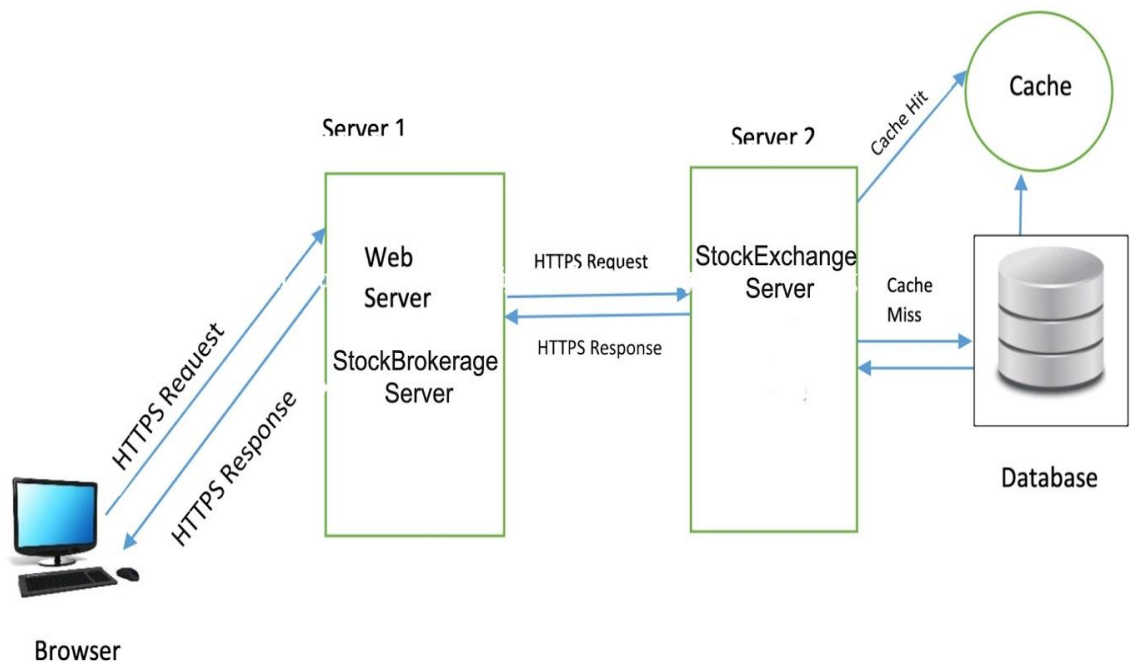


Stock Brokerage Web Application

Project Report

Architecture:

The Stock Brokerage is a **scalable** web-application based on **Service Oriented Architecture** (SOA). The architecture describing the complete flow of the web application is shown below:



Technologies Used:

- **Front-End:** HTML, CSS, JavaScript and Bootstrap
- **Server-Side:** NodeJS
- **Web Services:** NodeJS, PassPort
- **Database:** MySQL
- **Cache:** Redis Distributed Cache

Why NodeJS?

- Has robust technology stack
- Has fast-processing and event-based model
- Has scalable technology for microservices

- Has very good json support

Why Redis over Memcached?

- Redis only uses single cores while Memcached utilizes multiple cores - Thus, boasts a higher performance comparatively
- Efficient Memory usage compared to Memcached
- Supports complicated Data Structures and Operations

Functionalities Implemented:

1. New User Registration:

A new user can register by clicking on the 'SignUp' button on the homepage. Post clicking, the user will be redirected to a new page where he/she can fill out the required details. After submitting, if the SignUp is successful, the user will be redirected to the Profile page.

2. Existing user Login and Logout:

An existing user can login by clicking on the 'Login' button on the homepage. The user will be able to login to his account by entering the registered user id and password. On submitting, the details are passed to an Authentication Service. If the details are authenticated, the login will be successful and the user will be redirected to the Profile page. If not, 'Incorrect Login Info' will be displayed.

3. Reset Password:

If the user clicks on Forgot Password button, he will be redirected to the Forgot Password page where he will be asked to answer a security question. The answer entered will be verified against the one stored in the Database. If successful, the user will be redirected to Reset Password page, where he can reset the password.

4. Navigation Bar:

A navigation bar is present at the top of every page with 5 tabs: My Profile, My Stocks, Search Stocks, Buy Stocks and Sell Stocks. There is also a logout button present at the end of the navigation bar so that a user can easily logout of the application. This action will take the user back to the home page.

5. User Profile Information:

On this page, the user can see his/her profile details, i.e, Username, Email and Address. The user has an option to edit this information. Also, the current balance of the user and all the bank accounts linked to the user's account are displayed and are editable.

