1. INTRODUCTION TO PROJECT

In today's fast-paced financial world, managing investments efficiently and effectively is more crucial than ever. The "Investment Portfolio Management" system is designed to streamline and simplify the process of overseeing and optimizing investment portfolios. This project provides a robust and user-friendly platform that allows users to manage their financial assets, track performance, and make informed decisions.

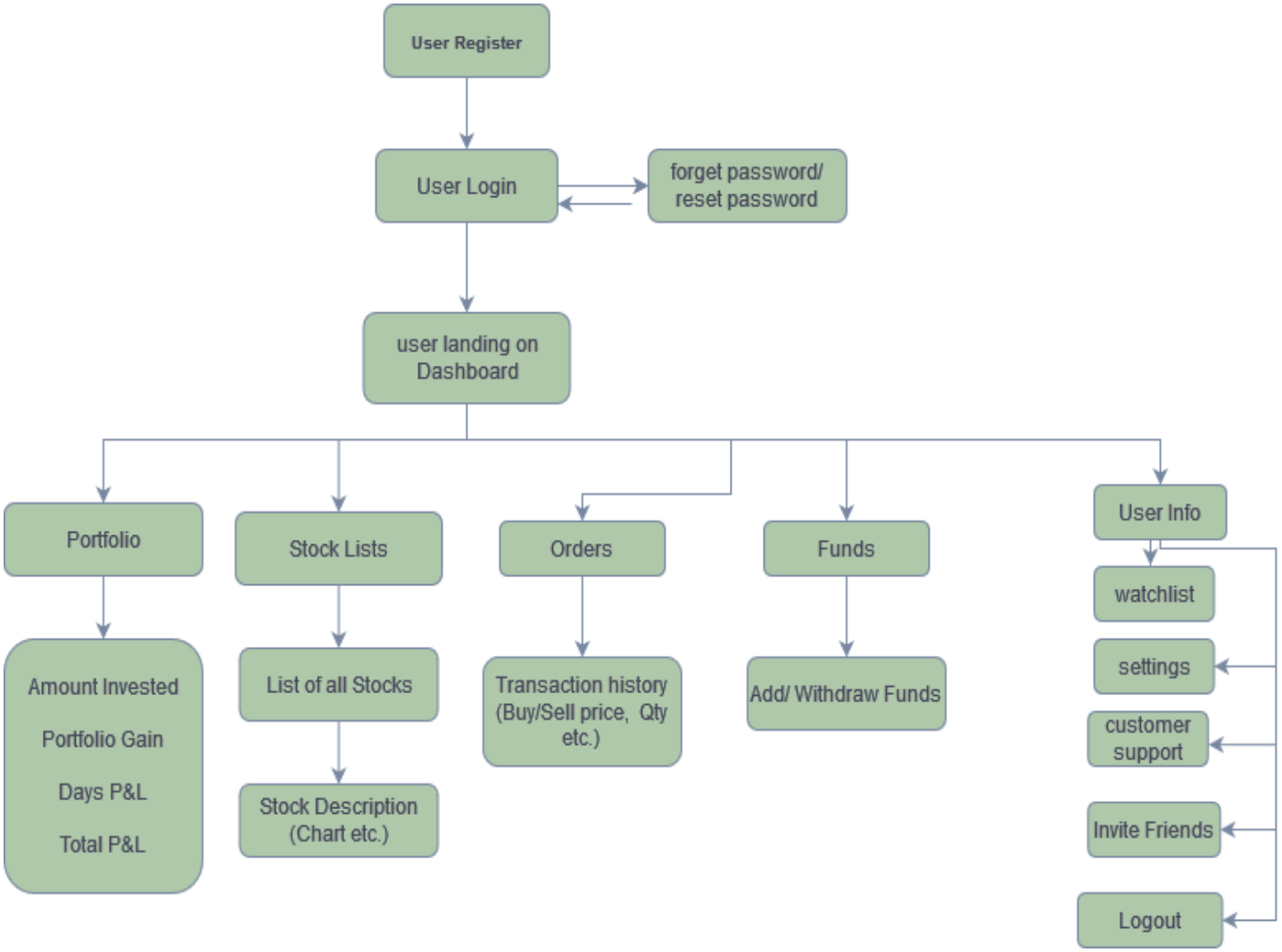
The web-based “Investment Portfolio Management” project is designed to streamline and enhance the management of investment portfolios. This system aims to provide users with a comprehensive platform to oversee, analyze, and optimize their financial assets.

The platform is built with a focus on providing a seamless user experience, leveraging modern web technologies to ensure scalability, security, and ease of use. By integrating advanced features and real-time data processing capabilities, the "Investment Portfolio Management" system aims to be a comprehensive solution for both novice investors and seasoned professionals.

The "Investment Portfolio Management" system is a modern solution to the challenges of managing investment portfolios. By combining the strengths of React and Spring Boot, the platform is designed to offer a powerful, secure, and user-friendly experience for all its users. This documentation will provide an in-depth look into the development process, key features, and the underlying technology stack that powers the platform.

2.REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS



User:

1. Implement user registration and login functionalities with proper authentication mechanisms.
2. Allow users to securely add and manage their bank details for transactions.
3. Provide options for users to add funds to their accounts through various payment methods.
4. Develop the buying and selling functionality for both stocks and mutual funds.
5. Allow users to create and manage watchlists for tracking favourite stocks and mutual funds.
6. Implement features to add and remove items from the watchlist easily.
7. Develop a dashboard for users to view their portfolio holdings and performance.
8. Display key metrics such as total investment value, gains/losses, and asset allocation.
9. Provide visualization tools such as charts and graphs to analyse portfolio performance over time.
10. Maintain a comprehensive transaction history for users to track their buying and selling activities.
11. Allow users to customize their account settings and preferences.
12. Provide options for profile management, including updating personal information and preferences.

2.1 User Registration

- **Purpose:** The registration feature allows new users to create an account in the system. This is a prerequisite for accessing the full range of portfolio management features.
- **Process:**
    1. **Registration Form:** New users must fill out a registration form that includes fields such as name, email address, password, contact number, and optionally additional personal information.
    2. **Email Verification:** Upon submitting the registration form, users receive a verification email to confirm their email address and activate their account.
    3. **Account Activation:** The user clicks on the verification link in the email to activate their account. The system then allows the user to log in.

2.2 Security Measures:

- **Password Encryption:** Passwords are stored securely using encryption techniques to ensure user data protection.
- **Email Validation:** The system checks the validity of the email address to prevent invalid or fraudulent registrations.

2.3 Password Management

- **Password Reset:** Users can request a password reset if they forget their password. This involves:
    1. **Password Reset Request:** Users enter their email address to receive a password reset link.
    2. **Reset Link:** An email with a secure reset link is sent to the user. Clicking the link directs them to a page where they can set a new password.
    3. **Password Update:** The user updates their password, which is then securely encrypted and stored.

Dashboard Overview

The Dashboard is the central interface of the “Investment Portfolio Management aka InvestLine” system, designed to provide users with a comprehensive and intuitive platform for managing their investments Upon logging in, users are immediately redirected to the Dashboard, where they can access a wide range of functionalities essential for effective portfolio management. The interface is organized to present real-time data and easy navigation, ensuring that users can quickly execute trades, monitor their investments, and stay updated with market trends.

The Dashboard in the “Investment Portfolio Management” system is developed using React, a leading JavaScript library for building dynamic and responsive user interfaces. React's component-based architecture allows the Dashboard to be highly modular, making it easy to manage and update individual sections like the Search Bar, Buy/Sell Interface, Watchlist, and Holdings Page.

Key features include a powerful search bar for quickly finding stocks, a buy/sell interface for executing trades, a watchlist for tracking preferred stocks, and a holdings page that provides a detailed view of the user’s current investments. Additionally, the Dashboard includes a market overview panel to keep users informed about broader market trends, as well as an order history section and account summary for easy access to past transactions and account settings.

The Dashboard is designed to provide a user-friendly experience, with real-time data and intuitive navigation, enabling users to make informed investment decisions efficiently. This comprehensive layout ensures that users have immediate access to all necessary features, enabling them to make informed investment decisions and efficiently manage their portfolios in real time.

Components Breakdown

1. **Search Bar**
    - **Functionality:** The Search Bar allows users to quickly find stocks by typing in the stock's name or ticker symbol. It provides real-time search results, displaying relevant stock information as the user types.
    - **Features:**
        - **Auto-suggestions:** As users type, the search bar provides auto-suggestions based on the input, making it easier to find specific stocks.
        - **Stock Data:** Displays the latest stock prices and key details in the dropdown as suggestions are generated.
        - **Navigation:** Users can click on a search result to view detailed stock information or initiate a transaction (buy/sell).

