

Software Architecture & Design Group Report

Members:

- Adam Toms-Sheridan
- Unais Qureshi
- Maciej Madejsza
- Rajababu Kushwaha

Introduction:

This project involves developing a flexible and scalable web application, where users can compare prices of different shares over time. This application allows users to compare share prices between different companies, to chart those prices on a graph, and view and compare share prices over different days between today and 2 years ago. This project uses Java for server-side operations and Pug as the templating engine for the front-end of the application.

To facilitate team coordination and task prioritisation, we will manage the project using Scrum techniques for a collaborative and agile development process. This approach will be supported by software development tools like Visual Studio and Git for version control, ensuring the application is scalable and deployable in diverse environments.

The application will serve as a valuable tool for our users by providing valuable tools and features used to purchase and sell stocks.

Aims:

Our team brainstormed ideas on which users we wanted to target for this program, which helped us focus our aims into logical steps. We created and developed use case diagrams and personas to see how this program would benefit from the user's point of view. These aims were developed by understanding the needs and preferences highlighted in our personas. Here are the refined aims of our project through two of our personas:

SAD Group Report



Name: Maxwell Anderson

Demographics

Gender: Male
Age: 28
Location: London
Relationship Status: Single
Title: Photographer
Education: BA Photography

Goals

- Maxwell wants to be able to access an easily understandable stock system to help get into stocks and shares trading.
- Maxwell suffers from colour blindness and the colours used on sites that he has found make them inaccessible to him.

Background Description

Maxwell is a photographer that got his degree at Westminster university. He has lived in London his whole life and has no aspirations to leave London, however due to the recent economical climate, the cost of living has become too much for him to afford as a photographer alone. Some of his friends have recently got into investing and have recommended it to him to help relieve some of the financial stress he is currently under.

Motivations

- Has a financial incentive
- Friends want to help him learn.

Frustrations

- Applications he has used haven't provided colour options for colour blindness.
- Wants to use the sites recommended by his friends but doesn't like the interface provided by the site.

Previous Experience:

- Hasn't got any actual experience in trading.
- Looking at other trading sites he hasn't liked a lot of their UIs, however there were some he liked without the accessibility options he needs.

Expectations:

- He expects accessibility to his colour blindness.
- He expects the ability to customize which site he uses when comparing stock prices.
- He expects a simple to navigate GUI.



Name: Tommy Dewey

Demographics

Gender: Male
Age: 32
Location: United Kingdom
Relationship Status: Married
Title: Banker
Education: BSc Economics, MSc Economics - King's College London

Goals

- Enter the trading scene.
- Be able to obtain daily price information on a given share.
- Find a software for checking share prices that he can rely on.

Background Description

Tommy is a banker working at Lloyds Bank. Tommy has always solely focused on his education; therefore, he hasn't explored other avenues of growing his wealth. However, his peers recently informed him that investing in stocks could prove useful to him as passive income, especially as he aims to hopefully start his own banking firm in the future. Tommy wants a simple program, with a simple interface, that he can use to be more familiar with the stock market and more confident in investing.

Motivations

- Wants to make as much money as possible.
- Wants to learn from his peers.

Frustrations

- Not being able to understand stocks with the information he is provided with.
- Complicated interfaces
- Lack of guidance or tutorials

Previous History/Experience:

- Beginner in trading
- Understands finances from his prior education in economics.

Expectations:

- Be able to understand stocks based on graphs and data presented.
- Be able to compare stocks easily.
- Be able to retrieve accurate data on stock prices based on two dates.

Objectives Overview:

To ensure the structured and effective development of our web-based stock trading application, our team defined clear objectives described in the sprint briefs on achieving our project aims. These objectives were systematically executed and tracked for progress.

Sprint 1:

- Identify the key requirement for this system and scope of the project.
- Task allocation to each member of the team.
- Identify architectural concepts encompassing Simple Architecture principles and outlining a component specification diagram for this system.
- Provide abstract implementation of simple architectural elements for this system in Java.

