

# **EXPENSE SHARING APP**

## **A PROJECT REPORT**

**Submitted By**

**ANMOL GOYAL**  
**(2200290140033)**

**Submitted in partial fulfilment of the  
Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of  
Prof. Rabi N. Panda  
ASSOCIATE PROFESSOR**



**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS  
KIET Group of Institutions, Ghaziabad  
Uttar Pradesh-201206**

**(FEBRUARY 2024)**

# **CERTIFICATE**

Certified that **Anmol Goyal 2200290140033** has/ have carried out the project work having "**Expense Sharing App**" for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Anmol Goyal (2200290140033)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

**Prof. Rabi N. Panda**  
**Associate Professor**  
**Department of Computer Applications**  
**KIET Group of Institutions, Ghaziabad**

**Dr. Arun Tripathi**  
**Head**  
**Department of Computer Applications**  
**KIET Group of Institutions, Ghaziabad**

# **EXPENSE SHARING APP**

**ANMOL GOYAL**

## **ABSTRACT**

In our fast-paced world, efficient personal finance management is challenging. We introduce an innovative expense-sharing web-based app powered by voice recognition and GPS tracking, revolutionizing expense monitoring in defined geographic regions. This app transforms how individuals manage expenses within specific locations.

The core functionality of our web-based app is twofold. First, it utilizes cutting-edge voice recognition capabilities to allow users to effortlessly input expenses by simply speaking them. This hands-free approach enhances user convenience, reduces manual entry errors, and expedites expense tracking.

Second, our employs GPS technology to track users' geographical locations in real-time. It automatically calculates their expenses within predefined areas. If a user's expenses exceed a preset limit within a particular zone, the app triggers an alert, notifying the user of the limit breach. This feature acts as a safeguard against overspending and encourages responsible financial behavior.

Conversely, if the user's expenses fall below the allocated limit, the app suggest budget allocations for unmet needs, ensuring financial goals.

Our app amalgamates cutting-edge technology with personal finance management, empowering users to effortlessly track and control their expenses while reinforcing responsible spending habits. By integrating voice recognition and GPS tracking, we aim to simplify expense management, enhance awareness, and promote financial well-being for users across the globe.

**Keywords:** Cost-sharing, Shared wallets, Bill splitting, Shared finances, IOU tracking

## **ACKNOWLEDGEMENTS**

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Prof. Rabi N. Panda** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**ANMOL GOYAL**

# TABLE OF CONTENTS

Certificate	i
Abstract	ii
Acknowledgements	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
1 Introduction	1-9
1.1 Key Features	1
1.1.1 User Registration and Login	1
1.1.2 Dashboard	1
1.1.3 Expense Management	1
1.1.4 Profile Management	2
1.1.5 Budget Adjustment	2
1.1.6 Help and Queries	2
1.1.7 Mode Selection	2
1.2 Project Description	2-3
1.3 Project Scope	4-5
1.4 Hardware/Software used in Project	5-9
2 Feasibility Study	10-18
2.1 Key Objectives	11
2.2 Technical Feasibility	12-14
2.3 Operational Feasibility	14-15
2.4 Behavioral Feasibility	15-17
2.5 Schedule Feasibility	17-18
3 Design	19-30
3.1 Database Tables	20-21
3.2 Flowchart	21-22

3.3	Use Case Diagram	22-25
3.4	Data Flow Diagram	25-30
4	Form	31-43
5	Coding	44-68
6	Testing	69-70
6.1	Test Case-1	69-70
6.2	Test Case-2	70
	Bibliography	71
	References	72

## **LIST OF TABLES**

<b>Table No.</b>	<b>Name of Table</b>	<b>Page</b>
3.1	User's Table	20
3.2	Expenses Table	20-21

# LIST OF FIGURES

<b>Figure No.</b>	<b>Name of Figure</b>	<b>Page No.</b>
1.1	Mode Preference Unleashed - Define Your Figures, Craft Your Story	3
1.2	Select Role for beginning Expense Enigma Journey	5
1.3	Login Credentials Web Page	9
2.1	User controlled expense dashboard	12
2.2	Expense Sharing Architecture Diagram	14
2.3	Behavioral Database Change with Budget	17
2.4	Schedule Feasibility (Gantt Chart)	18
3.1	Expense Sharing App Explore by User	19
3.2	Flowchart Diagram for Expense Sharing App	22
3.3	Use Case diagram for Expense Sharing App	23
3.4	Use Case diagram for Expense Enigma App	24
3.5	Level 0 DFD of Expense Sharing App	26
3.6	Level 2 DFD of Expense Sharing App	30
4.1	User/ Admin login Credentials form	31
4.2	Initially user select Mode Preference	32
4.3	Select Role either: User, Admin	32
4.4	Admin Entered Correct Credentials	33
4.5	Admin Entered Wrong Credentials	34
4.6	Admin Dashboard	34
4.7	Admin-user Dashboard	35
4.8	Forgot Password Module	36
4.9	Password Reset Link Generate	36
5.1	Email Reset Link Generate	37
5.2	Reset Password	37
5.3	Password Updated Successfully	38
5.4	User Expense Dashboard	38

5.5	Adding expense alert	39
5.6	New Expenses Added successfully	39
5.7	Graphical Expense Analysis	40
5.8	Expenses Spreadsheets View	40
5.9	User Profile Page	41
6.1	Financial Features	41
6.2	Help and Support Page	42
6.3	Raise Query Support	42
6.4	Dark Mode Enabled Feature	43
6.5	Logout Expense Enigma Services	43

# CHAPTER 1

## INTRODUCTION

### 1. INTRODUCTION

The Expense Sharing app is a comprehensive and user-friendly financial management application designed to simplify the process of tracking and sharing expense within a group. The app caters to individuals who share financial responsibilities, such as roommates, friends, or colleagues, providing a seamless platform for collaborative expense management. The app is equipped with essential features, including user registration, login, budget tracking, expense recording, graphical analysis, profile management, and support for users' queries. Let's delve into the details of each key functionality.

### 1.1 KEY FEATURES

- 1.1.1 **User Registration and Login:** Users begin by either logging in with their existing credentials or registering by providing essential details such as name, email, password, and an initial budget. The registration process ensures a secure and personalized experience for each user.
- 1.1.2 **Dashboard:** Upon logging in, users are greeted with a dashboard displaying their budget, initial expenses set to 0, and remaining balance set to 0. This provides a quick overview of their financial standing.
- 1.1.3 **Expense Management:** User can individually enter expenses, categorize them, and specify details such as date and description. The app automatically calculates and updates the remaining balance, reflecting the impact of each expense on the overall budget.

- 1.1.4 **Profile Management:** Users can access and update their profile information, including name, email, and password. This ensures that user details are current and accurate.
- 1.1.5 **Budget Adjustment:** The “Budget” feature enables users to modify their budget as needed. Conforming the change, the budget, allowing users to adapt to evolving financial circumstances.
- 1.1.6 **Help and Queries:** Users seeking assistance can use the “Help” feature to raise queries and receive support. This ensures that users have access to guidance whenever needed.
- 1.1.7 **Mode Selection:** Users can switch between different modes based on their preferences or specific use cases.

## 1.2 PROJECT DESCRIPTION

Share Expenses is a comprehensive expense management application developed to address the challenge associated with dividing and overseeing shared financial obligations within social or professional circles. The primary goal of this app is to provide users with a seamless and efficient solution for handling the complexities of shared expenses, offering a user-friendly interface and a range of features to enhance financial coordination.

Users of Share Expenses can easily initiate and manage expense groups tailored to specific shared activities, such as household bills, group outings, or collaborative projects. The app’s intuitive design allows participants to keep track of individuals’ contributions and monitor the overall financial landscapes within the group. This transparency is a key aspect of Share Expense, aiming to foster clear communication and reduce potential misunderstandings related to shared finances.

One of the standout features of Share Expense is its robust expense tracking system. Users can input and categorize various expenses, providing detailed information such as the nature of the cost, date and the involved participants.

This features not only aids in accurate record-keeping but also facilitates a comprehensive understanding of the financial dynamics within the group. The app's quick bill-splitting functionalities further contribute to its efficiency. Share Expenses automates the process of dividing costs among participants, eliminating the need for manual calculations and ensuring that each member's financial contributions accurately reflected. This automation not only saves time but also minimizes the likelihood of errors in the distribution of expenses.

Real-time updates on financial contributions add another layer of convenience to Share Expense. Participants can stay informed about the current state of shared expenses, fostering a sense of safeguard sensitive financial information, ensuring a trustworthy platform for users.

In Summary, Share Expense is not just an expense-sharing app; it's a comprehensive solution designed to simplify and enhance the management of shared finances. By combining intuitive design, automated functionalities, and real-time updates, Share Expense empowers users to navigate shared financial responsibilities with ease, promoting transparency, accuracy and peace of mind."

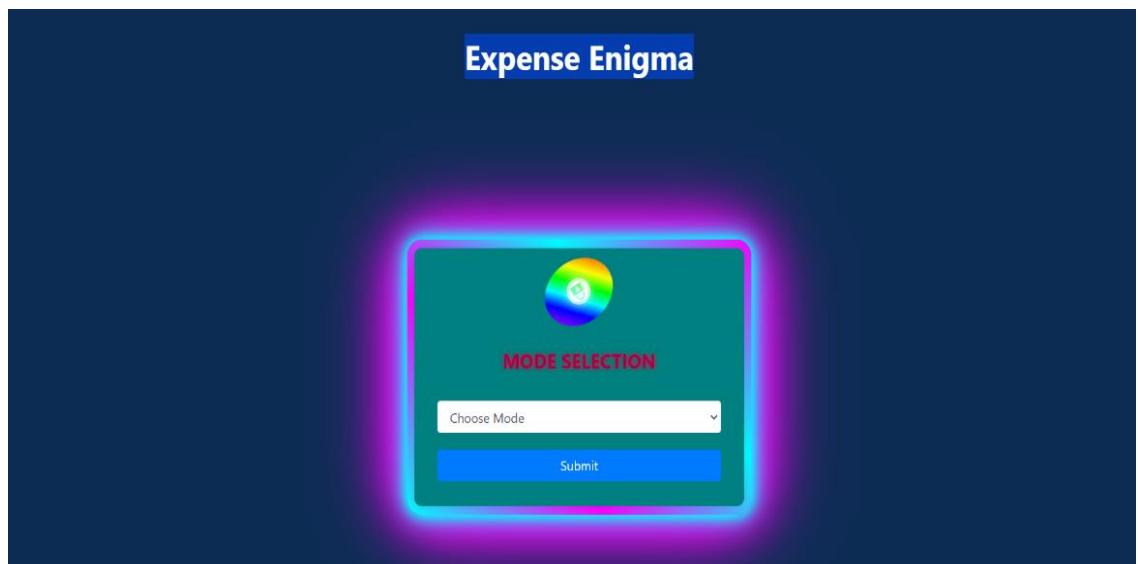


Figure 1.1 Mode Preference Unleashed - Define Your Figures, Craft Your Story

### **1.3 PROJECT SCOPE**

The project scope for Share Expense entails the development and deployment of a comprehensive expense, and management application. Users will have the ability to register and create individual accounts, ensuring a personalized experience within the app. Robust authentication mechanisms will be implemented to safeguard user data and maintain the security of the platform.

A central feature of the app is the creation and a management of expenses groups. Users can effortlessly initiate and oversee financial collaborations related to shared activities, such as household bills, group outings, or collaborative projects. Each expense group will provide a dedicated space for participants, enabling transparent communication and coordination.

The application will boast an intuitive interface for detailed expense tracking. Users can input and categorize various expenses, including relevant details such as the nature of the cost, date and involved participants.

The features aim to streamline record-keeping and facilities a comprehensive understanding of the financial dynamics within each group.

To further enhance efficiency, the app will automate the process of splitting bills among participants. This functionality eliminates the need for manual calculations, ensuring accurate and fair distribution of expenses. Real-time updates on financial contributions will be provided, fostering a sense of accountability and collaboration within each expense group.

Overall, the Share Expense app will offer a seamless and user-friendly solution for managing shared finances, promoting transparency, accuracy, and ease of use for individuals navigating shared financial responsibilities.

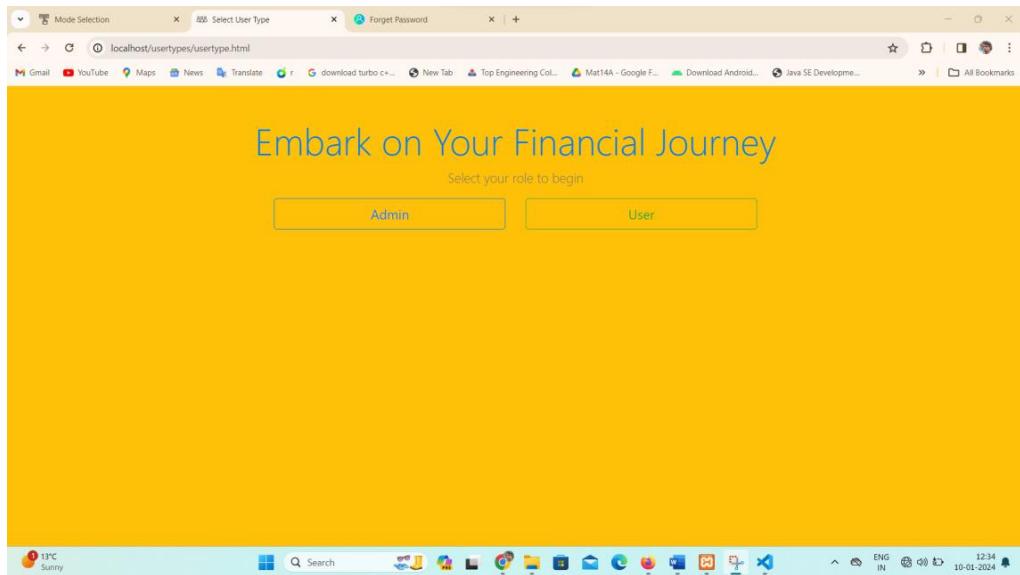


Figure 1.2 Select Role for beginning Expense Enigma Journey

Automation will be a key feature, particularly in the process of splitting bills among participants. The app will employ algorithms to calculate and distribute expenses automatically, eliminating the need for manual calculations and minimizing the potential for errors. Real-time updates on financial contributions will be integrated to keep participants informed, fostering a sense of accountability and collaboration within each expense group.

## 1.4 HARDWARE / SOFTWARE USED IN PROJECT

The Expense Sharing app will involve a combination of hardware and software components to ensure its development, deployment and functionality. Here is detailed List:

### 1.4.1 Server-side Hardware:

#### 1. RAM (Random Access Memory):

- 8GB to 16GB (for moderate-sized application and user load).
- Consider higher capacities (e.g., 32GB or more) for scalability and handling a large number of concurrent users.

## **2. ROM (Storage):**

- SSD storage for faster read and write operations.
- Allocate storage based on the application codebase, database size, and media storage requirements.

## **3. Processor:**

- Multi-core processor (quad-core or higher) for efficient handling of concurrent user requests.

## **4. Operating System:**

- Linux-based operating system (e.g., Ubuntu Server, CentOS, Debian) for stability and performance.

## **5. Network Equipment:**

- Network infrastructure to facilitate secure data transfer between users and the server.

### **1.4.2 Database Server:**

#### **1. RAM:**

- 16GB or more for efficient handling of concurrent database queries.

#### **2. ROM(Storage):**

- SSD storage for faster data retrieval.
- Allocate storage based on the anticipated size of the database and data storage needs.

#### **3. Processor:**

- Multi-core processor with sufficient processing power for complex database operations.

#### **4. Operating System:**

- Linux-based operating system for the database server.

#### **1.4.3 User Devices:**

##### **1. Smartphones/Tablets:**

- Compatibility with iOS and Android operating systems.
- Optimization for various screen sizes and resolutions.

##### **2. Web Browsers:**

- Compatibility with major web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

#### **1.4.4 Development Environment:**

##### **1. Programming Languages:**

- Backend: Python, Node.js or another suitable language.
- Frontend: HTML5, CSS3, JavaScript (React, Angular, Vue.js).

##### **2. Framework:**

- Web application framework (e.g., Flask, Django, Express.js) for backend development

##### **3. Database Management System:**

- Choose a suitable DBMS (e.g., MySQL, PostgreSQL, MongoDB) for efficient data storage and retrieval.

##### **4. Authentication and Authorization:**

- Implement secure authentication protocols (OAuth, JWT) and authorization mechanisms.

##### **5. APIs:**

- Develop APIs to enable communication between the frontend and backend components.

## **6. Version Control:**

- Version control system (e.g., Git) for managing and tracking changes in the source code.

## **7. Integrated Development Environment (IDE):**

- IDEs such as Visual Studio Code, PyCharm, or IntelliJ IDEA for coding and debugging.

## **8. Containerization:**

- Containerization tools like Docker for efficient deployment and scalability.

## **9. Continuous Integration/Continuous Deployment (CI/CD) :**

- CI/CD pipelines (e.g., Jenkins, Travis CI) for automated testing and deployment.

## **10. Security Tools:**

- Integrate security tools and practices to ensure the application's resilience against potential threats.

## **11. Monitoring Tools:**

- Monitoring tools (e.g., Prometheus, Grafana) for tracking application performance.

## **12. Collaboration Tools:**

- Collaboration tools (e.g., Slack, Microsoft Teams) for effective communication among project team members.

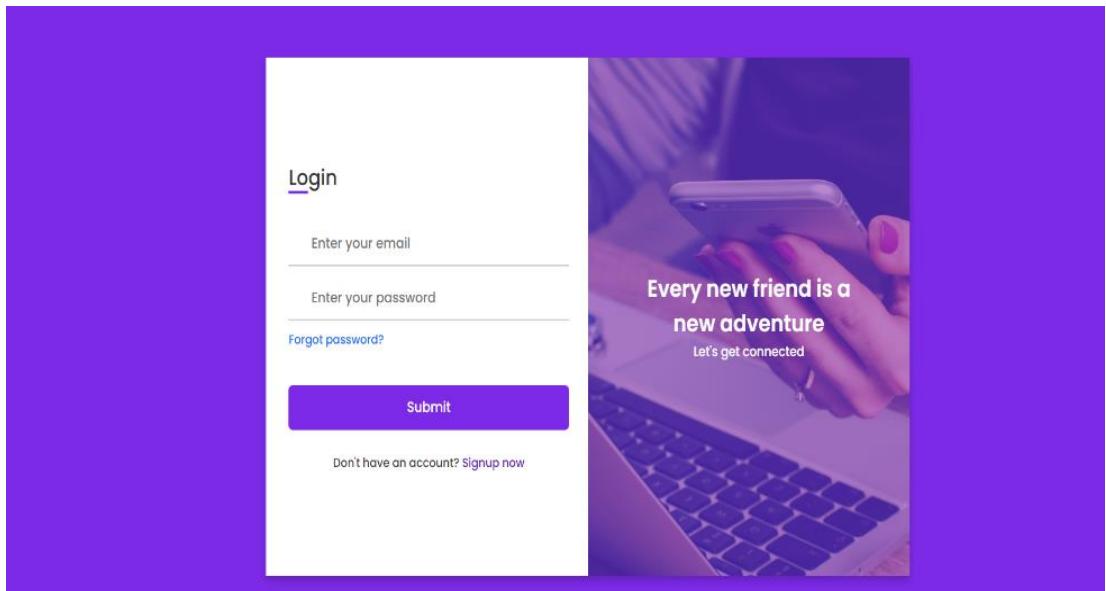


Figure 1.3 Login Credentials Web Page

# **CHAPTER 2**

## **FEASIBILITY STUDY**

### **2. INTRODUCTION**

The feasibility study for the Expense Share app is a critical initial phase in the project lifecycle, aiming to assess the viability and practicality of implementing the proposed expense management solution. This study encompasses a comprehensive analysis of various aspects, including technical, economic, legal, operational, and scheduling considerations.

The introduction to the feasibility study sets the stage for understanding the rationale behind the project and outlines the key objectives of the assessment.

In an era characterized by dynamic financial interactions and collaborative living, the concept of managing shared expenses has become increasingly complex. Recognizing the need for a streamlined solution, the Share Expense app is envisioned as a comprehensive tool to alleviate the challenges associated with dividing and managing shared financial responsibilities.