2. **Buy/Sell Interface**
    - **Functionality:** This component allows users to execute trades directly from the Dashboard. Users can choose to buy or sell stocks by specifying the quantity, price (market or limit), and other trade parameters.
    - **Features:**
        - **Order Types:** Supports various order types including market orders, limit orders, and stop-loss orders.
        - **Trade Confirmation:** Once an order is placed, the system provides a confirmation dialog showing the trade details before execution.
        - **Transaction History:** Post-execution, the transaction is logged in the user's order history, accessible from the Dashboard.

3. **Watchlist**
    - **Functionality:** The Watchlist component allows users to monitor selected stocks over time. Users can add stocks to their watchlist to track price movements and performance metrics.
    - **Features:**
        - **Stock Tracking:** Users can view price updates and volume for each stock on their watchlist.
        - **Customization:** Users can create watchlists, categorize stocks, and reorder items based on their preferences.
        - **Alerts:** The system can be configured to send notifications or alerts when a stock in the watchlist hits a certain price or meets predefined conditions.

4. **Holdings Page**
    - **Functionality:** The Holdings Page provides a detailed view of the user's current investments. It shows the user's portfolio, including the stocks they own, quantities, average purchase price, current market value, and overall profit/loss.
    - **Features:**
        - **Portfolio Overview:** Displays an aggregate view of the portfolio's total value, percentage gain/loss, and diversification across different asset classes.

- **Detailed Breakdown:** For each stock, the holdings page shows detailed information such as the number of shares, average buy price, current market price, and unrealized gains or losses.
- **Transaction History:** Users can click on individual stocks to view their full transaction history, including dates and types of transactions (buy/sell).

5. **Order History & Notifications**
    - **Functionality:** This component logs all the user's past transactions and system notifications related to their account and portfolio.
    - **Features:**
        - **Order Tracking:** Users can view detailed records of all past orders, including executed, pending, and canceled trades.
        - **Notifications:** The system sends alerts for significant events, such as order executions, price alerts, dividend payouts, or corporate actions.

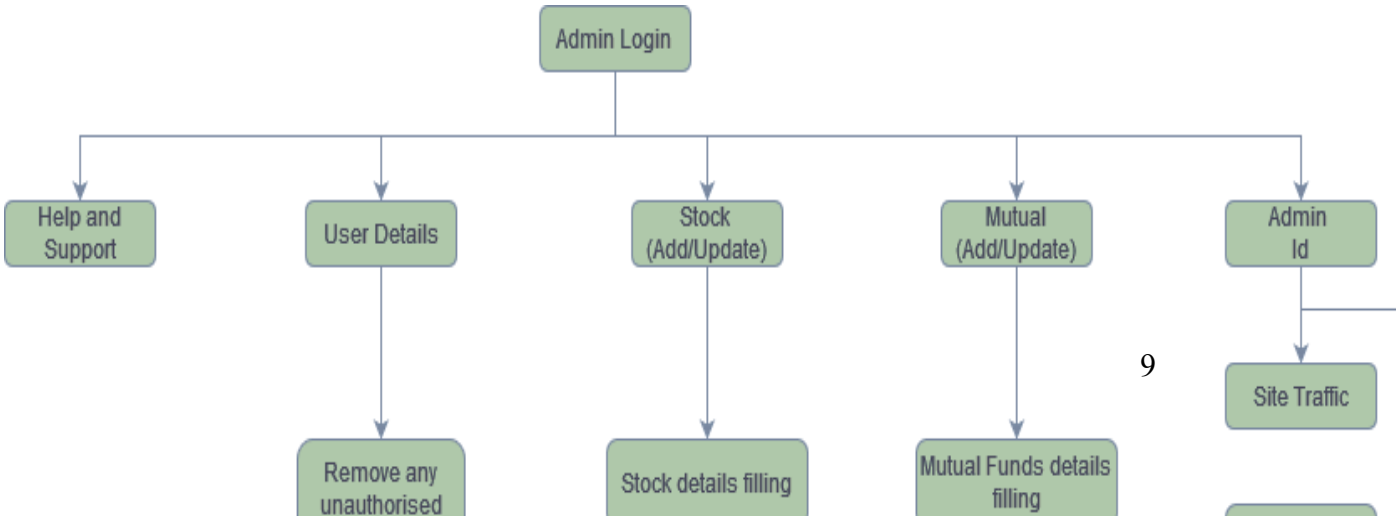
6. **Account Summary & Settings**
    - **Functionality:** This section provides a summary of the user's account balance, available funds, and access to account settings.
    - **Features:**
        - **Settings:** Users can access settings to update their profile, manage linked bank accounts, configure alerts, and set trading preferences.
7. **Funds**
    - **Functionality:**

- **Add Bank Details:** User can add his bank details to further add funds.
- **Deposit/Withdraw:** User can Desposit money in his Trading account and can also Withdraw Funds into his Bank accounts.
- **Account Balance:** Shows available funds, margin, and total account value.

Admin Side Server Overview:

The admin side of the "Investment Portfolio Management" system is designed to provide comprehensive management and support tools for system administrators. This backend server, developed using Spring Boot, enables administrators to maintain the integrity of the platform, manage user data, and ensure the smooth operation of the investment platform.

The admin interface offers various functionalities, including the ability to manage stock listings, oversee user accounts, and provide support to the client side. The admin side is critical to the overall system, ensuring that the platform remains up-to-date, secure, and responsive to user needs.



Components Breakdown

1. **Stock Management**
    - **Functionality:** Administrators have the capability to add, update, or delete stock listings within the system. This feature ensures that the stock database is accurate and reflects current market offerings.
    - **Features:**
        - **Add Stocks:** Admins can add new stocks to the system by inputting essential details such as stock name, ticker symbol, initial price, and other relevant data.
        - **Update Stocks:** The system allows for the modification of existing stock details, such as updating prices, changing stock information, or correcting errors.
        - **Delete Stocks:** If a stock is no longer available or relevant, admins can remove it from the system, ensuring the stock listings remain current and uncluttered.

2. **User Management**
    - **Functionality:** The admin interface allows administrators to oversee user accounts, including the ability to delete users who violate terms of service or are inactive.
    - **Features:**

- **View User Profiles:** Admins can access detailed user profiles, including their registration information, activity logs, and investment history.
- **Delete Users:** Administrators can remove users from the system, which is particularly useful for managing inactive accounts, resolving security concerns, or enforcing platform policies.
- **User Activity Monitoring:** The system logs user activity, allowing admins to monitor for unusual or suspicious behavior that might require intervention.

3. **Help and Support Management**
    - **Functionality:** This component is dedicated to providing support to users on the client side. Admins can manage support tickets, respond to user queries, and provide guidance on using the platform.
    - **Features:**
        - **Support Ticket System:** Users can submit support requests, which are then managed by admins. The system allows tracking of the status of each ticket, ensuring timely responses.
        - **Knowledge Base Management:** Admins can update or add new entries to a knowledge base, providing users with self-help resources that address common issues or questions.
        - **Communication Tools:** The admin interface includes tools for direct communication with users, such as sending messages or notifications regarding their queries or issues.

4. **System Maintenance and Security**
    - **Functionality:** Admins are responsible for maintaining the backend systems, ensuring the platform runs smoothly and securely.
    - **Features:**
        - **System Updates:** Admins can apply updates to the system, including software patches, feature upgrades, and security enhancements.
        - **Backup and Recovery:** The admin side includes tools for regular data backups and recovery, ensuring data integrity and availability in case of system failures.

5. **Analytics and Reporting**
    - **Functionality:** Administrators have access to a range of analytical tools to monitor system performance, user activity, and financial transactions within the platform.
    - **Features:**
        - **Usage Analytics:** Provides insights into how users are interacting with the platform, including metrics on active users, most accessed features, and overall platform engagement.
        - **Transaction Reporting:** Admins can generate reports on financial transactions, including buying/selling activities, volume of trades, and overall market activity on the platform.
        - **System Performance Metrics:** Monitors the performance of the server, including response times, uptime, and load management, ensuring that the platform remains efficient and responsive.

