Finance Management System

UCS503 Software Engineering Project Report

Submitted By:
Rehatman Kaur 102203730
Shiven Khare 102203852
Hrishita Dalal 102203863

B.E. Third Year – COE
Group: 3CO20

Submitted To: Ms. Pragya Mishra

Department of Computer Science and Engineering
Thapar Institute of Engineering and Technology

July – Dec 2024 (2425ODDSEM 5th Semester)

TABLE OF CONENTS

Software Bid Form

Group : 3CO20 Dated: 12 Aug, 2024

Team ID (will be assigned by Instructor):

Please enter the names of your Preferred Team Members. :

- You are required to form **a three to four person** teams
- Choose your team members wisely. You will not be allowed to change teams.

Name	Roll No	Project Experience	Programming Language used	Signature
Shiven Khare	102203852	Class XII, Employee Management System Fourth Semester, Library Management System	Python + MySQL PL/SQL	
Hrishita Dalal	102203863	Fourth Semester, Library Management System	PL/SQL	
Rehatman Kaur	102203730	Fourth Semester, Library Management System	PL/SQL	

Programming Language / Environment Experience

List the languages you are most comfortable developing in, **as a team**, in your order of preference. Many of the projects involve Java or C/C++ programming.

1.Python
2.MySQL
3.HTML
4.CSS
5.JavaScript

Choices of Projects:

Please select **4 projects** your team would like to work on, by order of preference: *[Write at-least one paragraph for each choice (motivation, reason for choice, feasibility analysis, etc.)]*

	Project Name	Unique Selling Point
First Choice	Finance Management System	The Finance Management System is designed to simplify financial management for users across all age groups. With an intuitive interface, it offers easy expense management, allowing users to track their daily spending and budget effectively. The system also integrates stock information, helping users stay informed about market trends and make educated investment decisions. Additionally, it serves as a comprehensive educational tool, providing valuable insights into various investment options, including mutual funds, to foster financial literacy. Whether users are seasoned investors or beginners, the system empowers them to take control of their finances with confidence.
Second Choice	Remote Work Collaboration Tool	The Remote Work Collaboration Tool is tailored for teams working in a remote or hybrid environment, enhancing their ability to collaborate seamlessly. It provides an integrated platform where team members can communicate, share files, and manage projects efficiently. The tool includes features like task

		assignment, real-time updates, and project tracking, ensuring that all team members are aligned and on track. By fostering a more organized and connected work environment, the tool reduces the challenges of remote collaboration, enabling teams to maintain productivity and achieve their goals, regardless of physical location.
Third Choice	Library Management System	Comprehensive functionalities for reporting detailed information on books, members, and publishers, enhancing staff's ability to access and utilize data efficiently; automated dues calculations through tailored procedures and functions streamline financial transactions, promoting timely returns and maintaining financial discipline among borrowers.
Fourth Choice	Health Tracker	The Health Tracker is a powerful tool for individuals looking to maintain a balanced and healthy lifestyle. It enables users to monitor their diet by tracking daily calorie intake and nutritional values, providing a clear picture of their dietary habits. The system also offers personalized exercise recommendations based on the user's diet, ensuring that their fitness routine complements their nutritional needs.

Additional Remarks/ Inputs

Please tell us about any other factors that we should take into consideration (e.g., if you really would like to work on a project for some particularly convincing reason).

As a team, we are particularly excited about the opportunity to work on the Finance Management System project. Our diverse skill sets in Computer Engineering, programming, and database management make us well-suited to tackle the challenges this project presents. We are united by a shared interest in developing innovative financial management solutions that are accessible to various age groups and are keen to apply our knowledge to enhance people's understanding of investments and mutual funds. Working on this project will not only allow us to grow professionally but also contribute meaningfully to simplifying financial decision-making for a wider audience.

Feasibility Study Report for Finance Management System

Project Overview

The Finance Management System is a project developed by a student team as part of a college Software Engineering course. The system aims to provide users with a comprehensive platform to manage their finances, track stock prices, and access information about the stock market and other investment options. Additionally, the project considers integrating a chatbot feature through WhatsApp to enhance user interaction and support.

Project Objectives

1. **Manage Expenditure**: Provide users with tools to record, track, and analyse their expenditures.
2. **Stock Prices**: Display stock prices for the last five days using the yfinance library to keep users updated on market trends.
3. **Stock Market Information and Investment Options**: Offer a static web page that discusses investment concepts, stock information, and other investment options.
4. **WhatsApp Chatbot Integration**: Explore the feasibility of integrating a chatbot feature via WhatsApp for user support and interaction.

Technical Feasibility

Programming Languages and Technologies Involved

• Python: Backend development, data processing, API integration (using yfinance), and logic implementation.
• HTML, CSS, JavaScript: Frontend development to create a responsive and user-friendly interface, including the static web page.
• MySQL: Database management for storing user data, expenditure records, and other relevant information.

1. **Manage Expenditure**
    - **Technology Stack**: Python for backend logic and data processing, MySQL for storing expenditure data, and HTML/CSS/JS for the frontend interface.
    - **Development Complexity**: Moderate. Tasks include creating a secure and efficient database, developing an intuitive user interface, and implementing data processing and analytics features.
    - **Resources Required**: Team members with skills in Python, MySQL, and frontend technologies (HTML, CSS, JS) will contribute to this component.

2. **Stock Prices (Last 5 Days)**
    - **Technology Stack**: Python for integrating the yfinance library to fetch stock prices, MySQL for storing fetched data, and HTML/CSS/JS for displaying the data on the frontend.
    - **Development Complexity**: Low to Moderate. The primary task is ensuring real-time or near-real-time data availability and displaying it in a user-friendly format.
    - **Resources Required**: Team members should have experience with Python, yfinance, MySQL, and frontend development to successfully implement this feature.

3. **Stock Market Information and Investment Options (Static Web Page)**
    - **Technology Stack**: HTML/CSS/JS for creating a static web page that discusses investment concepts, stock information, and other investment options.
    - **Development Complexity**: Low. The main task is to design an informative and visually appealing static web page.
    - **Resources Required**: Frontend developers within the team with expertise in HTML, CSS, and JavaScript, as well as content creators to develop the information presented on the page.
    -

4. **WhatsApp Chatbot Integration**
    - **Technology Stack**: Python for backend chatbot logic, MySQL for storing conversation history and user data, and integration with WhatsApp Business API via a third-party service (e.g., Twilio).
    - **Development Complexity**: High. Integrating WhatsApp requires compliance with WhatsApp's API policies, managing user conversations, and ensuring data privacy and security.
    - **Resources Required**: Team members with expertise in Python, API integration, chatbot development, and data security will focus on this advanced feature.

Operational Feasibility

• **User Interaction**: The system is designed with user-friendliness in mind. The expenditure management, stock price tracking, and static information page features are expected to be intuitive, requiring minimal training. The chatbot integration could further enhance the user experience by providing instant support and information.

• **Maintenance**: Regular updates will be required to keep stock market information current and ensure that the chatbot remains functional and relevant.

The static web page will require minimal updates unless the investment content needs revision.
• **Scalability**: The system is designed to be scalable, allowing for the addition of new features or the expansion of existing ones as user demand grows.

Economic Feasibility
• **Development Costs**: As a college project, the primary costs will be the time and effort invested by the student team. The team will account for any expenses related to server hosting, API subscriptions, or third-party services.
• **Operational Costs**: Ongoing costs will include server hosting, API subscriptions, and maintenance.

Legal Feasibility
• **Data Privacy**: The system must comply with data privacy laws, particularly when handling financial data and user interactions via WhatsApp. Implementing strong data encryption and secure storage practices is essential.
• **API Compliance**: Ensure compliance with the terms of service of any third-party APIs used, such as those from yfinance and WhatsApp.

Conclusion
The Finance Management System is technically feasible for development by a student team, with a moderate to high level of complexity. The inclusion of a WhatsApp chatbot feature adds significant value but also increases the development complexity. The static web page discussing investment concepts and options is straightforward and low-cost to implement. With a dedicated team familiar with Python, HTML/CSS/JS, MySQL, and yfinance, the project can be successfully implemented, providing a valuable tool for users to manage their finances and stay informed about the stock market.

Recommendations

1. **Proceed with Development**: Start with the core features (expenditure management, stock price tracking using yfinance, and the static information page) before moving on to the more complex WhatsApp chatbot integration.
2. **Pilot Testing**: Conduct pilot testing of the WhatsApp chatbot to evaluate user engagement and system performance before a full-scale launch.
3. **Legal Consultation**: Engage with your college's legal or ethics advisor to ensure compliance with data privacy regulations and third-party API terms of service.

PROJECT OVERVIEW

The Finance Management System is a dynamic platform developed to help users manage their finances efficiently. It provides tools for expenditure tracking, stock price monitoring, and educational content on various investment options. The system is designed to be scalable, offering the ability to add new features as needed.