Sprint 2:

- Create a Business Concept Model.
- Create a Use Case Model.
- Create a System Interfaces diagram.
- Create a Business Type Model.
- Create the Initial System Architecture.
- Create the Business Interfaces.
- Consider the principles of Clean Architecture and provide the implementation of components and interfaces in the program.

Sprint 3:

- Divide the architecture into compound components and apply basic styles for creating reusable and scalable UI components.
- Identify and implement the most appropriate domain independent style.
- Introduce principles of SOA, emphasising service modularity and interoperability.
- Define and carry out suitable test cases and document these cases and results.

Reflection:

The design of our system followed the n-tiered style within the compound components. We utilised the n-tiered style to follow the business model as there were multiple layers that fell into different areas of the business model. This would allow us to separate into clear boundaries what components would fall into what part of the business concept.

The compound components used were Stocks, Database, Graph, and User. There are outlying subcomponents that did not belong in the compound components, Notification System and Stock Site. The StockInformation subcomponent in the Stocks component gets data from the StockSite, in our case yahoo finance, and then temporarily stores the data. This information is then transferred into the Database component using the IGetStocks interface. The information is then permanently stored within the Database using the Update database class. Once the database has been updated the information is then retrieved using the retrieve database class. There is a split in the interface to notification system and the Graph component. When the data is sent to the notification system, it is

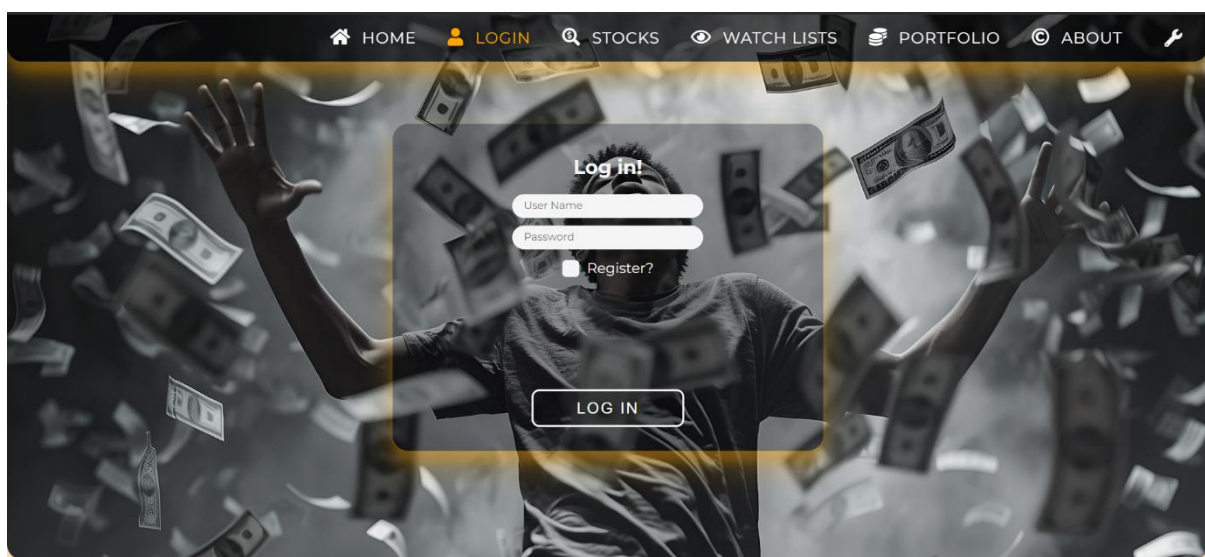
SAD Group Report

checked to see if the information has reached a certain price to then notify the user if they should buy the stock based on a set level. The Graph component gets the data from the database and filters it by using the Graph Filters compound. This filtered information is sent to the Plot Graph component, this then plots the graph for the users and sends the data to the GUI. The GUI then shows the user the frontend code and is the interactable part of the code. The GUI then connects the Login, User Preferences, Portfolio, Watchlist, and Stock Options components. The user is then able to visualise the data and access what stocks they desire.