The user has been given an option to transfer any amount to/from chosen account to the current balance. If required, the user can also add new bank accounts by entering the Bank Account Number and Routing Number and those will be linked to the user's account.

6. User's Stocks:

On navigating to My Stocks, the user can see all the stocks owned by him/her. The bought stocks are by default set to Non-Recurring. If the user wants, the user can buy/sell stocks on a recurring basis by editing the Recurring_buy and Recurring_sell fields.

For example, if the user sets the Recurring_buy field of the stock 'AMD' to 3, then every 3rd day the stock of 'AMD' will be bought automatically.

Additional checks are in place to ensure that the user has enough amount in the account balance to be allowed to do buy transactions.

7. Search Stocks:

On selecting a particular stock and a particular time range, on clicking submit button, the price history of the stock will be shown in a graphical format, along with the current price to the user.

Filters on the selling price history like current day or last business day, current week past week, Month-to-date, Year-to-date and past 5 years are made available to ensure the user gets the big picture about the stock's market performance.

This functionality is available to users that are not logged in as well. Clicking the "Search Stocks" button on the home page will bring the user to the Search Stocks page where they can then view the price of stocks over a time period. However, because of our authentication the user cannot access any other parts of the website.

8. Buy Stocks:

The buy stock service is an asynchronous web service functionality that is implemented using queue. The user can choose multiple stocks from different companies in a single transaction. All the 'buy requests' are pushed to a queue with the price of stock fixed at the time of buying. A service then knows when items have been added to the queue, and processes the buy transactions. Other user transactions and interactions are not delayed during the buying process.

Total price of the transaction is pre-calculated to ensure that the user has sufficient funds to be allowed to purchase the stocks.

9. Sell Stocks:

The sell stock service is an asynchronous web service functionality that is implemented using queue. The user can choose multiple stocks from different companies in a single transaction. All the 'sell requests' are pushed to a queue with the price of stock fixed at

the time of selling. A service then knows when items have been added to the queue, and processes the sell transactions. Other user transactions and interactions are not delayed during the selling process.

10. Stock Summary:

Page listing the summary of all the stocks held by the user with details of the buy/sell schedule, price at the time of buying and option to change schedule.

11. Caching:

A distributed cache stores the profile information. As most user information remains static, less cache refreshes are needed.

12. Client-Server Communication Encryption:

The communication between client, web site and web services are using SSL. This provides communication security over network.

13. Security:

The web service calls are authenticated using Passport. It allows login capabilities and allows setting of required permissions. Each api and service call also has a user token associated with it. Also, the password is encrypted before storing it to the database.

Database:

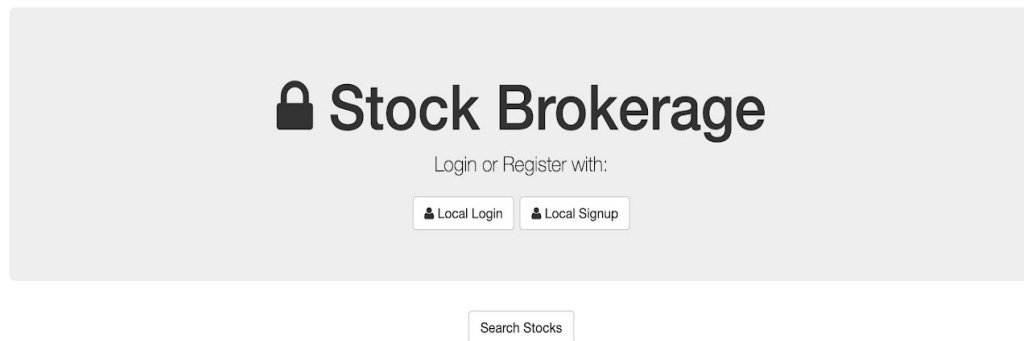
We used a relational MySQL database as all the data was structured. The schema had User, Stocks, Transaction and Current_Stock_Holdings tables to maintain the web application data. Events and triggers were used to add functionality like automatic buying/selling of stocks.

Issues Faced and Resolved:

We tried to decoupling the functionality and create a clean design that works independently. This was challenging at times as that meant more exports amongst modules. Using the model-view-controller architecture helped in creating a cohesive application while keeping the layers separate and clean.

Screenshots:

1. Initial Home Page:



2. Login Page:

➡ Login

Username

Password

Login

Need an account? [Signup](#)

Forgot password? [Forgot](#)

Or go [home](#).

3. SignUp Page:

➡ Signup

Username

Password

Signup

Already have an account? [Login](#)

Or go [home](#).

4. Forgot Password:

username

What is your favourite vacation destination?

Reset Password

5.Reset Password:

➡ Reset Password

Enter your new Password

Reset

[Login Here](#)[Login](#)

Or go [home](#).