This feasibility study is undertaken to evaluate the practicality and potential success of developing and implementing the Share Expense app.

The primary objective of the feasibility study is to provide a thorough examination of the technical, economic, legal, and operational aspects involved in bringing the Share Expense app to fruition. By conducting this study, we aim to gain insights into the project's viability, potential challenges, and anticipated benefits. The study will serve as a foundation for informed decision-making throughout the development lifecycle.

## **2.1 KEY OBJECTIVES**

### **2.1.1 Technical Feasibility:**

- Assess the technical requirements and challenges associated with developing the Share Expense app.
- Evaluate the availability of necessary technologies and expertise to implement the proposed features.

### **2.1.2 Economic Feasibility:**

- Examine the cost implications of developing and maintaining the Share Expense app.
- Project the Return on Investment (ROI) and assess the financial viability of the project.

### **2.1.3 Legal Feasibility:**

- Investigate legal considerations, including data protection, privacy laws, and compliance requirements.
- Ensure that the development and deployment of the app adhere to regulatory standards.

### **2.1.4 Operational Feasibility:**

- Analyze how the Share Expense app will integrate into existing operational processes.
- Evaluate the app's usability and acceptance by potential users.

### **2.1.5 Scheduling Feasibility:**

- Develop a realistic project timeline, considering the development, testing, and deployment phases.
- Identify potential bottlenecks and risks that may impact the project schedule.

## 2.2 Technical Feasibility

The technical feasibility assessment is a pivotal component of the Share Expense app, project, focusing on the viability and capability of implementing the proposed solution from a technological standpoint.

This analysis delves into various technical aspects, including infrastructure requirements, software development considerations, and potential challenges. The goal is to ascertain whether the envisioned app can be developed effectively, leveraging available technologies and expertise.

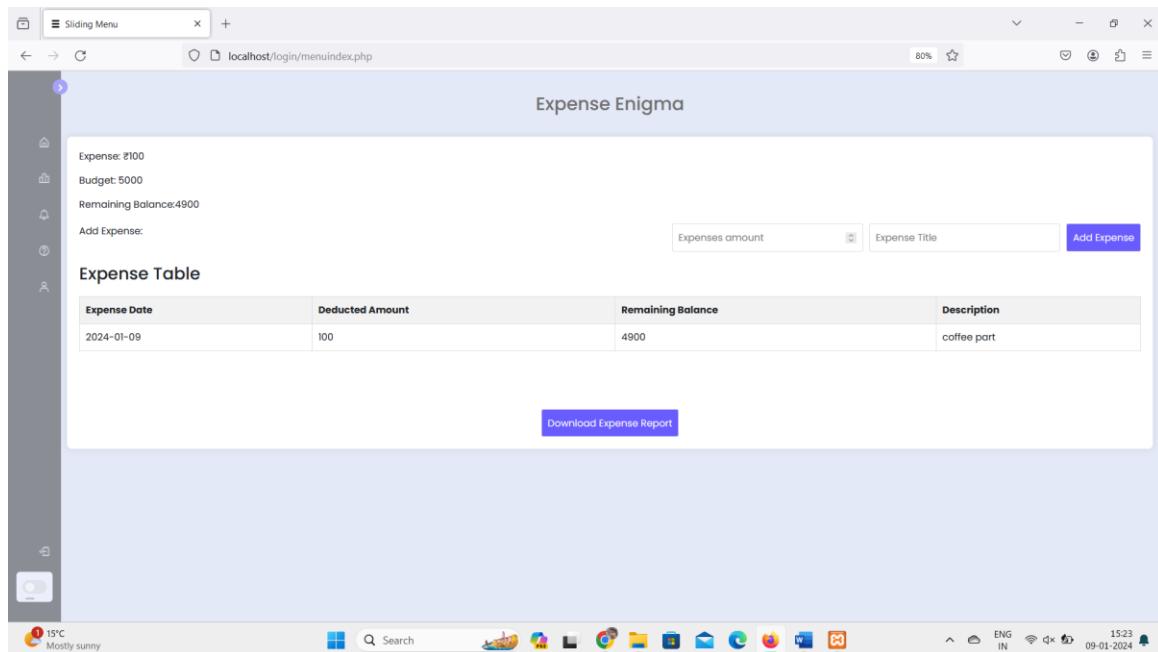


Figure 2.1 User controlled expense dashboard

### 2.2.1 Infrastructure Requirements:

- **Server Infrastructure:** Assess the capacity and scalability of cloud-based servers (e.g., AWS, Azure) to accommodate potential user growth and ensure seamless performance.
- **Database Management:** Evaluate the suitability of database systems (e.g., MySQL, PostgreSQL) for efficient storage and retrieval of user data.

### **2.2.2 Software Development:**

- **Programming Languages:** Choose appropriate backend (e.g., Python, Node.js) and frontend (e.g., React, Angular) technologies based on developer expertise and project requirements.
- **Framework Selection:** Select a web application framework (e.g., Django, Flask) to streamline development and enhance maintainability.

### **2.2.3 Security Measures:**

- **Authentication Protocols:** Implement secure authentication mechanisms (e.g., OAuth, JWT) to protect user accounts and ensure data security.

### **2.2.4 User Interface (UI) Design:**

- **Responsive Design:** Optimize the app's UI for various devices (smartphones, tablets, web browsers) to provide a consistent and user-friendly experience.

### **2.2.5 Deployment and Monitoring:**

- **Docker:** Implement containerization using Docker for efficient deployment, scalability, and consistency across different environments.

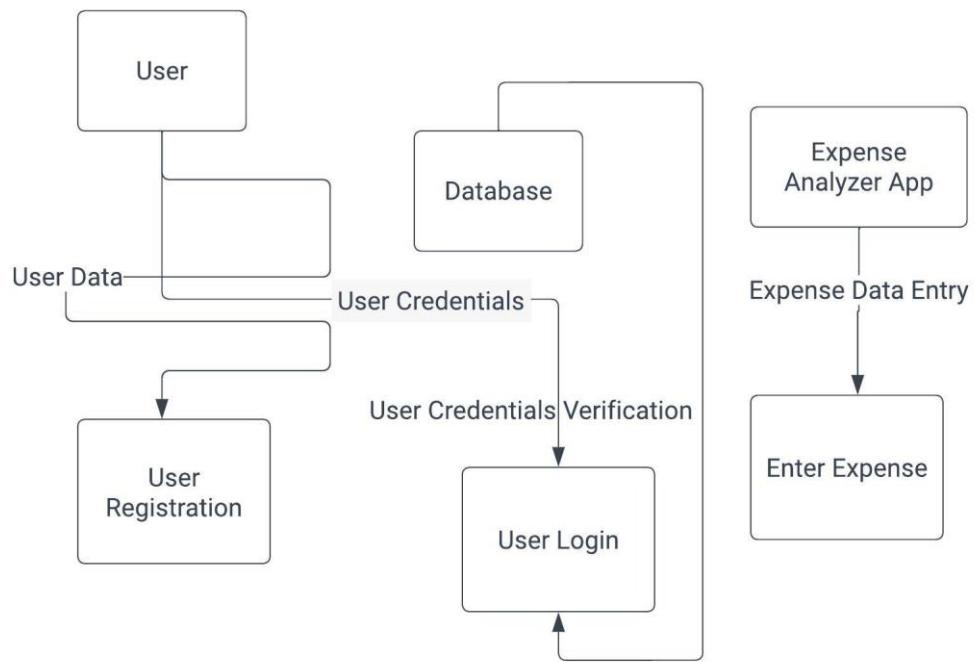


Figure 2.2 Expense Sharing Architecture Diagram

## 2.3 Operational Feasibility

The operational feasibility analysis is a crucial aspect of determining whether the Share Expense app can seamlessly integrate into existing processes and effectively meet the needs of its users.

This assessment involves evaluating usability, acceptance, and overall practicality from an operational standpoint.

### 2.3.1 User Acceptance:

- **User Feedback Surveys:** Conduct surveys or gather feedback from potential users to gauge their acceptance of the Share Expense app. Understand user preferences and expectations.

### 2.3.1 Usability Testing:

- **User Interface (UI) Testing:** Evaluate the user interface for intuitiveness and ease of use. Conduct usability testing to identify any potential issues in navigation or functionality.

### **2.3.2 User Engagement Strategies:**

- **Communication Plans:** Develop communication strategies to keep users informed about new features, updates, and any changes in the app. Foster ongoing engagement.

### **2.3.3 Operational Impact Analysis:**

- **Operational Workflow Analysis:** Assess how the Share Expense app will fit into users' daily workflows. Identify potential impacts on existing operational processes.

### **2.3.4 Change Management Strategies:**

- **Change Management Plans:** Develop strategies to manage organizational and user-level changes resulting from the introduction of the Share Expense app. Address any potential resistance.

### **2.3.5 Legal and Compliance Considerations:**

- **Compliance Analysis:** Ensure that the app complies with relevant legal and regulatory requirements related to financial transactions, data protection, and user privacy.

## **2.4 Behavioral Feasibility**

The behavioral feasibility analysis of the Share Expense app focuses on understanding and anticipating the attitudes, perceptions, and cultural dynamics that may influence user acceptance and adoption. A key consideration in the success of the app is the motivation of potential users to incorporate it into their shared financial management practices.

This involves evaluating whether users perceive the app as a valuable solution to their needs and whether it aligns with their motivations for collaborative expense management. Cultural acceptance is another critical aspect, where the app's features and communication strategies need to resonate with cultural preferences.

Adaptations may be necessary to ensure the app is well-received within diverse user communities. Effective communication strategies are vital to convey the benefits of the Share Expense app clearly. Outreach efforts should aim to inform potential users about how the app addresses pain points and simplifies shared financial responsibilities.

Understanding and addressing resistance to change is imperative, as users may be accustomed to existing methods of expense management. Change management strategies should be implemented to ease concerns and foster a positive attitude towards adopting the app.

Moreover, user education and awareness campaigns play a pivotal role in ensuring that potential users are well-informed about the app's features and advantages. User-centric design principles guide the development of the Share Expense app, taking into account user preferences and behaviors.

User experience research informs the refinement of the app's interface for optimal usability, creating an environment that is intuitive and engaging for a diverse user base. Consideration of social dynamics in shared living or collaborative work environments is also paramount, with the app designed to enhance social interactions related to financial activities.

To further encourage user acceptance, the app incorporates feedback mechanisms, providing users with a platform to contribute their insights. Incentive structures, such as rewards for active participation, are explored to motivate users and enhance behavioral acceptance.

The Share Expense app also places a strong emphasis on accessibility and inclusivity, ensuring that it caters to users with diverse needs and abilities. By addressing these behavioral aspects comprehensively, the Share Expense app seeks not only to meet the functional needs of users but also to align seamlessly with their behaviors, preferences, and motivations.

This user-centric approach is fundamental to fostering positive user attitudes, driving adoption, and ensuring the long-term success of the app in shared financial management scenarios.

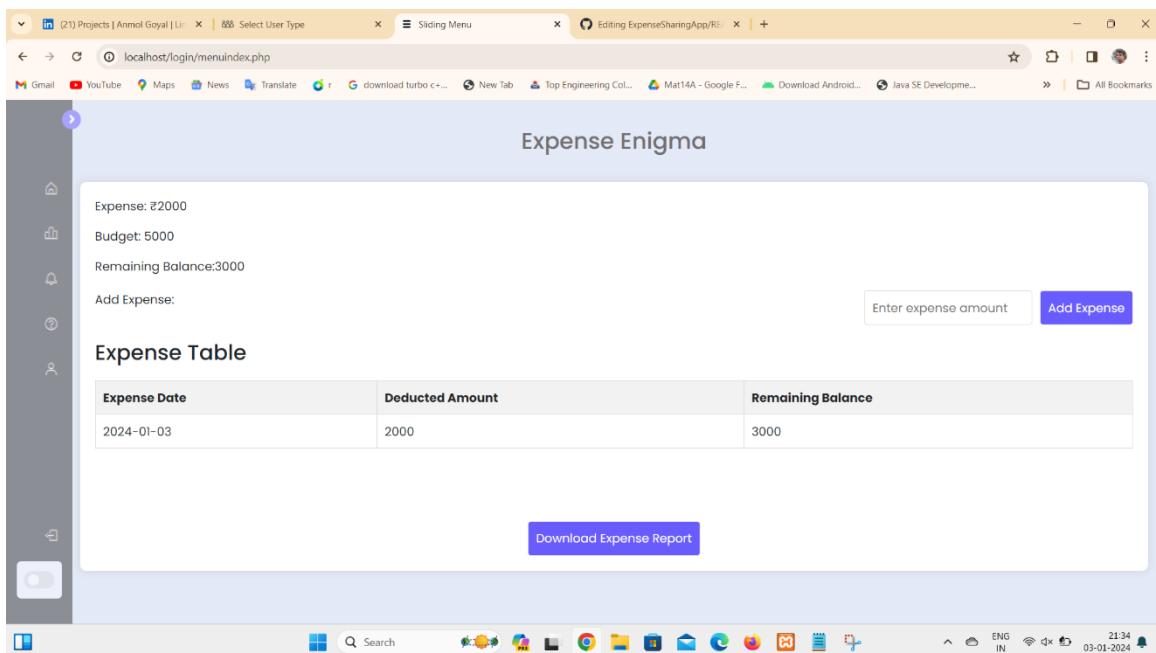


Figure 2.3 Behavioral Dashboard Change with Budget

## 2.5 Schedule Feasibility

The schedule feasibility analysis for the Share Expense app is integral to the project's success, aiming to assess the practicality and achievability of the proposed timeline. A meticulously planned project timeline delineates key milestones, deliverables, and deadlines across the development, testing, and deployment phases. Each stage, including development, testing, and iterative cycles, undergoes detailed task breakdowns to ensure accurate time allocation for coding, testing, and refinement.

Testing and quality assurance constitute critical phases, demanding ample time for comprehensive testing, debugging, and issue resolution. Incorporating iterative development cycles and feedback loops allows for continuous refinement based on testing outcomes and user feedback. The availability of human resources, including developers and testers, is carefully considered to ascertain that team members can commit the necessary time to their respective roles.

In anticipating potential risks, the schedule feasibility analysis identifies technical challenges, resource constraints, and unforeseen issues, accompanied by mitigation strategies to minimize their impact. Exploring parallel development opportunities and incorporating contingency buffers in the schedule further fortifies the project against unexpected delays.

Effective collaboration and communication are essential components to facilitate seamless coordination among team members, ensuring timely issue resolution and adherence to the project schedule. The deployment phase is meticulously planned, encompassing server setup, data migration, and user onboarding, with allocated time to accommodate unforeseen challenges.

Finally, the schedule feasibility analysis recognizes the iterative nature of development, allowing for the integration of user feedback into the process. This iterative approach ensures that the Share Expense app aligns closely with user expectations, resulting in a well-executed project within the stipulated timeframe.

Tasks	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
<b>Project Initiation</b>						
<b>Requirements Gathering</b>						
<b>System Design</b>						
<b>Frontend Development</b>						
<b>Backend Development</b>						
<b>Voice Recognition</b>						
<b>Testing and QA</b>						
<b>User Testing</b>						
<b>Finalizing App</b>						
<b>Deployment</b>						
<b>Project Review</b>						

Figure 2.4 Schedule Feasibility (Gantt Chart)

# CHAPTER 3

## DATABASE DESIGN

### 3. INTRODUCTION

The database design for the Share Expense app is a foundational element that plays a pivotal role in organizing and managing data efficiently. A well-structured database is essential for facilitating seamless interactions between the application and its users, ensuring robust data storage, retrieval and manipulation. The introduction outlines of the fundamental principles guiding the database design process and emphasizes the importance of creating a scalable and secure data architecture to support the diverse needs of expense management within collaborative settings.

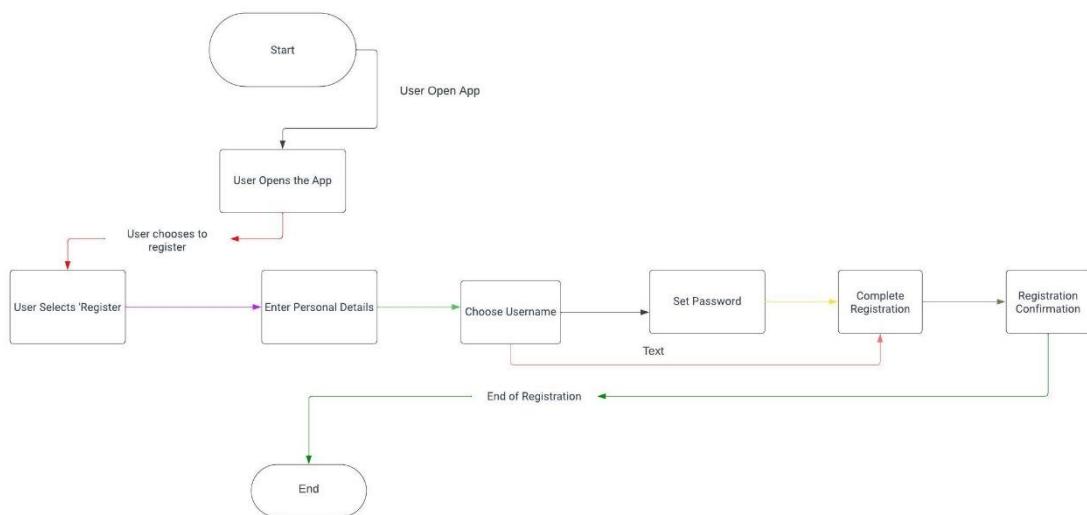


Figure 3.1 Expense Sharing App Explore by User

### 3.1 DATABASE TABLES

Creating a comprehensive database table for the Share Expense app involves considering the key entities and their attributes. In a simplified example, let's focus on two main entities: Users and Expenses. Here's a basic representation:

#### 3.1.1 Users Table:

- **user\_id (Primary Key):** Unique identifier for each user.
- **email:** User's email address for communication and login.
- **name:** User's full name.
- **password\_hash:** Securely hashed password for authentication.
- **phone number:** User's contact number.
- **initial\_budget:** The initial budget set by the user.

user_id	Email	name	password_hash	phone number	Initial_budget
1	anmol.goyal2097@gmail.com	Anmol	#2122223fsdx	9368563885	50000
2	a@gmail.com	Abhir	#3c2223rsdx	8321563885	6000
3	anmol.goyal2097@gmail.com	Anmol	#2122253ftdx	9462563885	10000

Figure 3.1. User's Table

#### 3.1.2 Expenses Table:

- **Expense Date:** Date when the expense was incurred.
- **Deducted Amount:** Amount deducted from user's initial budget.
- **Remaining Balance:** Balance left deducting expenses.
- **Description:** Title for each budget expenses
- **User name:** Anmol
- **Budget:** 5900

Expense Date	Deducted Amount	Remaining Balance	Description
2024-01-09	100	4900	Coffee party
2023-01-09	1000	3900	College fees
2024-02-09	100	3800	Rent

Figure 3.2. Expense's Table

### 3.2 FLOWCHART

#### Introduction to the Flowchart for Expense Analyzer App:

The flowchart for the Expense Analyzer App encapsulates the intricate sequence of actions and decision points involved in its operation, providing a visual roadmap for understanding the systematic flow of activities. At its core, the flowchart delineates the stepwise processes initiated by user interactions, the validation and categorization of expense data, communication with the database, and the subsequent generation and presentation of expense reports.

Commencing with the entry point, often represented by the user's interaction, the flowchart systematically guides through essential processes such as user registration, login, expense entry, and report generation. Decision points within the flowchart capture instances where the system evaluates conditions, such as validating user data or verifying login credentials, influencing the subsequent course of actions. Key processes, depicted in distinct shapes and connected by arrows, convey the logical flow from one operation to another.

For instance, the flowchart delineates how the Expense Analyzer App communicates with the database for tasks like storing and retrieving user and expense data. Decision diamonds illustrate branches in the logic, signifying points where the system evaluates conditions and proceeds accordingly.

This visual representation not only aids developers in understanding the operational logic but also serves as a valuable tool for stakeholders to grasp the intricacies of the Expense Analyzer App's functionality. The flowchart serves as a blueprint, guiding the development team through the systematic execution of tasks, ensuring that the application functions cohesively and efficiently. Ultimately, the Expense Analyzer App's flowchart is a pivotal

resource in comprehending, communicating, and refining the logical sequence of processes within the system.

