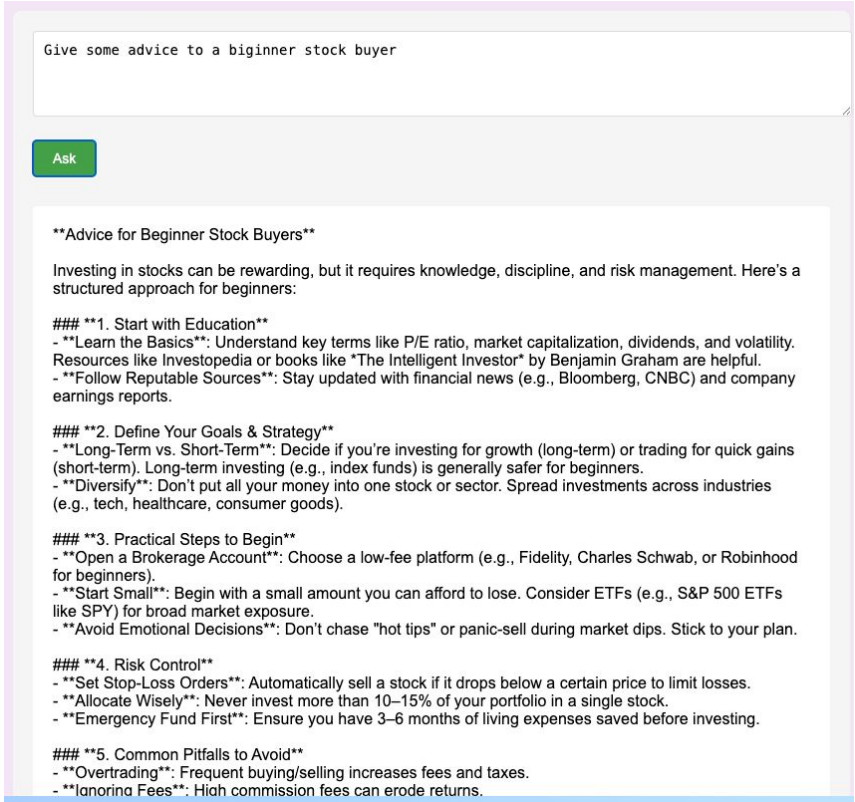
Analysis Content

- Overall Market Analysis
- Risk Assessment
- Technical Analysis
- Investment Strategy
- Final Recommendation



Single Stock Module - AI for general questions

1. **Check API Key**
2. **Initialize OpenAI with DeepSeek API**
3. **Define `/chat` API Endpoint**
4. **Receive User Message**
5. **Send Message to DeepSeek API**
6. **Extract AI Response**
7. **Print AI Response with Typing Effect**
8. **Send AI Response to Client**
9. **Handle Errors Gracefully**



Single Stock Module - Pseudocode for AI questions

1. Data preprocessing

```
Function prepareAnalysisData(stockData, stockTicker, basicAdvice, riskMetrics, riskComponents):
    Create a request body object
    analysisRequest = {
        stockInfo: {
            ticker: stockTicker,
            currentPrice: basicAdvice.currentPrice,
            strategy: basicAdvice.strategy,
            frequency: basicAdvice.frequency,
            quantity: basicAdvice.quantity
        },
        riskMetrics: {
            volatility: riskMetrics.volatility,
            maxDrawdown: riskMetrics.maxDrawdown,
            riskScore: riskMetrics.riskScore
        },
        riskComponents: {
            var: riskComponents.var,
            cvar: riskComponents.cvar,
            atr: riskComponents.atr
        },
        stockData: stockData
    }
    Return analysisRequest
```

2. AI Prompt creation

```
Function buildPrompt():
    prompt = {
        role: "system",
        content: """
            Where question is put
        """
    }
    Return prompt
```

Single Stock Module - Pseudocode for AI questions

1. AI analyze

```
Function analyzeStockData(analysisRequest, prompt):
    OpenAI API configuration
    response = OpenAI.chat.completions.create({
        model: "deeseek-4-turbo-preview",
        messages: [
            prompt,
            {role: "user", content: analysisRequest}
        ],
        temperature: 0.7
    })
    Return response
```

2. Process the results

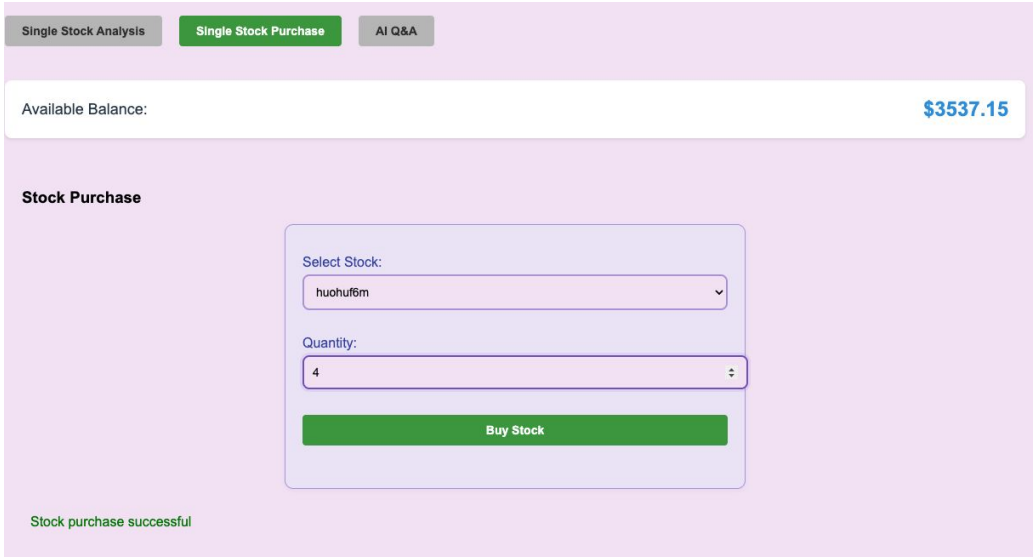
```
Function processAnalysisResult(response):
    If response.success:
        // Extract AI content
        analysis = response.choices[0].message.content

        // Return a success results
        Return {
            success: true,
            analysis: analysis
        }
    Else:
        // Return errors
        Return {
            success: false,
            error: "Analysis failed"
        }
```

