# Stock Market Trading Platform

## Execution Steps and Explanation

## 1. Prerequisites

Before running the application, ensure you have:

- Python 3.7 or higher installed
- Internet connection (for fetching real stock data)
- Git (optional, for cloning the repository)

## 2. Step-by-Step Execution Guide

### Step 1: Clone or Download the Repository

Clone the repository using Git:

```
git clone https://github.com/praveen19218/Stock-market-trading-platform-.git cd Stock-market-trading-platform-
```

Or download the ZIP file from GitHub and extract it.

### Step 2: Navigate to the Project Directory

Change to the Stock_market directory:

```
cd Stock_market
```

### Step 3: Install Required Dependencies

The application requires several Python packages. Install them using pip:

```
pip install flask mysql-connector-python requests yfinance plotly pandas numpy
```

## Explanation of Dependencies:

- Flask: Web framework for building the application
- mysql-connector-python: Database connector for MySQL
- requests: HTTP library for API calls
- yfinance: Yahoo Finance API wrapper for stock data
- plotly: Interactive charting library
- pandas: Data manipulation library
- numpy: Numerical computing library

### Step 4: Run the Application

Execute the main application file:

```
python app.py
```

## Expected Output:

```
* Running on http://127.0.0.1:5000 * Debug mode: on
```

### Step 5: Access the Application

Open your web browser and navigate to:

```
http://localhost:5000
```

## 3. Application Structure and Components

### Main Application File (app.py)

This is the core of the application containing:

- Flask Setup for web framework configuration
- Routes for different application functionalities
- Mock Data Structures for simulated user data
- Integration with Yahoo Finance API for real-time data

## Key Routes:

- /login and /signup for user authentication
- /stock_graph for interactive stock charts
- /portfolio for portfolio management
- /trade for buying/selling stocks
- /get_chart_data for real-time chart data
- /get_trading_signal for AI-powered trading signals
- /get_news for live market news

## Mock Data Structures:

- MOCK_USERS: Simulated user accounts
- MOCK_PORTFOLIO: Sample portfolio data
- MOCK_TRADING_HISTORY: Example trading history

## 4. Key Features Explained

### 1. Real-Time Stock Charts (/stock_graph)

- Displays interactive charts using Plotly.js
- Shows technical indicators (SMA, EMA, RSI, MACD)
- Supports multiple time frames (1D, 1W, 1M, etc.)
- Fetches real data from Yahoo Finance via yfinance

### 2. AI-Powered Trading Signals (/get_trading_signal)

- Analyzes stock trends using moving averages
- Generates BUY/SELL/HOLD recommendations
- Provides confidence metrics
- Uses technical analysis for decision making

### 3. Live News Updates (/get_news)

- Automatically refreshes every 5 minutes
- Generates news based on real market movements
- Shows top gainers/losers
- Provides market sentiment analysis

### 4. Portfolio Management (/portfolio)

- Tracks holdings and performance
- Calculates profit/loss in real-time
- Shows sector allocation
- Displays transaction history

## 5. Database Configuration (Optional)

The application uses mock data by default but can connect to a MySQL database:

- Install MySQL Server (if not already installed)
- Create Database: CREATE DATABASE stock_trading;
- Update Connection Details in app.py

### Connection Code Example:

```
conn = mysql.connector.connect( host='localhost', user='your_username',
password='your_password', database='stock_trading' )
```

## 6. Troubleshooting Common Issues

- Port Already in Use: python app.py --port 5001
- Missing Dependencies: pip install -r requirements.txt
- yfinance API Issues: Check internet connection or update yfinance
- Permission Errors: Run as administrator or adjust firewall settings

## 7. Testing the Application

Login with mock credentials:

- Username: admin
- Password: admin123

Explore Features:

- View stock charts for AAPL, MSFT, GOOGL
- Check AI trading signals
- Browse portfolio dashboard
- View trading history
- See live news updates

## 8. Stopping the Application

To stop the application, press Ctrl+C in the terminal where it's running.

## 9. Customization Options

- Adding New Stocks: Modify stock symbols in various routes
- Adjusting Time Frames: Edit period mappings in /get_chart_data route
- Modifying Technical Indicators: Update calculation logic in chart data functions

## Conclusion

This comprehensive guide should help you successfully execute and understand the Stock Market Trading Platform code. The application demonstrates real-world implementation of financial technology with modern web development practices.

Generated on December 05, 2025 at 00:45