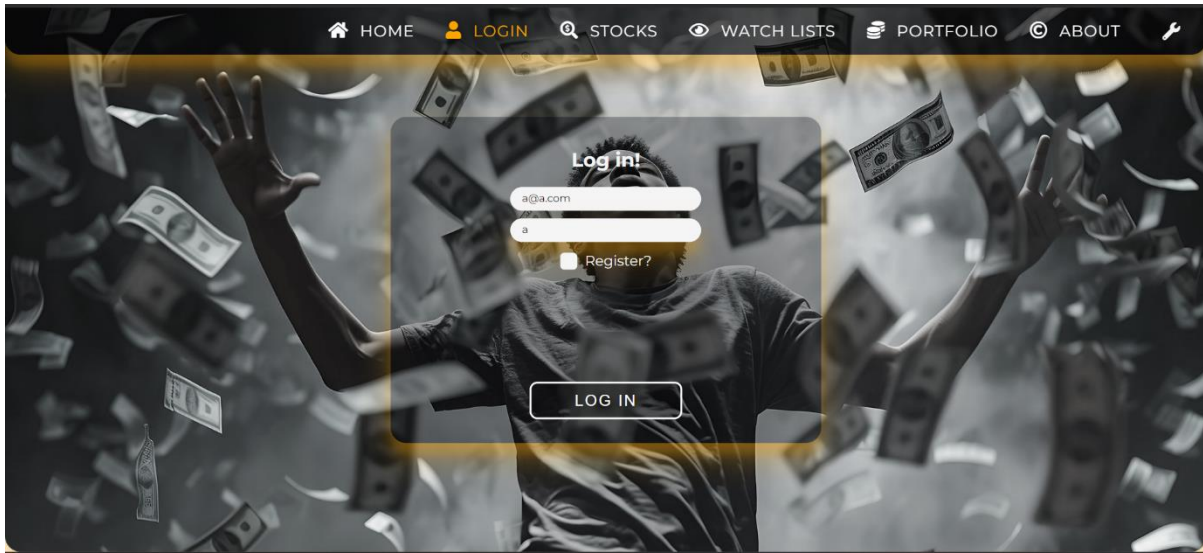
The architecture used in the system uses the main three components and then centralises them under the core management system. This core management system is the logic behind all the components. The core management system takes all of the data from the manager classes through interfaces and then sends the information to the core management system to then process the data.

The team meetings occurred three times a week since the beginning of the semester. They would be every Sunday (7.00pm), Wednesday (10.30 pm), and Friday (9.00 pm). No one missed a meeting during this. The first week of meetings were to get to know each other and see what everyone's strengths were and what everyone was happy doing. The second week focused on architecture and how group members were getting on with their work, the third week was finalising the GitHub setup and the code of conduct as well as the architecture. Sprint 2 commenced on week 4 and we hadn't been given much information how to start the work, so we started planning everything that week, week 5 started focusing on the business concept model and anything extra we could start working on. Week 6 carried on with the development of the business concept models and initial system architecture and focused on cleaning up the Java code following SOLID Principles, week 7 and 8 was spend finalising designs and development of software to provide a very basic terminal interface. Week 9 was spend trying to develop the advanced compound component styles alongside week 10 to ensure that we were happy with the style the architecture has gone in.

Test Case 1: Logging in with email "a@a.com" and password "a".