Single Stock Module - Single Stock purchase

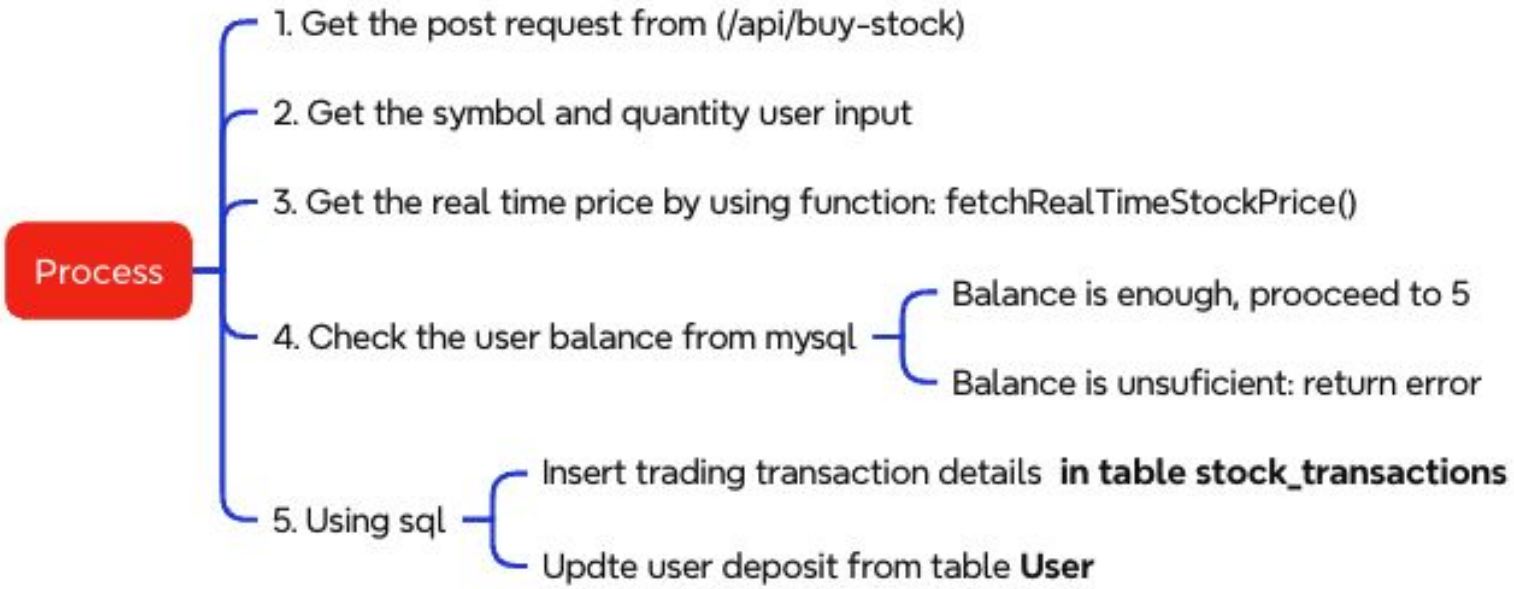
Input

- Stock Symbol: user input
- Quantity: user input

If user purchases successfully, it will give information: stock purchase successful