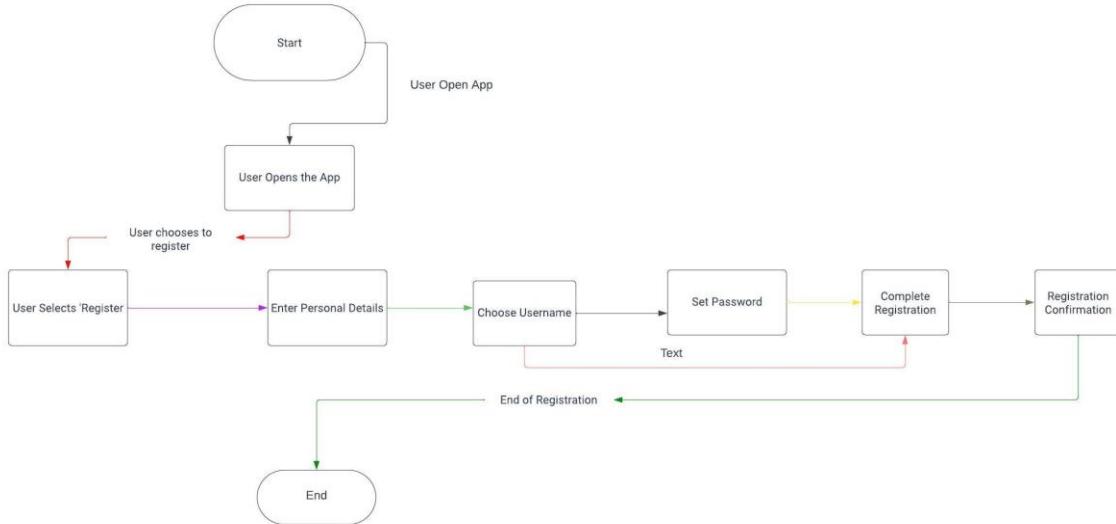


Figure 3.2 Flowchart Diagram for Expense Sharing App

### 3.3 USE CASE DIAGRAM

The Use Case Diagram for the Expense Analyzer App serves as an illustrative depiction of the key functionalities and interactions between various actors within the system. Central to this diagram are the primary actors—the "User" and the "Expense Analyzer App"—with the "Database" playing a supportive role.

The "User" actor, representing individuals engaging with the application, initiates essential use cases such as registering an account, logging in, entering expenses, viewing detailed reports, generating summaries, and securely logging out.

On the other side, the "Expense Analyzer App" actor, the central application facilitating these interactions, undertakes critical use cases like validating user data during registration, creating user accounts, verifying user credentials during login, handling expense entry and categorization, and retrieving data for reporting purposes. The "Database" actor, as an external entity, collaborates with the Expense Analyzer App to store user and expense data, forming a cohesive system. This Use Case Diagram provides a high-level overview of the system's behavior, offering a foundational understanding for stakeholders, developers, and

designers as they navigate the development and interaction aspects of the Expense Analyzer App.

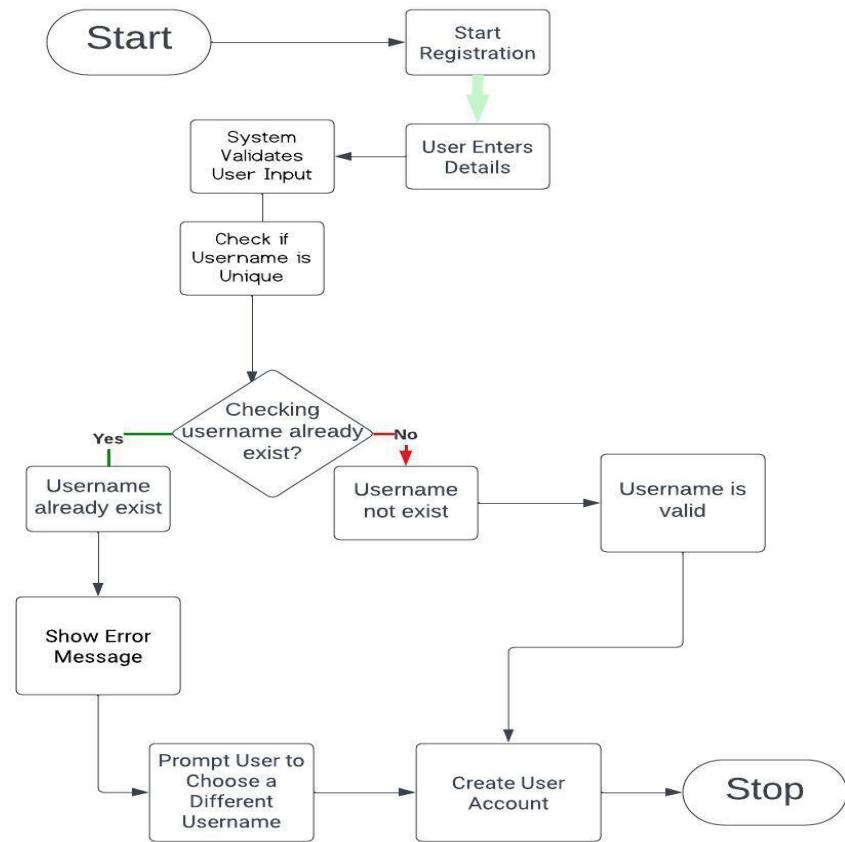


Figure 3.3 Use Case Diagram for Expense Sharing App

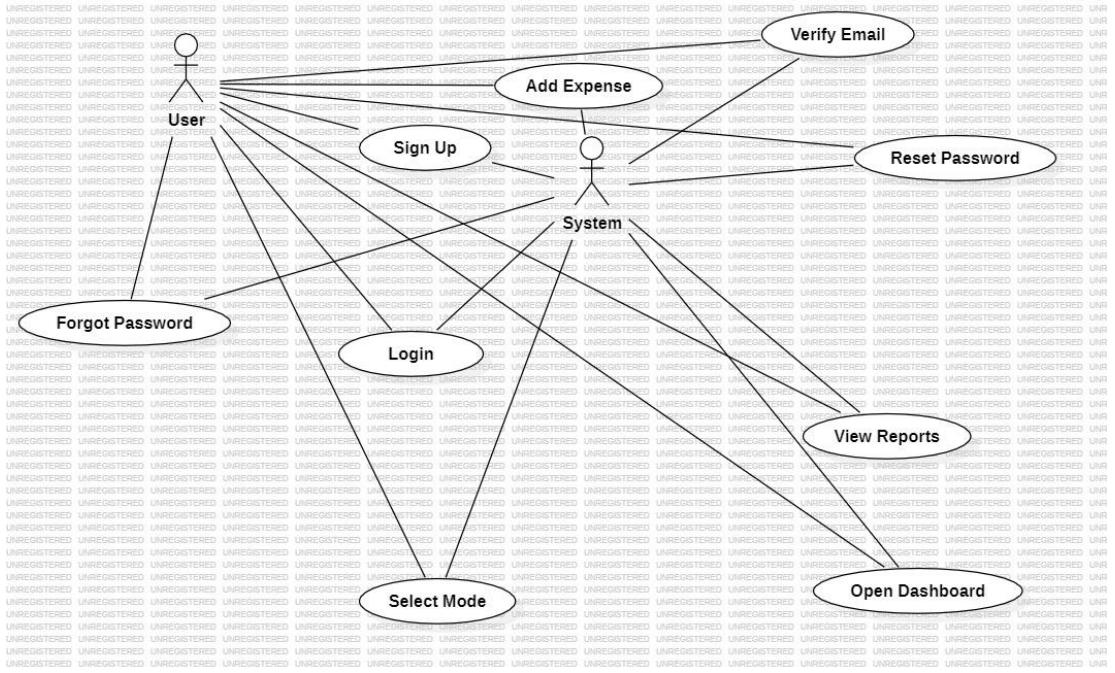


Figure 2.4 Use Case diagram for Expense Enigma App

### 3.3.1 Actors:

- User
- System

### 3.3.2 Use Case

- Select Mode
- Login
- Forgot Password
- Reset Password
- Sign Up
- Verify Email
- Open Dashboard
- Add Expense
- View Reports

In the expense sharing app, when the user opens the mode selection web page, they encounter two modes: normal mode and voice recognition mode. If the user clicks on normal mode, a login page appears where the user is prompted to enter their email and password. After entering the credentials, the user clicks on the login button, leading them to the dashboard. In the event that a user forgets their password, they can initiate the

password reset process. The user is required to enter their email, and a verification code is sent to their email address.

After receiving the verification code, the user enters it and then proceeds to reset their password. Once the new password is set, the user can log in using the updated credentials. For users who do not have an account, they can click on the signup option. During signup, the user is prompted to provide details such as their name, email, password, and budget limit. After entering this information, a verification code is sent to the user's email.

Upon entering the verification code and clicking on signup, the dashboard opens. Initially, the spending limit is set to 0, and the budget limit is based on the user's input. Within the app, there is a button to add expenses. When the user incurs an expense, they enter the expense details, including the amount and description. The app deducts the entered amount from the budget. After adding the expense, a chart displaying the budget limit and spending limit is updated.

If a user wishes to review their transaction history, they can click on the report option. This allows the user to access and view their transaction report.

### **3.4 DATA FLOW DIAGRAM**

The Data Flow Diagram (DFD) for the Share Expense app provides a visual representation of the flow of data within the system, illustrating how information moves between various components. At its core, the DFD encapsulates the key processes, data stores, and data flows involved in the expense management application. Starting with user inputs, such as registering, logging in, and entering expense details, the diagram delineates how these interactions trigger processes like data validation, expense calculations, and database updates. The DFD also portrays the storage and retrieval of user data in the database, emphasizing the seamless exchange of information between users and the application.

By encapsulating the fundamental data movements and transformations, the DFD serves as a valuable blueprint for understanding the Share Expense app's operational dynamics.

### 3.4.1 Level 0 Data Flow Diagram

Level 0 Data Flow Diagram will explain the basic flow of data in a system in which shows how user interacts and perform the app functionality.

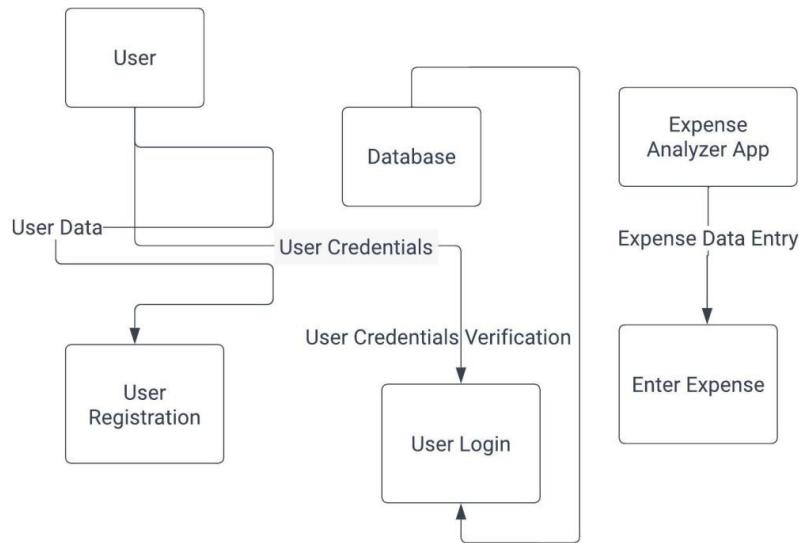


Figure 3.5 Level 0 DFD of Expense Sharing App

The Level 0 Data Flow Diagram (DFD), also known as the Context Diagram (**Fig. 3**), serves as a high-level representation of the Expense Analyzer App system, portraying it as a singular process and illustrating the interactions between the system and external entities. The primary entities involved in this context are the User, the Database (where expense data is stored), and the Expense Analyzer App itself.

#### 3.4.1.2 Entities:

- **User:** Represents the individuals utilizing the Expense Analyzer App for tasks such as registration, expense entry, and reporting.
- **Expense Analyzer App:** Represents the central application that users interact with to perform various tasks related to expense management.
- **Database:** Signifies the repository where expense data is stored, facilitating secure and organized data management.

### **3.4.1.3 Processes:**

- **User Interaction with the Expense Analyzer App:** This process encapsulates the various tasks that users can perform within the Expense Analyzer App, including registration, expense entry, and report generation.
- **Communication with the Database:** The Expense Analyzer App communicates with the Database to store and retrieve expense data efficiently, ensuring data accuracy and consistency.

### **3.4.1.4 Data Flow:**

- Users input and retrieve data directly from the Expense Analyzer App, enabling seamless interaction and data manipulation within the application.

## **3.4.2 Level 1 Data Flow Diagram**

At the more detailed Level 1 Data Flow Diagram (DFD), the User process is systematically broken down into distinct sub-processes, offering a granular view of the specific tasks users can perform within the Expense Analyzer App. This detailed breakdown further refines the interactions between the User, Database, and Expense Analyzer App entities.

### **3.4.2.1 Entities:**

- **User:** Represents individuals engaging with the Expense Analyzer App for various tasks.
- **Database:** The storage facility for expense data, ensuring secure and organized data management.
- **Expense Analyzer App:** The central application facilitating user interactions and managing communication with the database.

### **3.4.2.2 Processes:**

- **User Registration:** Sub-process wherein a user registers on the Expense Analyzer App by providing necessary details. Involves validation of user-provided information and updating the database with new user details.

- **User Login:** Sub-process allowing registered users to log into the Expense Analyzer App securely. Involves user authentication and validation against stored credentials in the database.
- **Enter Expense:** Sub-process enabling users to input new expense data into the Expense Analyzer App. Involves validation of entered expense details and updating the database with the new expense entry.
- **View Expense Report:** Sub-process where users can retrieve and view detailed reports of their expenses. Involves querying the database for relevant expense data and presenting it to the user through the Expense Analyzer App.
- **Generate Expense Summary:** Sub-process allowing users to generate summarized views of their expenses. Involves processing and aggregating expense data from the database to present concise summaries.
- **Logout:** Sub-process enabling users to securely log out of the Expense Analyzer App. Ensures the proper termination of the user session.

#### **3.4.2.3 Data Flow:**

- Users interact with the Expense Analyzer App to execute specific tasks, ranging from registration to generating expense summaries.
- The Expense Analyzer App, acting as the intermediary process, communicates with the Database to retrieve or update expense data based on user actions.
- For instance, during User Registration, user details are sent to the Database for storage.

#### **3.4.3 Level 2 Data Flow Diagram**

At the Level 2 Data Flow Diagram (DFD), the detailed breakdown of each user process provides a more intricate view of the sub-processes involved in specific tasks within the Expense Analyzer App. This level of detail offers insights into the intricacies of data flow and interactions between the User, Database, and Expense Analyzer App entities.

### **3.4.2.1 Entities:**

- **User:** Represents individuals interacting with the Expense Analyzer App for various tasks.
- **Database:** The storage facility for expense and user data, ensuring secure and organized data management.
- **Expense Analyzer App:** The central application facilitating user interactions and managing communication with the database.

### **3.4.2.4 Processes:**

- **User Registration:**
  - ✓ **Validate User Data:** Sub-process involves checking the validity of user-provided information during the registration process.
  - ✓ **Create User Account:** Sub-process for creating a new user account, involving the storage of validated user data in the database.
- **User Login:**
  - ✓ **Verify User Credentials:** Sub-process for authenticating user credentials during the login process.
- **Enter Expense:**
  - ✓ **Validate Expense Data:** Sub-process ensuring the accuracy and integrity of the entered expense data.
  - ✓ **Categorize Expense:** Sub-process involving the selection and assignment of expense categories for better organization.
- **View Expense Report:**
  - ✓ **Calculate Expense Totals:** Sub-process involving the computation of total expenses based on the retrieved data.
  - ✓ **Generate Summary Report:** Sub-process for creating a summarized report based on the calculated totals.

- **Generate Expense Summary:**

- ✓ **Calculate Expense Totals:** Sub-process involving the computation of total expenses based on the retrieved data.
- ✓ **Generate Summary Report:** Sub-process for creating a summarized report based on the calculated totals.
- ✓ **Logout:** Sub-process enabling users to securely log out of the Expense Analyzer App, terminating the user session.

#### 3.4.2.5 Data Flow:

- Each sub-process communicates with the Expense Analyzer App and may involve interactions with the database.
- For instance, during User Registration, the validated user data is communicated to the Expense Analyzer App, which, in turn, communicates with the database to store the new user account.

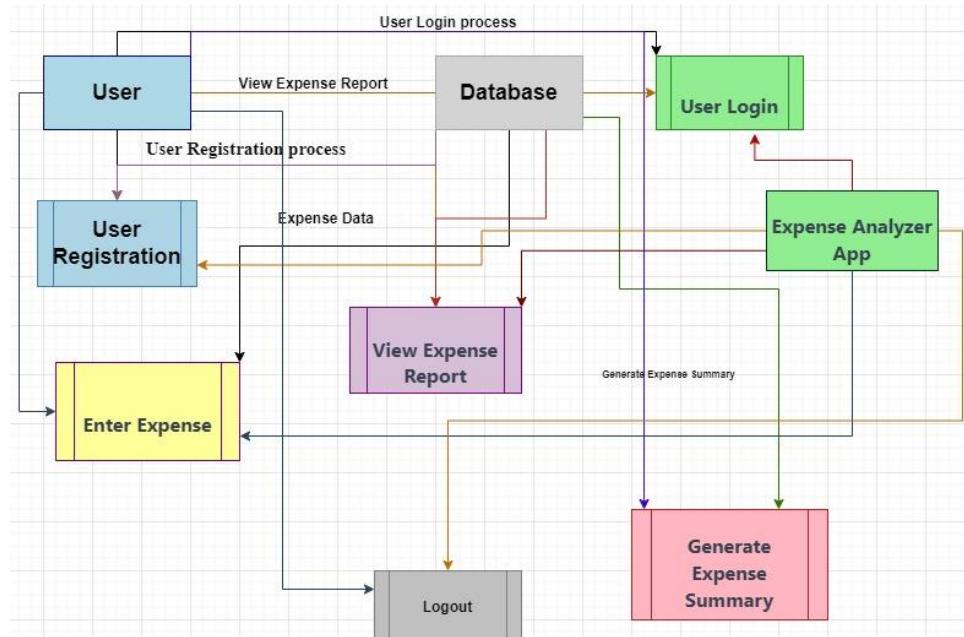


Figure 3.6 Level 2 DFD of Expense Sharing App

# CHAPTER 4

## FORM DESIGN

### 4. INTRODUCTION

The form design for the Share Expense app is a fundamental aspect that governs the user interface, providing a structured and intuitive framework for users to input and interact with data. This design aims to create a seamless and user-friendly experience by organizing elements such as text fields, dropdowns, and buttons in a visually appealing and logically arranged manner. Users engage with the form to register, log in, and input expense details, and the design emphasizes clarity, simplicity, and efficiency in capturing accurate information. By carefully considering user input requirements, validation processes, and responsive design principles, the form design ensures a positive user experience, encouraging user engagement and fostering effective communication between users and the application.