SAD Group Report



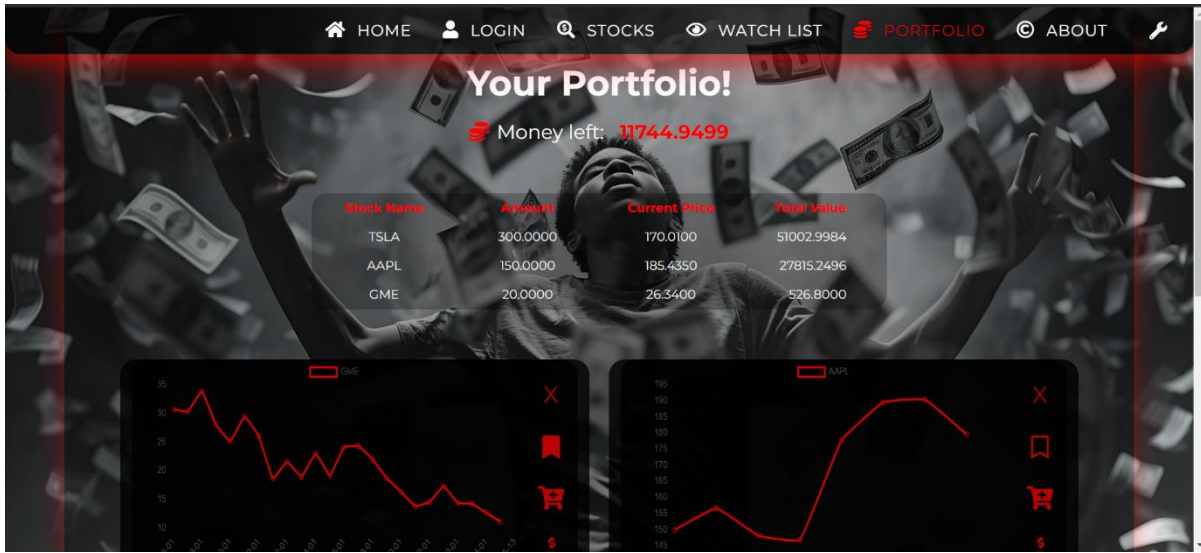
```
PROBLEMS 1 OUTPUT TERMINAL ... java + - [ ] [ ] ... ^ x

0Engineering%20Year%202/Software%20Architecture%20and%20Design/SAD
%20Group%20Coursework/Architecture_and_Design-CW_-Java-Javalin-JS
/sad/target/classes/resources/js/'
[sad.Main.main()] INFO io.javalin.Javalin - Listening on http://lo
calhost:3001/
[sad.Main.main()] INFO io.javalin.Javalin - You are running Javali
n 6.1.3 (released March 2, 2024).
name: a@a.com
password: a
[ ]

0 Java: Ready Ln 19, Col 22 Spaces: 2 UTF-8 CRLF {} Java [ ]
```

Test Case 2: Buying 20 GameStop (GME) stocks.