Single Stock Module - Single Stock purchase workflow

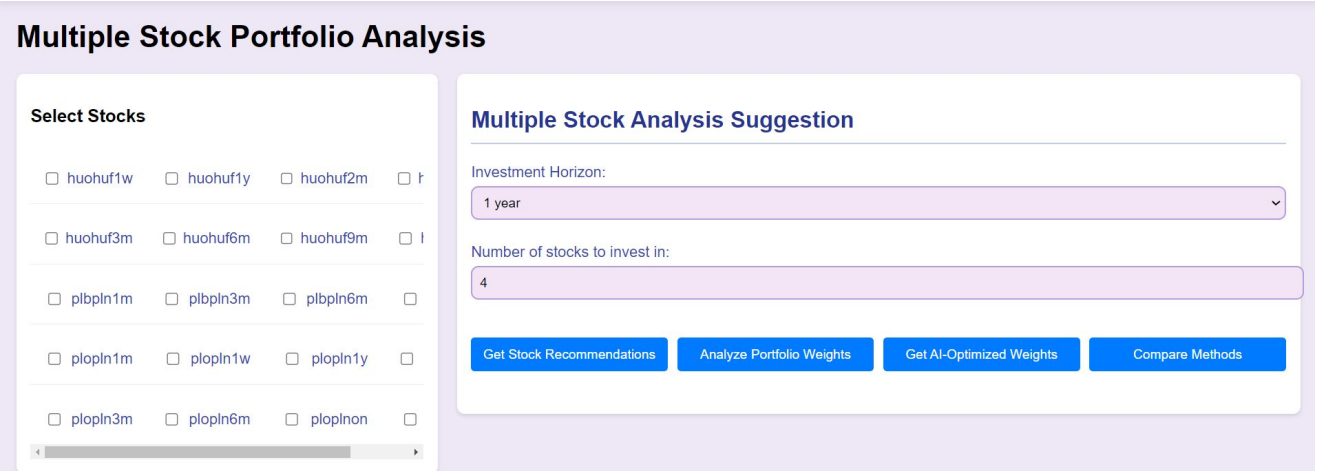


Multiple stock module

Multiple stock module - Multiple stock prediction

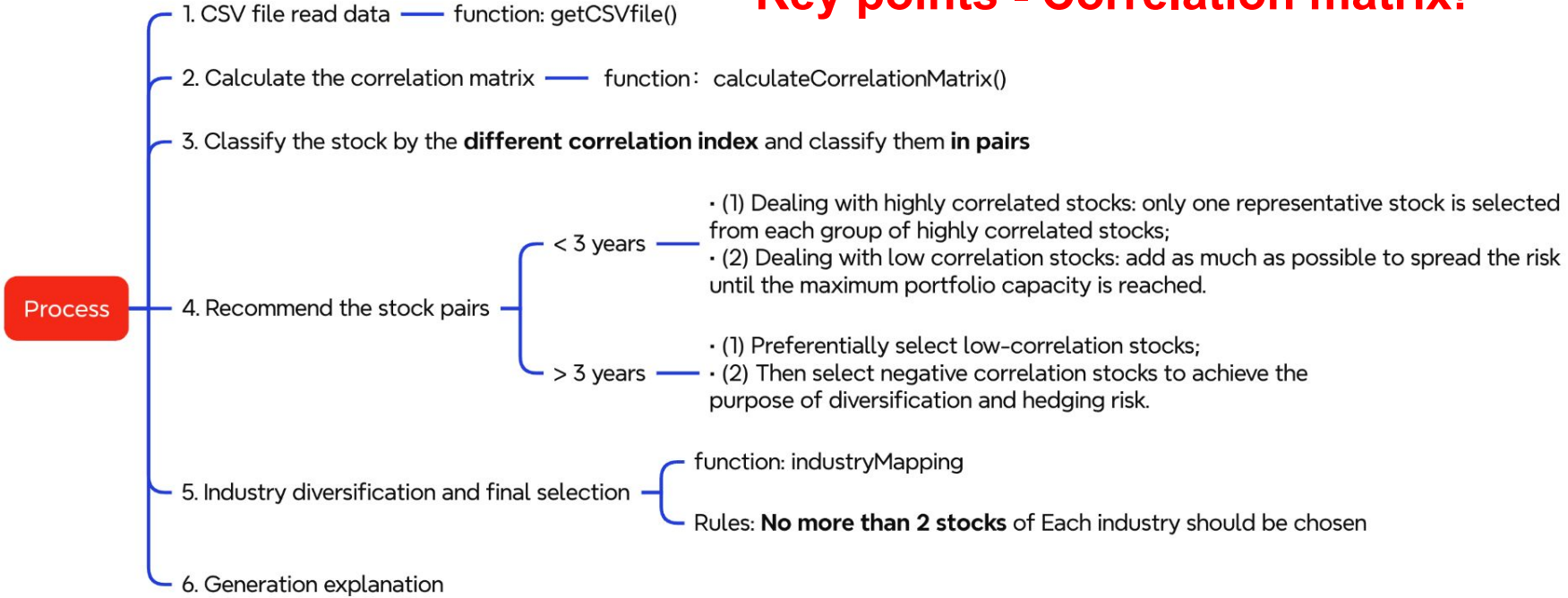
Input:
Investment horizon,
The multiple of stocks you want to input,

Output:
Recommend portfolios,
Portfolio weights,
AI optimized weights(traditional and AI method)



Multiple stock module - Prediction process

Key points - Correlation matrix!



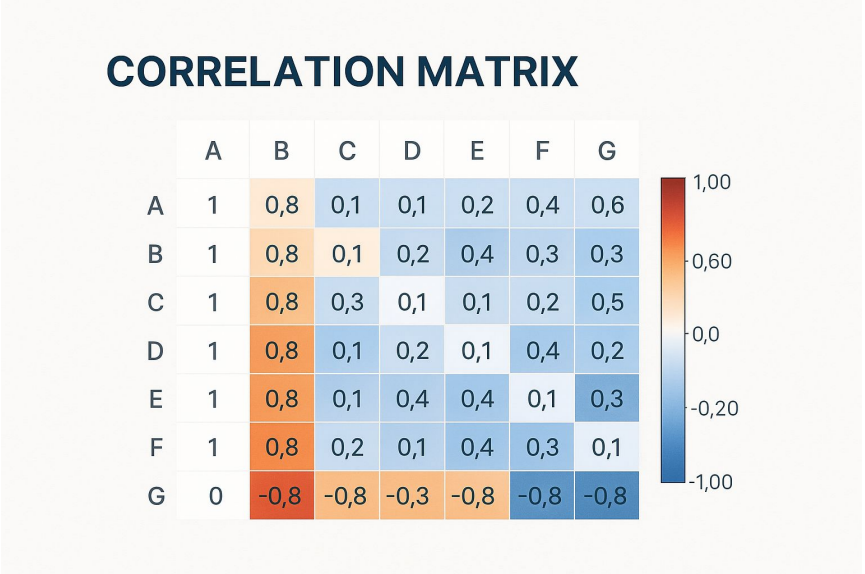
Multiple stock module - Prediction correlation matrix

Correlation matrix is a symmetric matrix used to represent pair-to-pair correlations between a set of variables,

- **1: Perfectly positive correlation.**
- **-1: Completely negative correlation.**
- **0: No correlation.**

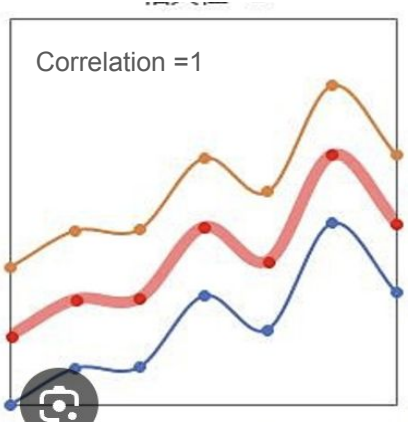
$$\text{Correlation}(A, B) = \frac{\text{Cov}(A, B)}{\sqrt{\text{Var}(A) \times \text{Var}(B)}}$$

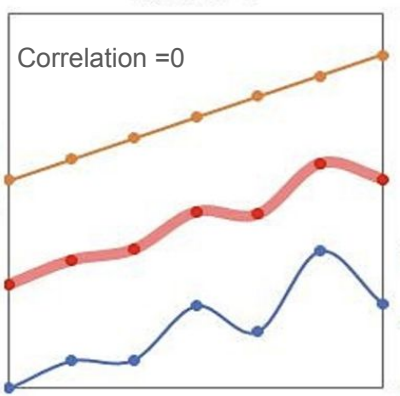
• **Covariance**: Measures the degree to which the prices of two stocks move in tandem.
• **Variance**: A measure of how much the price of a single stock fluctuates.