6. **Spring Framework**
    - User Authorization with Spring Security
    - In the "Investment Portfolio Management" project, we have integrated Spring Security to handle user authentication and authorization. This implementation ensures that only authenticated users can access specific resources and perform actions based on their roles.
    - Key Components:
    - Security Configuration:
    - We configured WebSecurityConfigurerAdapter to customize the security settings.
    - Defined httpSecurity to secure endpoints, enforce authentication, and authorize access based on user roles (e.g., USER, ADMIN).
    - User Authentication:
    - Implemented custom UserDetailsService to load user-specific data from the database.
    - Passwords are stored securely using BCrypt hashing.
    - JWT (JSON Web Token) Integration:
    - Utilized JWT for stateless authentication, generating tokens upon successful login.
    - Configured filters to validate JWT tokens in incoming requests, ensuring that only authenticated requests can access protected endpoints.
    - Role-Based Access Control (RBAC):
    - Defined role-based access to different endpoints, ensuring that users can only perform actions permitted by their roles. For example, ADMIN users can manage stocks and users, while USER roles have access to stock trading functionalities.
    - Exception Handling:
    - Implemented custom handlers for authentication and access-related exceptions to provide meaningful responses to the client.
    - This setup provides a robust and scalable security framework, ensuring the protection of user data and operations within the application.

3. DESIGN

3.1 Database Design

The following table structures depict the database design.

Table1: User_Info

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
creation_time	datetime(6)	YES		NULL	
dob	date	YES		NULL	
email	varchar(30)	YES	UNI	NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
password	varchar(255)	NO		NULL	
phone_number	varchar(255)	YES		NULL	
role	varchar(30)	YES		ROLE_CUSTOMER	

Table2: Bank Details

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
creation_time	datetime(6)	YES		NULL	
account_number	varchar(255)	NO		NULL	
bank_address	varchar(255)	NO		NULL	
bank_name	varchar(255)	NO		NULL	
ifsc_code	varchar(255)	NO		NULL	
user_id	bigint	NO	MUL	NULL	

Table3: Watchlists

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
creation_time	datetime(6)	YES		NULL	
w_name	varchar(255)	YES		NULL	
user_id	bigint	YES	MUL	NULL	

Table4: Trade_History

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
creation_time	datetime(6)	YES		NULL	
price_at_transaction	double	NO		NULL	
quantity	int	NO		NULL	
trade_date	datetime(6)	YES		NULL	
transaction_type	varchar(255)	YES		NULL	
stock_id	bigint	NO	MUL	NULL	
user_id	bigint	NO	MUL	NULL	

Table5: Orders

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
creation_time	datetime(6)	YES		NULL	
order_status	varchar(255)	YES		NULL	
order_type	varchar(255)	YES		NULL	
price	double	NO		NULL	
quantity	int	NO		NULL	
stock_id	bigint	NO	MUL	NULL	
user_id	bigint	NO	MUL	NULL	

Table6: Portfolio

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
creation_time	datetime(6)	YES		NULL	
average_purchase_price	double	NO		NULL	
quantity	int	NO		NULL	
stock_id	bigint	NO	MUL	NULL	
user_id	bigint	NO	MUL	NULL	

Table7: Stocks

Field	Type	Null	Key	Default	Extra
stock_id	bigint	NO	PRI	NULL	auto_increment
company_name	varchar(255)	YES		NULL	
created_on	date	YES		NULL	
instrument_key	varchar(255)	YES		NULL	
updated_on	datetime(6)	YES		NULL	

Table8: nse_stocks

Field	Type	Null	Key	Default	Extra
instrument_key	varchar(255)	NO	PRI	NULL	
exchange	varchar(255)	YES		NULL	
instrument_type	varchar(255)	YES		NULL	
last_price	varchar(255)	YES		NULL	
name	varchar(255)	YES		NULL	
tradingsymbol	varchar(255)	YES		NULL	

4. CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	Person, BankVault, SMSMessage, Dept	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Camel	getDetails, updateStore	Methods should use **verbs** or verb phrases.
Parameter	Camel	personName, bankCode	Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	Disposable	Do not use the '_' sign
Property	Pascal	ForeColor, BackColor	Use a noun or noun phrase to name properties.
Associated private member variable	_camelCase	_foreColor, _backColor	Use underscore camel casing for the private member variables
Exception Class	Pascal with "Exception" suffix	WebException,	

Comments

- Comment each type, each non-public type member, and each region declaration.
- Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
- Separate comments from comment delimiters (apostrophe) or // with one space.
- Begin the comment text with an uppercase letter.
- End the comment with a period.
- Explain the code; do not repeat it.

5. TEST REPORT

Another group called Linux did the testing and the report of the testing is given hereunder.

GENERAL TESTING:

SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	Register Page	Redirected to Next page	OK	Nothing
2	Login Page	Pop-up will come	Ok	Please enter username and password again .

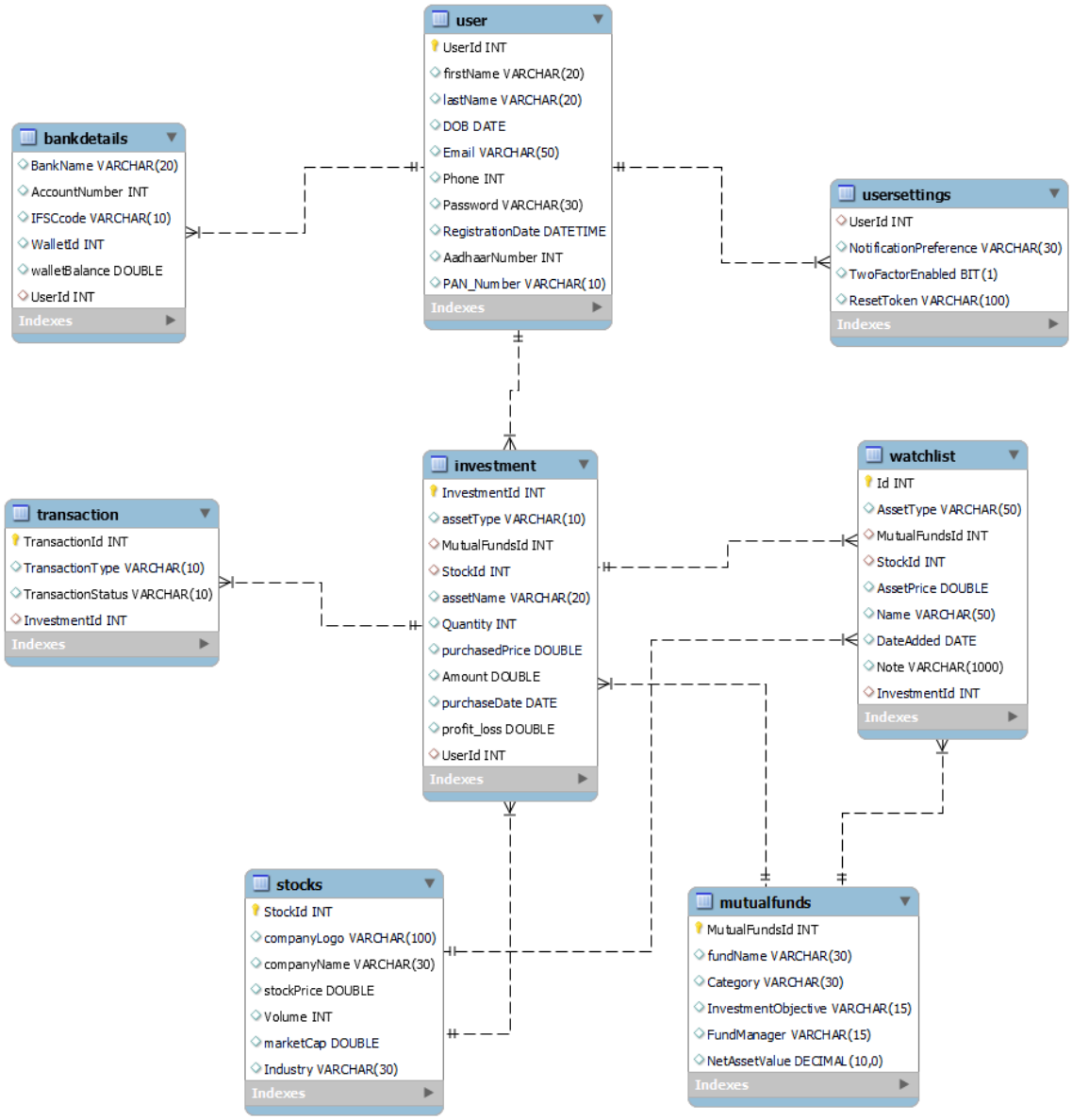
3	Reset login	Only users password will be reseted	Ok	Nothing
4	Quick search Stocks	Gives all details	Ok	Nothing
5	Add Stocks	Add Stocks to watchlists	Ok	Nothing
6	Checking login or not	User is logged in or not	Ok	Nothing
7	Add Bank details	User bank details will be added	Ok	Nothing
8	Goto Holding page	Redirect to holdings page	Ok	Nothing
9	Buy Stock	Buys stock at given price	Ok	Nothing
10	Transaction	Reverts to Orders page	Ok	Nothing
11	View transaction done	It shows you all transactions done previously	Ok	Nothing
12	Logout	It will logout from user profile.	Ok	Nothing
	STATIC TESTING			
SR-NO	**Deviation**	**Program**		
1	Commenting not followed	All Web Application		