Key Features:

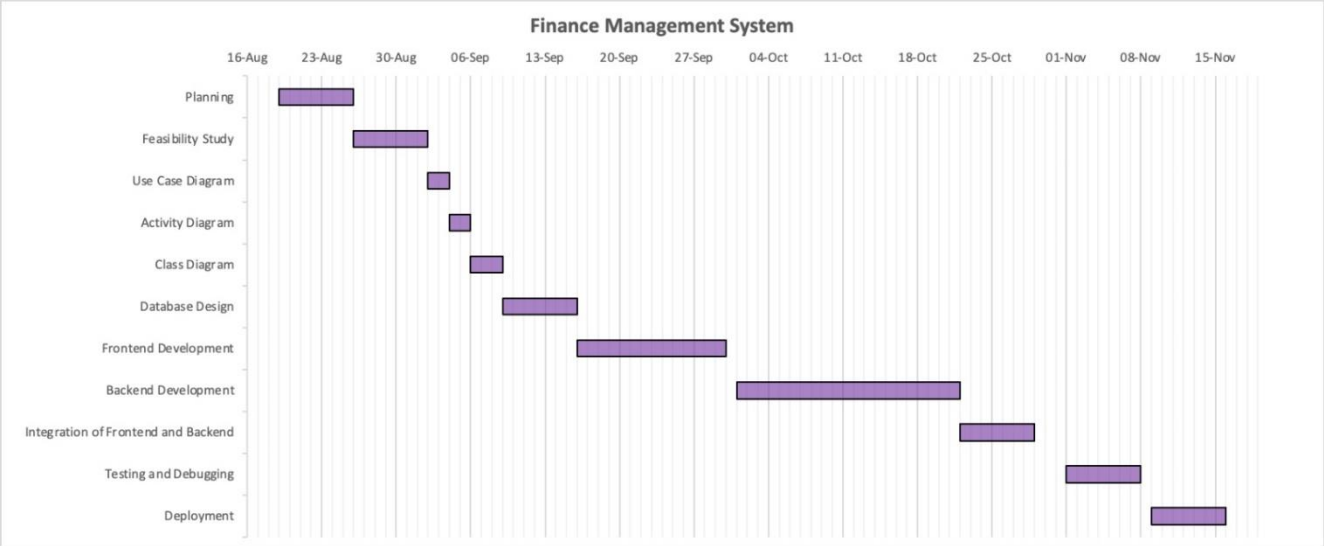
Expenditure Management:
Users can record, track, and analyze their daily, monthly, or annual expenses. This feature offers insights into spending habits, helping users manage and optimize their financial decisions. The user will be able to see the expenditure for various categories like food , transport, rent, EMIs , education and create any custom fields as per need.

Stock Price Tracking with Graphs:
The system integrates the yfinance library to fetch historical stock price data for a user-specified number of years (n-years). Users can view these stock trends in visual graph formats, helping them analyze long-term market performance and make informed investment decisions.

Investment Information and Options:
A static web page offers comprehensive information on various investment options such as stocks, mutual funds, fixed deposits, and other financial instruments. This content helps users understand the different avenues for growing their wealth.

WhatsApp Chatbot Integration :
The system will also be integrating a chatbot via WhatsApp for user interaction and support. The chatbot will provide personalized reminders, assist users with financial tracking. The chatbot eases the process of data entry on daily basis for the user.

GANTT CHART



USE CASES

Expenditure Management

- **Record Expenses**: Users can log their expenses into predefined or custom categories for easy tracking.
- **Track Expenses**: The system allows users to view their expense reports for selected periods (daily, monthly, yearly).
- **Analyse Spending**: Insights into spending habits help users identify high-spending areas and optimize their budget.
- **Custom Fields Creation**: Users can create new categories to track specific expenses according to their needs.

Stock Price Tracking with Graphs

- **Fetch Stock Data**: Users can retrieve historical stock price data for specified symbols and time periods using yfinance.
- **View Stock Trends**: Stock performance is visualized through graphs, helping users analyse market trends.

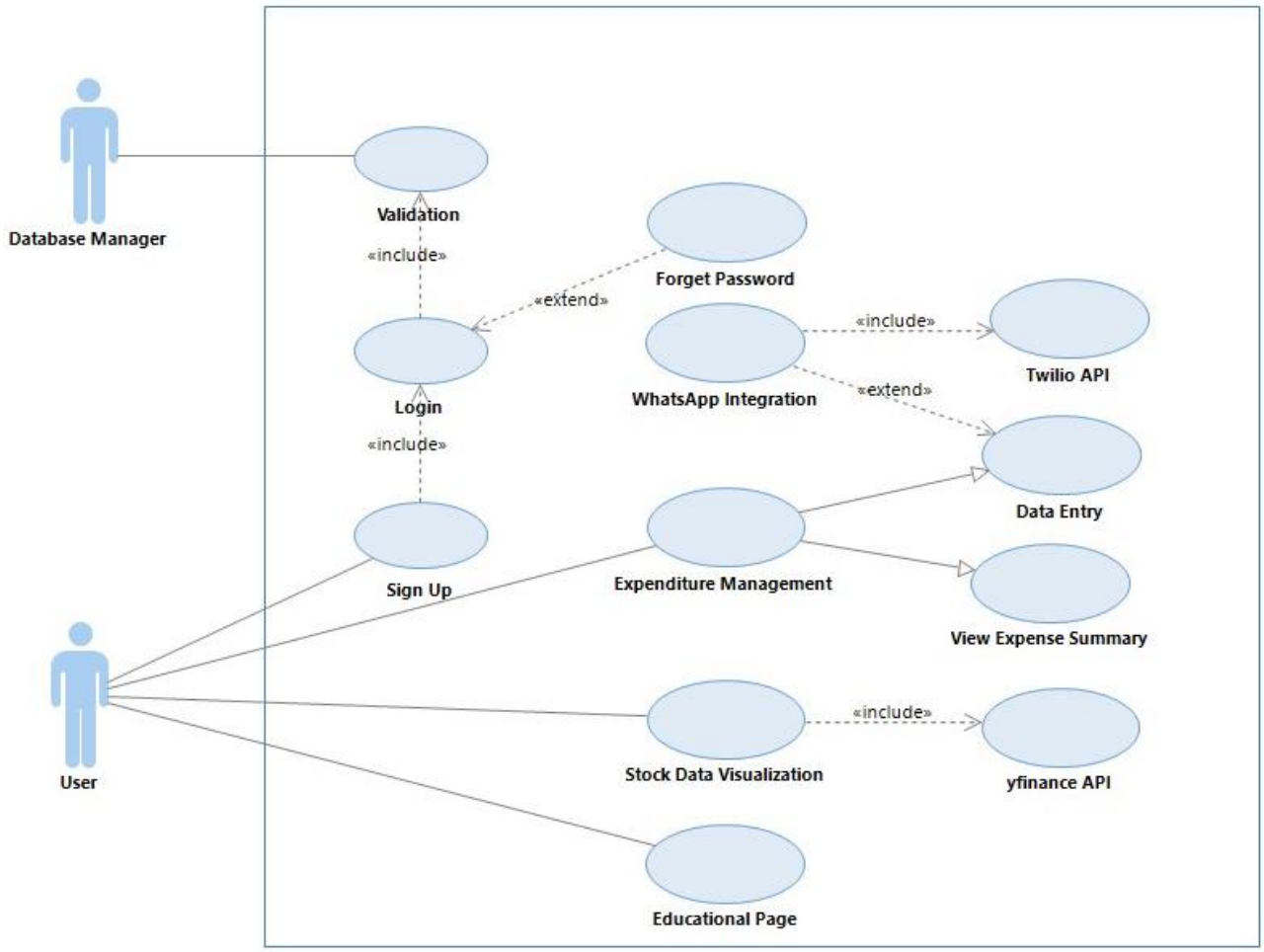
Investment Information and Options

- **Access Investment Information**: A static web page provides comprehensive details on various investment avenues like stocks, mutual funds, and FDs.
- **Understand Investment Strategies**: Users can explore benefits, risks, and strategies for each type of investment instrument.

WhatsApp Chatbot Integration

- **Daily Data Entry**: The chatbot allows users to log expenses conveniently through WhatsApp.
- **Reminders for Financial Tracking**: Personalized WhatsApp reminders help users maintain consistent financial tracking.
- **Receive Financial Insights**: Users can request financial summaries or specific insights directly from the chatbot.

USE CASE DIAGRAM



USE CASE TEMPLATE

Use Case Template

1. Use Case Title	Stock Data Visualization
2. Abbreviated Title	Stock Data Visualization
3. Use Case ID	UC004
4. Actors	User

5. Description:
This use case involves the visualization of stock performance over a period of chosen number of years in graphical form.

5.1. Pre Conditions
- The user must be signed up.
- The user's device must have a stable internet connection.

5.2. Task Sequence
1. Log In:
   - Trigger: The user launches the website.
   - Action: The website displays the main dashboard.
2. Stock Data Visualization:
   - Trigger: The user selects the Stock Data option on the dashboard and chooses the name of the stock from the drop-down menu and the number of years to see stock performance.
   - Action: The user gets the stock performance in graphical form over a period of chosen number of years.