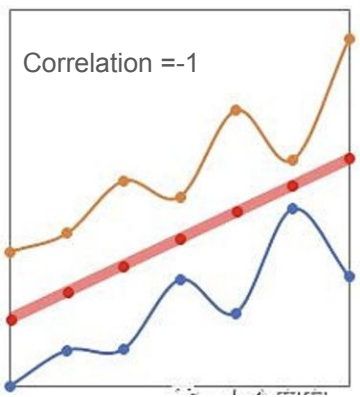


Multiple stock module - Prediction correlation matrix

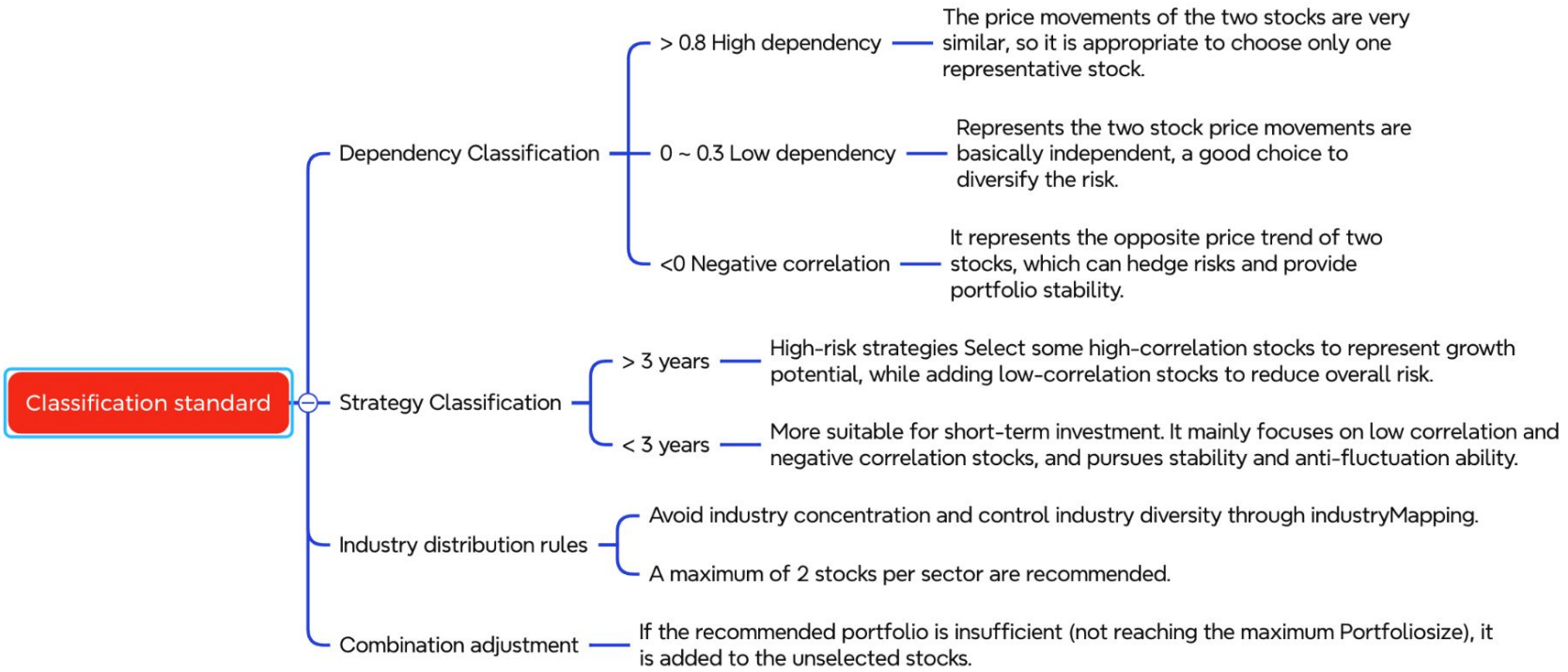
- If two stocks are **highly correlated** their price changes may be similarly affected by the same market event, and risk cannot be spread.
- If two stocks are negatively correlated, price movements can be partially hedged, reducing the volatility of the portfolio.







Multiple stock module -Prediction classification standard



Multiple stock module - Logistics of Module Portfolio weights

1. Calculate Returns for Each Stock

- Compute the **standard deviation** (risk)
- Compute the **Sharpe ratio** (risk-adjusted return)

2. Calculate the Correlation Matrix

- Measure the correlation between all pairs of stocks

3. Initial Weight Calculation

- **Risk-based adjustment**: `RiskAdjustment = risk / totalRisk`
- **Sharpe ratio adjustment**: `SharpeAdjustment = 1 + (sharpeRatio / 2)`
- **Combined adjustment**: `InitialWeight = riskAdjustment * sharpeAdjustment`

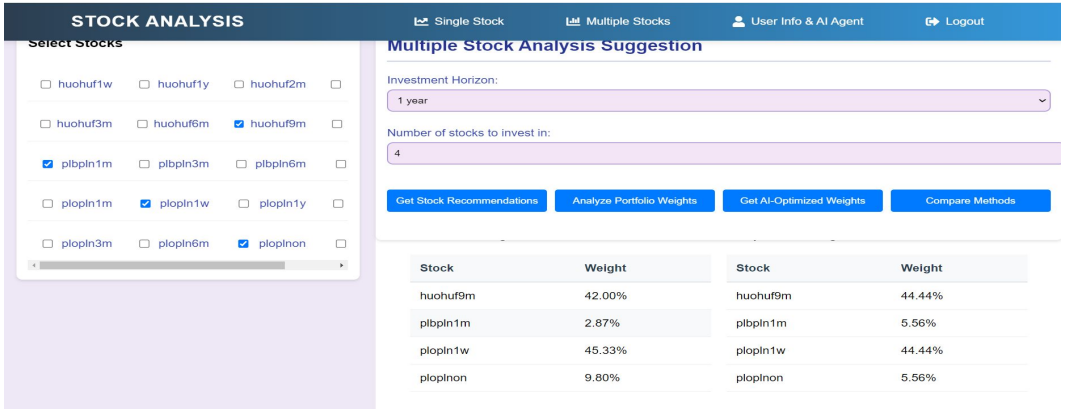
4. Correlation Adjustment

- Adjust weights based on the correlation matrix
- Stocks with high correlations will have their weights reduced

5. Weight Normalization

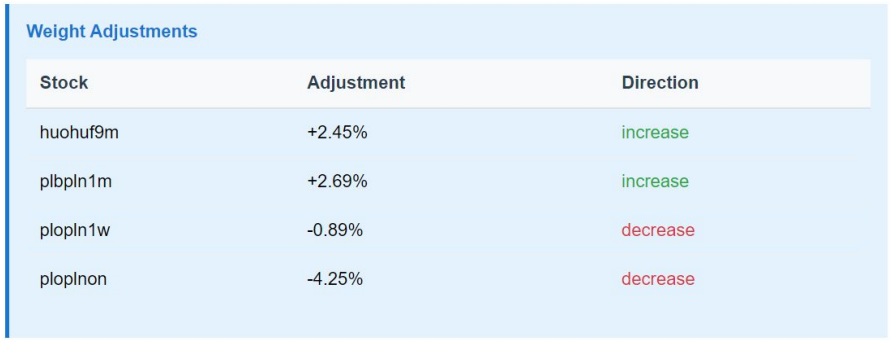
- Ensure the sum of all weights equals 1

Multiple stock module - Calculate weights by AI



Input:
Multiple stocks

Output:
Weights of multiple stocks(Adjustment and AI analysis)



AI Analysis: AI has optimized the portfolio for short-term gains by increasing weights in more volatile stocks that show strong momentum.