6. Profile Page:

Profile Page

[My Profile](#)[My Stocks](#)[Search Stocks](#)[Buy Stocks](#)[Sell Stocks](#)[Logout](#)

Profile

View and Edit Profile

Personal Information

Username: qw

Password: *****

Email: pranjaldeshmukh95@gmail.com

Address: 17817

Edit Profile

Add Bank Accounts

View Bank Accounts

Transfer Balance

7. My Stocks: (Current stocks owned by the user)

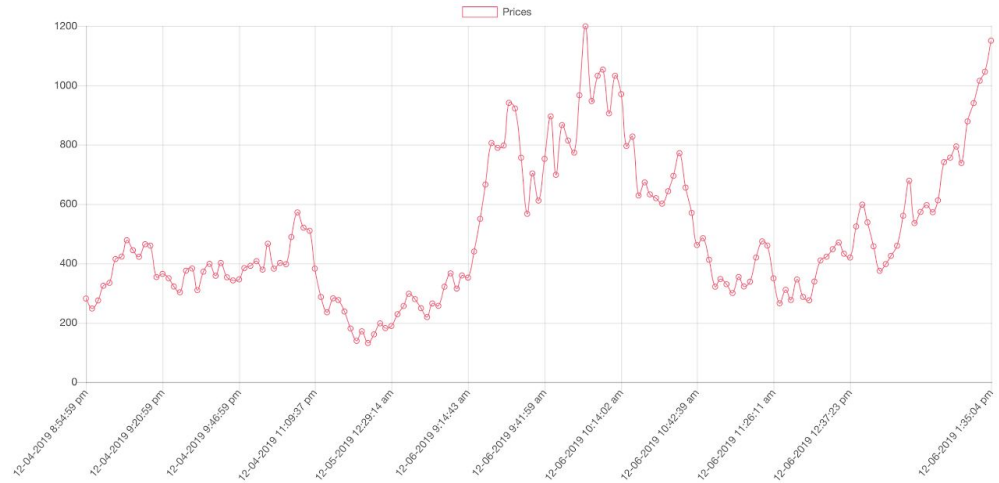
Profile Page							
My Profile		My Stocks		Search Stocks		Buy Stocks	
Sell Stocks							

AAPL

Company: Apple Inc.
Time Period: Past Week

Search StocksBuy Stocks

Current Price: \$1150.37



9.Buy Stocks:

My ProfileMy StocksSearch StocksBuy StocksSell StocksLogout

Buy Selected Stocks

Stock	Company	Price	Number of Stocks
<input type="checkbox"/> GILD	Gilead Sciences, Inc.	65.07	
<input type="checkbox"/> KHC	The Kraft Heinz Company	31.30	
<input type="checkbox"/> VIAB	Viacom Inc.	23.06	
<input type="checkbox"/> SCHW	The Charles Schwab Corporation	44.35	
<input type="checkbox"/> PG	The Procter & Gamble Company	120.54	
<input type="checkbox"/> LYFT	Lyft, Inc.	43.04	
<input type="checkbox"/> CZR	Caesars Entertainment Corporation	12.96	
<input type="checkbox"/> SDC	SmileDirectClub, Inc.	8.60	
<input type="checkbox"/> PTON	Peloton Interactive, Inc.	27.23	
<input type="checkbox"/> ITUB	Itau Unibanco Holding S.A.	8.56	
<input type="checkbox"/> HBI	Hanesbrands Inc.	15.57	
<input type="checkbox"/> RF	Regions Financial Corporation	16.52	
<input type="checkbox"/> MRVL	Marvell Technology Group Ltd.	26.86	
<input type="checkbox"/> SBUX	Starbucks Corporation	84.21	
<input type="checkbox"/> ZNGA	Zynga Inc.	6.28	
<input type="checkbox"/> GRUB	Grubhub Inc.	39.79	
<input type="checkbox"/> CBS	CBS Corporation	38.57	
<input type="checkbox"/> EQ	Equinix Inc.	64.70	

10.Sell Stocks:

My Profile

My Stocks

Search Stocks

Buy Stocks

Sell Stocks

Logout

SELL Selected Stocks

Stock	Company	Price	Number of Stocks	Stocks to sell
<input type="checkbox"/>	PG	The Procter & Gamble Company	117.8	1
<input type="checkbox"/>	SCHW	The Charles Schwab Corporation	1.59	4
<input type="checkbox"/>	VIAB	Viacom Inc.	34.2	3
<input type="checkbox"/>	KHC	The Kraft Heinz Company	19.49	1
<input type="checkbox"/>	GILD	Gilead Sciences, Inc.	15.62	12