5.3. Post Conditions
User can see the stock performance in graphical manner for the chosen stock.

6. Modification History: N/A

7. Author: Rehatman Kaur, Shiven Khare, Hrishita Dalal

Flow of Events

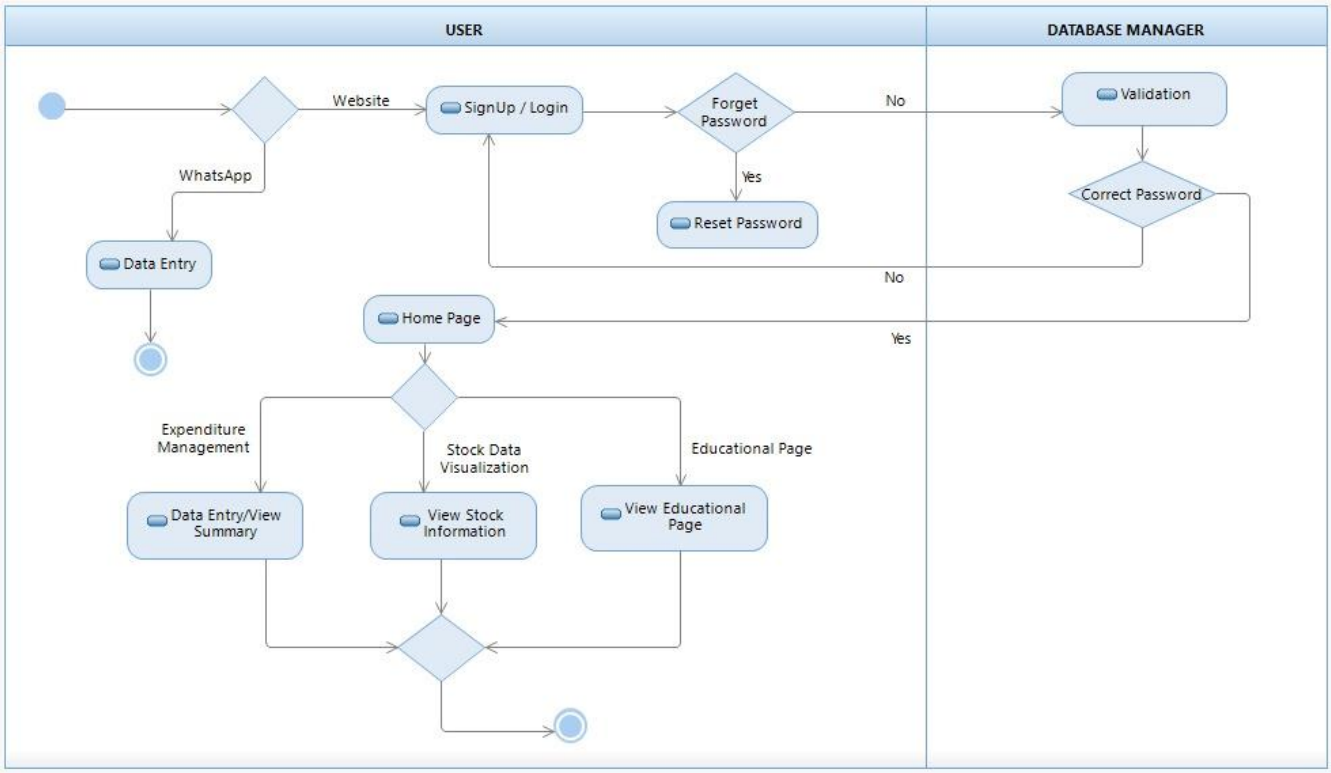
1. User Acquisition
    - New User
      SR: The user is asked to sign up.
      AA: The user enters a valid phone number.
      SR: OTP sent to the user for verification.
      AA: The user enters the OTP.
      SR: The user is directed to home page.
    - Existing User
      SR: The user is asked to log in.
      AA: The user logs in.
      SR: The user is directed to home page.

2. Expenditure Management
    - Logged in on Website
      AA: The user chooses Expenditure Management from the home page.
      SR: Choose to enter data or view expense summary.
      AA: The user either enters data by giving input in the various fields or create custom fields; or view the expense summary.
    - WhatsApp
      SR: The chatbot will send a notification to enter the data.
      AA: The user enters the data.

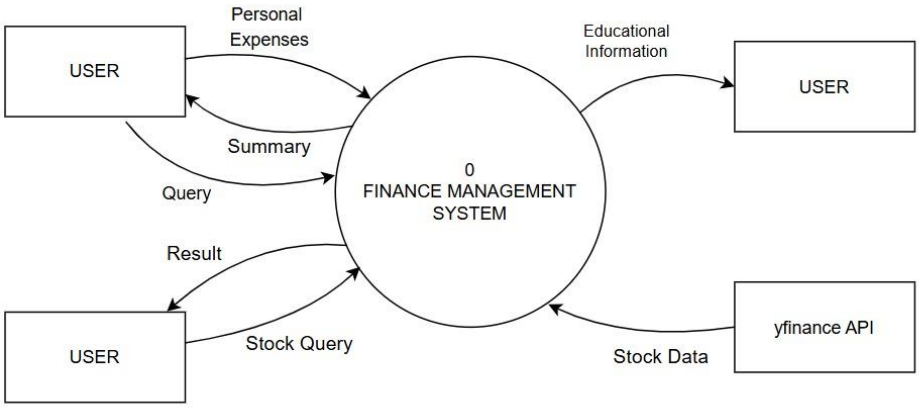
3. Stock Data Visualization
      AA: The user chooses the Stock Data Visualization option from the Dashboard.
      SR: The user is prompted to choose the stock name and the number of years from drop down menu for which the user wants to see stock performance.
      AA: The user chooses the stock name and number of years from the drop-down menu shown.
      SA: The stock performance is represented graphically to the user.

4. Educational Page
      AA: The user chooses the Educational Information page.
      SA: The information about different investment options is displayed on screen.

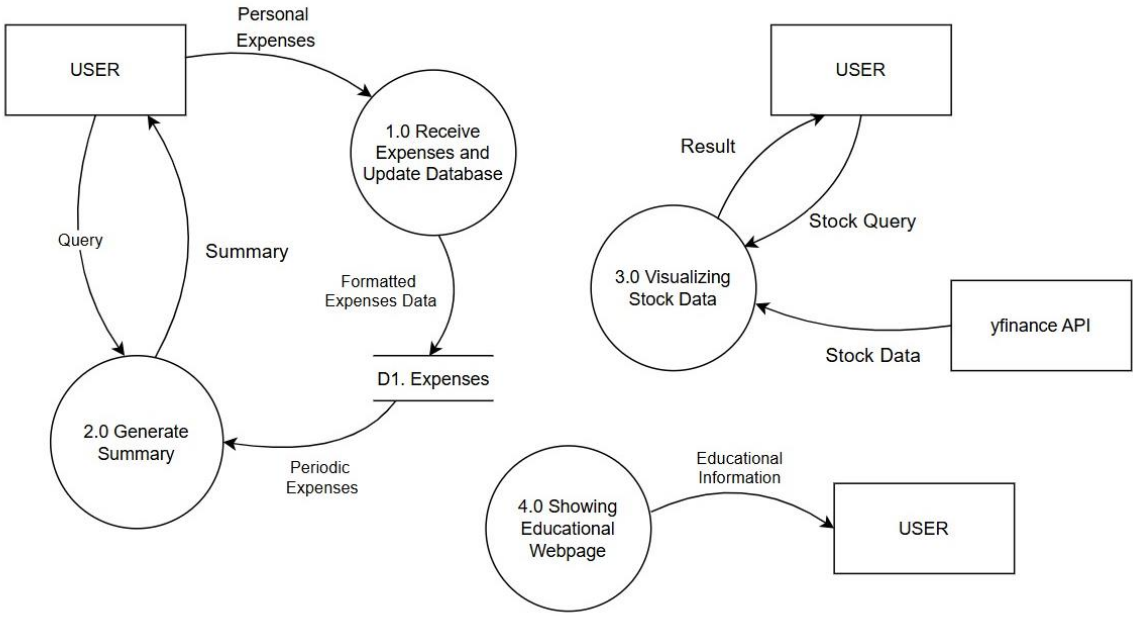
ACTIVITY DIAGRAM



DATA FLOW DIAGRAM



Level 0



Level 1

Table of Contents (SRS)

SRS DOCUMENT IEEE FORMAT

1.Introduction

1.1 Purpose of this Document

This Software Requirements Specification (SRS) document outlines the detailed requirements and design specifications of the Finance Management System, a web-based tool designed to help users manage personal finances efficiently. The document includes all the information required by the developers, testers, and stakeholders to understand how the system will work. It describes the functional and non-functional requirements, the system's constraints, dependencies, and interactions with external components. The goal is to create a clear roadmap to ensure the successful implementation of the project.

1.2 Scope of the Development Project

The Finance Management System is designed to help users log and track their expenditures, visualize stock data from the National Stock Exchange (NSE), and learn about various investment options such as mutual funds (MFs) and fixed deposits (FDs). Additionally, the system integrates with WhatsApp to remind users to log their daily expenses and send a thank you message after they input their expenses.

The main features of the project include:

1. **Expenditure Management:** Users will input daily expenses, and the system will store and categorize them. Users can review their expenditure history in summarized views (daily, weekly, or monthly).
2. **Stock Data Visualization:** Users will be able to enter the name of an NSE stock, and the system will retrieve and display stock performance data in a graphical format.
3. **Educational Page:** This page will serve as a knowledge hub, offering educational content about different types of investments such as stocks, MFs, and FDs.
4. **WhatsApp Integration:** Using the Twilio API, the system will send users daily reminder messages to input their expenses and a thank you message once the data is submitted.

The system will be built using Python, JavaScript, HTML, CSS, and MySQL, along with libraries like yfinance, matplotlib, and Twilio for API integration.

1.3 Definitions, Abbreviations, and Acronyms

Table 1 : Acronyms , their Full Forms and Definitions

S.No.	Acronym	Full Form	Definition
1.	NSE	National Stock Exchange	Indian stock exchange where stock data will be sourced.
2.	MFs	Mutual Fund	A type of investment vehicle made up of a pool of money collected from many investors to invest in securities.
3.	FDs	Fixed Deposits	Fixed Deposits – a financial instrument provided by banks or NBFCs that provides investors with a higher rate of interest.
4.	SRS	Software Requirements Specification	The document that outlines the detailed requirements for a software system.
5.	API	Application Programming Interface	It allows interaction between software components.

1.4 References

[1]NSE.Link:https://www.investopedia.com/terms/n/national_stock_exchange.asp#:~:text=The%20National%20Stock%20Exchange%20of%20India%20(NSE)%20is%20India's%20largest,as%20measured%20by%20market%20capitalization.

[2]yfinance. Link: https://pypi.org/project/yfinance/

[3]Matplotlib. Link: https://matplotlib.org/

1.5 Overview

The Overview section briefly explains what the following sections will cover:

Section 2 provides a general description of the system, including its purpose, product functions, and key features. Section 3 outlines the specific functional and non-functional requirements for the system. Section 4 documents any changes made during the development.
Section 5 lists the document approvers.

2. Overall Description

2.1 Product Perspective

The Finance Management System is a standalone web-based application that users will access through a browser. The system will be built using Python and Flask on the backend, with MySQL as the database for storing expenses and user data. JavaScript, HTML, and CSS will be used to develop the frontend for a responsive user interface.

Users will be able to log expenses, view stock data, and read educational content about investment options. Additionally, the system will send WhatsApp messages to users reminding them to input daily expenses. After they log their expenses, the system will send a thank you message. This messaging functionality will be powered by the Twilio API.