AI Recommendations: Consider monitoring market conditions weekly and be prepared to adjust positions based on short-term market movements.

Multiple stock module - Calculate weights by AI

Import modules: Uses Express and OpenAI.

Initialize OpenAI API with `apiKey`.

Define POST endpoint `/ai-predict`:

- Accepts stock features from client.
- Constructs a prompt based on these features.
- Sends the prompt to deepseek for portfolio weight suggestions.
- Parses and normalizes AI's response.
- Returns the weights as a JSON array summing to 1.

Multiple stock module - AI Consideration Advantages

AI Analysis Preparation

- Constructs structured prompts
- Incorporates all stock feature information
- Sets analysis objectives and constraints

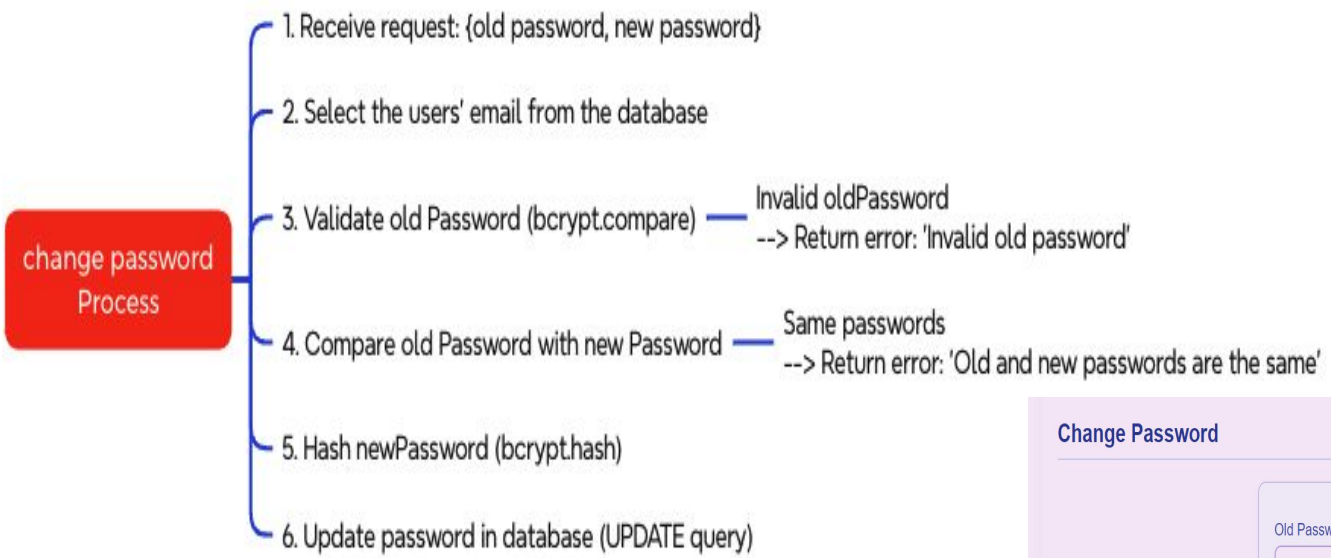
AI Model Processing

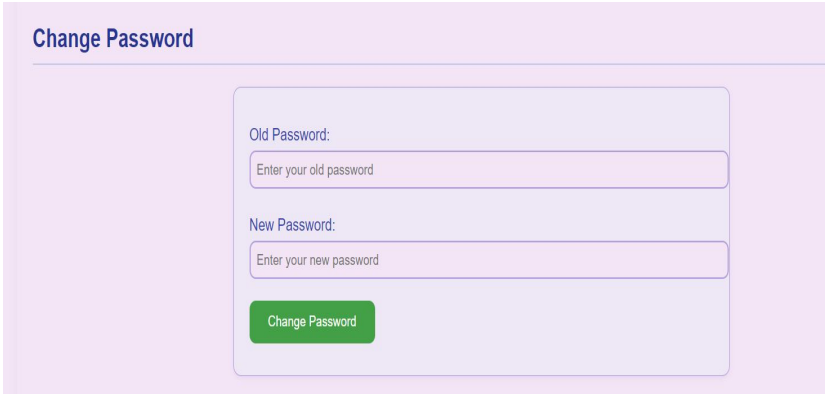
- Utilizes deepseek model for analysis
- Considers risk-adjusted returns
- Evaluates stock correlations
- Analyzes overall portfolio risk
- Takes market conditions into account

Factor	Traditional Method	AI Method
Returns Analysis	✓ Basic return calculation	✓ Advanced return analysis with market context
Risk Assessment	✓ Standard deviation	✓ Multiple risk metrics (VaR, Drawdown)
Market Conditions	✗ Not considered	✓ Current market environment analysis
Correlation Analysis	✓ Basic correlation matrix	✓ Advanced correlation with market trends
Dynamic Adjustment	✗ Static weights	✓ Dynamic weight adjustment