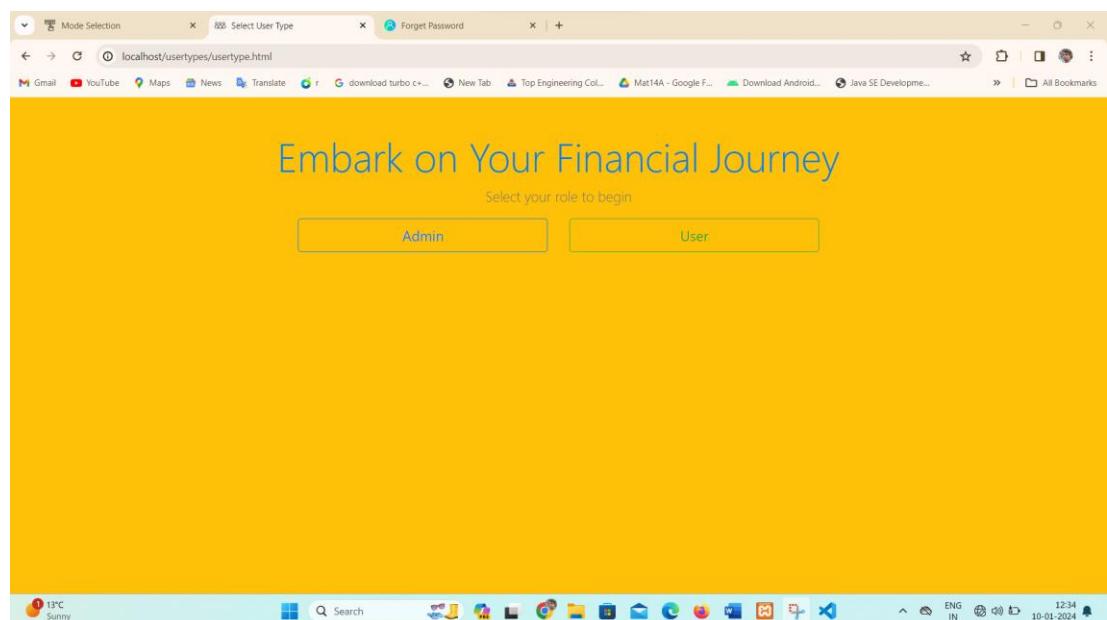


Figure 4.1 Users/Admin login Credentials form

## 4.1 INPUT/OUTPUT FORM (Screenshot)

### 4.1.1 Mode Selection Form

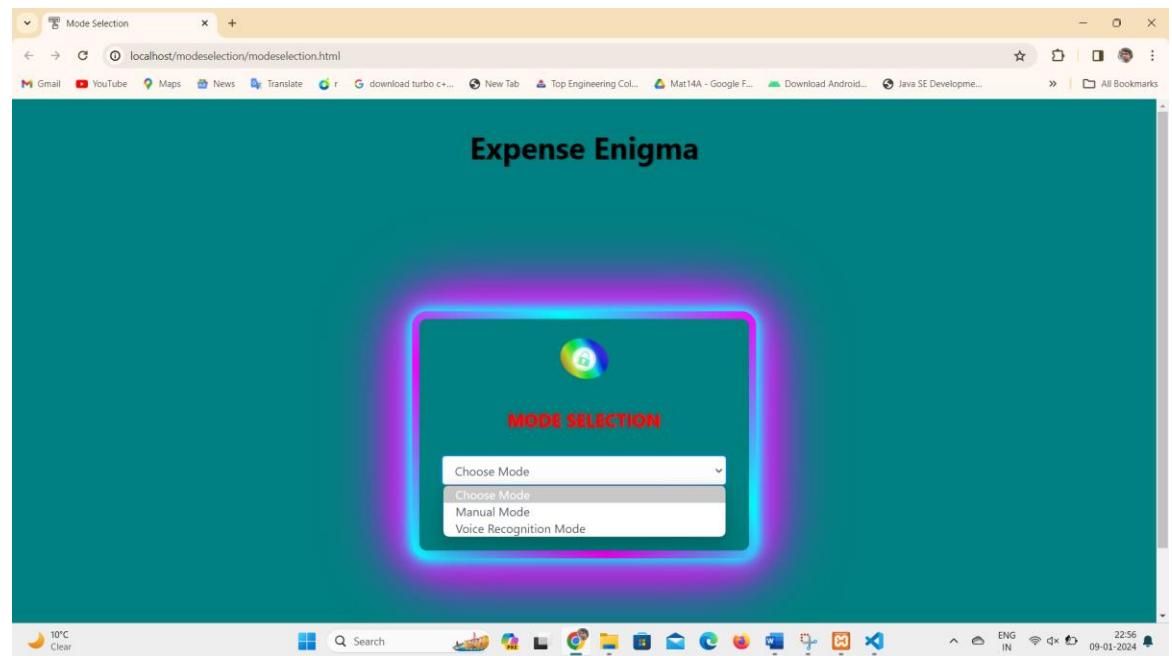


Figure 4.2 Initially user select Mode Preference

### 4.1.2 Role Selection Form

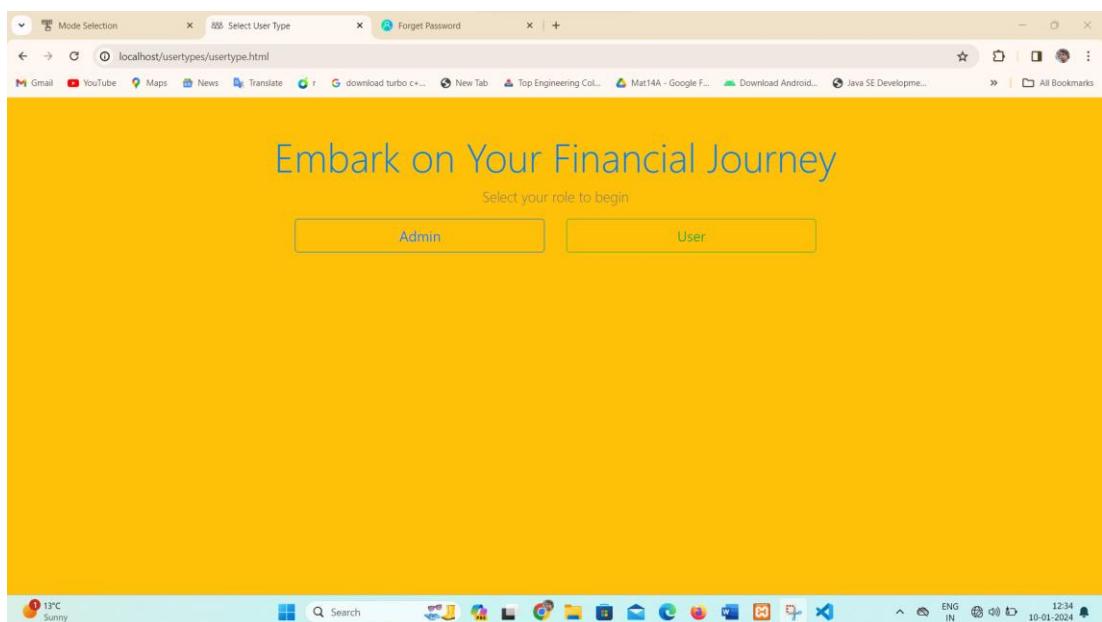


Figure 4.3 Select Role either: User, Admin

#### 4.1.3 Admin Module: Admin can login with the Credentials

- ✓ **Email:** [anmol@gmail.com](mailto:anmol@gmail.com)
- ✓ **Password:** Anmol@1819
- ✓ If admin entered incorrect credentials, then alert will generated but if credentials are matched with the admin credentials then admin can login.

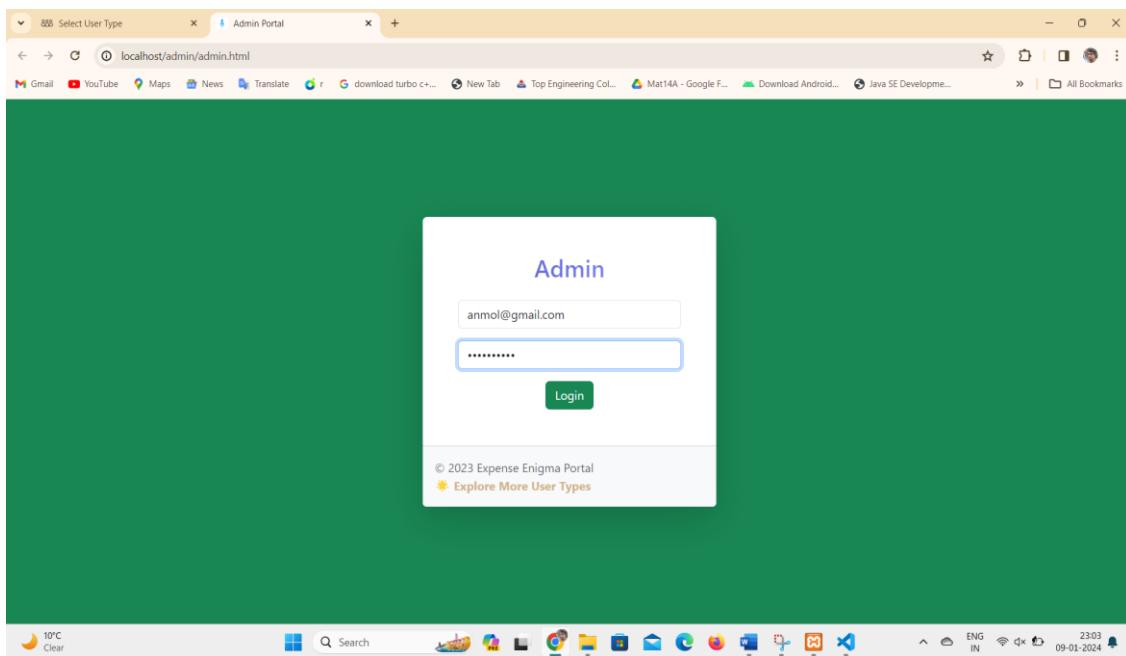


Figure 4.4 Admin Entered Correct Credentials

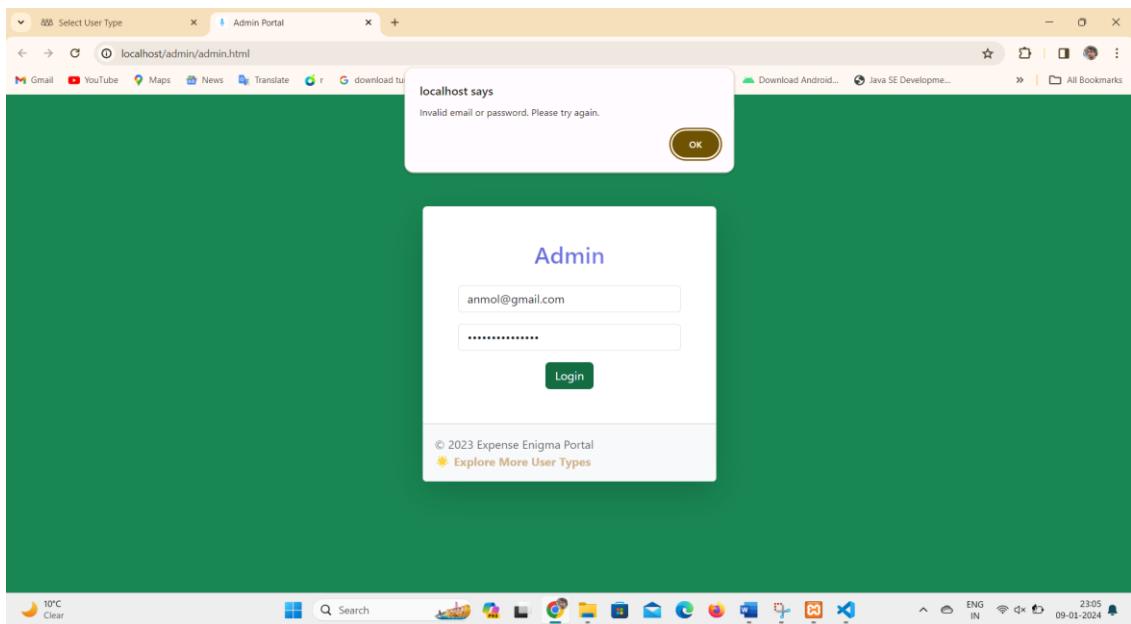


Figure 4.5 Admin Entered Wrong Credentials

#### 4.1.4 Admin Dashboard:

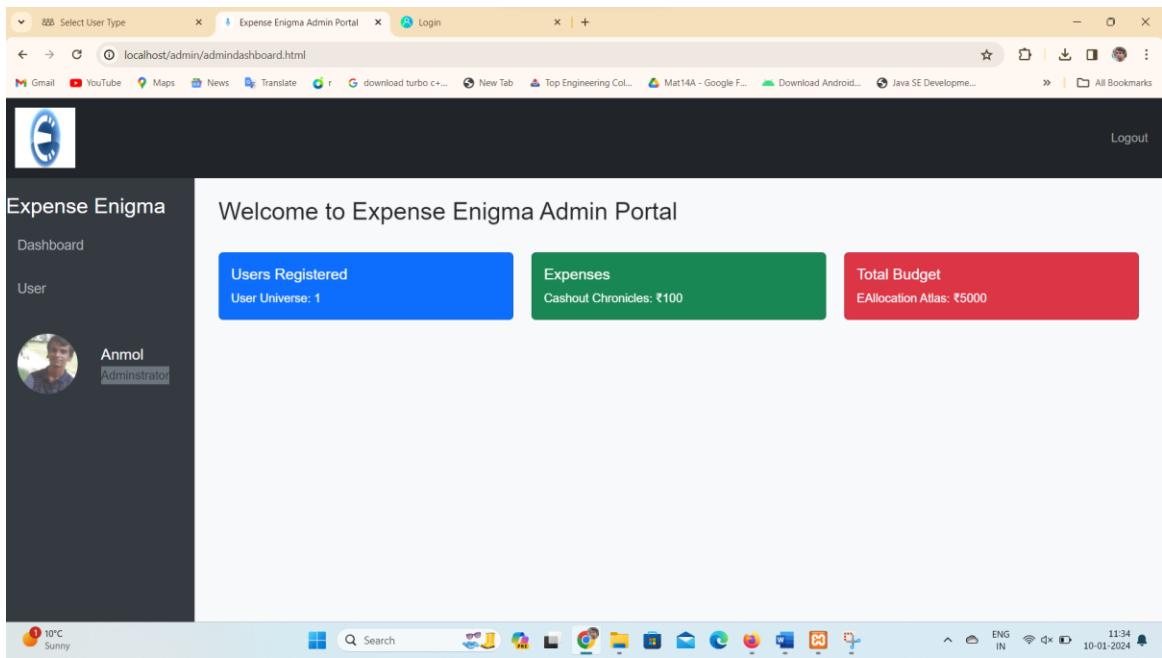


Figure 4.6 Admin Dashboard

#### 4.1.5 Admin- users Dashboard:

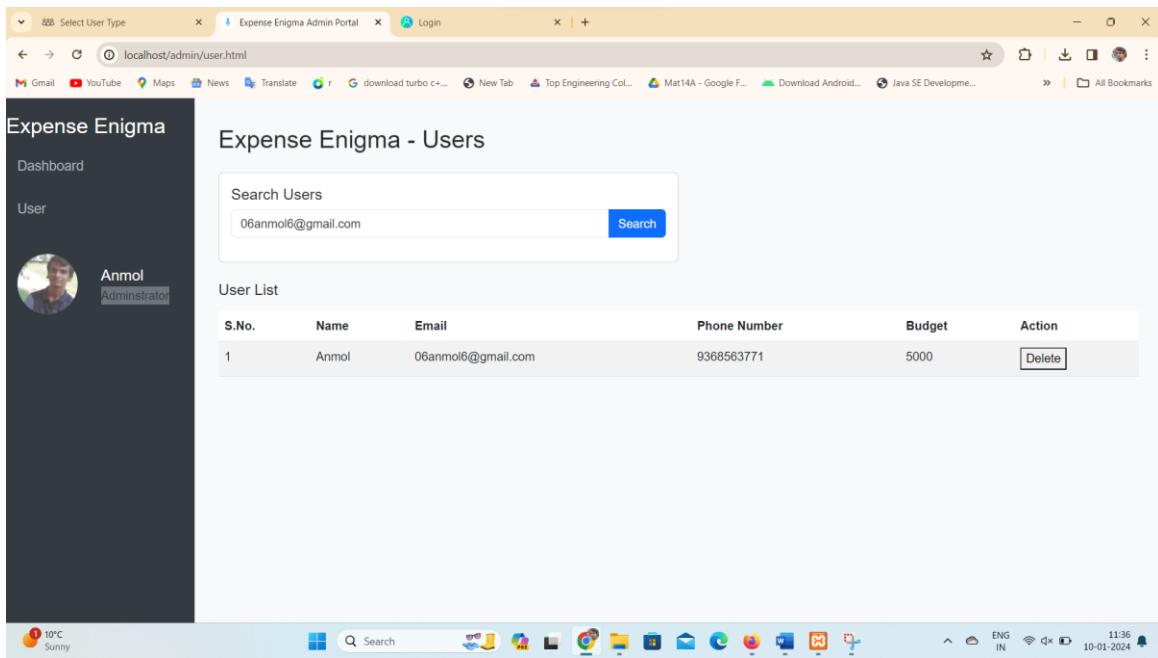


Figure 4.7 Admin-user Dashboard

#### 4.1.6 User Login Form:

- **Email:** User has to enter their email for login in Expense Enigma App.
- **Password:** User enter the password which they have entered during registration.
- **Forgot Password:** If user forget his password, then user click on forget password. After clicking on forget password the user has to enter the email then a verification link will be come to user email for resetting the password.
- **Update Password:** User can update their password through reset password option.