2.2 Product Functions

The product will support the following functionalities:

Manage Expenditures: Users will log their expenses via a simple input form, and the data will be stored in the database. Users will be able to categorize their expenses (e.g., groceries, entertainment) and view reports that summarize their spending habits over different periods (daily, weekly, monthly).

Stock Data Visualization: Users will search for stock data by entering the NSE stock name. The system will retrieve data from yfinance, a financial data API, and display stock performance as graphs (e.g., line charts, bar graphs) using matplotlib.

Educational Page: The system will feature an educational page that provides easy-to-understand content about different investment options. This will include descriptions of stocks, mutual funds, and fixed deposits, as well as guidance on the risks and rewards associated with each investment type.

WhatsApp Integration: Users will receive a daily reminder message via WhatsApp prompting them to input their expenses. After successfully inputting the expenses, the system will send a thank you message. The integration is powered by the Twilio API.

2.3 User Characteristics

The primary users of the system are individuals looking to manage personal finances. The expected user characteristics include:

Basic computer literacy: Users should be familiar with how to use a web browser, input data into forms, and navigate through different pages of the website.
Basic understanding of personal finance: Users should have a basic understanding of financial terms such as expenditures, stocks, mutual funds, and fixed deposits. However, the system will aim to educate users through its educational pages.

2.4 General Constraints, Assumptions, and Dependencies

This section outlines the constraints and assumptions that affect the system:

Constraints:
The system will require internet access for API calls to retrieve stock data (yfinance) and send WhatsApp messages (Twilio).
Users need a valid phone number to receive WhatsApp notifications.

Assumptions:
The system assumes users have basic knowledge of finance and are capable of interacting with the website to input expenses and search for stock data.
The system will be able to handle moderate traffic levels (up to 100 concurrent users) without performance issues.

Dependencies:
The system depends on external APIs such as yfinance for fetching stock data and Twilio for sending WhatsApp messages.

2.5 Apportioning of Requirements

The system will be developed in multiple phases:

Phase 1: Basic features such as expenditure management and stock data visualization will be developed.
Phase 2: The educational content and WhatsApp integration will be implemented.
Phase 3: Comprehensive testing and deployment of the system.

3. Specific Requirements

3.1 External Interface Requirements

User Interface: A simple, responsive interface that includes an expenditure logging form, a search bar for stock data, and access to educational content. The website should be accessible on desktops and mobile devices.

API Integration: The system will interface with the yfinance API to retrieve stock data and Twilio API to send WhatsApp messages (reminders and thank you messages).

3.2 Detailed Description of Functional Requirements

Table 2 : Functional Requirements for Expenditure Management

Purpose	Expenditure Management
Inputs	Users log expenditures, categorize them, and track expenses.
Processing	The system categorizes and stores expenses in the MySQL database.
Outputs	Users can view daily, weekly, and monthly summaries of their expenses.

Table 3 : Functional Requirements for Stock Data Visualization

Purpose	Stock Data Visualization
Inputs	Users enter the NSE stock name to retrieve data.
Processing	The system fetches data from yfinance and visualizes it using matplotlib.
Outputs	A graph of stock performance is displayed on the dashboard.

Table 4 : Functional Requirements for Educational Page

Purpose	Educational Page
Inputs	Users navigate to the educational section to view investment information.
Processing	The page serves static content, educating users on different investment options.
Outputs	Users view content on stocks, MFs, FDs, and other financial instruments.

Table 5 : Functional Requirements for WhatsApp Integration

Purpose	WhatsApp Integration
Inputs	Users set up WhatsApp notifications by providing their phone numbers.
Processing	Twilio API sends a daily reminder to input expenses. After users input their expenses, the system sends a thank you message via WhatsApp.
Outputs	Users receive reminder to input their expenses and thank you messages.

3.3 Performance Requirements

The system should load all pages (dashboard, expenditure logging, stock data, etc.) within 2-3 seconds for optimal user experience.
Data retrieval for stock information and graph rendering should be completed within 5 seconds.

3.4 Logical Database Requirements

The system will use a MySQL database to store:

Expense Data: Logs of user expenses, including amount, date, and category.

User Information: Data related to user profiles and WhatsApp preferences (phone number).
The database schema will ensure secure storage and efficient retrieval of data.

3.5 Quality Attributes

Usability: The system will have an intuitive interface that is easy to navigate, even for users with limited technical skills.
Reliability: The system will be designed to support up to 100 concurrent users without performance degradation.
Security: Sensitive user data, such as expense details and phone numbers, will be encrypted to ensure privacy and protection from unauthorized access.

3.6 Other Requirements

At this point, there are no additional requirements for the system.

4. Document Approvers

SRS for Finance Management System approved by :

Ms. Pragya Mishra
Project Mentor
Date:

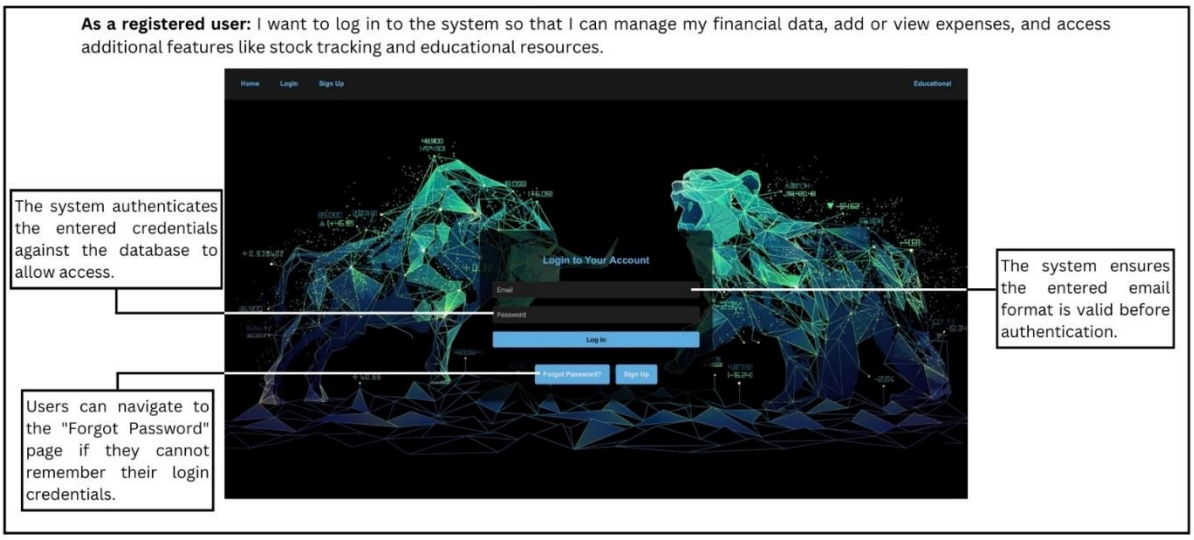
User Stories and Story Cards

User Stories for Finance Management System
User Stories Description:
As a customer:
I want to manage my daily expenses, view spending summaries, and analyze my expenditure trends to better manage my finances, get stock details and information regarding investments.

As an administrator:
I want to ensure users can log their expenses, view visual summaries, and securely access their accounts with reliable authentication and password reset functionality.

User Story: Front of Card

As a registered user: I want to log in to the system so that I can manage my financial data, add or view expenses, and access additional features like stock tracking and educational resources.