6. PROJECT MANAGEMENT RELATED STATISTICS

DATE	WORK PERFORMED	___*SLC Phase*	Additional Notes
APR 18,2024	Project Allotment and User Requirements Gathering	Feasibility Study	Our team met the client Mr. Nitinkudale (CEO, SIIT Pune) to know his requirements.
MAY 14,2024	Initial SRS Document Validation And Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better
MAY 15,2024	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces	Requirement Analysis & Design Phase	Database Design completed
JUNE 19,2024	Business Logic Component design Started	Design Phase	----------
JUNE 20,2024	Coding Phase Started	Coding Phase	70% of Class Library implemented.
JUNE 27,2024	Implementation of Web Application and Window Application Started	Coding Phase	Class Library Development going on.
JULY 02, 2024	Off	Off	Off
JULY 10, 2024	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	Class Library Modified as per the need.
JULY 16, 2024	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	--

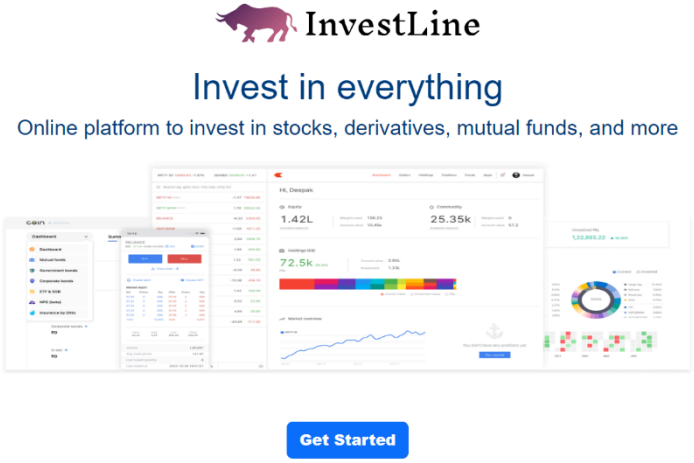
JULY 22, 2024	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project Manager
JULY 28, 2024	The Project was Tested by the respective Team Leaders and the Project Manager	Testing Phase (Module Testing)	--
AUG 10, 2024	The Project was Submitted to Other Project Leader of Other Project Group For Testing	Testing Phase (Acceptance Testing)	The Project of Other Team was Taken up by the Team for Testing
AUG 10-12, 2024	The Errors Found were Removed	Debugging	The Project was complete for submission
AUG 16, 2024	Final Submission of Project		

Entity Relationship Diagram

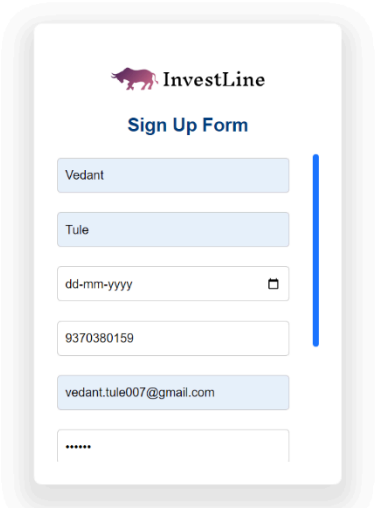


Appendix B

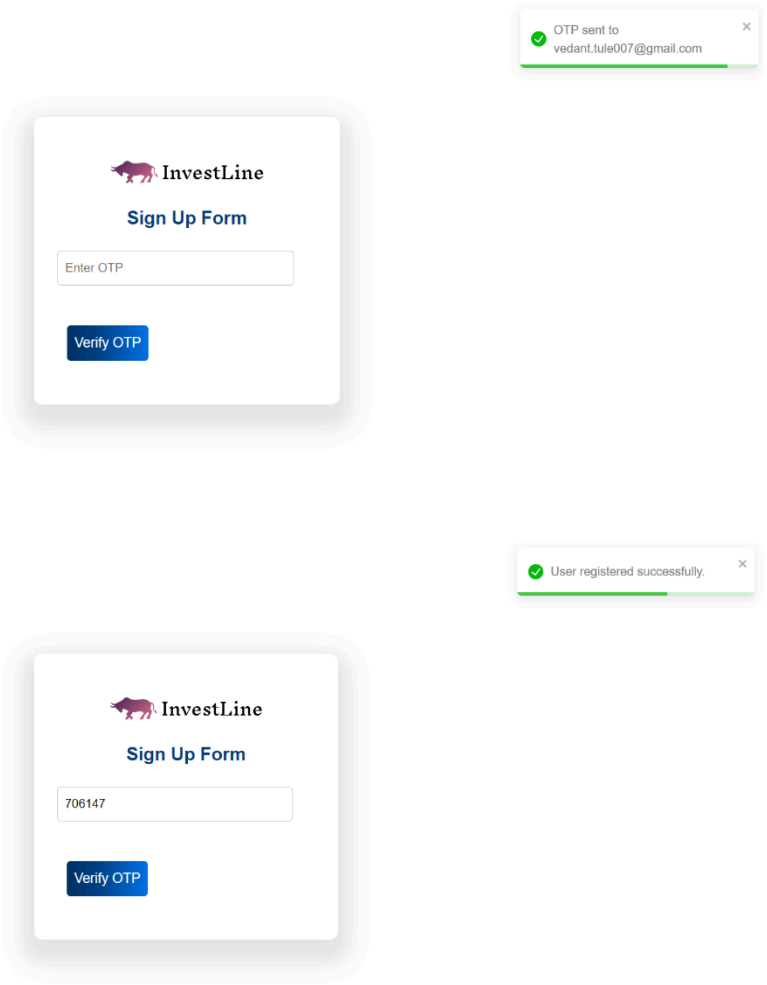
MainPage:



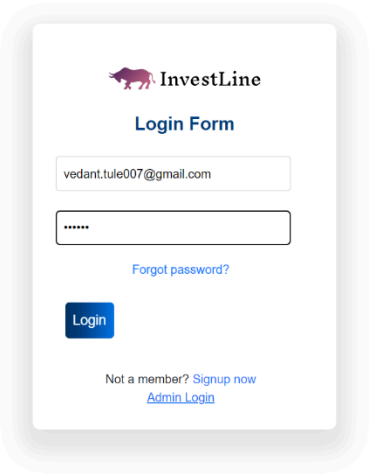
User Registration Page:



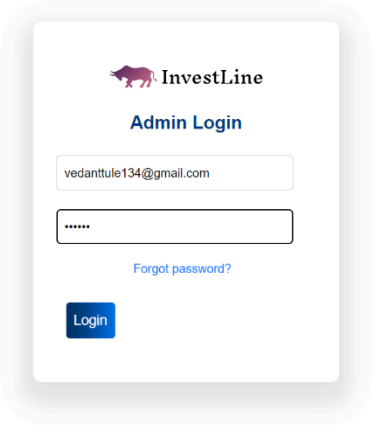
OTP Verification:



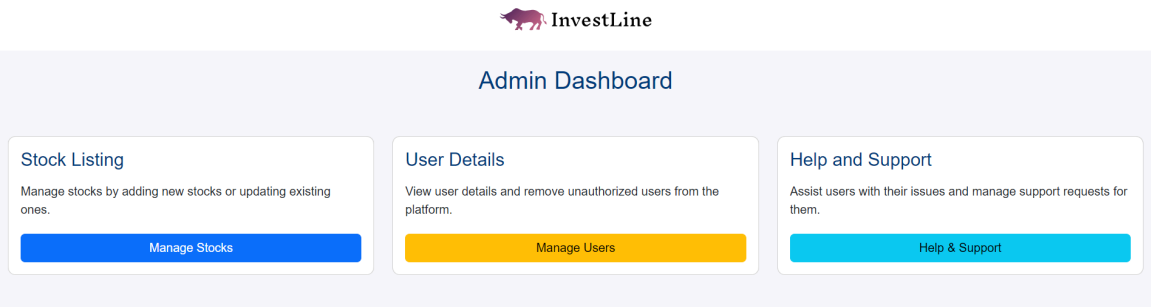
Login User:



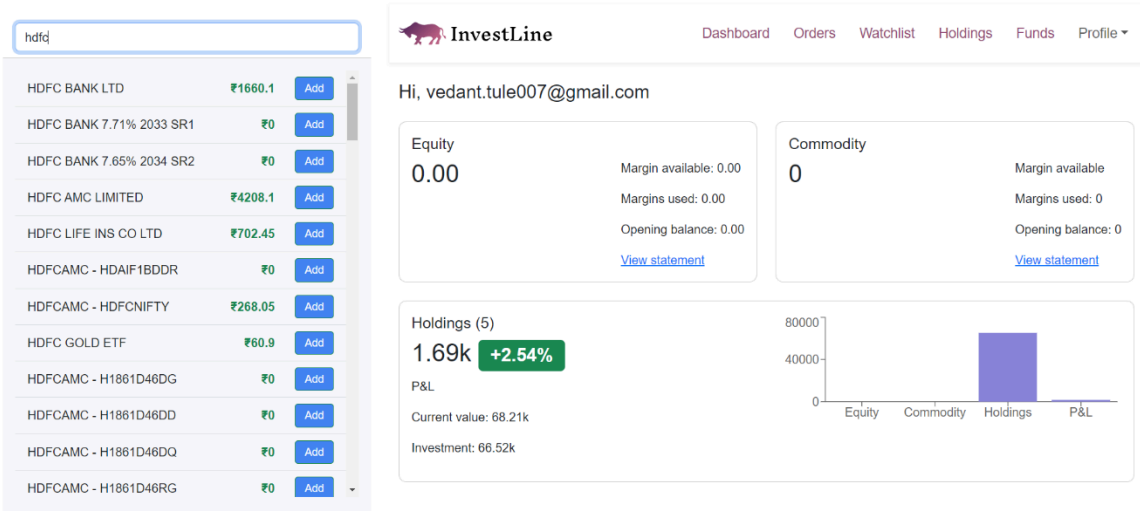
Admin Login:



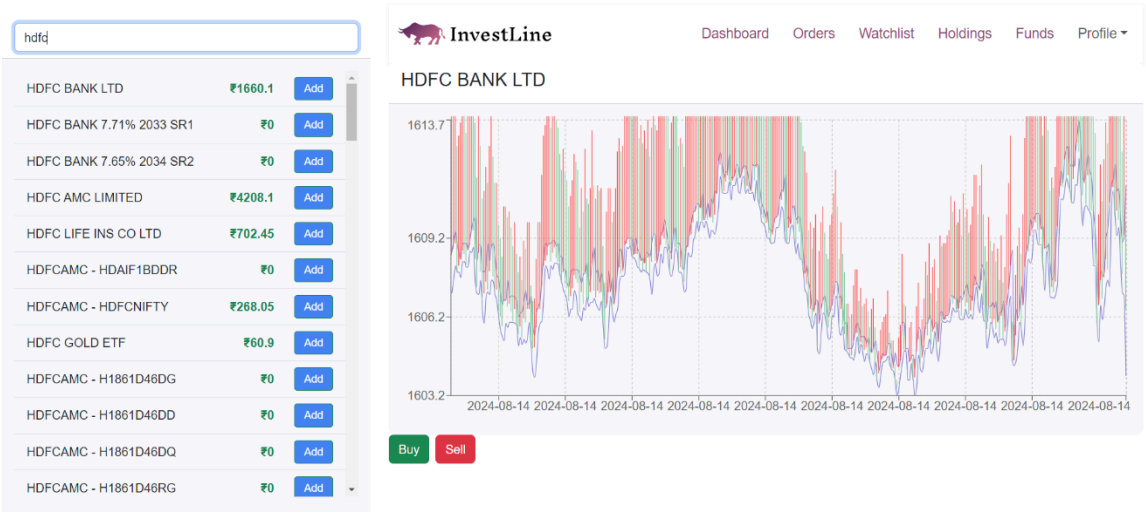
Admin Dashboard:

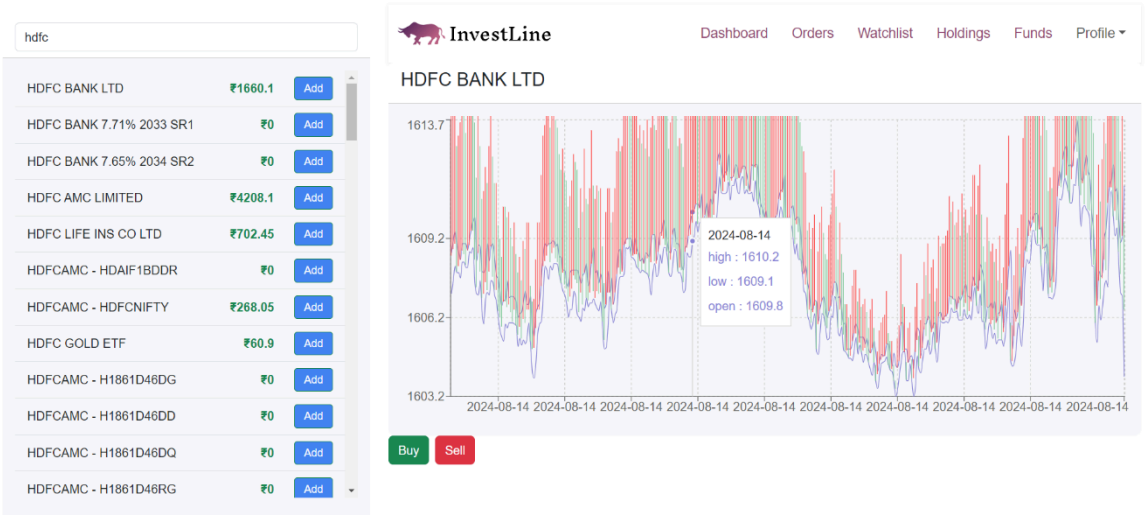


User Dashboard:

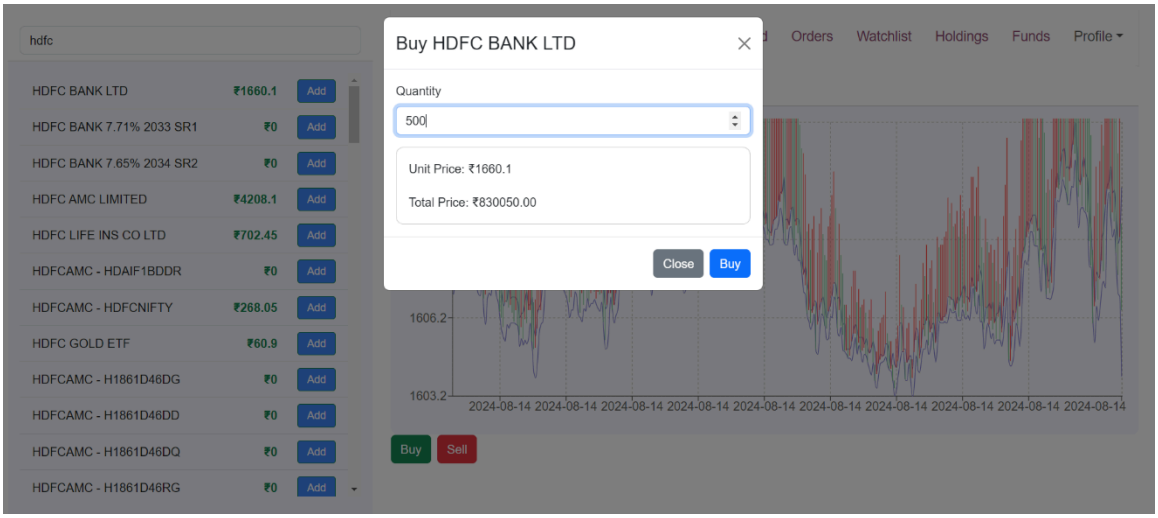


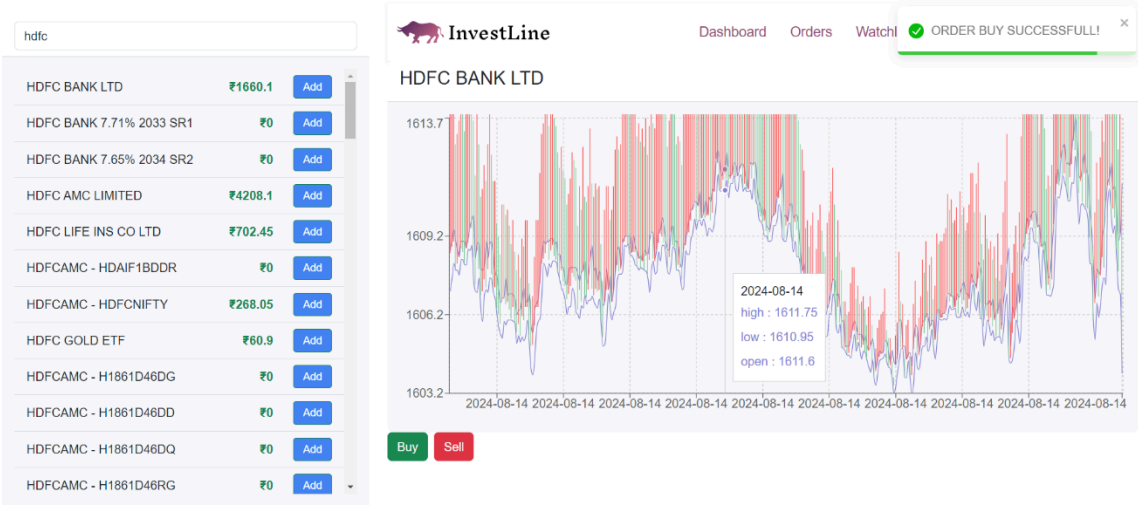
Stocks with Historical Candle Chart:

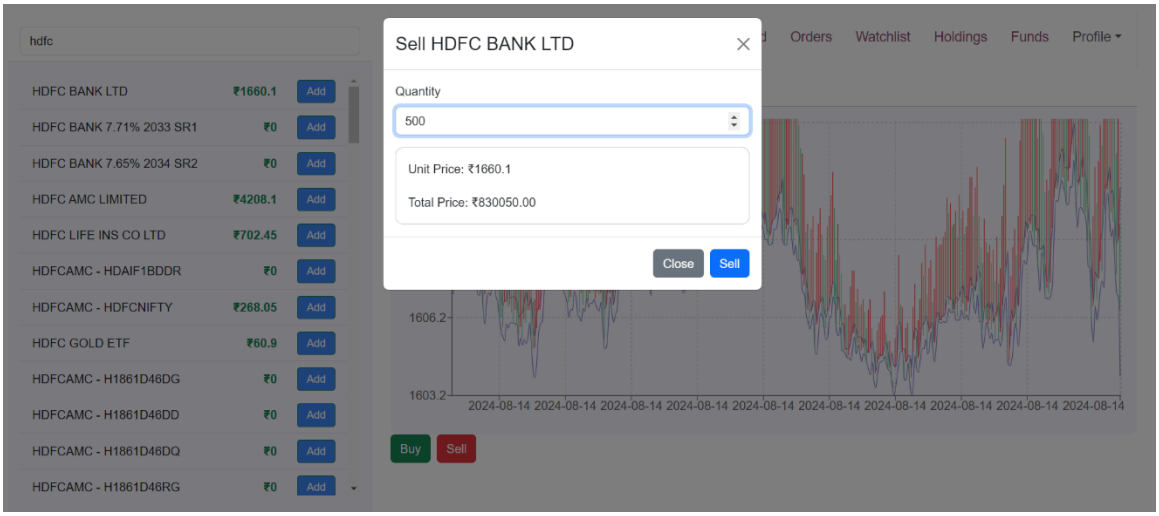


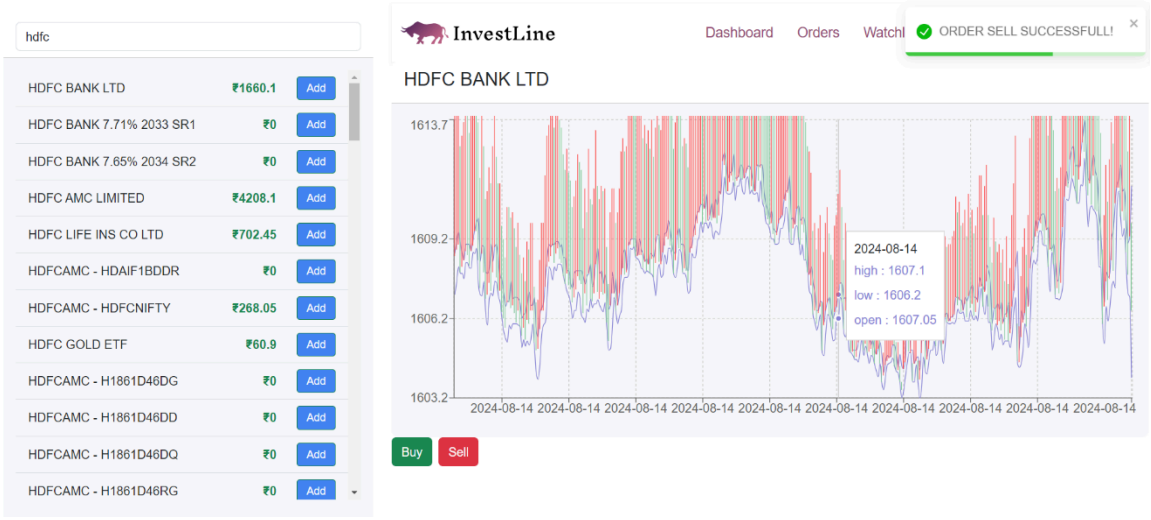


Options to Buy/Sell Stocks:

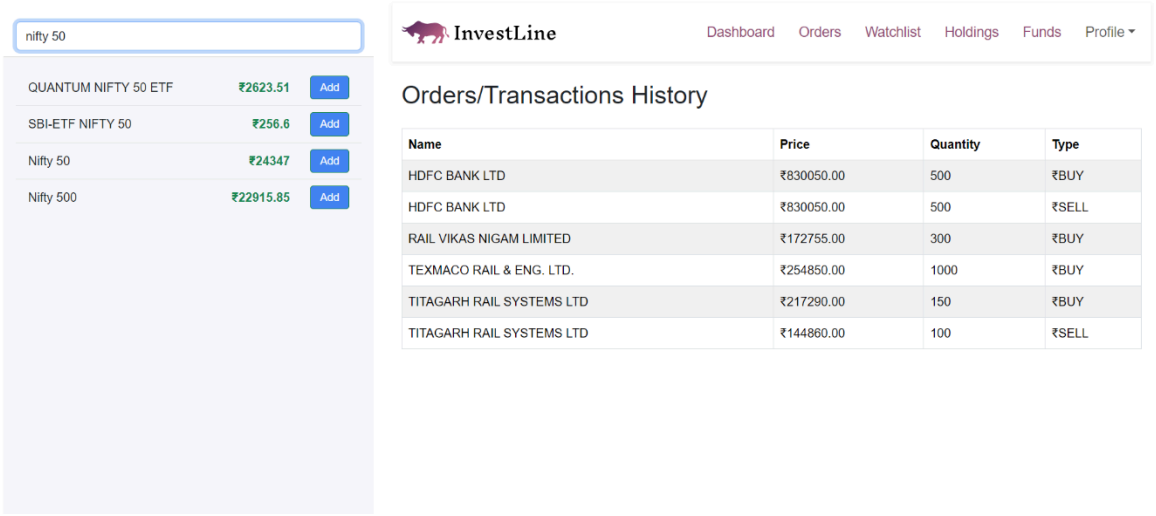

hdfc
HDFC BANK LTD ₹1660.1 Add
HDFC BANK 7.71% 2033 SR1 ₹0 Add
HDFC BANK 7.65% 2034 SR2 ₹0 Add
HDFC AMC LIMITED ₹4208.1 Add
HDFC LIFE INS CO LTD ₹702.45 Add
HDFCAMC - HDAIF1BDDR ₹0 Add
HDFCAMC - HDFCNIFTY ₹268.05 Add
HDFC GOLD ETF ₹60.9 Add
HDFCAMC - H1861D46DG ₹0 Add
HDFCAMC - H1861D46DD ₹0 Add
HDFCAMC - H1861D46DQ ₹0 Add
HDFCAMC - H1861D46RG ₹0 Add
Buy HDFC BANK LTD
Quantity
500
Unit Price: ₹1660.1
Total Price: ₹830050.00
Close
Buy
Orders
Watchlist
Holdings
Funds
Profile
1606.2
1603.2
2024-08-14
Buy
Sell