User information module

User information module - Change password

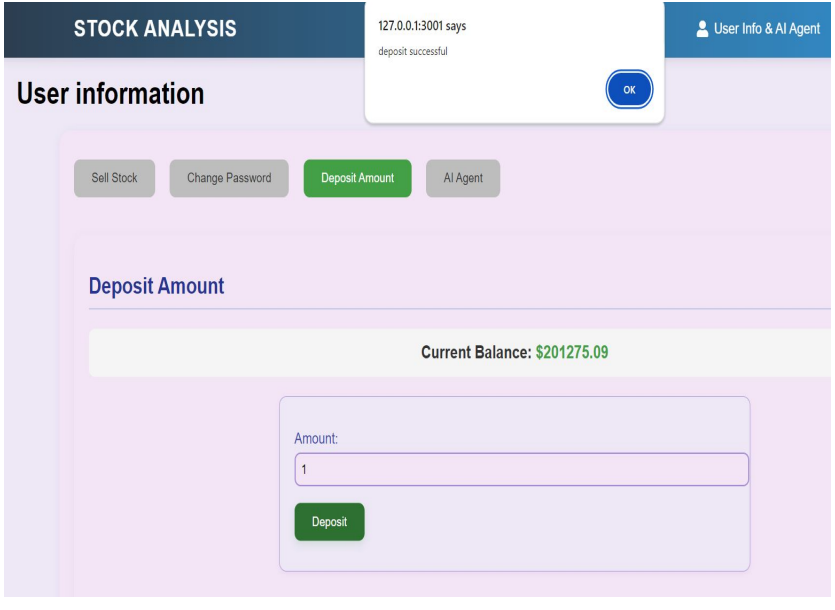




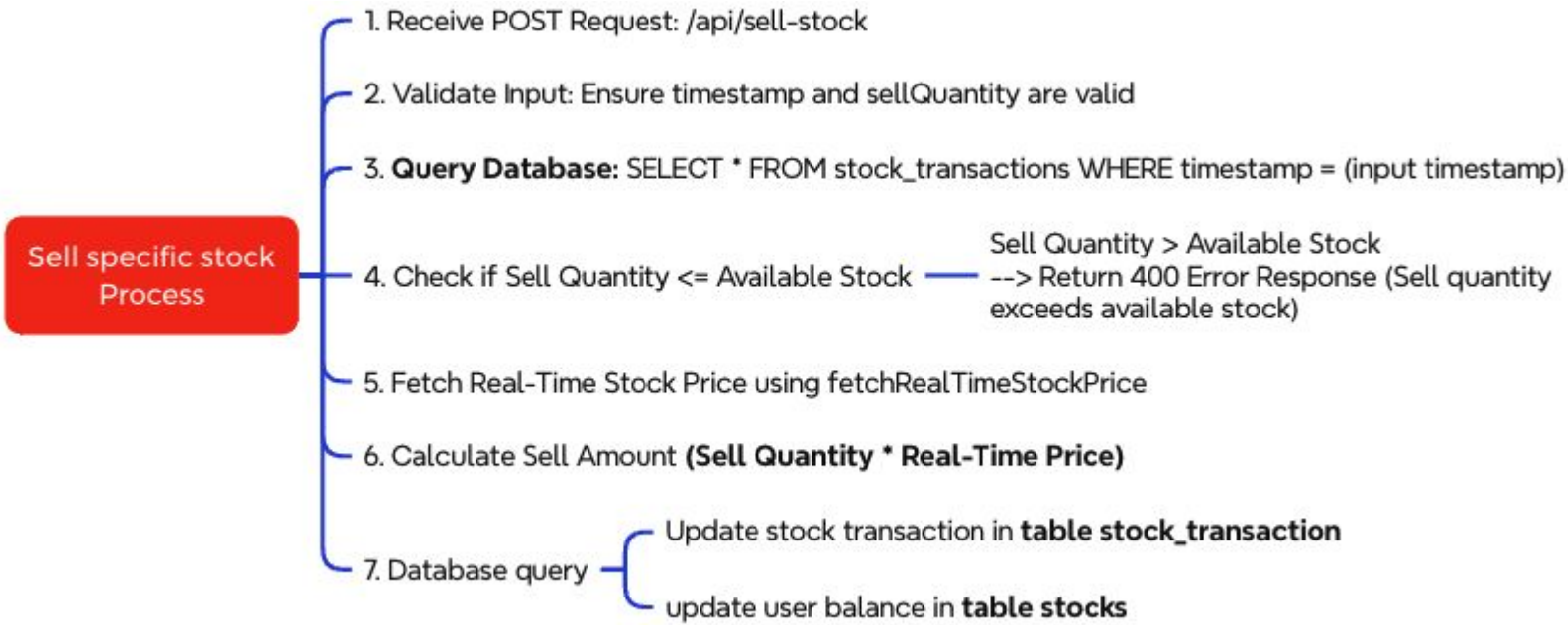
User information module - deposit money

Deposit money process
Process

1. Receive Request: { amount }
2. Update user balance in database (UPDATE query)
3. Release database connection



User information module - Sell specific stock



2025-03-03 02:29:45	huohuf1y	11	Sell
2025-03-03 02:30:40	huohuf1y	12	Sell

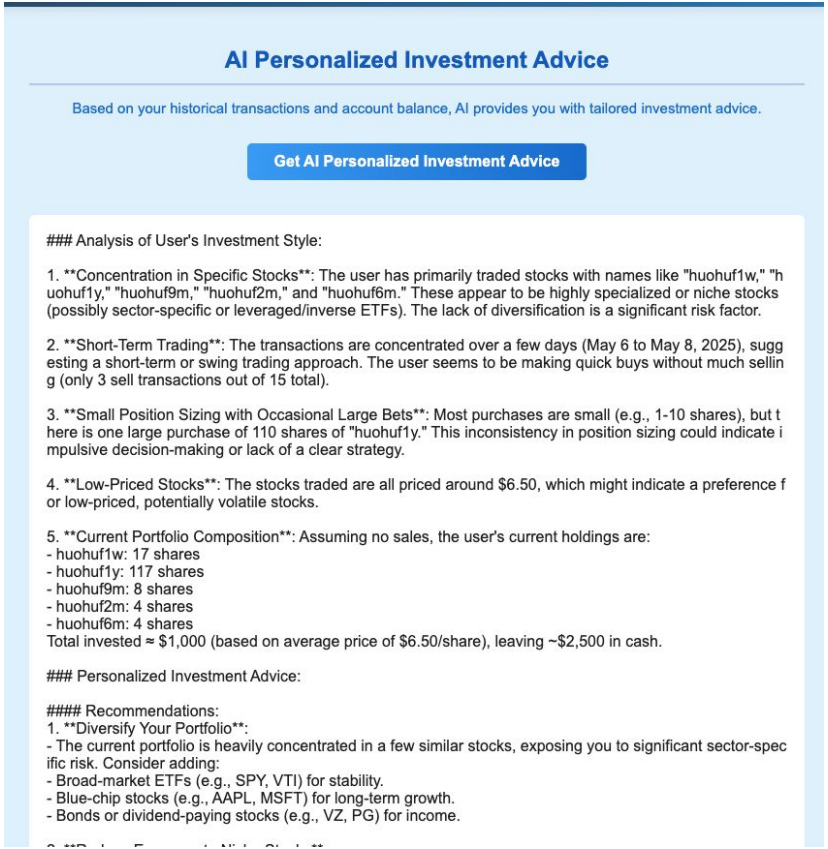
AI personalized investment advice