SAD Group Report



```
PROBLEMS 1 OUTPUT TERMINAL ... java + - [ ] [X] ... ^ X

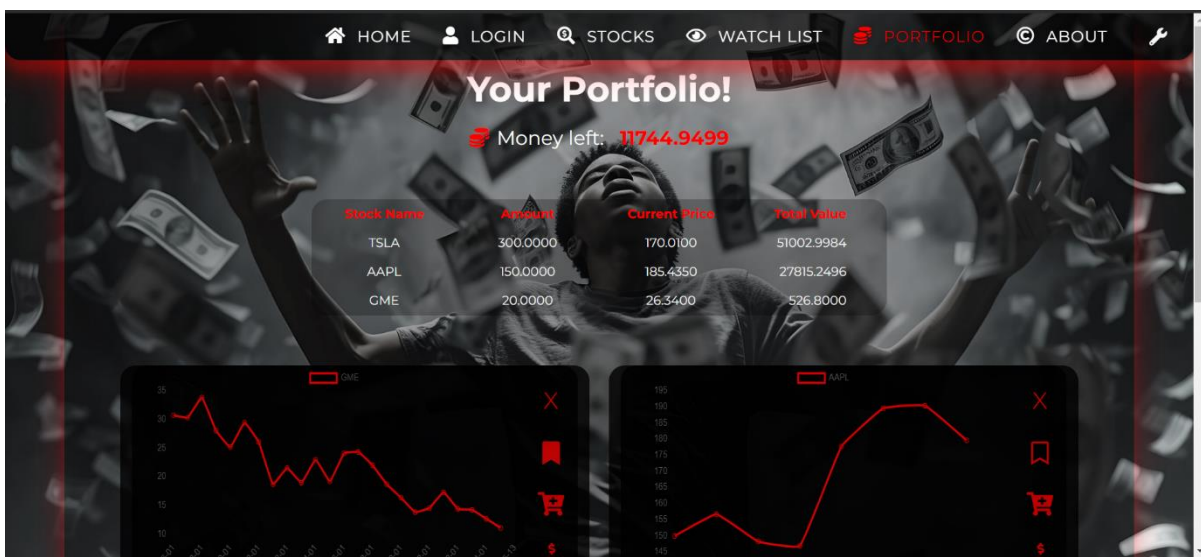
name: a@a.com
password: a

portfolio.push [{"name": "AAPL", "amount": 200}, {"name": "TSLA", "amount": 300}, {"name": "GME", "amount": 20, "value": 526.8000030517578}]

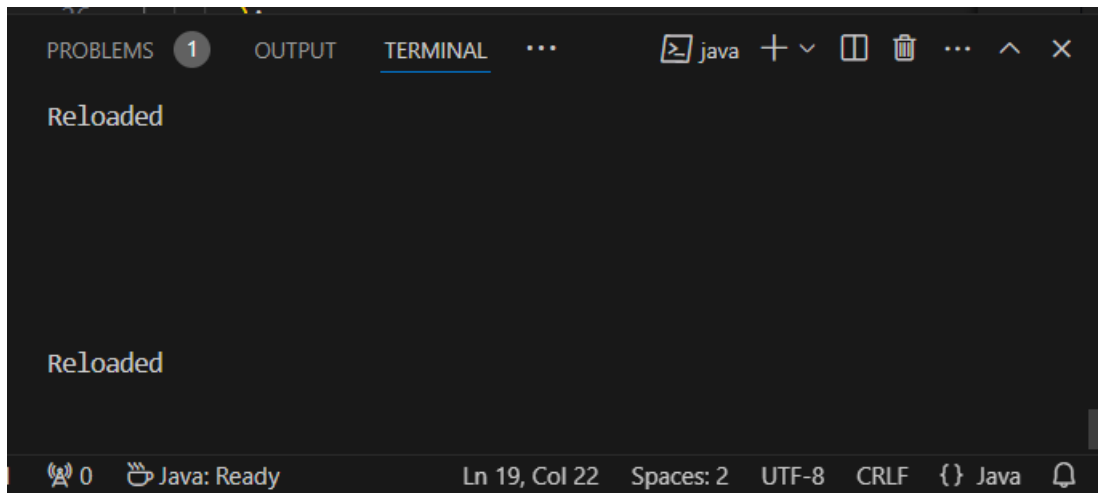
Reloaded

1 0 Java: Ready Ln 19, Col 22 Spaces: 2 UTF-8 CRLF {} Java
```