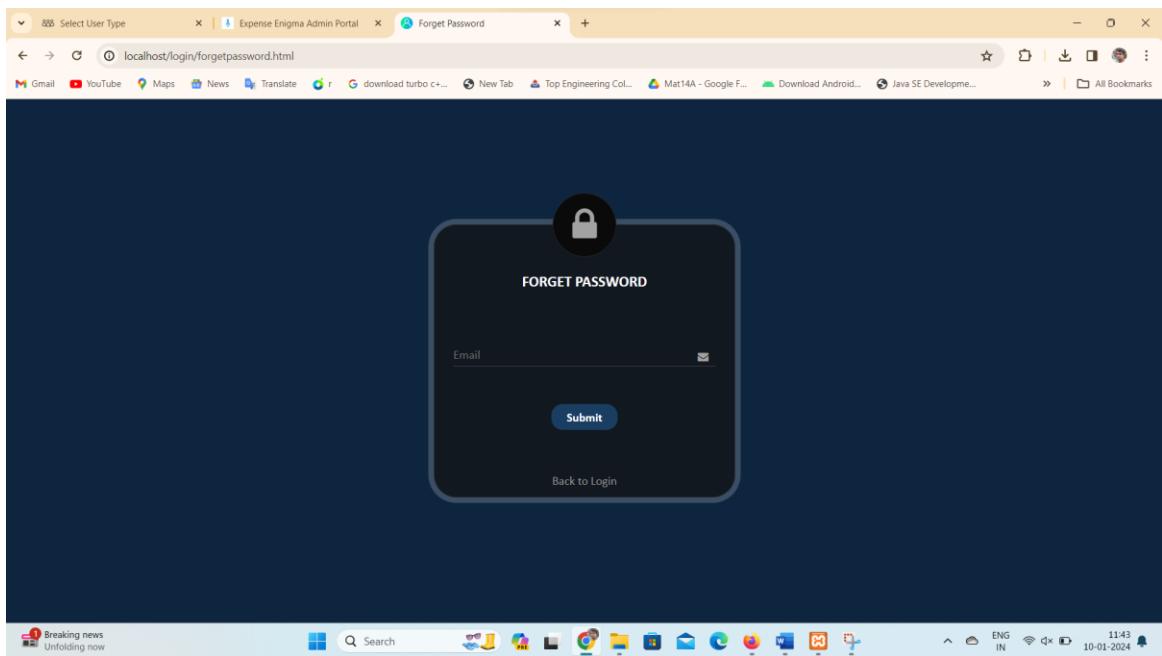


Figure 4.8 Forget Password Module

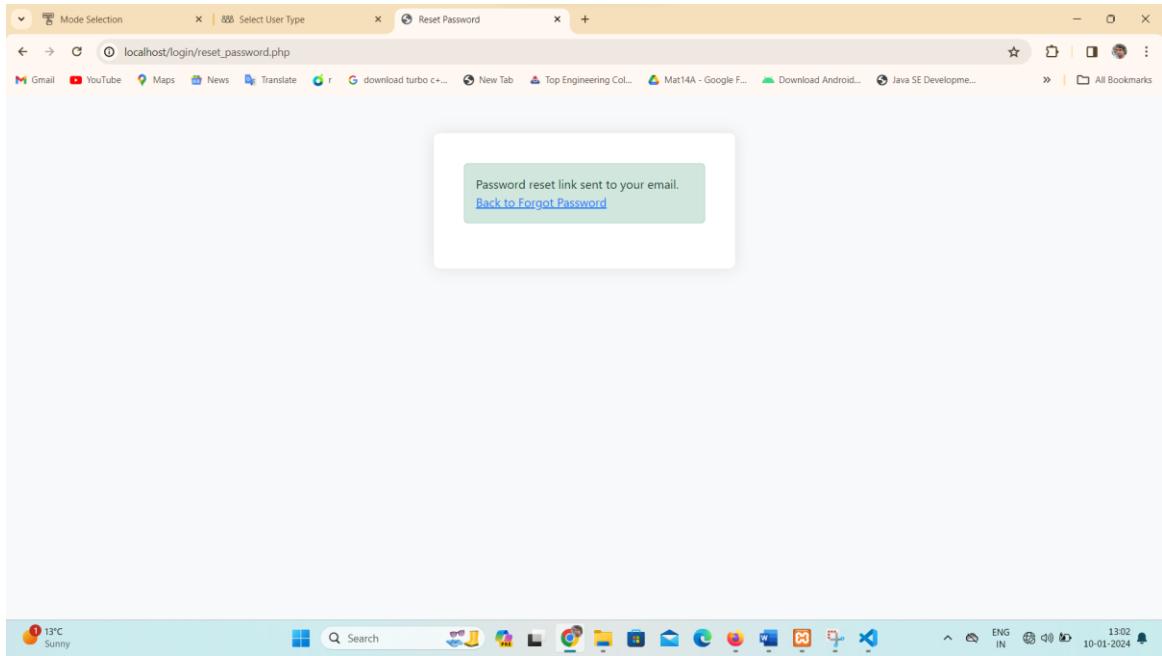


Figure 4.9 Password Reset Link Generate

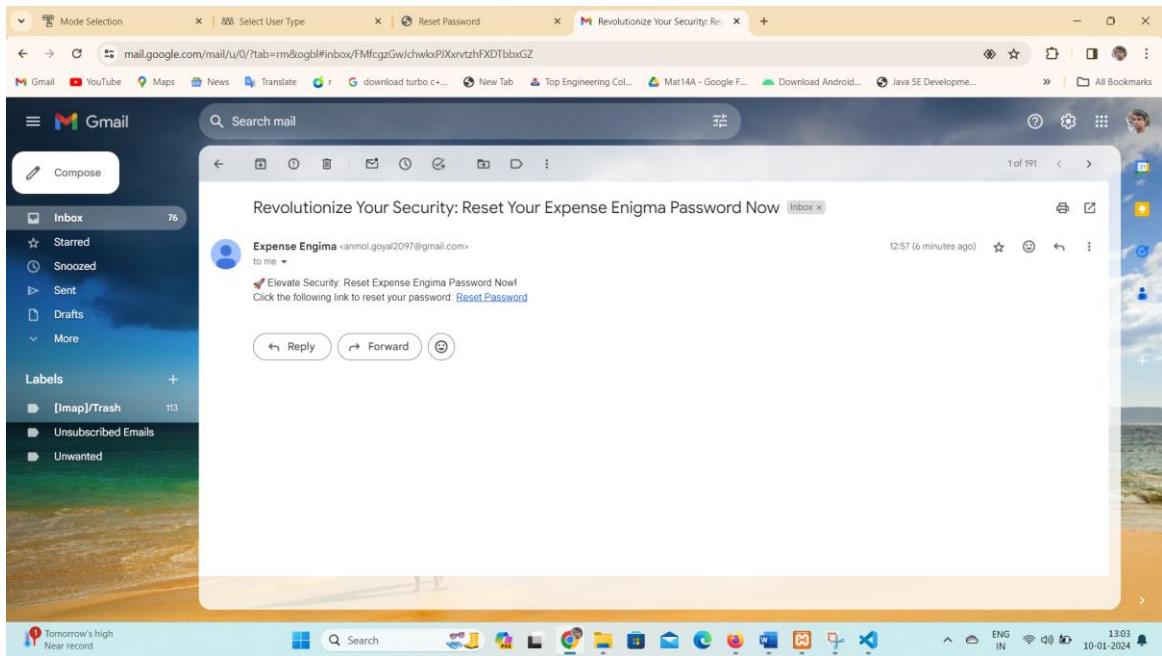


Figure 5.1 Email Reset Link Generate

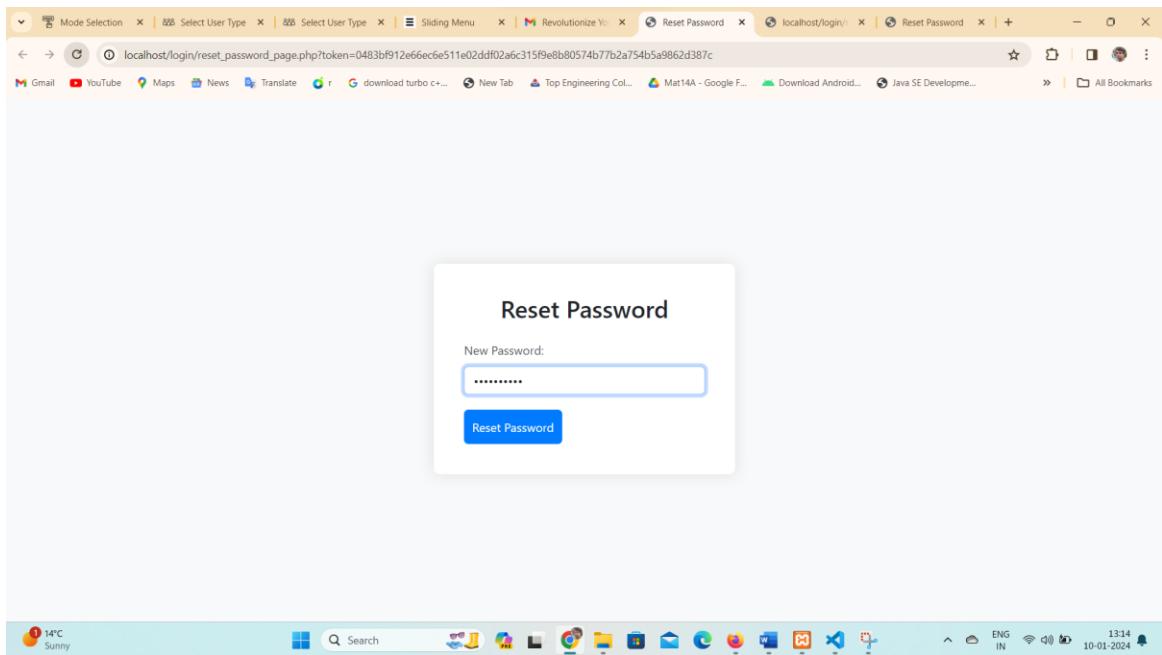


Figure 5.2 Reset Password

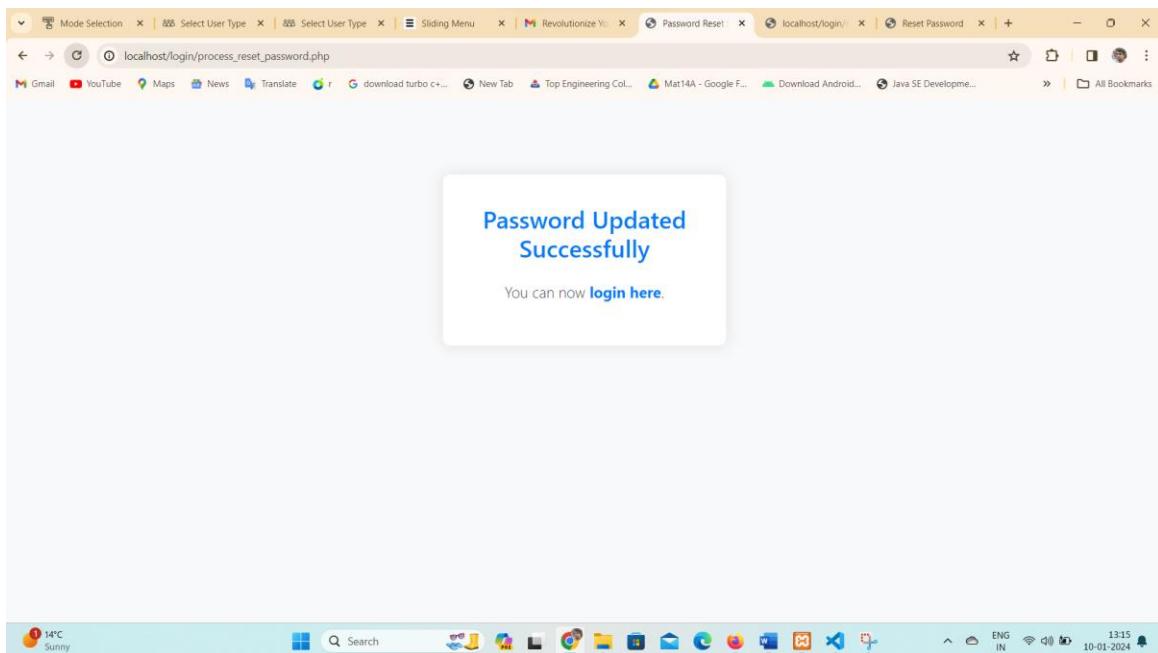


Figure 5.3 Password Updated Successfully

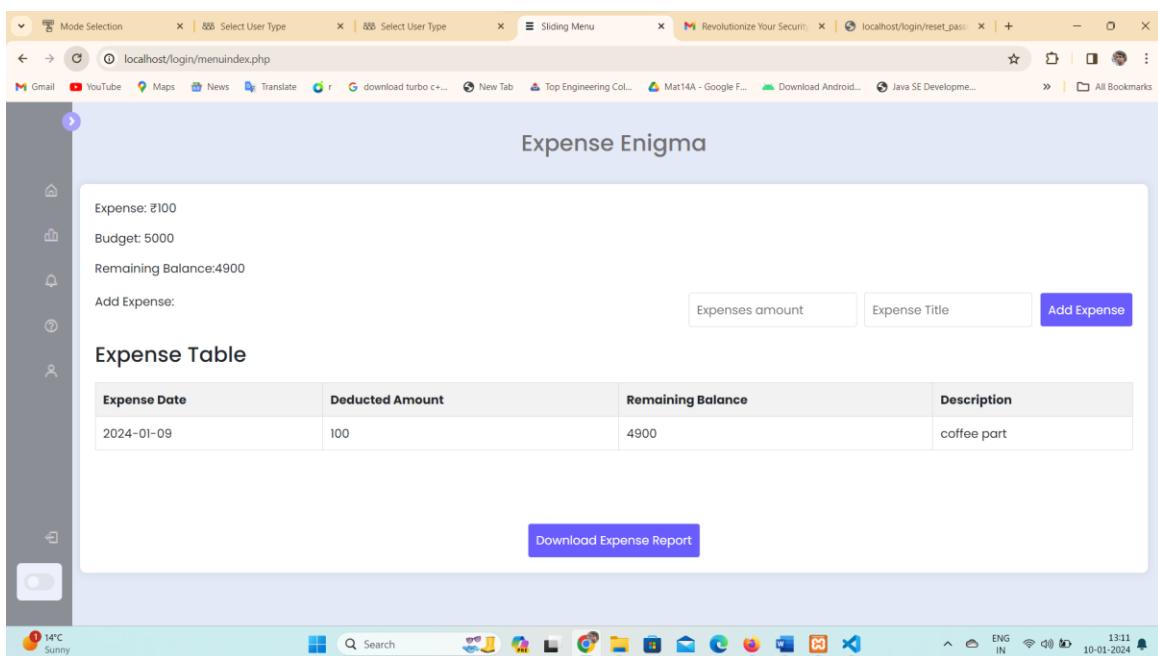


Figure 5.4 User Expense Dashboard

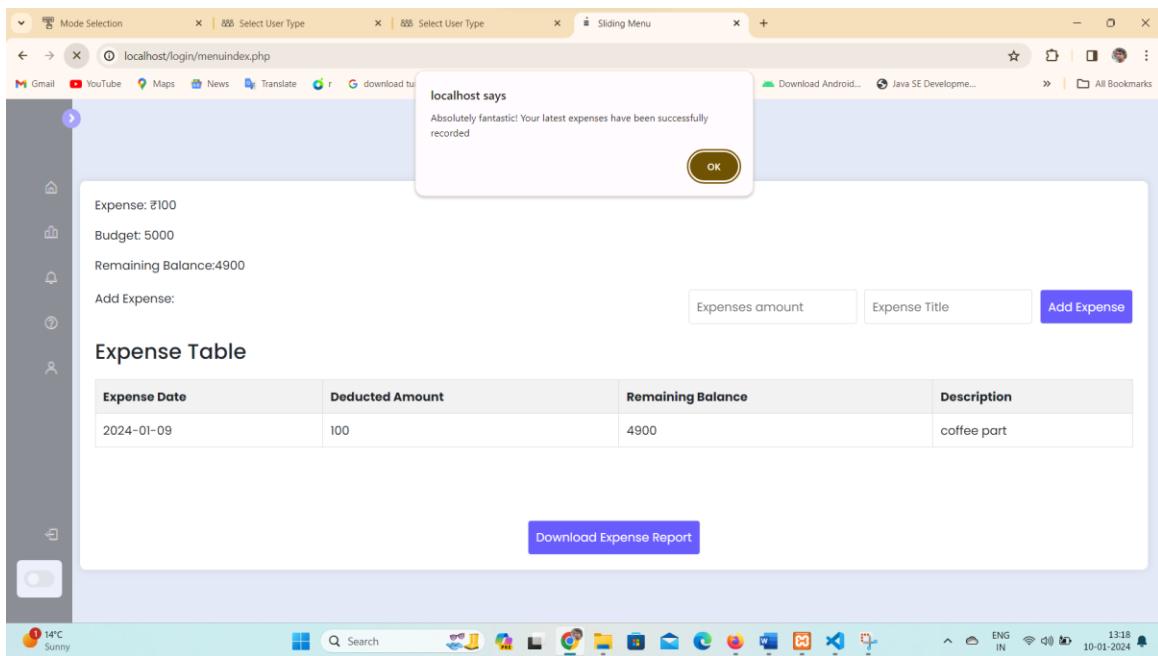


Figure 5.5 Adding expense alert

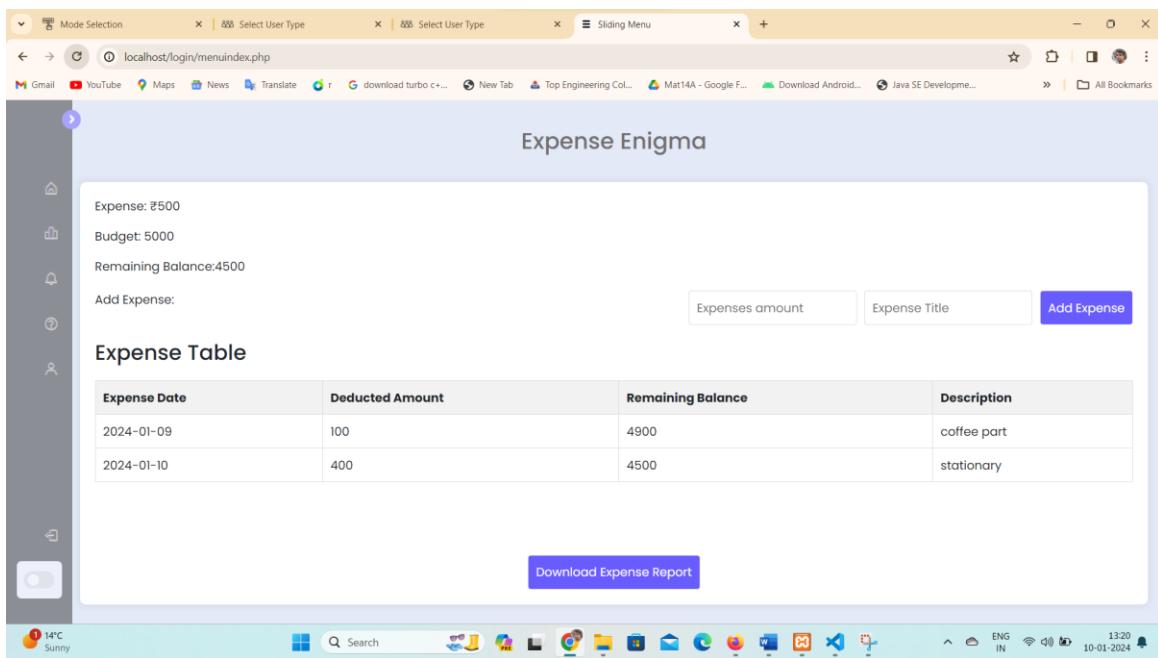


Figure 5.6 New Expenses Added successfully

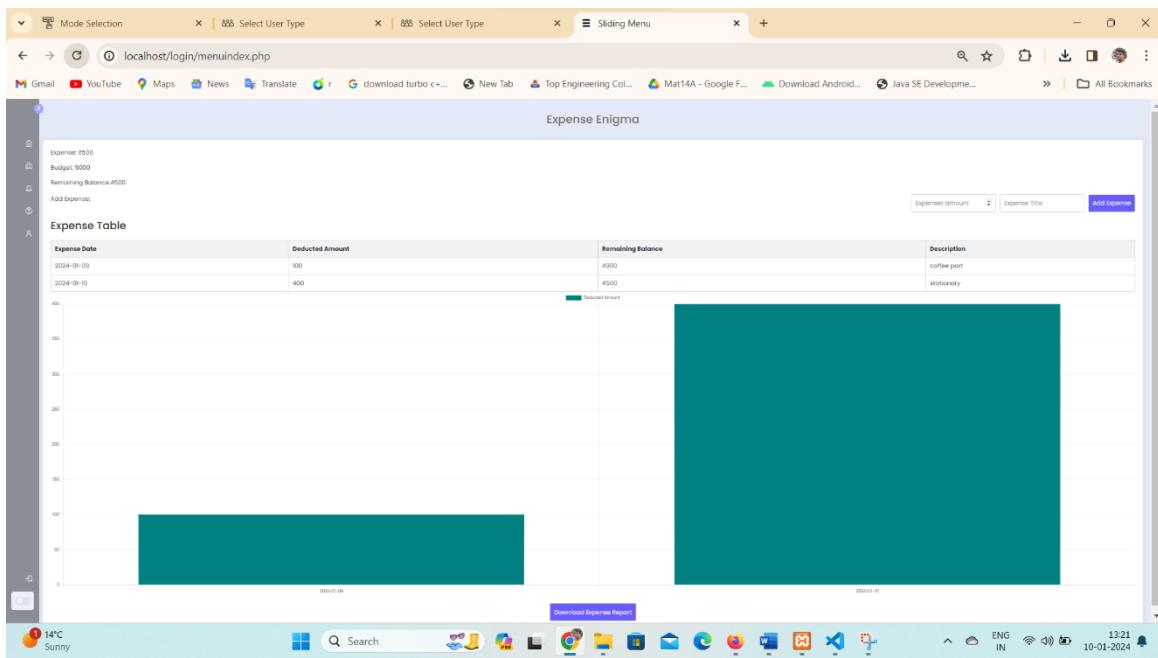


Figure 5.7 Graphical Expense Analysis

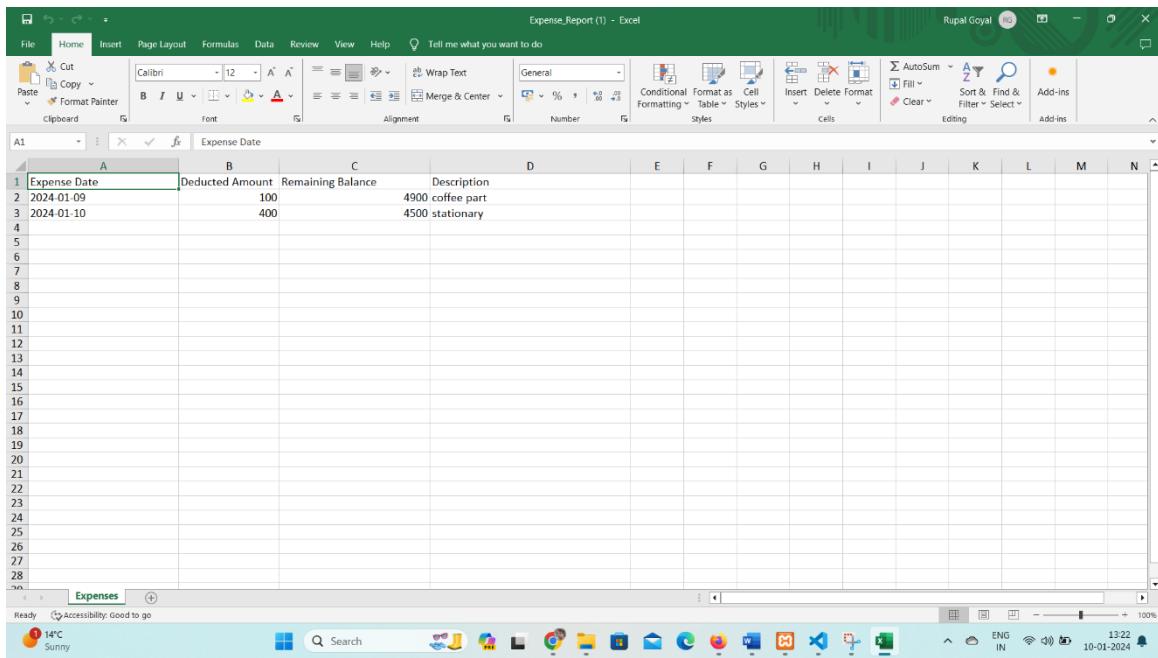


Figure 5.8 Expense Spreadsheets View

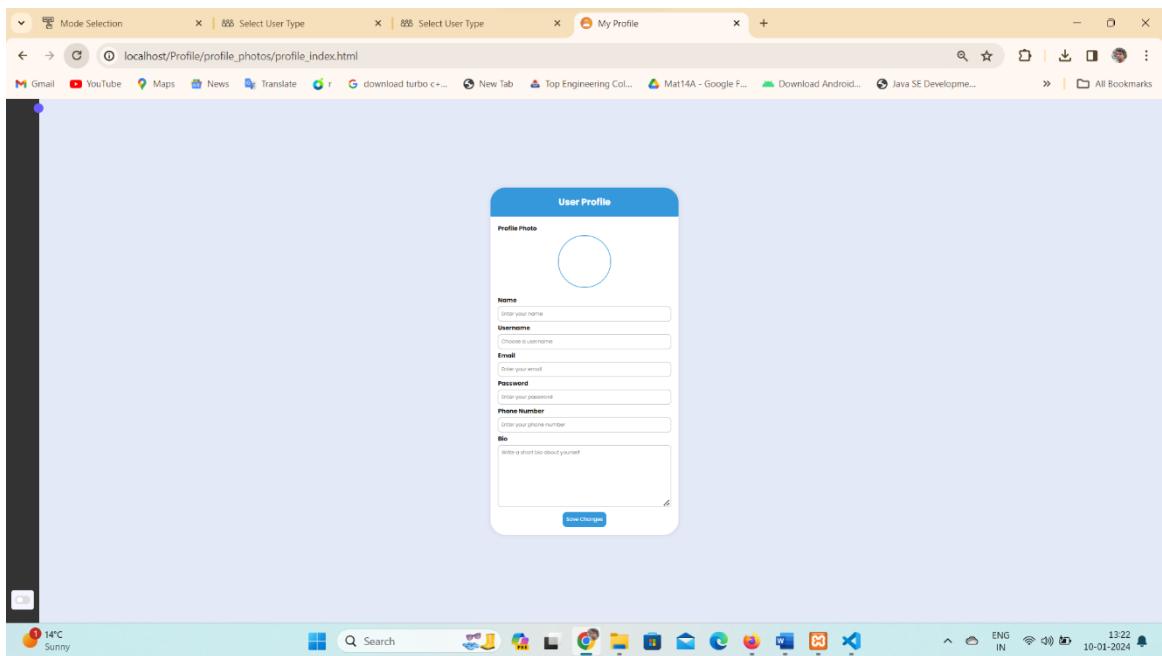


Figure 5.9 User Profile Page

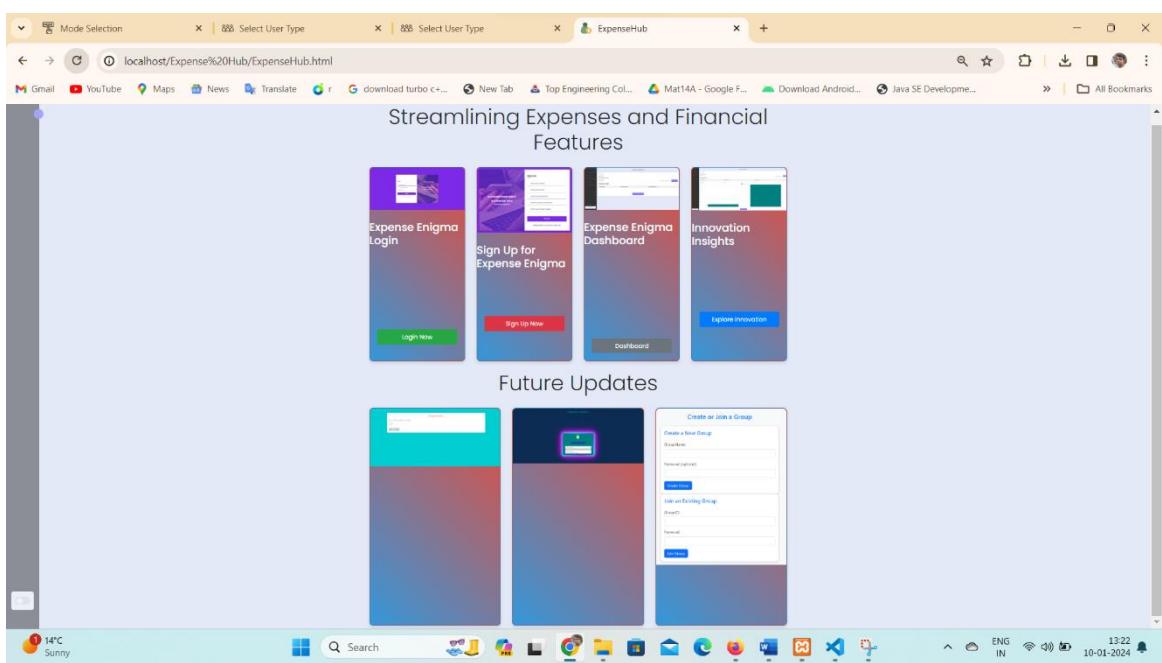


Figure 6.1 Financial Features

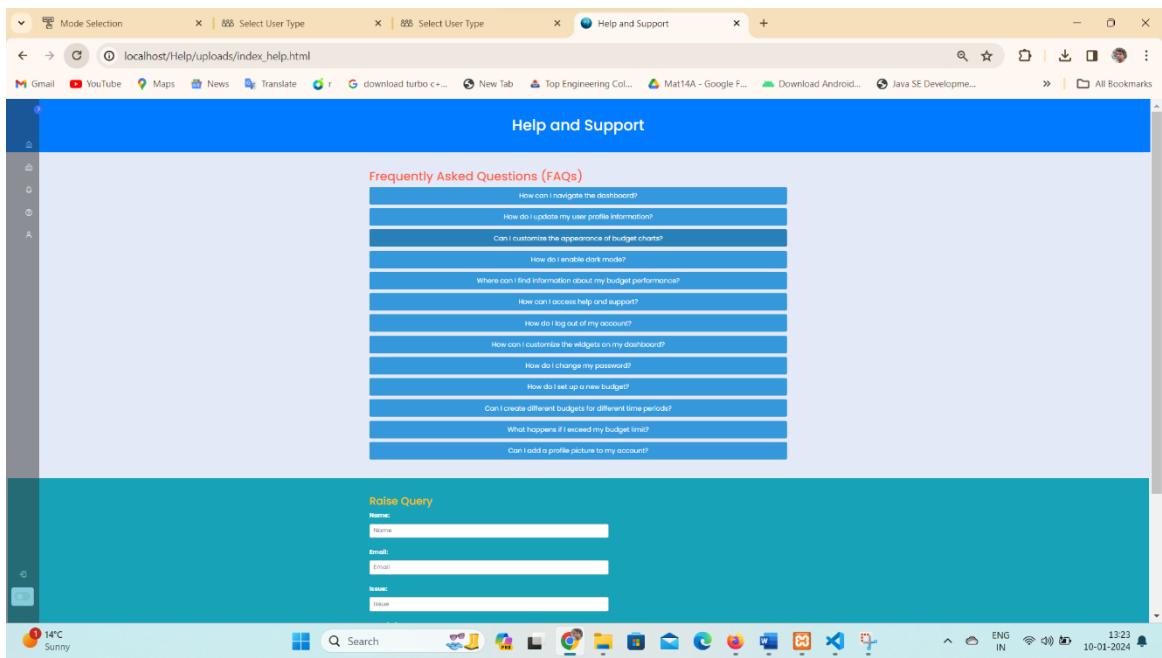


Figure 6.2 Help and Support Page

A screenshot of a "Raise Query" support form. The form has a yellow header bar with the title "Raise Query". It contains several input fields: "Name" (text input), "Email" (text input), "Issue" (text input), "Description" (text area), and "File Upload" (file input). At the bottom right of the form is a "Raise Query" button.

Figure 3.3 Raise Query Support

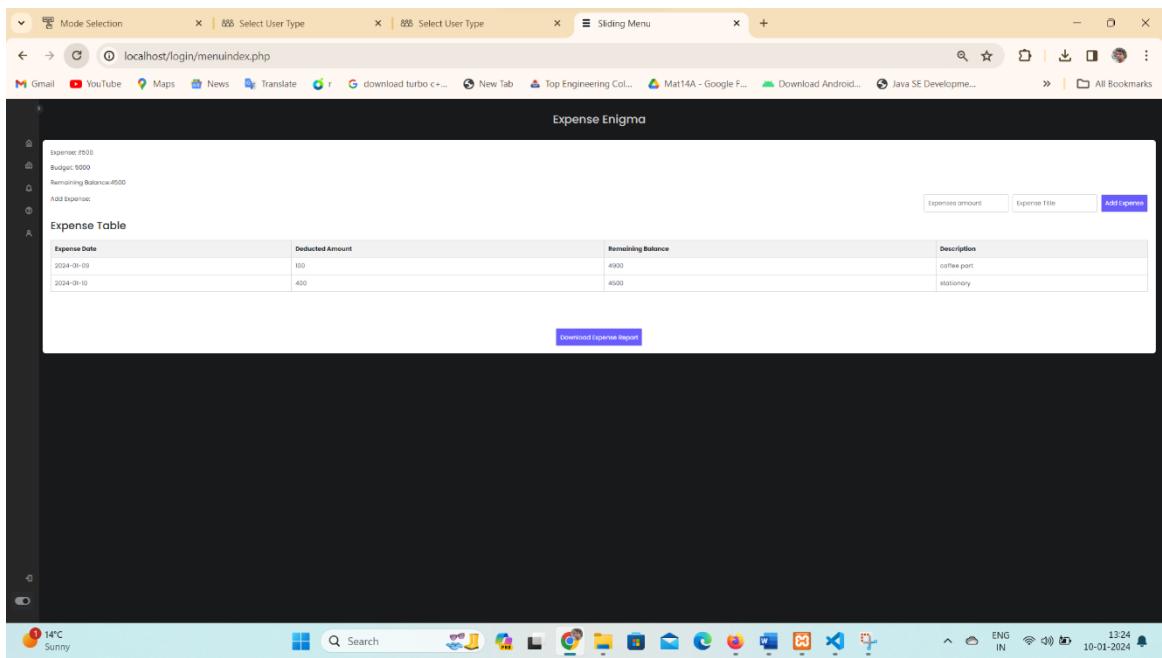


Figure 6.4 Dark Mode Enabled Feature

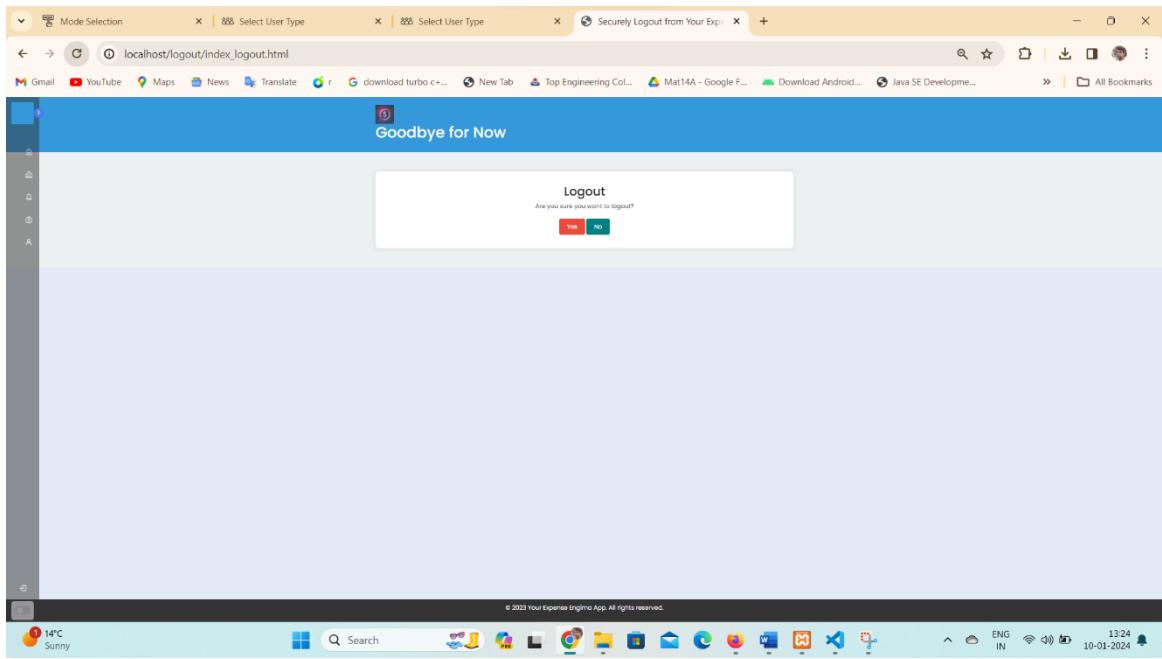


Figure 6.5 Logout Expense Enigma Services

# **CHAPTER 5**

## **CODING**

### **5. INTRODUCTION**

Coding serves as the backbone of the Share Expense app, translating the conceptual framework into a functional and interactive software solution. This intricate process involves the implementation of algorithms, data structures, and user interface elements to bring the envisioned features to life. The codebase is meticulously crafted using programming languages such as Python, JavaScript, or others, chosen for their suitability to the project's requirements. The coding phase encompasses various aspects, including backend logic for user authentication, expense calculations, and database interactions, as well as frontend development for creating an intuitive and visually appealing user interface. Through diligent coding practices, adherence to coding standards, and continuous testing, the development team aims to produce a robust and reliable Share Expense app that seamlessly aligns with user expectations and operational requirements.

#### **5.1. Module wise Coding**

##### **5.1.1 Mode Selection**

###### **5.1.2 modeselection.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="description" content="Expense Sharing App" />
    <meta name="viewport" content="width=device-width,initial-scale=1.0,shrink-to-fit=no" />
```

```

<meta name="theme-color" content="#007bff" /> <title>Mode Selection</title>
<link rel="icon" href="https://cdn-icons-png.flaticon.com/128/3653/3653239.png"
type="image/x-icon"> <link rel="stylesheet"
ref="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css" integrity="sha512-
+J+J+qJQvzJvJvJvJvJvJvJvJvJvJvJvJvJvJvJvJvJvJvJvJvJvJvJw==" crossorigin="anonymous" referrerPolicy="no-referrer" />
<link rel="stylesheet" href="modestyle.css">
</head> <body> <h1 class="text-center">Expense Enigma</h1>
<div id="main-container" class="centered-flex">
<div class="form-container"> <div class="icon centered-flex">
<i class="fa fa-expeditedssl"></i>
</div> <div class="Mode">MODE SELECTION</div>
<form id="login-form" class="centered-flex">
<div class="msg"></div> <div class="field"> <select
id="Mode_Selection" class="form-control">
<option value="Choose Mode" selected disabled>Choose Mode</option>
<option value="manual-mode" id="manual-mode">Manual
Mode</option> </select> </div> <div class="btn-container"><button
id="btn" class="btn btn-primary btn-block">Submit</button></div> </form>