User input

- Uer email
- User transaction history
- User current balance

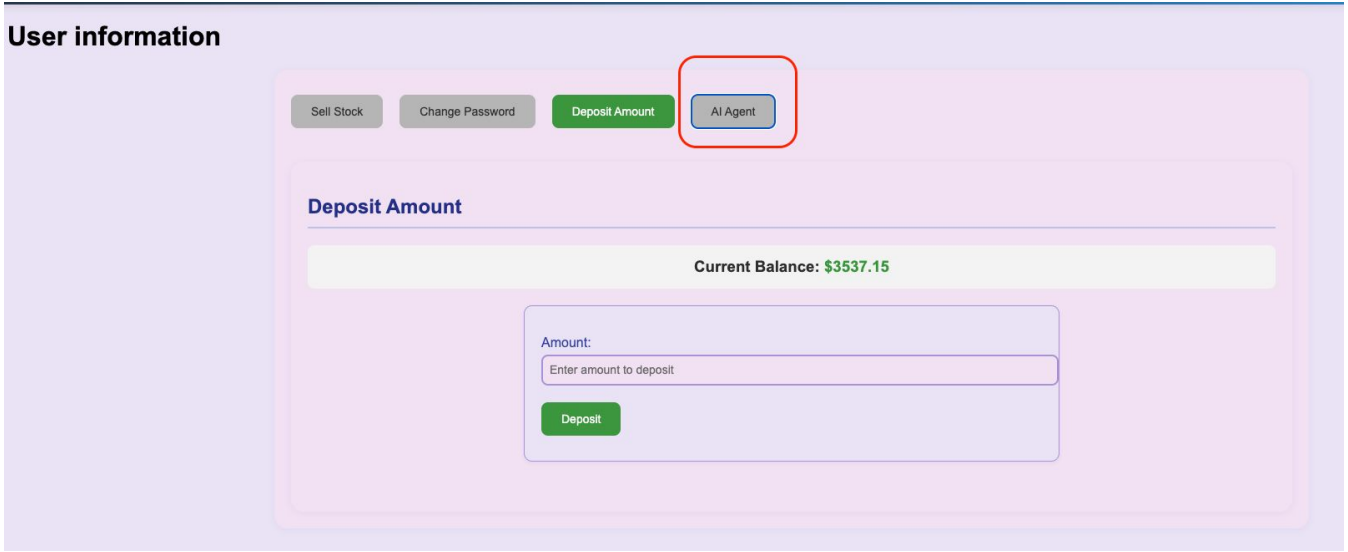
Analysis Content

- Investment style
- Analysis of Trading Patterns
- Risk analyze
- Recommendations for action steps



AI Agent part

User information module - AI Agent



- **AI Agent**: click the button and it will go to the http:localhost:5001

User information module - AI Agent Demo

User information module - AI Agent

API Configuration:

DeepSeek API key

Model Name

Basic Settings:

Number of Agents

Total Simulation Days

Daily Trading Sessions

Stock Initial Settings

Stock A/B initial Price

Agent Initial Property

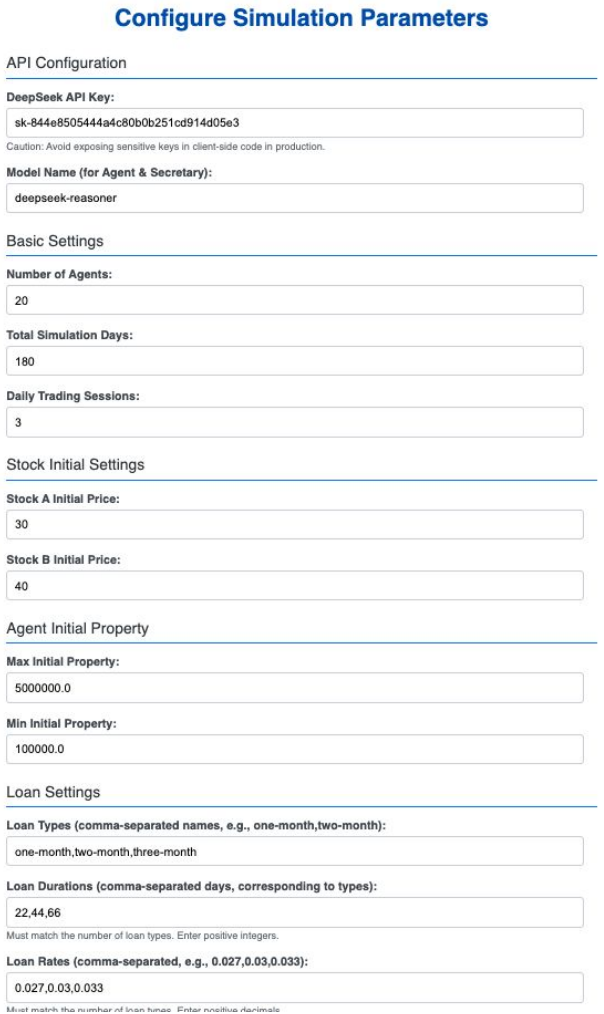
Max/Min initial property

Loan Settings

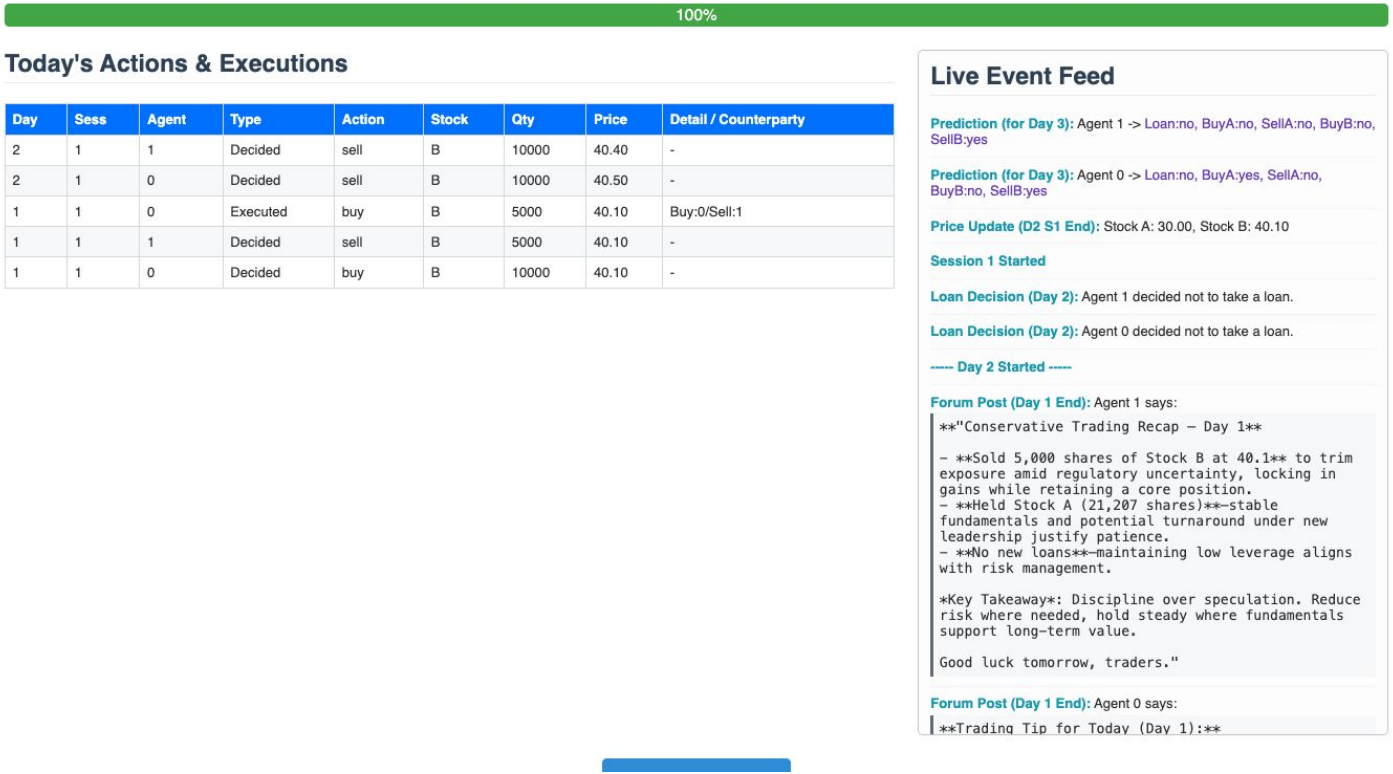
Features:

It will analyze two stocks

It will make a more comprehensive inference based on the financial reports and the market



User information module - AI Agent



- **AI Agent**: click the button and it will go to the http:localhost:5001

User information module - AI Agent

Forum Post (Day 2 End): Agent 1 says:

```
**Conservative Trading Recap – Day 2**

- **Sold 10,000 shares of Stock B at 40.4** to lock
in gains and reduce regulatory risk exposure, while
retaining a core position.
- **Held Stock A (21,207 shares)**–awaiting clearer
signs of turnaround under new leadership.
- **No new loans**–maintaining low leverage aligns
with risk discipline.

*Key Takeaway*: Patience and selective profit-
taking. Trim speculative positions (B) and hold
steady where fundamentals support long-term value
(A).

Good luck tomorrow, traders.
```

Forum Post (Day 2 End): Agent 0 says:

```
**Trading Tips for Today:**

- **Stock A (Chemicals):** Showing signs of
turnaround with new leadership and government
support. Revenue stabilizing, cash flow improving.
Good long-term value at current price. **Bullish.**
- **Stock B (Tech):** Strong growth but concerns
remain about past reporting. Taking partial profits
here after recent run-up. Still positive outlook but
cautious. **Neutral/Slightly Bearish short-term.**
- **Portfolio Adjustment:** Trimmed some B to
increase A exposure. Holding cash for potential
opportunities.

*Trade wisely and manage risk!* 🚀
```

Forum Post (Day 1 End): Agent 1 says:

```
**"Conservative Trading Recap – Day 1**

- **Sold 5,000 shares of Stock B at 40.1** to trim
exposure amid regulatory uncertainty, locking in
gains while retaining a core position.
- **Held Stock A (21,207 shares)**–stable
fundamentals and potential turnaround under new
leadership justify patience.
- **No new loans**–maintaining low leverage aligns
with risk management.

*Key Takeaway*: Discipline over speculation. Reduce
risk where needed, hold steady where fundamentals
support long-term value.

Good luck tomorrow, traders."
```

Forum Post (Day 1 End): Agent 0 says:

```
**Trading Tip for Today (Day 1):**

*"Stock B (Tech) remains a strong buy–despite past
governance concerns, growth projections (+20%
revenue) and government support signal upside. Stock
A (Chem) is a hold; wait for clearer signs of
turnaround under new leadership. No rush to add debt
–cash reserves are sufficient for now. Monitor B's
price action closely tomorrow."*

*(Posted publicly to trader forum.)*
```

Frontend design

HTML: overall layout

Two forms: Login form and registration form, with the login form displayed by default, and the registration form shown when a button is clicked.

Element breakdown:

Input fields: Used for the user to enter their username and password (<input> tag).

Buttons: Used for the user to perform login or registration actions (<button> tag).

Technical Details

- Frontend: HTML, CSS, JavaScript for user interaction.
- Backend: Node.js with Express.js for server-side logic.
- Database: MySQL for storing user and stock data.
- APIs: Facilitate communication between frontend and backend.
- Utility Libraries:
  - bcrypt.js (Hash users’ password)
  - Csv-parser (Read CSV file)
  - Moment.js
  - Math.js (Used to calculate some functions for stock broadcast)

Contribution

Juncheng Zhao

- Frontend Design: Page design
- Backend: Login module design, Single stock module design, Multiple Stock Module(AI Design)
- PPT design

Siyi Chen

- Backend Design: User information module, Multiple stock module, Database creation and operation
- Frontend Design: Page design
- PPT design

Yiwei Li

- AI Agency Design
- Backend Design: Single stock purchase design, User information design
- PPT Design

Thank you