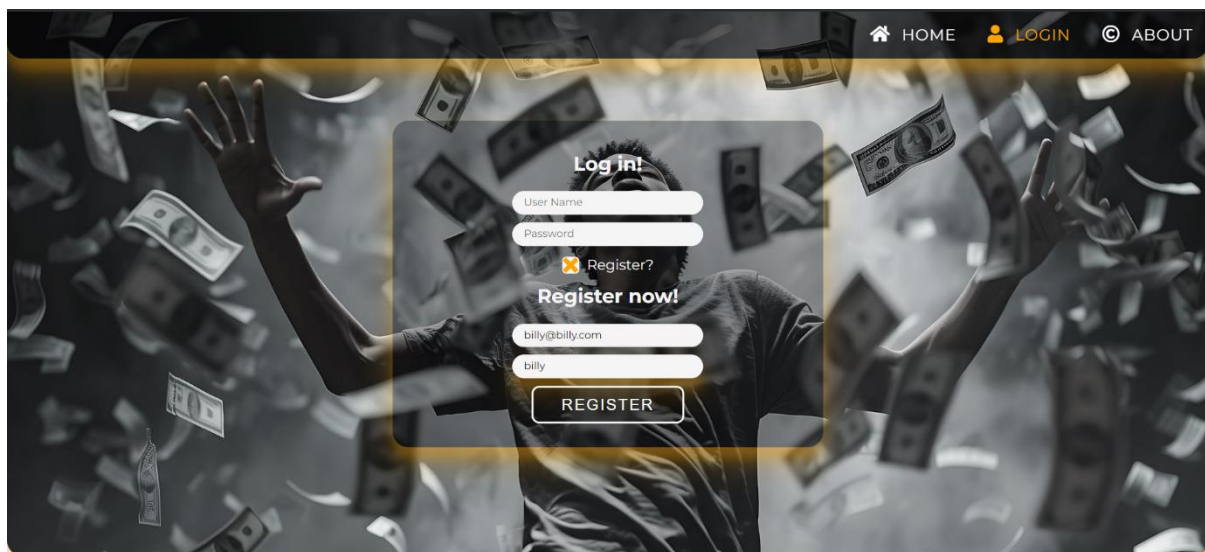
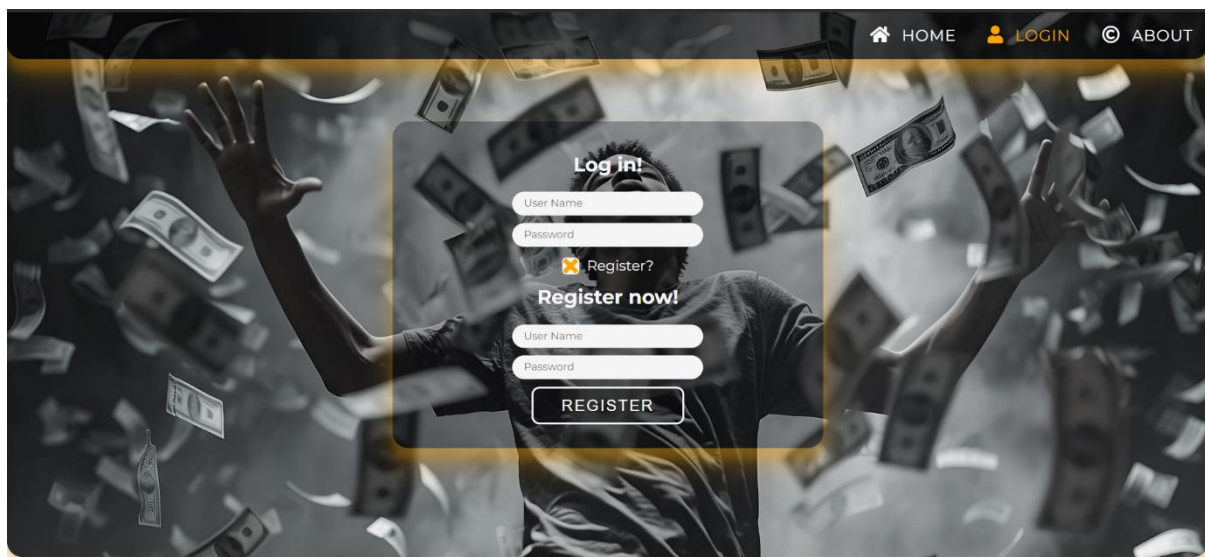
Test Case 3: Selling 50 Apple (AAPL) stocks.