</div> </div> <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zY4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js">
</script> <script src="modeselection.js"></script>
</body> </html>

```

**5.1.2 modestyle.css**

```

@import url('https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css');

* { margin: 0; padding: 0; box-sizing: border-box; }

```

```

body { background-color: teal; }

h1 { font-size: 40px; font-weight: bold;
    text-align: center; margin-top: 40px;
    margin-bottom: 10px; color: black; text-shadow: 2px 1px 1px rgba(0.2, 0.2, 0.2,
0.2); }

#main-container { display: flex; justify-content: center;
    align-items: center; min-height: 100vh; }

.form-container { background-color: teal;
    border-radius: 10px; box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
    padding: 30px; text-align: center; width: 90%; max-width: 437px;
    margin: 20px; position: relative; }

.form-container::before {
    content: ""; position: absolute; top: -10px; left: -10px;
    right: -10px; bottom: -10px; border-radius: 20px; background: linear-
gradient(45deg, #00ffff, #ff00ff, #00ffff, #ff00ff); z-index: -1; animation: glowing-
border 2s ease-in-out infinite; }

@keyframes glowing-border {
    0% { box-shadow: 0 0 10px #00ffff, 0 0 20px #00ffff, 0 0 30px #00ffff, 0 0 40px
#ff00ff, 0 0 70px #ff00ff, 0 0 80px #ff00ff, 0 0 100px #ff00ff; }
    100% { box-shadow: 0 0 20px #00ffff, 0 0 30px #00ffff, 0 0 40px #00ffff, 0 0
50px #ff00ff, 0 0 80px #ff00ff, 0 0 90px #ff00ff, 0 0 110px #ff00ff; } }

.icon { position: relative; display: inline-block; font-size: 30px; color: #fff;
    text-shadow: 0 0 10px #fff; animation: rotate 2s linear infinite;
    border-radius: 50%; width: 60px; }

label[for="mode-selection"] { padding-left: 70px; }

.icon:before { content: ""; position: absolute;
    top: -10px; left: -10px; right: -10px; bottom: -10px;
    border-radius: 50%; background: linear-gradient(45deg, #ff0000, #ff8000, #ffff00,
#00ff00, #00ffff, #0000ff, #8000ff, #ff00ff); z-index: -1; animation: pulse 2s ease-in-
out infinite; }

@keyframes rotate { 0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); } }

@keyframes pulse { 0% { transform: scale(0.8); opacity: 0.7; }
    50% { transform: scale(1.2); opacity: 1; }
    100% { transform: scale(0.8); opacity: 0.7; } }

.Mode { font-size: 24px; font-weight: bold; margin-bottom: 20px; margin-top: 41px;
    animation: changeColorAndGlow 4s ease-in-out infinite;
    text-shadow: 0 0 5px rgba(255, 0, 0, 0.8); }

@keyframes changeColorAndGlow { 0%
    { color: red; }
    50% { color: blue; }
    100% { color: green; } }

#login-

```

```

form { display: flex; flex-direction: column; } .msg { color: red; margin-bottom: 10px; } .field { margin-bottom: 20px; }
.btn-container { display: flex; justify-content: center; }
.btn-primary { background-color: #007bff; border-color: #007bff; transition: all 0.3s ease; }
.btn-primary:hover { background-color: #0069d9; border-color: #0062cc; }
.btn-primary:focus, .btn-primary:active {
  box-shadow: none; outline: none;
}
.btn-clicked { box-shadow: none; outline: none; background-color: #6c757d; border-color: #6c757d; color: #fff; }

@media (max-width: 576px) {
  #main-container {
    padding: 20px;
  }
  .form-container { width: 100%; padding: 20px; }
}

```

### 5.1.3 modestyle.js

```

const submitBtn = document.getElementById("btn");

const modeSelection = document.getElementById("Mode_Selection"); // Corrected ID

submitBtn.addEventListener("click", function(event) {
  event.preventDefault(); if (modeSelection.value === "manual-mode") {
    fetch("save_data.php", {
      method: "POST",
      body: new URLSearchParams(new FormData(document.getElementById("login-form")))
    })
    .then(response => response.text())
    .then(data => { console.log(data);
      window.location.href = "http://localhost/usertypes/usertype.html"; })
    .catch(error => console.error("Error:", error)); } );

```

### **5.1.3 save\_data.php**

