

PROJECT REPORT
On
“E-TRACK Website”

Submitted by
Mr. Siddhesh Chalke

in partial fulfilment for the award of the degree
of
BACHELOR OF SCIENCE
in
COMPUTER SCIENCE

under the guidance of

Prof. Pradnya Kharade

Department of Computer Science



SHETH L.U.J. COLLEGE OF ARTS AND SIR M.V. COLLEGE
OF
SCIENCE AND COMMERCE

SEM V
2024-2025

DECLARATION

I, **Mr. Siddhesh Chalke** hereby declare that the project entitled “**E-TRACK Website**” submitted in the partial fulfilment for the award of Bachelor of Science in Computer Science during the academic year 2024-2025 is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Signature of the Student:

Place:

Date:

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my esteemed project guide, **Prof .Pradnya Kharade**, for their invaluable guidance and support throughout the development of this project. Their encouragement and patience have been instrumental in shaping my understanding of the project's objectives and enabling me to navigate the challenges that arose during the implementation process.

I would also like to extend my sincere appreciation to our Principal **Dr. Mahendra Kanojia** , for providing the necessary resources to support our project endeavours. The college's commitment to fostering a nurturing learning environment has played a significant role in our academic growth and professional development.

Furthermore, I am grateful to the entire teaching staff for their unwavering support and dedication to our academic success. Finally, we would like to thank everyone who has helped us directly or indirectly in our project. This project imparted us with a great deal of practical experience, which will prove a boon to us in the near future.

ABSTRACT

E-Track is a powerful expense tracking tool that simplifies the process of managing your finances. It offers a user-friendly interface for easily recording expenses, categorizing them into different groups, and generating detailed reports. With E-Track, you can gain valuable insights into your spending habits, identify areas for improvement, and make informed financial decisions.

Whether you're a small business owner or an individual looking to track your personal expenses, E-Track provides the necessary tools to effectively manage your finances. The application's intuitive design and robust features make it a valuable asset for anyone seeking to improve their financial well-being.

INDEX

| Sr No | Content | Page No | Sign |
|-------|---|---------|------|
| 1. | Introduction | 1 | |
| 2. | Objectives | 2 | |
| 3. | Stakeholders | 3 | |
| 4. | Scope | 4 | |
| 5. | Methodology | 5 | |
| 6. | Requirement Analysis | 6 | |
| 7. | Timeline | 8 | |
| 8. | Gantt Chart | 10 | |
| 9. | Entity Relationship Diagram | 11 | |
| 10. | Class Diagram | 14 | |
| 11. | Use Case Diagram | 17 | |
| 12. | Sequence Diagram | 20 | |
| 13. | System Flow Chart | 22 | |
| 14. | Tools & Technology | 25 | |
| 15. | Database Description | 26 | |
| 16. | Database Description Table (Attribut, Datatype And Constrain) | 27 | |
| 17. | Module Description | 28 | |
| 18. | Code (snippet) | 30 | |
| 19. | Expected Outcome | 43 | |
| 20. | Screenshots | 44 | |
| 21. | Conclusion | 48 | |
| 22. | Resources | 49 | |
| 22. | References | 50 | |

INTRODUCTION

In today's fast-paced world, managing personal finances effectively is more crucial than ever. Whether tracking daily expenses or planning for future financial goals, individuals need reliable tools to help them navigate their financial landscape. **E-TRACK** is an innovative, all-in-one expense tracking platform designed to empower users to take control of their finances with ease and confidence.

At E-TRACK, we understand that personal finance management goes beyond just keeping track of expenses; it's about fostering financial well-being and making informed decisions. Our platform streamlines the entire expense management process—from logging daily expenditures to generating insightful reports. With user-friendly features that simplify budgeting, categorizing expenses, and visualizing spending patterns, E-TRACK equips users with the necessary tools to enhance their financial literacy and achieve their goals.

But E-TRACK is more than just a functional application; it is a vibrant community of individuals committed to improving their financial health. Our platform offers resources, expert tips, and collaborative spaces where users can share experiences, explore innovative financial trends, and learn best practices for managing their money effectively. Through forums, blogs, and engaging content, E-TRACK fosters connections and encourages users to support one another on their financial journeys.

At E-TRACK, we believe that managing finances should be an empowering experience. By leveraging technology to streamline the expense tracking process, we strive to make every financial decision more informed, every budget more manageable, and every goal more achievable. Together, we aim to create a thriving ecosystem where individuals can achieve financial success and enjoy peace of mind in their everyday lives.