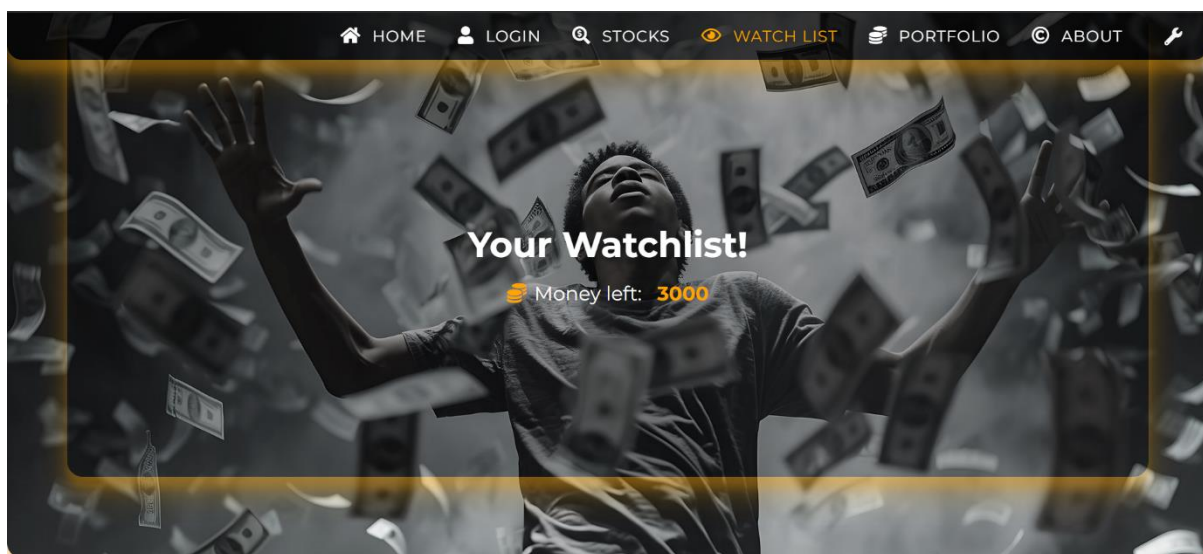
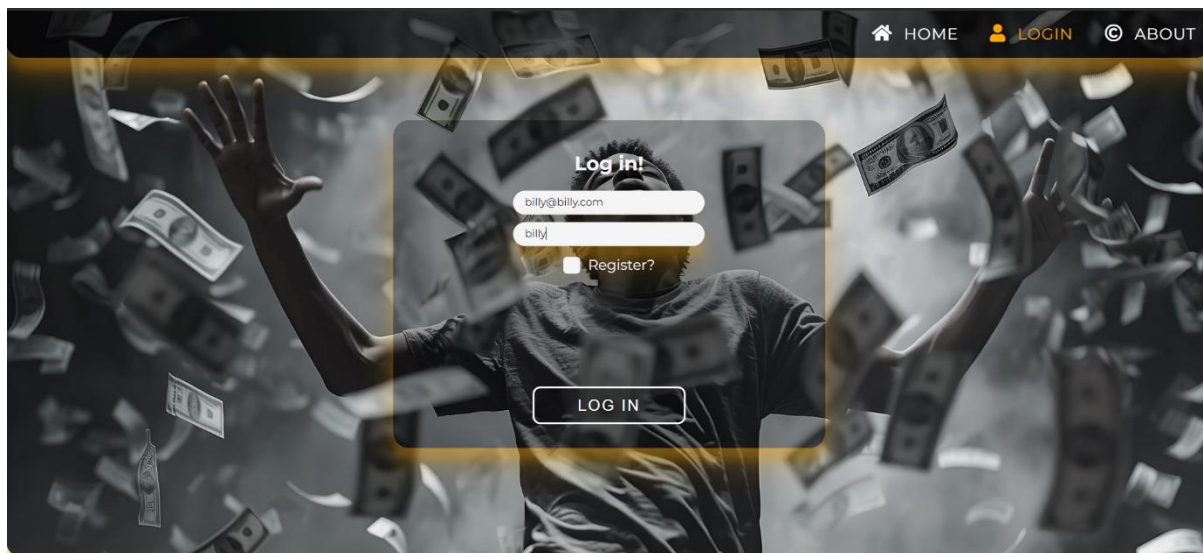
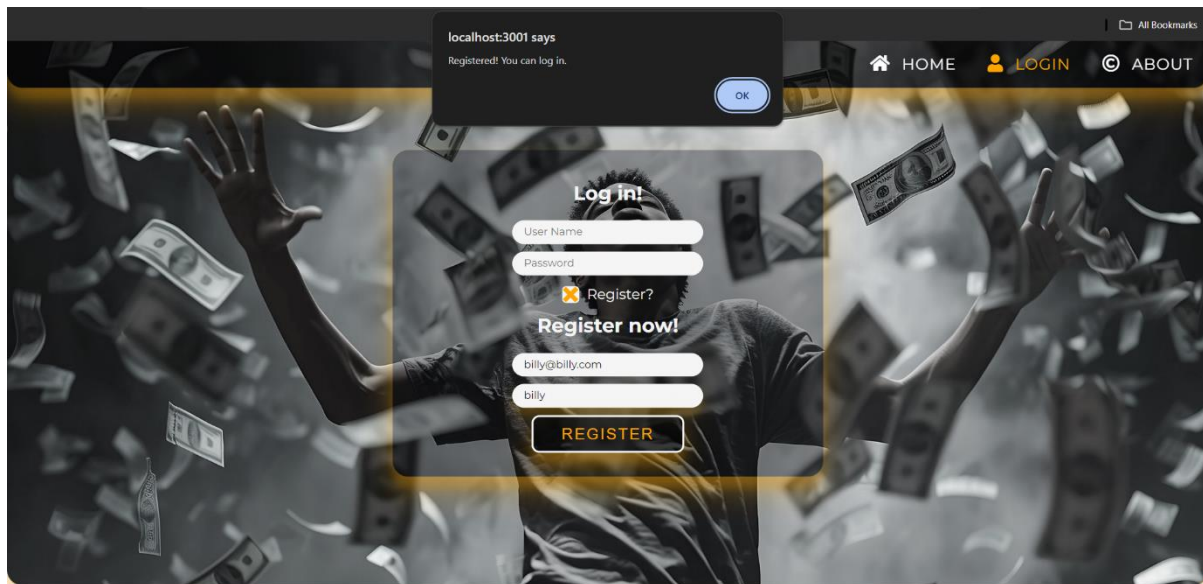
SAD Group Report