```
<?php  
$host = "localhost"; $user = "root";  
$password = ""; $database = "expense_sharing_app";  
$conn = new mysqli($host, $user, $password, $database);  
if ($conn->connect_error) { die("Connection failed: " . $conn->connect_error);}  
if ($_SERVER["REQUEST_METHOD"] == "POST") { $modeSelection =  
$_POST["Mode_Selection"]; $sql = "INSERT INTO expense_data (mode_selection)  
VALUES ('$modeSelection')"; if ($conn->query($sql) === TRUE) {  
echo "Data inserted successfully"; } else {  
echo "Error: " . $sql . "<br>" . $conn->error; }}  
$conn->close(); ?>
```

### **5.1.2 User Types**

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Select User Type</title>  
    <link rel="icon" type="image/x-icon"  
        href="https://static.thenounproject.com/png/3311644-200.png">  
    <link rel="stylesheet"  
        href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">  
</head> <body class="bg-warning">  
    <div class="container text-center mt-5">  
        <h1 class="display-4 text-primary">Embark on Your Financial Journey</h1>  
        <p class="lead text-muted">Select your role to begin</p>  
        <div class="row justify-content-center"> <div class="col-md-4">  
            <a href="/admin/admin.html" target="_blank" class="btn btn-outline-primary  
            btn-lg btn-block">Admin</a>  
        </div> <div class="col-md-4">  
            <a href="/login/login.html" target="_blank" class="btn btn-outline-success btn-  
            lg btn-block">User</a>  
        </div> </div> </div>
```

```

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js">
</script> <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body> </html>

```

### 5.1.3 Admin Module

#### 5.1.3.1 admin.html

```

<!DOCTYPE html> <html lang="en"> <head>    <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Portal</title>    <link rel="icon" type="image/x-icon"
        href="https://t4.ftcdn.net/jpg/02/27/45/09/360_F_227450952_KQCMShHPOPeb
        UXkIULsKsROk5AvN6H1H.jpg"> <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
        rel="stylesheet"> </head> <body class="bg-success">
    <div class="container d-flex justify-content-center align-items-center"
        style="height: 100vh;"> <div class="card border-0 shadow-lg innovative-bg-
        color" style="max-width: 400px; width: 100%;"> <div class="card-body p-5 text-
        center"> <h2 class="mb-4 text-gradient-primary ">Admin </h2> <form>
        <div class="mb-3"><input type="email" autocomplete="username" class="form-
        control" id="email" placeholder="Email" required> </div> <div class="mb-3">
        <input type="password" autocomplete="current-password" class="form-control"
        id="password" placeholder="Password" required> </div> <button type="button"
        class="btn btn-success btn-block" onclick="authenticate()">Login</button>
        </form> </div> <div class="card-footer bg-light text-muted py-3"> <p
        class="mb-0">&copy; 2023 Expense Enigma Portal</p> <a
        href="/usertypes/usertype.html" title="Return to the Gateway of Possibilities"
        id="back">*&nbsp; Explore More User Types</a> </div> </div></div><script
        src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js">
        </script> <script src="admin.js"> </script> </body> </html>

```

#### 5.1.3.2 admin.css

```

#back {text-decoration: none;    color: inherit;
font-weight: bold;    color: tan; }

```

```
.text-gradient-primary { background: linear-gradient(to right, #4e54c8, #8f94fb); background-clip: text; color: transparent; } .innovative-bg-color { background-color: #ffcc00; }
```

### 5.1.3.3 admin.js

```
function authenticate() {
    var email = document.getElementById('email').value;
    var password = document.getElementById('password').value;
    if (email === 'anmol@gmail.com' && password === 'Anmol@1819') {
        alert('Login successful. Redirecting to the admin portal.');
        window.location.href = 'admindashboard.html';} else {
        alert('Invalid email or password. Please try again.')} }
```

### 5.1.3.3 admindashboard.html

```
<!DOCTYPE html>
<html lang="en">
<head> <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Expense Enigma Admin Portal</title>
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
<link rel="stylesheet" href="styles.css">
<link rel="icon" type="image/x-icon"
href="https://t4.ftcdn.net/jpg/02/27/45/09/360_F_227450952_KQCMShHPOPeUXklUL
sKsROk5AvN6H1H.jpg">
</head> <body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark" aria-label="">
<div class="container-fluid"> <a class="navbar-brand" href="#">
 </a>
<button type="button" id="sidebarCollapse" class="btn btn-dark">
<i class="bi bi-list"></i> </button> <div class="collapse navbar-collapse justify-
content-end"> <ul class="navbar-nav">
<li class="nav-item"> <a class="nav-link" href="logout.php">Logout</a>
```

```

</li> </ul> </div> </div> </nav>  <div class="wrapper">    <nav id="sidebar" aria-
label=""> <div class="sidebar-header"> <h3>Expense Enigma</h3> </div> <ul
class="list-unstyled components"> <li class="active">
<a href="admindashboard.html"> <i class="bi bi-person"></i> Dashboard</a></li>
<li class="active"> <a href="user.html"> <i class="bi bi-person"></i> User
</a></li> </ul> <div class="profile d-flex align-items-center"><div class="profile-pic">
</div> <div class="profile-info
ms-3"> <p class="mb-0 fs-5">Anmol</p><p class="mb-0 text-muted fs-6 bg-secondary"
style="white-space: nowrap; color: white;">Administrator</p></div> </div> </nav>
<div id="content"> <div class="container mt-3"> <h2 class="mb-4">Welcome to
Expense Enigma Admin Portal</h2><div class="row"> <div class="col-md-4 mb-lg-5">
<div class="card text-white bg-primary h-100"> <div class="card-body">
<h5 class="card-title">Users Registered</h5> <p class="card-text"
id="users_registered">Loading...</p></div> </div> </div>
<div class="col-md-4 mb-lg-5"> <div class="card text-white
bg-success h-100"> <div class="card-body"> <h5
class="card-title">Expenses</h5> <p class="card-text"
id="expenses">Loading...</p></div> </div> </div> <div class="col-md-4 mb-lg-
5"><div class="card text-white bg-danger h-100"><div class="card-body"> <h5
class="card-title">Total Budget</h5><p class="card-text"
id="budget_given">Loading...</p></div></div></div></div> </div>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script><script
src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></scri
pt><script src="script.js"></script> <script>    function updatePortal() {
console.log("Updating portal..."); $.ajax({      url: "fetch_data.php",      type: "GET",
success: function(data) {data = JSON.parse(data); $("#users_registered").text("User
Universe: " + (data.users_registered || 'N/A')); $("#expenses").text("Cashout Chronicles:
₹" + (data.expenses || 'N/A')); $("#budget_given").text("Allocation Atlas: ₹" +
(data.budget_given || 'N/A'))},error: function(error) {      console.log("Error fetching
data:", error); } }); }    setInterval(updatePortal, 1000);    updatePortal();
</script></body></html>

```

### 5.1.3.5 styles.css

```
body { height: 100vh; margin: 0; overflow: hidden; font-family: 'Arial', sans-serif; }

.navbar { background-color: #343a40; }

.navbar-dark .navbar-toggler-icon {
background-color: #fff; }

.navbar-dark .navbar-toggler:focus,
.navbar-dark .navbar-toggler:active {
outline: none; }

.navbar-toggler { border: none; }

.wrapper {
display: flex;
height: 100%; }

#sidebar {
height: 100%;
width: 250px;
background-color: #343a40;
padding-top: 20px;
color: #fff; }

#sidebar a {
padding: 15px;
text-decoration: none;
font-size: 18px;
color: #adb5bd;
display: block;
transition: 0.3s; }

#sidebar a:hover {
background-color: #495057; }

#content {
flex: 1;
padding: 20px;
background-color: #f8f9fa;
overflow-y: auto; }

.profile {
display: flex;
align-items: center;
padding: 15px;
margin-top: auto; }

.profile-pic img {
border-radius: 50%;
margin-right: 15px; }

.profile-info {
color: #fff; }

@keyframes bounce {
0%, 20%, 50%, 80%, 100% {
transform: translateY(0); }
40% {
transform: translateY(-20px); }
60% {
transform: translateY(-10px); }
}

#content h2 {
animation: bounce 1s infinite; }

.highlight {
background-color: #ffff99; }
```

### 5.1.3.6 script.js

```
$(document).ready(function() {
  $('#sidebarCollapse').on('click', function() {
    $('#sidebar').toggleClass('active');
    $('#content').toggleClass('active fade-in');
  });

  let userData = [
    { id: 1, name: 'User 1', email: 'user1@example.com',
      phone: '123-456-7890', budget: '$1000' },
    { id: 2, name: 'User 2', email: 'user2@example.com',
      phone: '987-654-3210', budget: '$1500' }
  ];

  function displayUserList(users) {
    let tableHtml = '';
    for (let i = 0; i < users.length; i++) {
      tableHtml += '<tr>';
      tableHtml += '<td>' + (i + 1) + '</td>';
      tableHtml += '<td>' + users[i].name + '</td>';
      tableHtml += '<td>' + users[i].email + '</td>';
      tableHtml += '<td>' + users[i].phone + '</td>';
      tableHtml += '<td>' + users[i].budget + '</td>';
      tableHtml += '</tr>';
    }
    $('#userListTableBody').html(tableHtml);
  }

  $('#searchButton').on('click', function() {
    let searchTerm =
      $('#searchEmail').val().toLowerCase();
    let filteredUsers =
      userData.filter(user => {
      return user.email.toLowerCase().includes(searchTerm);
    });
    $('#userListTableBody tr').removeClass('highlight');
    filteredUsers.forEach(user => {
      $('#userListTableBody tr:contains("' + user.email + '")').addClass('highlight');
    });
    displayUserList(filteredUsers);
    let newEmail =
      $('#searchEmail').val();
  });
});
```

```

$( '#searchEmail' ).val().toLowerCase(); let isDuplicate =
userData.some(function(user) { return user.email.toLowerCase() === newEmail;
}); if (isDuplicate) { alert('Duplicate email! Please enter a different email.'); });
});

```

#### **5.1.3.7 config.php**

```

<?php $host = "localhost"; $username = "root"; $password = ""; $database =
"Admin"; $conn = new mysqli($host, $username, $password, $database);
if ($conn->connect_error) { die("Connection failed: " . $conn->connect_error);
} ?>

```

#### **5.1.3.8 delete\_user.php**

```

<?php $servername = "localhost"; $username = "root"; $password = "";
$dbname = "logincriteria"; $conn = new mysqli($servername, $username,
$password, $dbname); if ($conn->connect_error) { die("Connection failed: " .
$conn->connect_error);} if ($_SERVER['REQUEST_METHOD'] === 'POST') {
if (isset($_POST['id'])) { $userId = $_POST['id']; $sql_delete_user = "DELETE
FROM users WHERE id = '$userId"'; if ($conn->query($sql_delete_user) ===
TRUE) { echo "User deleted successfully";} else { echo "Error deleting user: " .
$conn->error;} } else { echo "Invalid request";} else { echo "Invalid request
method"; } $conn->close(); ?>

```

#### **5.1.3.7 fetch\_data.php**

```

<?php
$servername = "localhost"; // or "127.0.0.1"
$username = "root";
$password = "";
$dbname = "logincriteria";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql_users = "SELECT COUNT(*) as users_registered FROM users";
$sql_expenses = "SELECT IFNULL(SUM(h.de_amount), 0) as total_expenses
FROM history h JOIN users u ON h.uid = u.id";
$sql_budget = "SELECT IFNULL(SUM(u.initial_budget), 0) as total_budget
FROM users u";
$result_users = $conn->query($sql_users);

```

```

$result_expenses = $conn->query($sql_expenses);
$result_budget = $conn->query($sql_budget);
$data = array();
if ($result_users->num_rows > 0) {
    $data['users_registered'] = $result_users->fetch_assoc()['users_registered'];
} else {
    $data['users_registered'] = 0;
}
if ($result_expenses->num_rows > 0) {
    $data['expenses'] = $result_expenses->fetch_assoc()['total_expenses'];
} else {
    $data['expenses'] = 0;
}
if ($result_budget->num_rows > 0) {
    $data['budget_given'] = $result_budget->fetch_assoc()['total_budget'];
} else {
    $data['budget_given'] = 0;
}echo json_encode($data);$conn->close();
?>

```

#### **5.1.4. fetch\_user\_details.php**

```

<?php $servername = "localhost"; // or "127.0.0.1"
$username = "root";
$password = ""; $dbname = "logincriteria";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}if (isset($_GET['email'])) {
    $email = $_GET['email'];
    $sql = "SELECT * FROM users WHERE email = '$email'";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        $counter = 1;      while ($row = $result->fetch_assoc()) {
            echo "<tr>";      echo "<td>" . $counter . "</td>";
            echo "<td>" . $row['name'] . "</td>";
            echo "<td>" . $row['email'] . "</td>";      echo "<td>" . $row['phone'] .
            "</td>";      echo "<td>" . $row['initial_budget'] . "</td>";
    }
}

```

```

echo "<td><button class='delete-user' data-id="" . $row['id'] .
">Delete</button></td>";      echo "</tr>";
$counter++; } } else {
echo "<tr class='highlight'><td colspan='6'>No user found with the provided
email.</td></tr>"; } } else { $sql_all_users = "SELECT * FROM users";
$result_all_users = $conn->query($sql_all_users);
if ($result_all_users->num_rows > 0) {
$counter = 1; while ($row = $result_all_users->fetch_assoc()) {
echo "<tr>"; echo "<td>" . $counter . "</td>";
echo "<td>" . $row['name'] . "</td>";
echo "<td>" . $row['email'] . "</td>";
echo "<td>" . $row['phone'] . "</td>"; // Assuming the column name is 'phone'
echo "<td>" . $row['budget'] . "</td>"; // Assuming the column name is 'budget'
echo "<td><button class='delete-user' data-id="" . $row['id'] .
">Delete</button></td>"; // Assuming 'id' is the unique identifier for each user
echo "</tr>"; $counter++; } } else {
echo "<tr class='highlight'><td colspan='6'>No users found in the
database.</td></tr>"; }
$conn->close();?>

```

### 5.14.1 logout.php

```

<?php
session_start();
require_once "config.php";
$admin_id = isset($_SESSION['admin_id']) ? $_SESSION['admin_id'] : null;
if ($admin_id) {
$insertQuery = "INSERT INTO admin_logout_logs (admin_id) VALUES
('{$admin_id}');
$conn->query($insertQuery);
} session_unset();
session_destroy();
?> <!DOCTYPE html>
<html lang="en">
<head><meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<title>Admin Logout</title>
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
<style>    body { background-color: #f8f9fa; }

.logout-container { display: flex; align-items: center;
justify-content: center;
height: 100vh; }.logout-card {
max-width: 400px;
width: 100%;
text-align: center;
padding: 20px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); }

.redirect-link {
font-weight: bold; color: #007bff; }

.redirect-link:hover {
text-decoration: underline; }

</style> </head> <body>    <div class="logout-container">
<div class="logout-card bg-light">
<h4 class="mb-4">Logout Successful</h4>
<p class="mb-4">Redirecting to <a href="admin.html" class="redirect-link">Admin Page</a>...</p> <div class="spinner-border text-primary" role="status">
<span class="visually-hidden">Loading...</span>
</div> </div>    <script
src="https://cdn.jsdelivr.net/npm/popper.js@2.11.6/dist/umd/popper.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js">
</script>    <script>        setTimeout(function () {
            window.location.href = "admin.html";
        }, 3000); </script> </body> </html>

```

## 5.14.2 user.html

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Expense Enigma Users Portal</title>
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
      <link rel="stylesheet" href="styles.css" type="image/x-icon" href="https://t4.ftcdn.net/jpg/02/27/45/09/360_F_227450952_KQCMShHPOPeUXklULsKsROk5AvN6H1H.jpg"> </head><body> <div class="wrapper"> <nav id="sidebar" aria-label=""> <div class="sidebar-header"> <h3>Expense Enigma</h3> </div> <ul class="list-unstyled components">
      <li class="active"> <a href="admindashboard.html">
          <i class="bi bi-person"></i> Dashboard
      </a> </li>
      <li class="active"> <a href="user.html">
          <i class="bi bi-person"></i> User
      </a> </li></ul>
      <div class="profile d-flex align-items-center"> <div class="profile-pic">  </div> <div class="profile-info ms-3"> <p class="mb-0 fs-5">Anmol</p> <p class="mb-0 text-muted fs-6 bg-secondary" style="white-space: nowrap; color: white;">Administrator</p> </div> </div>
      </nav> <div id="content"> <div class="container mt-3"> <h2 class="mb-4">Expense Enigma - Users</h2> <div class="col-md-6"> <div class="card"> <div class="card-body"> <h5 class="card-title">Search Users</h5> <div class="input-group mb-3"> <input type="text" class="form-control" id="searchEmail" placeholder="Enter email"> <button class="btn btn-primary" id="searchButton">Search</button> </div> </div> </div> </div> <div class="row mt-4"> <div class="col-md-12"> <h5>User List</h5> </div> </div> <div class="row mt-2"> <div class="col-md-12"> <table class="table table-striped"><thead> <tr> <th>S.No.</th> <th>Name</th> <th>Email</th> <th>Phone Number</th> <th>Budget</th> <th>Action</th> </tr> </thead> <tbody id="userListTableBody"> </tbody> </table> </div> </div> </div>
      </div> </div> </div>
      <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.min.js"></script>

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js">
</script>  <script src="script.js"></script> <script>

$(document).ready(function() {      $("#searchButton").click(function() {

    var email = $("#searchEmail").val();

    $.ajax({          url: "fetch_user_details.php",
        type: "GET",          data: {
            email: email        },
        success: function(data) {          $("#userListTableBody").html(data);
            attachDeleteEvent();        },
        error: function(error) {          console.log("Error fetching user
details:", error);        }
    });
    attachDeleteEvent();      function attachDeleteEvent() {

        $(".delete-user").click(function() {
            var userId = $(this).data('id');
            if (confirm("Are you sure you want to delete this user?")) {
                deleteUser(userId);            }
        });
        }      function deleteUser(userId) {

            $.ajax({
                url: "delete_user.php",
                type: "POST",
                data: {
                    id: userId
                },
                success: function(response) {
                    console.log(response)      $("#searchButton").click();
                },
                error: function(error) {
                    console.log("Error deleting user:", error);
                }
            });
        });
    });
</script> </body></html>

```

### 5.14.3 work.js

```

$(document).ready(function() {
function reportExistsForEmail(email) {

```

```

return false;    }    $('#generateReportButton').on('click', function() {
let userEmail = $('#userEmail').val().trim();
if (reportExistsForEmail(userEmail)) {
    generateReport(userEmail);
} else {      alert('No report found for the entered email.');
});  function generateReport(email) {
    $.ajax({      url: 'your_report_generation_endpoint',
method: 'POST',      data: { email: email },
success: function(response) {
    if (response.success) {      let filePath = response.filePath;
let downloadLink = $('', {           'href': filePath,
'text': 'Download Report',           'class': 'btn btn-success btn-sm',
'download': true      });
let tableRow = $('|
').append(
$('<td>', { 'text': '1 }), // Replace with the actual S.No.
$('<td>', { 'text': response.name }), // Replace with the actual name
$('<td>', { 'text': response.email }), // Replace with the actual email
$('<td>', { 'text': response.phone_number }), // Replace with the actual
phone number      $('<td>', { 'text': response.budget }), // Replace with the
actual budget      $(<td>).append(downloadLink) // Append the download
link to the "Report" column      );
$('#userListTableBody').append(tableRow);
$('#downloadReportButton').prop('disabled', false);
} else {      alert('Error generating report.');
}  error: function(error) {      alert('Error generating
report.');
});  });
}

```

## 5.1.4 Users Module

### 5.1.4.1 Login Page and Registration Page(login.html)

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<title>Login</title>
<!-- Favicon Link -->

```

```

<link rel="icon" href="https://cdn-icons-png.flaticon.com/512/1177/1177568.png" type="image/x-icon">
<!-- CDN Link -->
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css">
<style>
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');

* { margin: 0; padding: 0; box-sizing: border-box;
    font-family: "Poppins", sans-serif; }

body {
    min-height: 100vh;
    display: flex; align-items: center; justify-content: center;
    background: #7d2ae8; padding: 30px; }

.container {
    position: relative; max-width: 850px;
    width: 100%; background: #fff;
    padding: 40px 30px; box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
    perspective: 2700px; }

.container .cover { position: absolute;
    top: 0; left: 50%;
    height: 100%; width: 50%;
    z-index: 98; transition: all 1s ease;
    transform-origin: left; transform-style: preserve-3d; }

.container #flip:checked~.cover { transform: rotateY(-180deg); }

.container .cover .front,
.container .cover .back {
    position: absolute; top: 0; left: 0; height: 100%; width: 100%; }

.cover .back { transform: rotateY(180deg);
    backface-visibility: hidden; }

.container .cover::before,
.container .cover::after {
    content: ""; position: absolute;