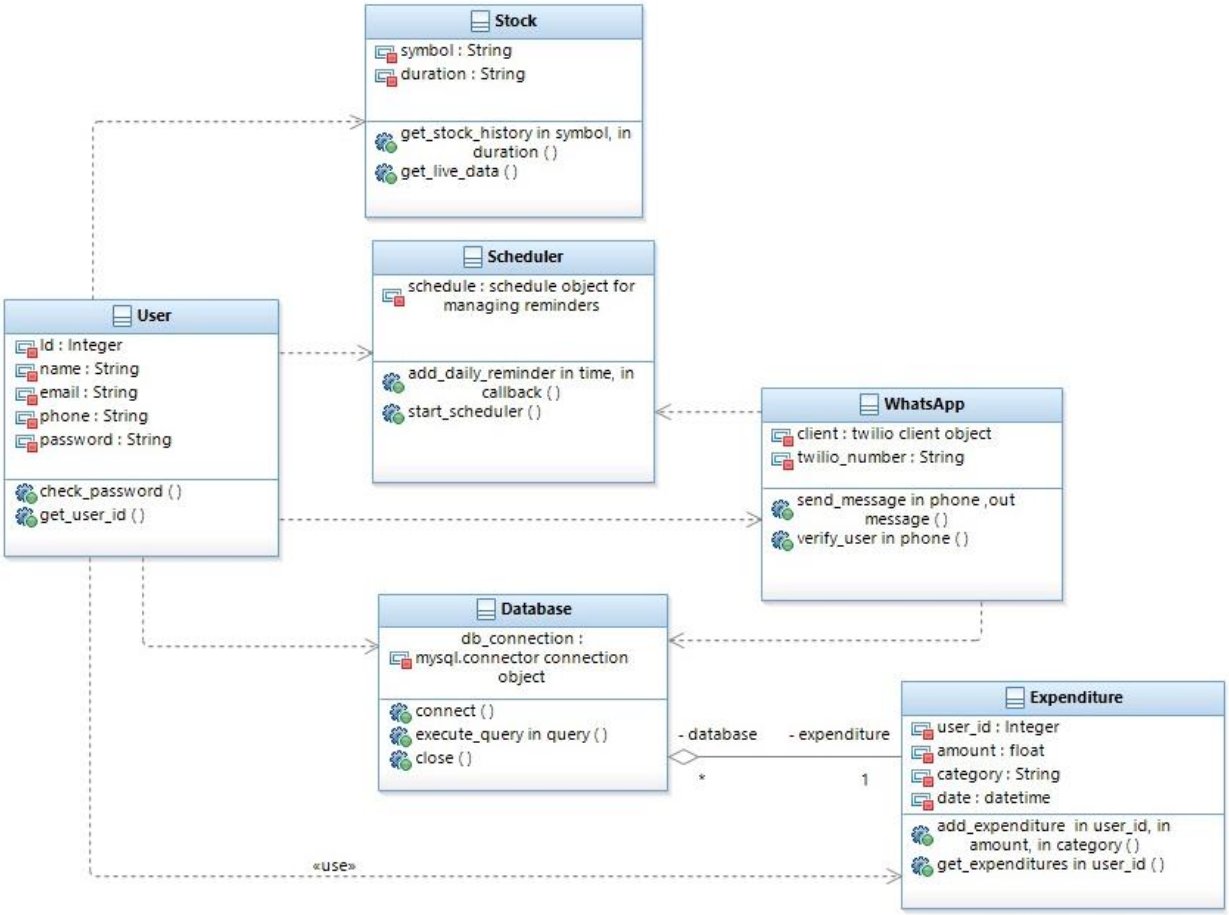


User Story: Back of Card

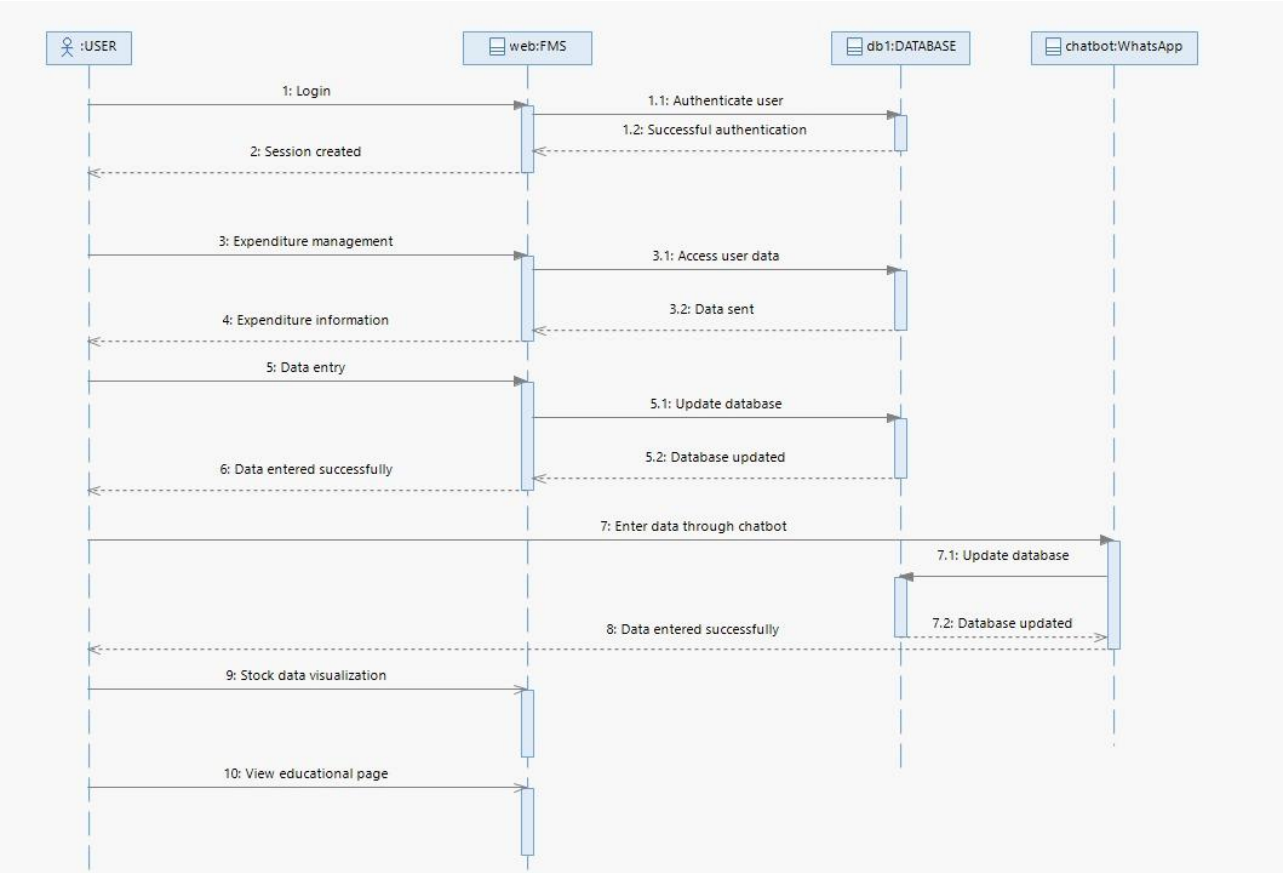
Confirmation:

- **Success:** Upon successful login, the user is redirected to the dashboard where they can manage financial data, add or view expenses, track stocks, and access educational resources.
- **Failure:**
  1. "Fill out the fields"
  2. "Enter an email address"
  3. "Invalid Email or Password"