hdfc
HDFC BANK LTD ₹1660.1 Add
HDFC BANK 7.71% 2033 SR1 ₹0 Add
HDFC BANK 7.65% 2034 SR2 ₹0 Add
HDFC AMC LIMITED ₹4208.1 Add
HDFC LIFE INS CO LTD ₹702.45 Add
HDFCAMC - HDAIF1BDDR ₹0 Add
HDFCAMC - HDFCNIFTY ₹268.05 Add
HDFC GOLD ETF ₹60.9 Add
HDFCAMC - H1861D46DG ₹0 Add
HDFCAMC - H1861D46DD ₹0 Add
HDFCAMC - H1861D46DQ ₹0 Add
HDFCAMC - H1861D46RG ₹0 Add
InvestLine
Dashboard
Orders
ORDER BUY SUCCESSFULL!
HDFC BANK LTD
1613.7
1609.2
1606.2
1603.2
2024-08-14
high : 1611.75
low : 1610.95
open : 1611.6
Buy
Sell

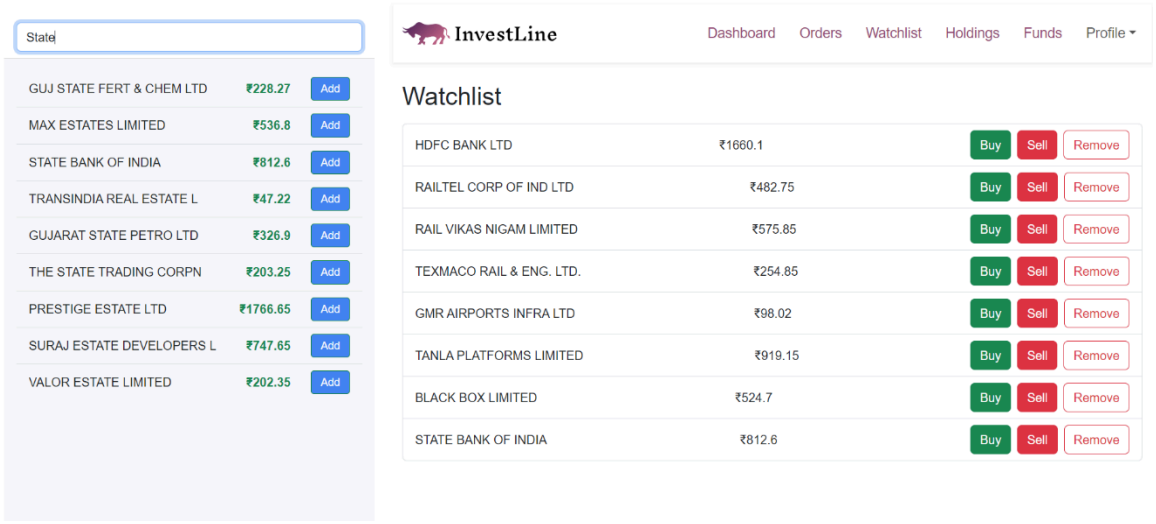

hdfc
HDFC BANK LTD ₹1660.1 Add
HDFC BANK 7.71% 2033 SR1 ₹0 Add
HDFC BANK 7.65% 2034 SR2 ₹0 Add
HDFC AMC LIMITED ₹4208.1 Add
HDFC LIFE INS CO LTD ₹702.45 Add
HDFCAMC - HDAIF1BDDR ₹0 Add
HDFCAMC - HDFCNIFTY ₹268.05 Add
HDFC GOLD ETF ₹60.9 Add
HDFCAMC - H1861D46DG ₹0 Add
HDFCAMC - H1861D46DD ₹0 Add
HDFCAMC - H1861D46DQ ₹0 Add
HDFCAMC - H1861D46RG ₹0 Add
Sell HDFC BANK LTD
Quantity
500
Unit Price: ₹1660.1
Total Price: ₹830050.00
Close
Sell
Orders
Watchlist
Holdings
Funds
Profile
1606.2
1603.2
2024-08-14
Buy
Sell


hdfc
HDFC BANK LTD ₹1660.1 Add
HDFC BANK 7.71% 2033 SR1 ₹0 Add
HDFC BANK 7.65% 2034 SR2 ₹0 Add
HDFC AMC LIMITED ₹4208.1 Add
HDFC LIFE INS CO LTD ₹702.45 Add
HDFCAMC - HDAIF1BDDR ₹0 Add
HDFCAMC - HDFCNIFTY ₹268.05 Add
HDFC GOLD ETF ₹60.9 Add
HDFCAMC - H1861D46DG ₹0 Add
HDFCAMC - H1861D46DD ₹0 Add
HDFCAMC - H1861D46DQ ₹0 Add
HDFCAMC - H1861D46RG ₹0 Add
InvestLine
Dashboard
Orders
Watchl
ORDER SELL SUCCESSFULL!
HDFC BANK LTD
1613.7
1609.2
1606.2
1603.2
2024-08-14
high : 1607.1
low : 1606.2
open : 1607.05
Buy
Sell

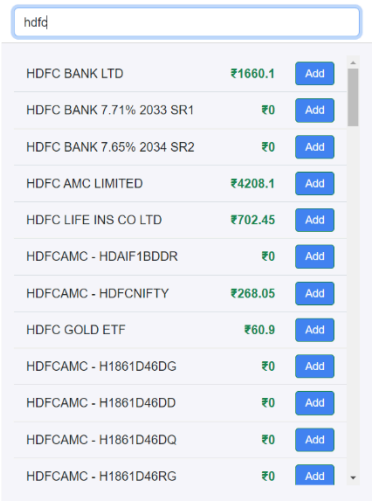
Orders Page:

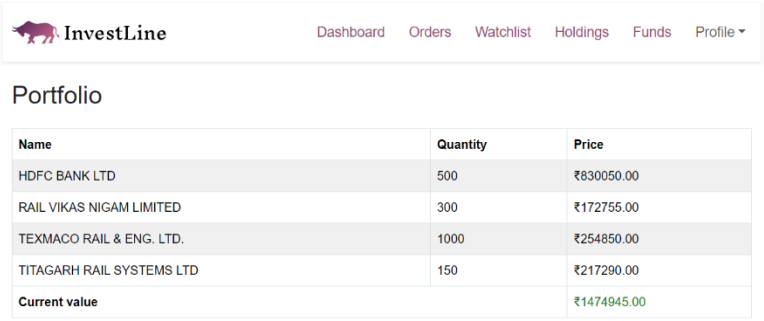


Watchlist Page:

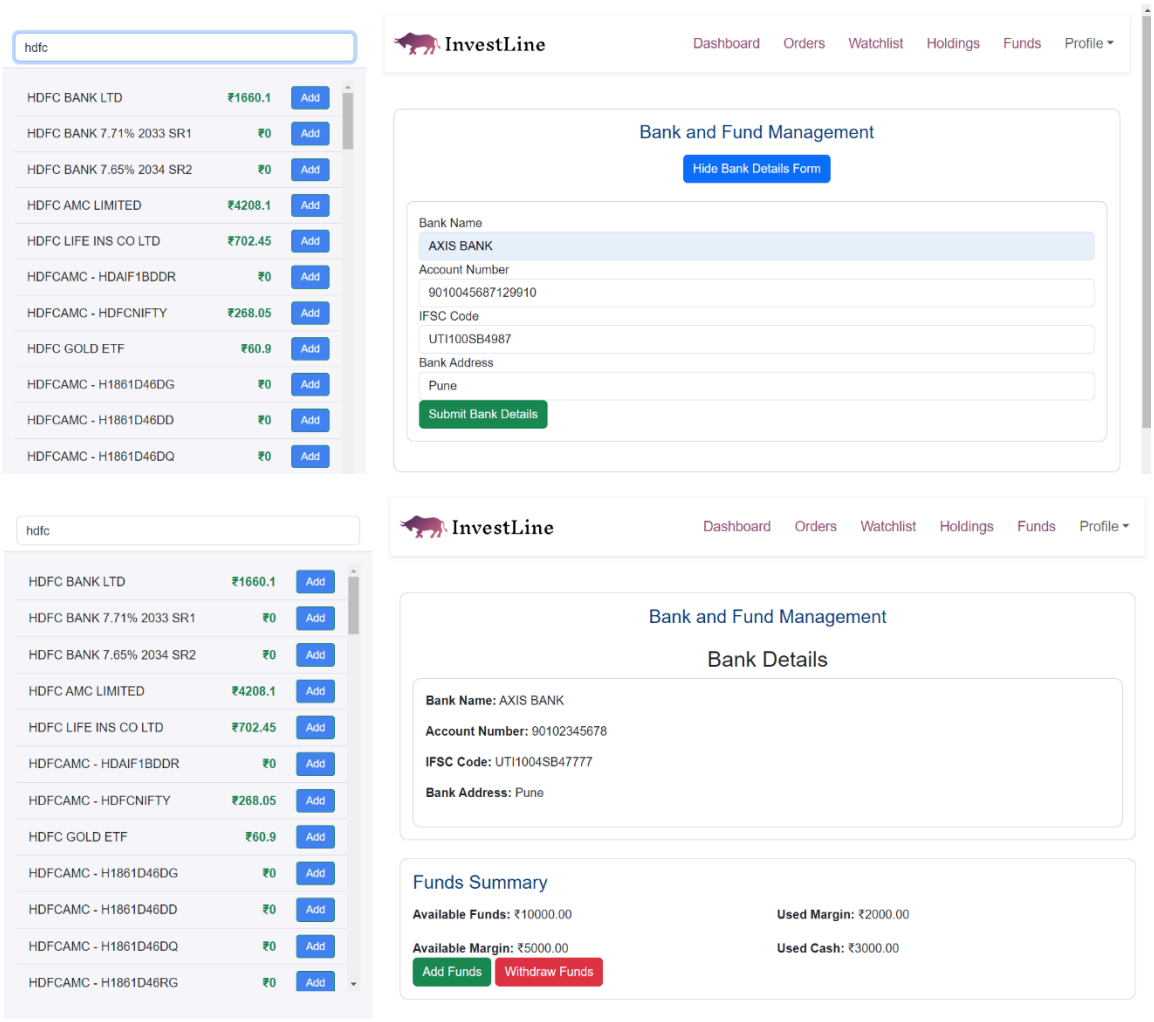


Holdings/Portfolio Page:

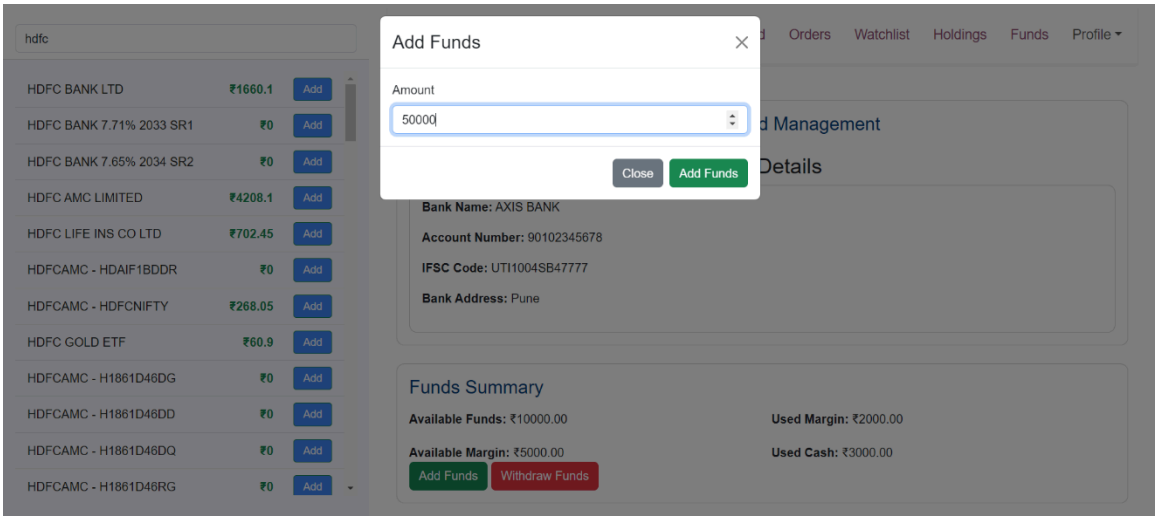


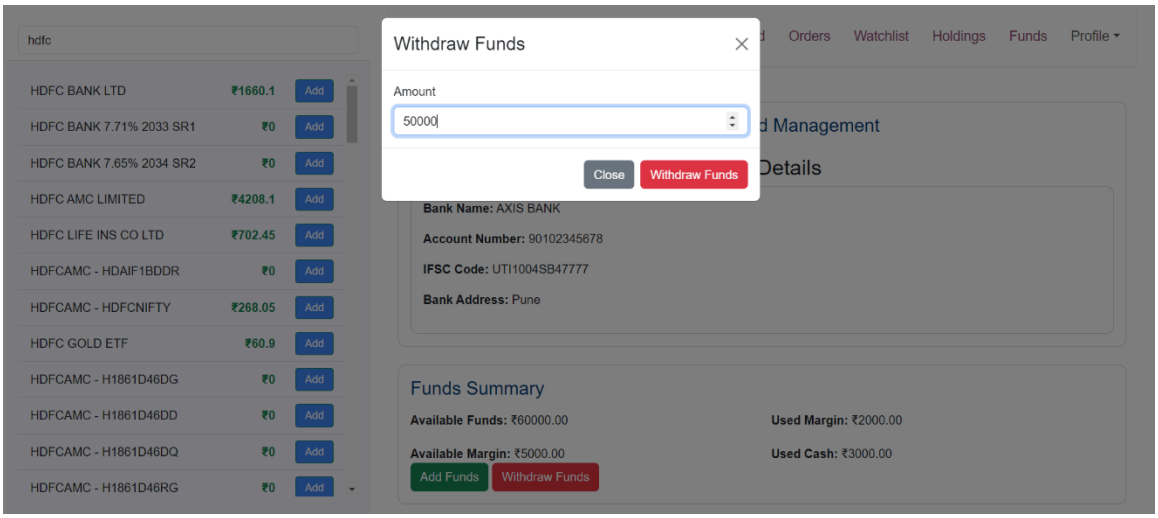


Band Details and Funds Page:

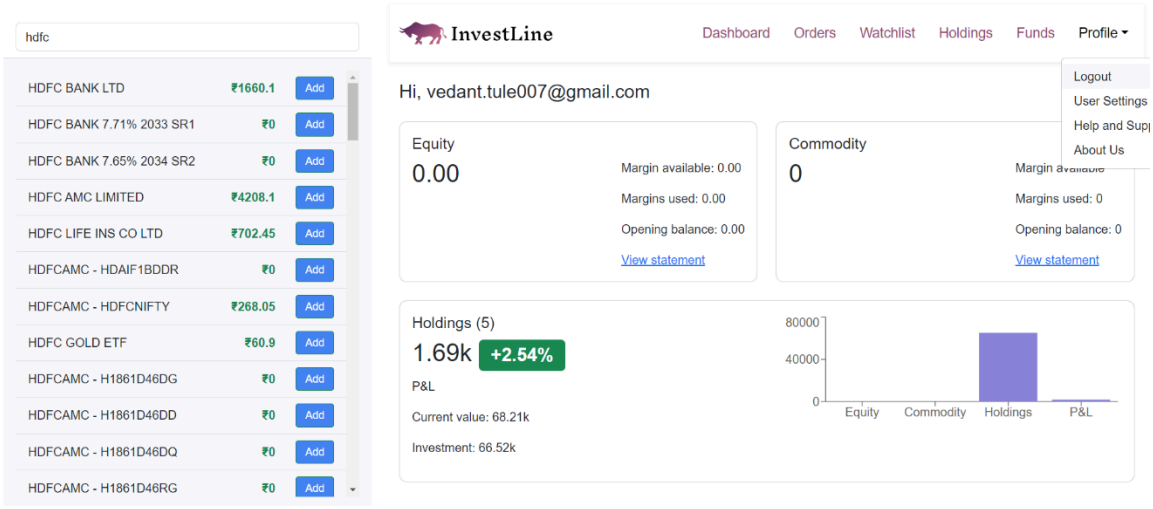


Add and Withdraw Funds:





Profile Page:



7.REFERENCES:

http://www.google.com

https://upstox.com/developer/api-documentation/

https://react.dev/

https://kite.zerodha.com/

https://www.alphavantage.co/