Test Case 4: Register a user with email “billy@billy.com” and password “billy”



SAD Group Report



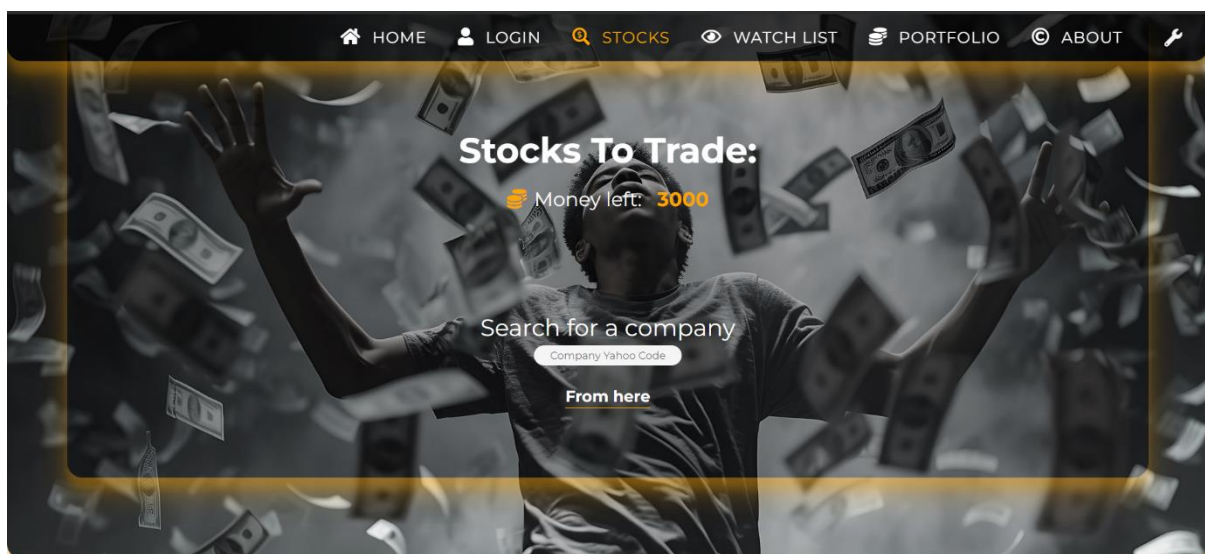
SAD Group Report

```
PROBLEMS 3 OUTPUT TERMINAL ... java + - [ ] [ ] ... ^ X

[sad.Main.main()] INFO io.javalin.Javalin - You are running Javalin 6.
1.3 (released March 2, 2024).
name: a@a.com
password: a
Received body: {"name":"billy@billy.com","password":"billy"}
Extracted login: billy@billy.com
Extracted password: billy
name: billy@billy.com
password: billy
[ ]

Java: Ready Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Pug [ ]
```

Test Case 5: Adding a stock to the watchlist



SAD Group Report