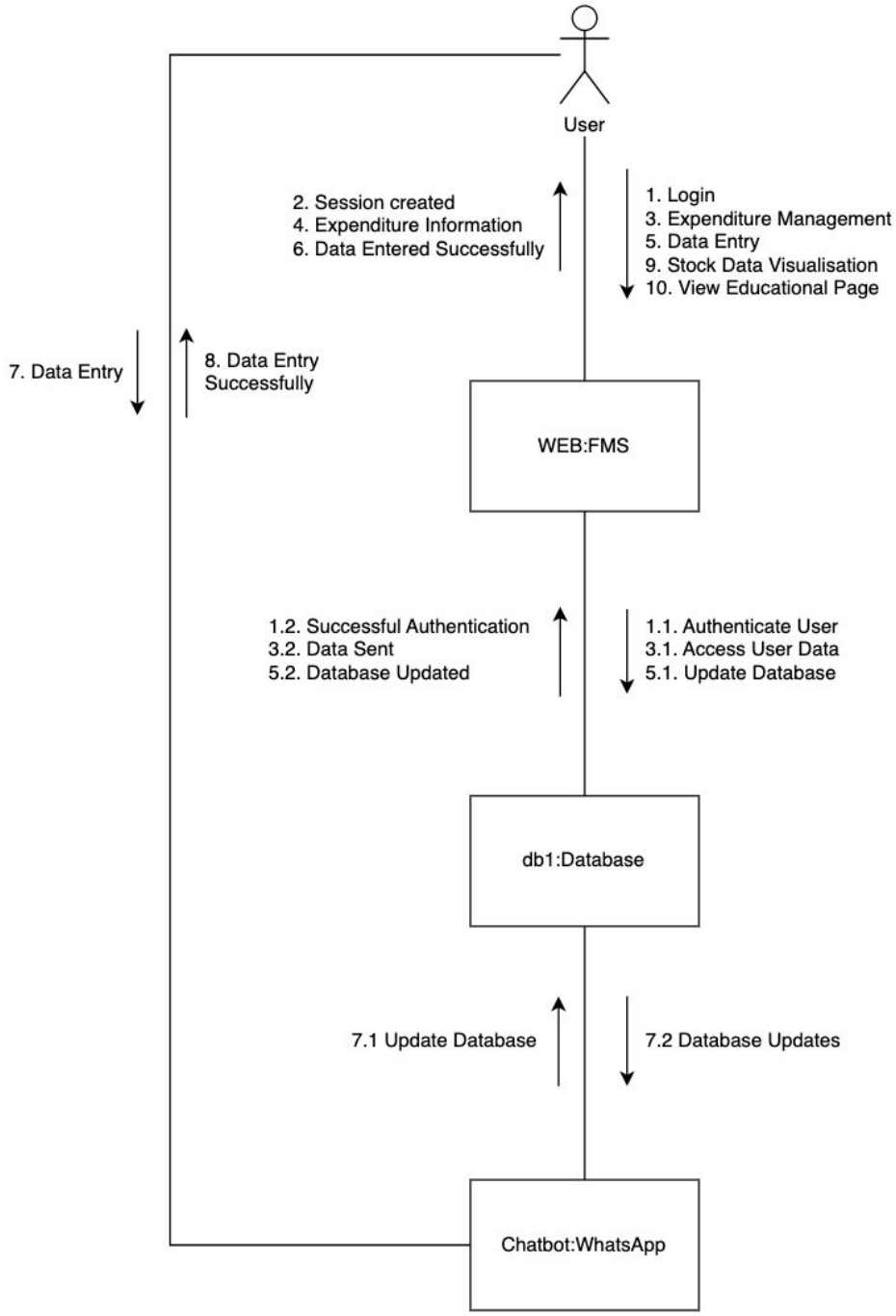
CLASS DIAGRAM



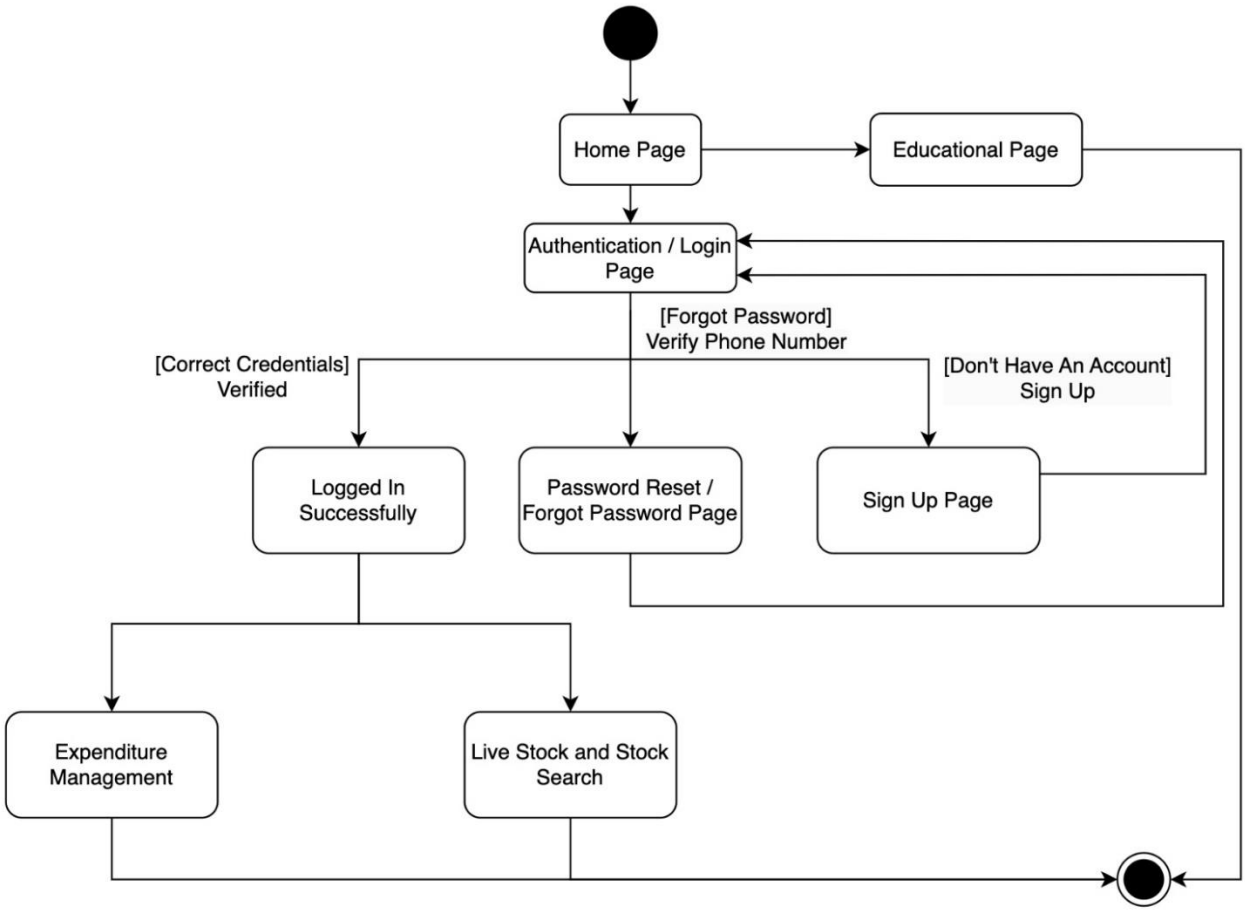
SEQUENCE DIAGRAM



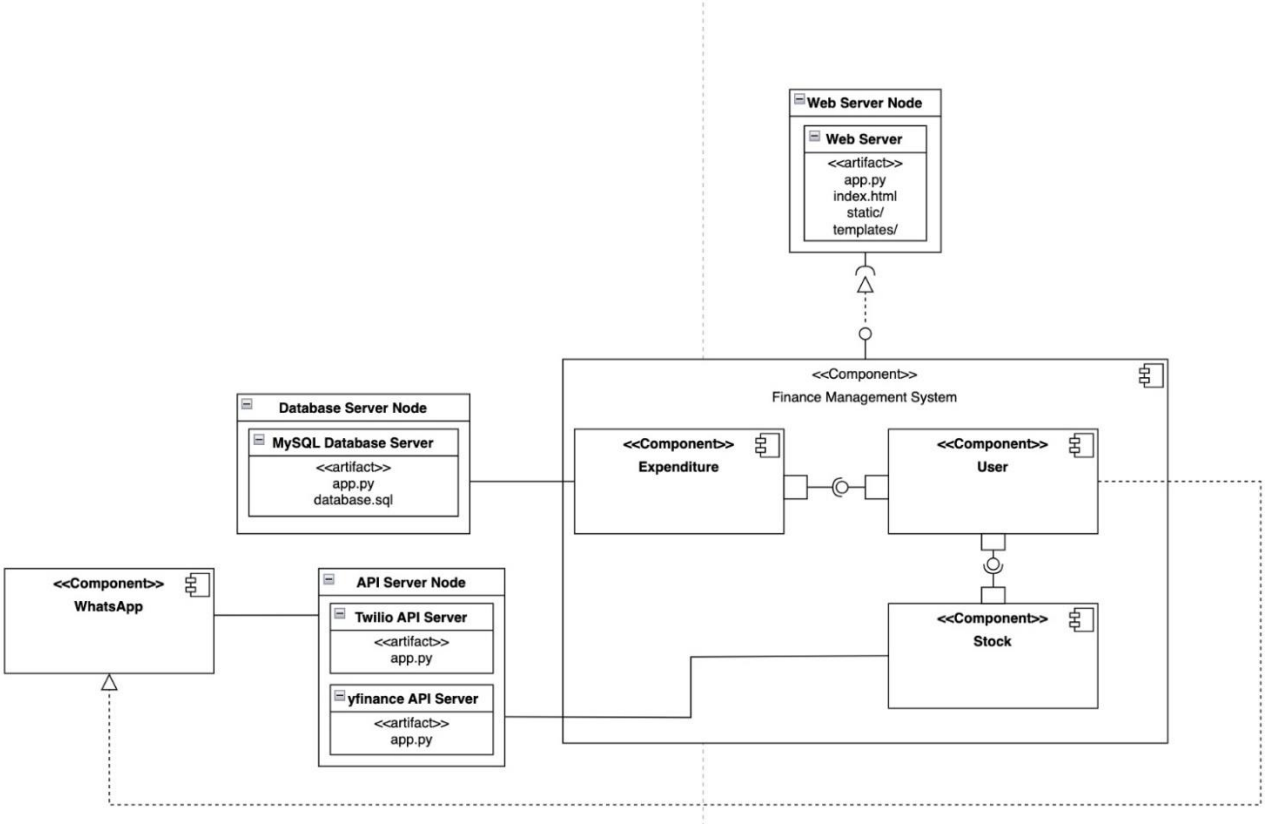
COLLABRATION DIAGRAM



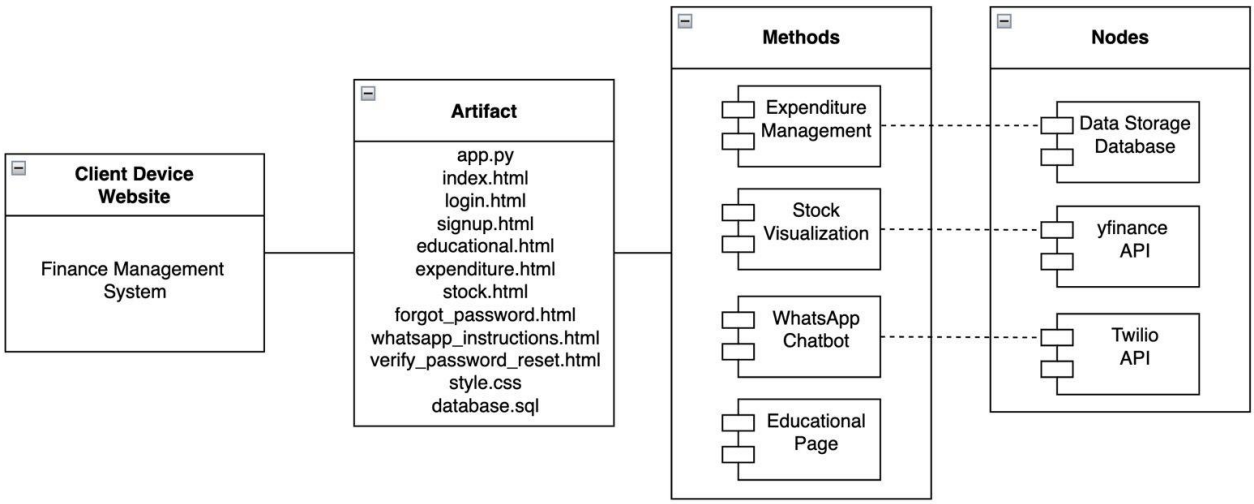
STATE CHART DIAGRAM



COMPONENT DIAGRAM



DEPLOYMENT DIAGRAM



SCREENSHOTS

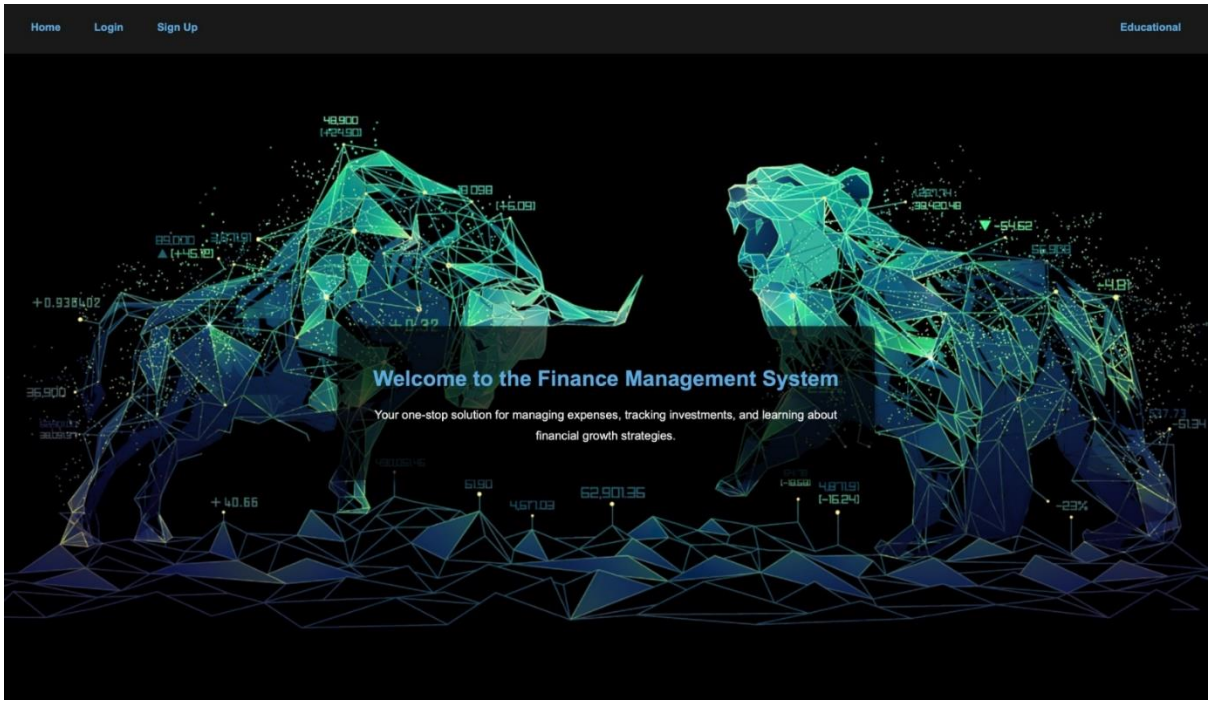


Fig 1 : Welcome Page of the Website

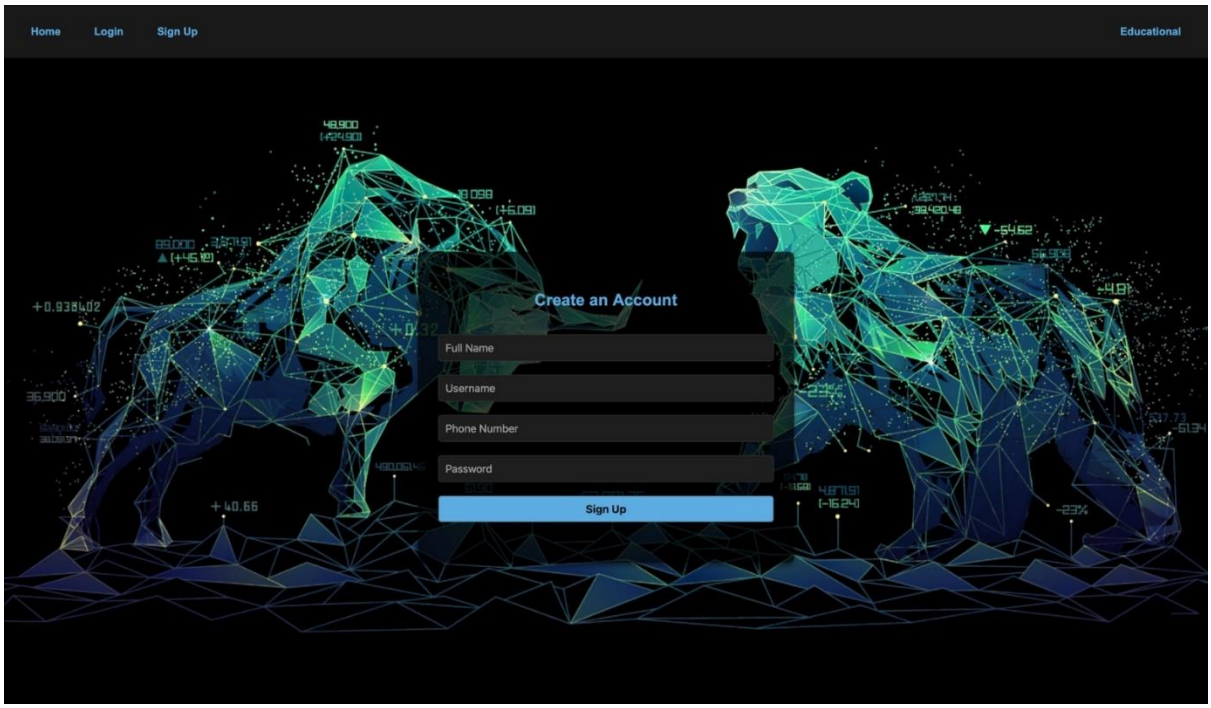


Fig 2 : Sign Up Page for Account Creation on Website

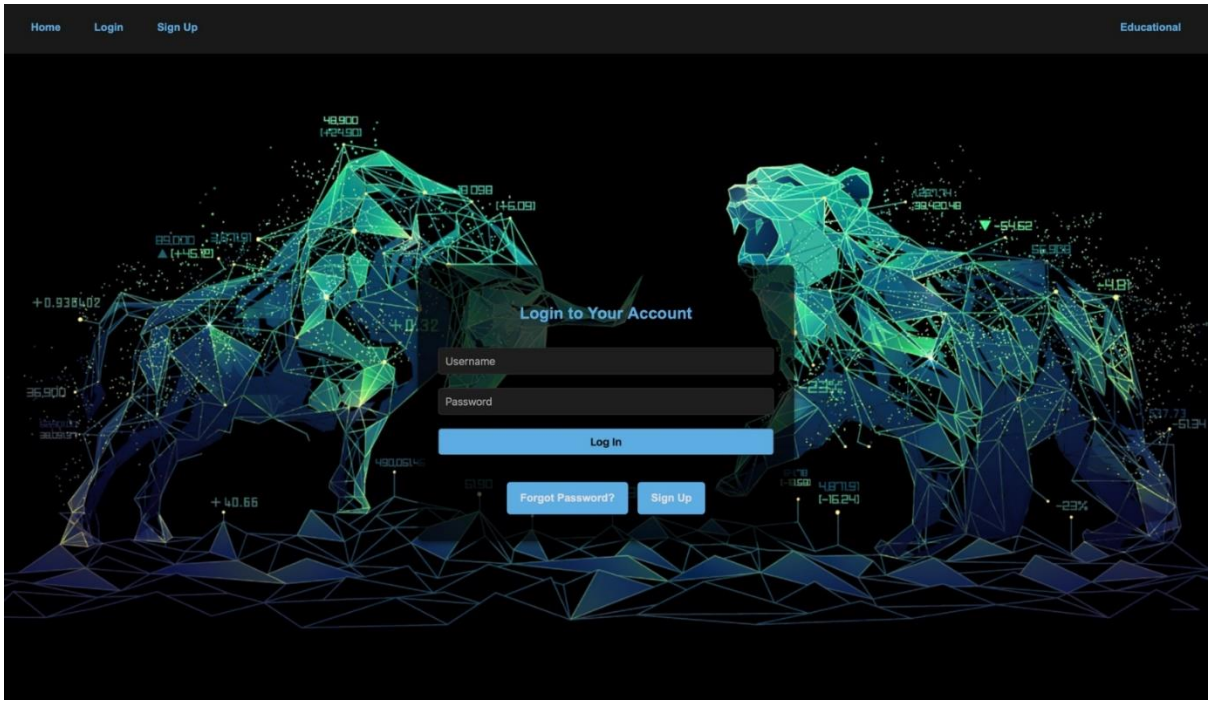


Fig 3 : Login Page of the Website

Fig 4 : Expenditure Records Entry and Viewing on Website

Fig 5 : Viewing Stock Performance on Website

Fig 6 : Viewing Stock Performance on Website for a Particular Stock

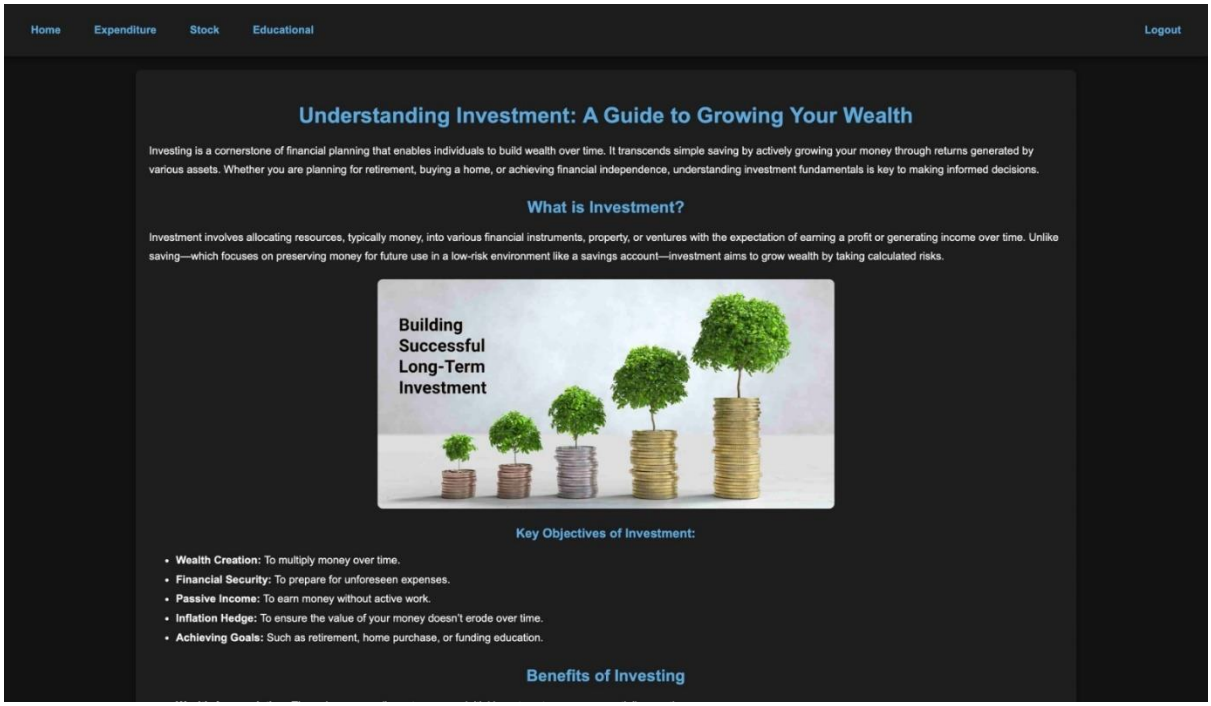


Fig 7 : Education Page on Website

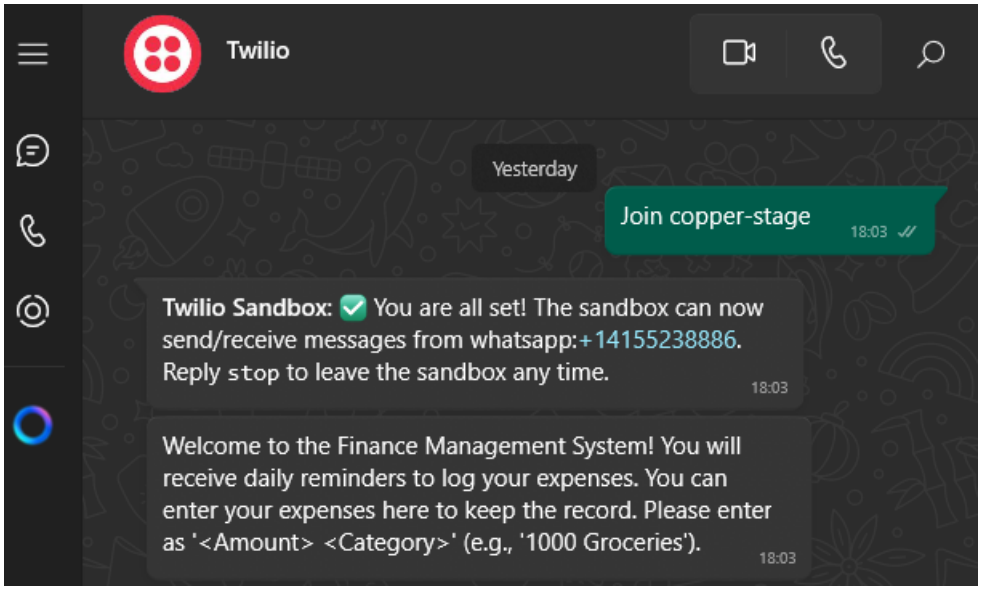


Fig 8 : Joining Twilio on WhatsApp for Expenditure Entry

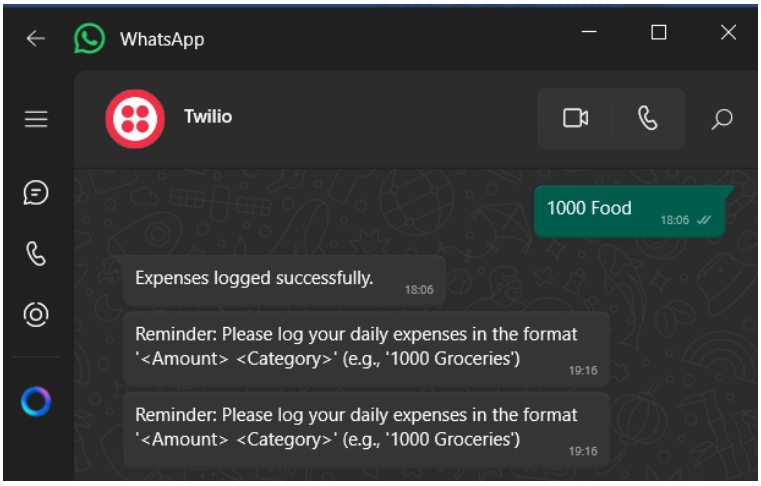


Fig 9 : Expenses Entry and Reminders on WhatsApp

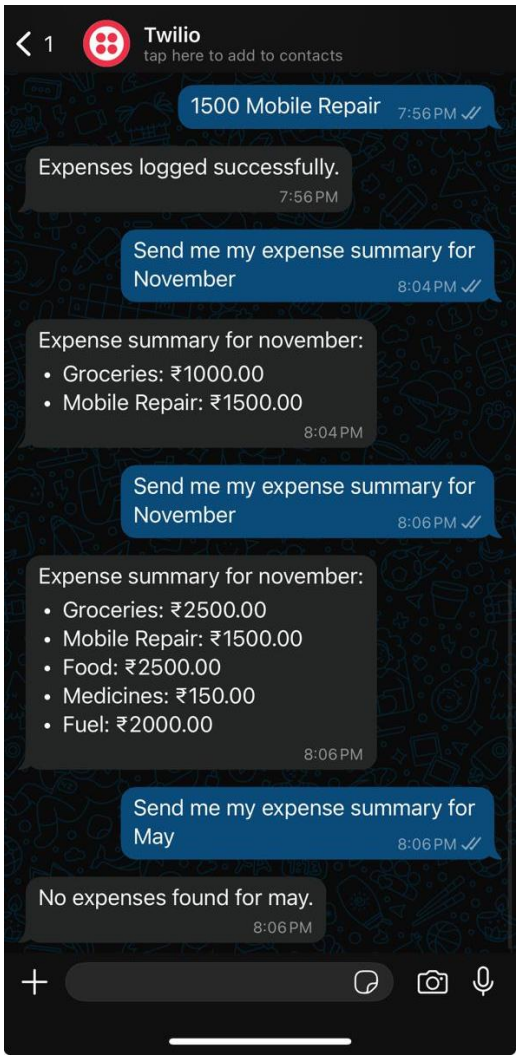


Fig 10 : Viewing Expenses Summary on WhatsApp

TEST CASES

Test Case #1: Login

Test Case #: 1 System: The web application handles user authentication, including verifying credentials, redirecting users upon success, and displaying error messages for invalid attempts. Designed by: Rehatman Kaur, Shiven Khare, Hrishita Dalal Executed by: Rehatman Kaur, Shiven Khare, Hrishita Dalal	Test Case Name: Verify User Login Subsystem: Authentication Short Description: Test the user login functionality with valid and invalid credentials. Design Date: 17 Nov, 2024 Execution Date: 17 Nov, 2024

Pre-conditions:
- User should have an existing account.
- The login page should be accessible.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Navigate to the login page	Login page loads successfully	**Pass**	Login page loaded successfully
2	Enter valid username and password	User is redirected to the home page, and gets access to the features	**Pass**	User successfully logged in and redirected to the home page.
3	Enter invalid username and password	An error message is displayed	**Pass**	Error message displayed as expected: 'Invalid Email or Password.
4	Leave fields blank and click login	A validation error message is displayed	**Pass**	Validation message displayed as expected: “Fill out this field”.