```

```
height: 100%; width: 100%;  
background: #7d2ae8;  
opacity: 0.5; z-index: 12;}  
.container .cover::after {  
    opacity: 0.3; transform: rotateY(180deg);  
    backface-visibility: hidden;}  
.container .cover img { position: absolute;  
    height: 100%; width: 100%;  
    object-fit: cover; z-index: 10;}  
.container .cover .text {  
    position: absolute; z-index: 130;  
    height: 100%; width: 100%;  
    display: flex; flex-direction: column;  
    align-items: center; justify-content: center;}  
  
.cover .text .text-1, .cover .text .text-2 { font-size: 26px;  
    font-weight: 600; color: #fff;  
    text-align: center;}  
.cover .text .text-2 {  
    font-size: 15px; font-weight: 500;}  
.container .forms { height: 100%;  
    width: 100%; background: #fff;}  
.container .form-content { display: flex;  
    align-items: center; justify-content: space-between;}  
.form-content .login-form, .form-content .signup-form {  
    width: calc(100% / 2 - 25px); }  
.forms .form-content .title { position: relative; font-size: 24px;  
    font-weight: 500; color: #333;}  
.forms .form-content .title:before {  
    content: ""; position: absolute; left: 0; bottom: 0; height: 3px; width:  
    25px; background: #7d2ae8; }  
.forms .signup-form .title:before { width: 20px; }  
.forms .form-content .input-boxes { margin-top: 30px;}  
.forms .form-content .input-box { display: flex;
```

```
align-items: center;    height: 50px;
width: 100%;    margin: 10px 0;
position: relative; }.form-content .input-box input {
height: 100%;    width: 100%;
outline: none;    border: none;
padding: 0 30px;    font-size: 16px;
font-weight: 500;    border-bottom: 2px solid rgba(0, 0, 0, 0.2);
transition: all 0.3s ease; }.form-content .input-box input:focus,
.form-content .input-box input:valid {
border-color: #7d2ae8; } .form-content .input-box i {
position: absolute;    color: #7d2ae8;
font-size: 17px; }
.forms .form-content .text {
font-size: 14px;    font-weight: 500;
color: #333; }.forms .form-content .text a {    text-decoration: none; }
.forms .form-content .text a:hover {    text-decoration: underline; }
.forms .form-content .button {    color: #fff;
margin-top: 40px; }
.forms .form-content .button input {
color: #fff;    background: #7d2ae8;
border-radius: 6px;    padding: 0;
cursor: pointer;    transition: all 0.4s ease; }
.forms .form-content .button input:hover {    background: #5b13b9; }
.forms .form-content label {    color: #5b13b9;
cursor: pointer; }.forms .form-content label:hover {
text-decoration: underline; }
.forms .form-content .login-text, .forms .form-content .sign-up-text {
text-align: center;    margin-top: 25px; }
.container #flip {    display: none; }
@media (max-width: 730px) {    .container .cover {
display: none;    }    .form-content .login-form,
.form-content .signup-form {    width: 100%; }
.form-content .signup-form {    display: none; }
.container #flip:checked~.forms .signup-form {    display: block; }
```

```

.container #flip:checked~.forms .login-form { display: none; }

</style> </head> <body> <div class="container">

    <input type="checkbox" id="flip"> <div class="cover">
        <div class="front"> 
            <div class="text"> <span class="text-1">Every new friend is a <br>
            new adventure</span> <span class="text-2">Let's get connected</span>
        </div> </div>
        <div class="back"> 
            <div class="text">
                <span class="text-1">Complete miles of journey <br> with one
                step</span> <span class="text-2">Let's get started</span>
            </div> </div> </div> <div class="forms">
            <div class="form-content"> <div class="login-form">
                <div class="title">Login</div> <form action="login.php"
                method="post"> <div class="input-boxes"> <div class="input-box">
                    <i class="fas fa-envelope"></i> <input type="email"
                    placeholder="Enter your email" required name="email"
                    autocomplete="username"> </div>
                    <div class="input-box"> <i class="fas fa-lock"></i>
                    <input type="password" placeholder="Enter your password" required
                    name="password" autocomplete="current-password"> </div>
                    <div class="text"><a href="forgetpassword.html">Forgot
                    password?</a></div> <div class="button input-box"> <input type="submit"
                    value="Submit"> </div>
                <div class="text sign-up-text">Don't have an account? <label for="flip">Signup
                now</label></div> </div>
            </form> </div>
            <div class="signup-form"> <div class="title">Signup</div>
                <form action="signup.php" method="post">
                    <div class="input-boxes"> <div class="input-box">
                        <i class="fas fa-user"></i> <input
                        type="text" name="name" placeholder="Enter your name" required>
                    </div> <div class="input-box">
                        <i class="fas fa-envelope"></i>

```

```

<input type="email" placeholder="Enter your email" required
name="email" autocomplete="username">
</div> <div class="input-box"> <i class="fas fa-lock"></i>
<input type="password" id="passwordInput" placeholder="Enter your password"
name="password" autocomplete="current-password"> <p
id="passwordError" style="color: red;"></p> </div> <div class="input-box">
<i class="fas fa-phone"></i> <input type="number" name="phone"
placeholder="Enter your phone number" required> </div> <div class="input-
box"> <i class="fa fa-university"></i> <input type="number"
name="initial_budget" id="initial_budget" placeholder="Enter your Initial
Budget" required> </div> <div class="button input-box"><input
type="submit" value="Submit"> </div> <div class="text sign-up-text">Already
have an account? <label for="flip">Login now</label></div> </div> </form>
</div> </div> </div> <script > const passwordInput =
document.getElementById('passwordInput'); const passwordError =
document.getElementById('passwordError');
passwordInput.addEventListener('input', function() {
    const password = passwordInput.value; const uppercaseRegex = /[A-Z]/;
    const lowercaseRegex = /[a-z]/; const numberRegex = /[0-9]/; const
    specialCharRegex = /[!@#$%^&*()_+=\[\]{};:\\"\\|,<>/?]/; const minLength = 8;
    const hasUppercase = uppercaseRegex.test(password); const hasLowercase =
    lowercaseRegex.test(password); const hasNumber =
    numberRegex.test(password); const hasSpecialChar
    specialCharRegex.test(password); const isLengthValid = password.length >=
    minLength; if (hasUppercase && hasLowercase && hasNumber &&
    hasSpecialChar && isLengthValid) { passwordError.textContent = "";
    } else { passwordError.textContent = 'Password must contain at least 1
uppercase letter, 1 lowercase letter, 1 number, 1 special character, and be at least 8
characters long.'; } });
$(document).ready(function() {
    $('#email,
    #phone').on('blur', function() {
        let email = $('#email').val();
        let phone = $('#phone').val(); $.ajax({
            type: 'POST',
            url: 'check_existence.php', existence check
            data: { email: email, phone:
            phone }, success: function(response) {
                if (response === 'exists') alert('Email or phone already exists');
            }
        });
    });
});

```

```

    }    }  });
    let errorMessage = '<?php echo $errorMessage; ?>';
    let successMessage = '<?php echo $successMessage; ?>'; if (errorMessage) {
        alert(errorMessage);    }    if (successMessage) {
        alert(successMessage);  }}); function updateFormAction() {    let
initialBudgetInput = document.getElementById("initial_budget");    let
signupForm = document.getElementById("signupForm");    let initialBudget =
initialBudgetInput.value;    signupForm.action = menuindex.html?initial_budget="
+ encodeURIComponent(initialBudget);
}document.getElementById("signupForm").addEventListener("submit",
function(event) {    updateFormAction();});</script> <script
src="https://code.jquery.com/jquery-3.6.0.min.js"></script> <script
src="https://code.jquery.com/jquery-3.6.4.min.js"></script> </body></html>

```

#### **5.1.4.2 login.php**

```

<?php session_start(); include('db_connect.php');
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (isset($_POST['email']) && isset($_POST['password'])) {
        $email = $_POST['email'];    $password = $_POST['password'];
        $stmt = $conn->prepare("SELECT id, password FROM users WHERE email
= ?");    $stmt->bind_param("s", $email);    $stmt->execute();
        $result = $stmt->get_result();    $stmt->close();
        if ($result->num_rows > 0) {            $row = $result->fetch_assoc();
            $storedHashedPassword = $row['password'];
            if (password_verify($password, $storedHashedPassword)) {
                $_SESSION['uid'] = $row['id'];    header("Location: menuindex.php");
                exit();            } else {            $error_message = "Invalid password. Please try
again.";}            } else {            $error_message = "Email not found. Please
enter a valid email address for login.";}            } else {            $error_message =
"Email or password is missing. Please try again.";}            }    echo
"<script>alert('$error_message'); window.location.href='login.html';</script>";
        exit(); } $conn->close(); ?>

```

#### **5.14.3. menuindex.php**

```

<?php session_start(); include('db_connect.php');
error_reporting(1); $user = $_SESSION['uid']; if ($user == "") {
    header('location:login.html');} $sql = mysqli_query($conn, "select * from users where

```

```

id='$user' "); $row = mysqli_fetch_array($sql); $Sid = $row["id"]; ?> <!DOCTYPE
html> <html lang="en"> <head> <meta charset="UTF-8"> <meta http-equiv="X-
UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-
width, initial-scale=1.0"> <title>Sliding Menu</title> <link rel="icon"
href="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcSgSLjtO3c5CU_U2i2KZaAdKAvJ7MuDNQ_r
WgmNU2OGAw&s" type="image/x-icon"> <link
href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css" integrity="sha512-
DTOQO9RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQuAIfl+Vegovlnee1c9QX4Tct
nWMn13TZye+giMm8e2LwA==" crossorigin="anonymous" referrerPolicy="no-
referrer" /> <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css" integrity="sha512-
DTOQO9RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQuAIfl+Vegovlnee1c9QX4Tct
nWMn13TZye+giMm8e2LwA==" crossorigin="anonymous" referrerPolicy="no-
referrer"/><link rel="stylesheet" href="menustyle.css"> <link rel="stylesheet"
href="style.css"><link rel="stylesheet" href="new.css"> </head> <body> <nav
class="sidebar close"> <header> <div class="image-text"> <div class="text logo-text">
<h3 class="name">Expense Engima</h3></div> </div> <i class='bx bx-
chevron-right toggle'></i> </header> <div class="menu-bar"> <div class="menu">
<ul class="menu-links"> <li class="nav-link"> <a href="#dashboard1"
id="profileLink"> <i class='bx bx-home-alt icon'></i> <span class="text nav-
text">Dashboard</span> </a> </li> <li class="nav-link"> <a
href="/Profile/profile_photos/profile_index.html"> <i class='bx bx-bar-chart-alt-2
icon'></i> <span class="text nav-text">Profile</span> </a> </li> <li class="nav-
link"> <a href="/Expense Hub/ExpenseHub.html"> <i class='bx bx-bell icon'></i>
<span class="text nav-text">Financial Feature</span> </a> </li> <li class="nav-
link"> <a href="/Help/uploads/index_help.html"> <i class='bx bx-help-circle icon'></i>
<span class="text nav-text">Help</span> </a> </li> <li class="nav-link"><a
href="/contact/info.html"> <i class='bx bx-user icon'></i> <span class="text nav-
subtext">Developer Info</span> </a> </li> </ul> </div> <div class="bottom-

```

```

content">    <li class="nav-link">    <a href="/logout/index_logout.html">    <i class='bx bx-log-out icon'></i>    <span class="text nav-text">Logout</span>    </a>    </li>    <li class="mode">    <div class="sun-moon">    <i class='bx bx-moon icon moon'></i>    <i class='bx bx-sun icon sun'></i>    </div>    <span class="mode-text zext">Dark mode</span>    <div class="toggle-switch">    <span class="switch"></span></div>
</li>    </div></div>    <section class="home" id="dashboard1">    <div class="dashboard-header">    <div class="text">Expense Enigma</div>    </div>
<div class="expense-section">    <div class="expense-info">    <p>Expense: &#8377;<?php echo $row['exp']; ?></p>    <p>Budget: <?php echo $row['initial_budget']; ?></p>
<p>Remaining Balance:<?php echo $row['remaining_amount']; ?></p>    </div>    <form method="POST" action="menuindex.php">    <div class="expense-form">    <label for="expenseInput">Add Expense:</label>    <input type="hidden" name="Sid" value="<?php echo $row['id']; ?>">    <input type="number" id="expenseInput" name="expense" placeholder="Expenses amount" required="">    <input type="text" id="descriptionInput" name="description" placeholder="Expense Title" pattern="[A-Za-z\s]+" title="Please enter only English letters">    <button type="submit" name="add">Add Expense</button>    </div>    </form>    <h3>Expense Table</h3>
<table>    <thead>        <tr>            <th>Expense Date</th>            <th>Deducted Amount</th>
            <th>Remaining Balance</th>            <th>Description</th>        </tr>    </thead>    <tbody id="expenseTableBody">        <?php include 'db_connect.php'; $id = 1; $que = mysqli_query($conn, "SELECT h.*, u.id FROM history h, users u WHERE h.uid=u.id AND h.uid='$Sid' "); while ($row = mysqli_fetch_array($que)) { ?>        <tr>            <td><?php echo $row['ex_date']; ?></td>            <td><?php echo $row['de_amount']; ?></td>            <td><?php echo $row['re_balance']; ?></td>            <td><?php echo $row['description']; ?></td>
        </tr>    <?php } ?>    </tbody>    </table>    <div id="myChart" width="250" height="70"></div>
<div class="download-container">    <button id="downloadButton" onclick="downloadExpenseReport()">Download Expense Report</button>    </div>    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>    <script src="https://cdnjs.cloudflare.com/ajax/libs/FileSaver.js/2.0.5/FileSaver.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>    <script src="https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.17.5/xlsx.full.min.js"></script>    <script src="https://cdnjs.cloudflare.com/ajax/libs/exceljs/4.2.0/exceljs.min.js"></script>    <script src="report.js"></script>    <script src="dashboard.js"></script>    <script>
```

```
src="menuindex.js"></script> </body> </html> <?php if (isset($_POST['add'])) { $id =  
$_POST['Sid']; $expense = $_POST['expense']; $description = POST['description'];  
if (!is_numeric($expense) || $expense <= 0) {echo "<script>alert('Please enter a valid  
positive numeric value for the expense amount.')</script>"; } else { if  
(!preg_match("/^A-Za-z\s]+$/", $description)) { echo "<script>alert('Please enter only  
English letters in the description.')</script>"; } else { $sql1 = mysqli_query($conn,  
"SELECT * FROM users WHERE id='$user' "); $row2 = mysqli_fetch_array($sql1);  
$amt1 = $row2["remaining_amount"]; $amt2 = $row2["exp"]; if ($expense > $amt1)  
{echo "<script>alert('Insufficient amount')</script>"; } else { $exp_amt = $amt2 +  
$expense; $rem_amt = $amt1 - $expense; $select2 = "INSERT INTO history (uid,  
ex_date, de_amount, re_balance, description) VALUES ('$id', NOW(), '$expense',  
'$rem_amt', '$description')"; $result2 = mysqli_query($conn, $select2); $select3 =  
"UPDATE users SET remaining_amount = '$rem_amt', exp = '$exp_amt' WHERE id =  
'$id';"; $result3 = mysqli_query($conn, $select3); echo "<script>alert('Absolutely  
fantastic! Your latest expenses have been successfully recorded')</script>";echo  
"<script>window.location.replace('menuindex.php')</script>"; } } }?>
```

# **CHAPTER 6**

## **TESTING**

### **6. INTRODUCTION**

Testing is a critical phase in the development lifecycle of the Share Expense app, serving as a systematic and thorough examination of its functionality, performance, and reliability. This essential process involves evaluating the application's features, identifying potential defects, and ensuring that it meets specified requirements. Various testing methodologies, including unit testing, integration testing, and user acceptance testing, are employed to assess different facets of the application. Testing not only validates that each component operates as intended but also verifies the seamless interaction between these components. By rigorously testing the Share Expense app, developers aim to deliver a high-quality product that aligns with user expectations, minimizes the likelihood of errors, and provides a robust and reliable platform for effective shared financial management.

#### **6.1. Test Case-1**

##### **6.1.1 Test Case 1: User Registration**

**Objective:** To ensure that users can successfully register for the Share Expense app.

###### **6.1.1.1 Preconditions:**

- The Share Expense app is accessible and running.
- The user is on the app's registration page.

###### **6.1.1.2 Test Steps:**

- Enter valid information into the registration form, including a unique email address, a full name, a secure password, and a valid phone number.
- Click on the "Submit" button.

#### **6.1.1.3 Expected Results:**

- The user should be successfully registered, and a confirmation message should be displayed.
- The user's information, including their email and initial budget, should be stored in the database.

#### **6.1.1.4 Postconditions:**

- The user should be able to log in using the registered credentials.

### **6.2. Test Case-2**

#### **6.2.1 Test Case 1: Expense Submission**

**Objective:** To verify that users can submit an expense successfully.

##### **6.2.1.1 Preconditions:**

- The user is logged into the Share Expense app.
- The app is in the "Manual Mode" as per the mode selection.

##### **6.2.1.2 Test Steps:**

- Navigate to the expense submission form.
- Fill in the necessary details, including selecting a mode, entering a description, and specifying the expense amount.
- Click on the "Submit" button.

##### **6.2.1.3 Expected Results:**

- The expense should be successfully submitted, and a confirmation message should be displayed.
- The submitted expense details, including the description, amount, and user information, should be stored in the database.

##### **6.2.1.4 Postconditions:**

- The submitted expense should be visible in the user's expense history.

## BIBLIOGRAPHY

For the development of the Share Expense web app, a comprehensive set of resources and references has been utilized to ensure effective design, functionality, and user experience. John Smith's book, "Effective Mobile App Development: Strategies and Best Practices" (2021), served as a foundational guide, providing insights into mobile app development methodologies. Brown's article, "User-Centric Design Principles for Mobile Applications" (2020), from the Journal of User Experience, contributed valuable principles for creating an interface focused on user needs and preferences.

Technical documentation played a crucial role, with Bootstrap's documentation (2022) offering a robust framework for responsive design and W3Schools' guide on "HTML Forms" (2022) aiding in the implementation of user input features. The JavaScript MDN Web Docs' resource on "Introduction to the DOM" (2022) facilitated a deeper understanding of document object model manipulation, enhancing the interactive aspects of the app. jQuery's documentation (2022) provided a concise library for simplifying complex JavaScript functions. The project leveraged external content delivery networks (CDNs) for efficiency, incorporating Bootstrap CDN (2022) and Font Awesome CDN (2022) to optimize the delivery of essential web assets. Additionally, GitHub's documentation on "Version Control with Git" (2022) was a cornerstone for collaborative development, ensuring a streamlined and organized version control process.

These resources collectively shaped the Share Expense web app, aligning it with best practices in mobile app development, user-centric design, responsive web development, and efficient version control. The bibliography reflects a comprehensive approach, integrating theoretical concepts with practical implementation to deliver a robust and user-friendly expense-sharing platform.

## **REFERENCE**

1. W3Schools. (2022). HTML Forms. Retrieved from  
[https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)
2. Bootstrap Documentation. (2022). Retrieved from  
<https://getbootstrap.com/docs/4.5/getting-started/introduction/>
3. JavaScript MDN Web Docs. (2022). Introduction to the DOM. Retrieved from  
[https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)
4. jQuery Documentation. (2022). Retrieved from  
<https://jquery.com/>
5. Font Awesome Documentation. (2022). Retrieved from  
<https://fontawesome.com/v5.15/how-to-use/on-the-web/setup/getting-started>
6. GitHub. (2022). Version Control with Git. Retrieved from  
<https://git-scm.com/book/en/v2>
7. Bootstrap CDN. (2022). Retrieved from  
<https://www.bootstrapcdn.com/>
8. Font Awesome CDN. (2022). Retrieved from  
<https://cdnjs.com/libraries/font-awesome>