Post-conditions:
- User is successfully logged in if credentials are valid.
- User remains on the login page if credentials are invalid.

Test Case #2: Delete Expenditure

Test Case #: 2 System: The web application manages expenditure records, allowing users to delete specific records from the database with confirmation to avoid accidental deletions. Designed by: Rehatman Kaur, Shiven Khare, Hrishita Dalal Executed by: Rehatman Kaur, Shiven Khare, Hrishita Dalal Short Description: Test if the user can delete expenditure entries.	Test Case Name: Verify Deletion of Expenditure Subsystem: Expenditure Management Design Date: 17 Nov, 2024 Execution Date: 17 Nov, 2024

Pre-conditions:
- User must be logged in.
- The expenditure table must have entries.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Navigate to the expenditure page	Expenditure page loads successfully	Pass	Expenditure page loaded without errors, displaying existing entries.
2	Click the "Delete" button	A confirmation dialog appears	Pass	Confirmation dialog displayed, asking for user confirmation before deletion.
3	Confirm the deletion	Entry is deleted and removed from the table	Pass	Selected entry successfully deleted from the table and database.
4	Cancel the deletion	Entry remains in the table	Pass	Selected entry remains in the table, no changes were made.

Post-conditions:
- The selected entry is deleted if confirmed.
- No changes are made if deletion is canceled.

Test Case #3: Forgot Password

Test Case #: 3
System: The web application enables users to reset their passwords through a secure process, including verifying their identity via phone number or email and allowing password updates.
Designed by: Rehatman Kaur, Shiven Khare, Hrishita Dalal
Executed by: Rehatman Kaur, Shiven Khare, Hrishita Dalal

Test Case Name: Verify Password Reset Functionality
Subsystem: Authentication
Design Date: 18 Nov, 2024
Execution Date: 18 Nov, 2024
Short Description: Test the password reset process.

Pre-conditions:
- User must have a registered account.
- The forgot password page must be accessible.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Navigate to the forgot password page	Forgot password page loads successfully	Pass	The page loaded with no issues.
2	Enter a registered phone number	Message for resetting the password is sent through WhatsApp	Pass	The reset password message was successfully sent via WhatsApp.
3	Enter an unregistered phone number	An error message is displayed	Pass	Error message "Invalid or Unregistered Phone Number" was displayed.
4	User enter's a valid password	The password reset is successful	Pass	The password was reset successfully and the confirmation sent via WhatsApp.
5	User enter's an invalid password	The user receives the message suggesting that the entered password doesn't meet the validations	Pass	The validation error message sent correctly via WhatsApp

Post-conditions:
- User can log in with the new password.
- Reset request fails if verification fails.

Test Case #4: Searching Stock

Test Case #: 4
System: The web application allows users to search for stock information, fetches relevant stock data from an external API, and displays it along with a historical chart based on the selected duration.
Designed by: Rehatman Kaur, Shiven Khare, Hrishita Dalal
Executed by: Rehatman Kaur, Shiven Khare, Hrishita Dalal
Short Description: Test the stock search feature for valid and invalid stock symbols.

Test Case Name: Verify Stock Search Functionality
Subsystem: Stock Management
Design Date: 18 Nov, 2024
Execution Date: 18 Nov, 2024

Pre-conditions:
- User must be logged in.
- The stock search feature must be available.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Navigate to the stock search section	Stock search page loads successfully	Pass	Stock search page displayed correctly with input fields and options visible.
2	Enter a valid stock symbol/company name	Stock ticker name fetched correctly and details and chart are displayed	Pass	Stock ticker details retrieved successfully, and the chart is displayed as expected.
3	Enter an invalid stock symbol/company name	An error message is displayed	Pass	Error message displayed, indicating no matching stock symbol or company was found.
4	Leave the search field blank	A validation error message is displayed	Pass	Validation error message displayed, prompting the user to enter a stock symbol or company name.

Post-conditions:
- Stock details are shown for valid symbols.
- Errors are handled gracefully for invalid inputs.