OBJECTIVES

1. **To create an intuitive online platform for expense management:**
 - **Purpose:** Develop a user-friendly website where individuals can efficiently track their daily expenses, manage budgets, and analyse financial data using HTML, CSS, JavaScript, PHP, and MySQL.
 - **Benefits:** Simplify personal finance management, reduce manual record-keeping, and enhance overall financial awareness for users.
2. **To provide comprehensive resources for financial literacy:**
 - **Purpose:** Equip users with valuable insights and best practices on budgeting, expense tracking, and savings strategies.
 - **Benefits:** Improve financial decision-making, promote responsible spending, and empower users to achieve their financial goals.
3. **To encourage sustainable financial practices:**
 - **Purpose:** Promote mindful spending and saving habits by encouraging users to prioritize essential expenses and minimize wasteful spending.
 - **Benefits:** Foster a culture of financial responsibility, improve resource allocation, and position users as proactive in their financial management.
4. **To enhance user satisfaction and engagement:**
 - **Purpose:** Provide tools for users to create personalized budgeting experiences, including customizable expense categories, tailored financial advice, and real-time notifications.
 - **Benefits:** Boost user loyalty, improve retention rates, and generate positive feedback and reviews of the platform.
5. **To build a collaborative community for personal finance management:**
 - **Purpose:** Create an online space where users can exchange ideas, share experiences, and access industry trends related to personal finance.
 - **Benefits:** Foster a supportive network, encourage knowledge-sharing, and facilitate peer learning among individuals managing their finances.
6. **To streamline market access and user acquisition:**
 - **Purpose:** Help users attract more opportunities for financial growth by providing tools for savings goals, investment tracking, and personalized financial planning.
 - **Benefits:** Increase savings potential, stabilize financial health, and reduce reliance on traditional financial management methods.
7. **To leverage technology for informed financial decisions:**
 - **Purpose:** Utilize data analytics to provide users with insights into spending patterns, budgeting effectiveness, and savings opportunities.
 - **Benefits:** Enable data-driven decision-making, optimize budgeting strategies, and enhance overall financial satisfaction.
8. **To facilitate efficient expense tracking and management:**
 - **Purpose:** Streamline the expense logging process by offering intuitive features for real-time expense entry and categorization.

STAKEHOLDERS

1. Users (Individuals Managing Finances)

- **Role:** Primary users of the E-TRACK platform, responsible for tracking personal expenses, managing budgets, and analyzing financial data.
- **Interests:** Simplify expense management, improve financial literacy, achieve savings goals, and make informed financial decisions through an intuitive platform.

2. Financial Advisors

- **Role:** Professionals who provide guidance and support to users on effective budgeting, investment strategies, and overall financial planning using E-TRACK's resources.
- **Interests:** Enhance client satisfaction, build a strong client base, and promote sound financial practices through user engagement with the platform.

3. Data Analysts

- **Role:** Utilize data collected through the E-TRACK platform to generate insights, trends, and reports that help users optimize their financial management.
- **Interests:** Provide accurate analysis, improve data quality, and deliver valuable insights that drive user engagement and satisfaction.

4. Investors and Financial Institutions

- **Role:** Provide financial support for the development and scaling of the E-TRACK platform.
- **Interests:** Achieve a return on investment, support E-TRACK's growth through innovative solutions, and promote sustainable financial practices.

5. Technology Partners

- **Role:** Offer technical solutions, tools, and support for the development, maintenance, and enhancement of the E-TRACK platform.
- **Interests:** Ensure platform functionality and reliability, collaborate on technological advancements, and enhance user experiences.

6. Marketing and Advertising Agencies

- **Role:** Promote E-TRACK to potential users, increasing brand visibility and user engagement through targeted marketing campaigns.
- **Interests:** Drive user acquisition, enhance platform presence in the personal finance industry, and implement effective marketing strategies.

7. Customer Support Teams

- **Role:** Assist users with any issues or inquiries regarding the use of the E-TRACK platform, including technical support and feedback resolution.
- **Interests:** Ensure high user satisfaction, resolve issues promptly, and provide a seamless experience for all users on the platform.

8. Financial Education Organizations

- **Role:** Collaborate with E-TRACK to provide resources, workshops, and content that enhance financial literacy among users.
- **Interests:** Promote financial education, improve community financial well-being, and support users in achieving their financial goals.

SCOPE

1. **Personal Finance Management Platform Development**

- Comprehensive budgeting tools for tracking expenses, incomes, and savings.
- User registration and secure authentication for individual users and financial advisors.
- Seamless data entry process for logging transactions and monitoring financial health.
- Advanced analytics and reporting features to visualize spending habits and trends.

2. **Educational Resources for Users**

- Best practices in personal finance management, including budgeting techniques and debt reduction strategies.
- Guidance on financial literacy topics such as investment options, savings plans, and retirement planning.
- Training materials and webinars to improve users' financial knowledge and decision-making skills.

3. **User Engagement and Feedback**

- User feedback and review system to gather insights on platform usability and features.
- Personalized recommendations based on user behavior and financial goals.
- Real-time notifications for budgeting alerts, reminders, and financial tips.

4. **Market Access for Financial Tools**

- Direct connections between users and financial service providers, increasing accessibility to products like loans, insurance, and investment opportunities.
- Partnerships with financial institutions to offer exclusive deals and services to users.

5. **Data Security and Privacy Solutions**

- Implementation of robust security measures to protect user data and ensure privacy.
- Compliance with financial regulations and data protection laws to maintain user trust.

6. **Data Analytics**

- Insights into spending patterns, savings goals, and investment performance.
- Performance metrics to help users optimize their financial strategies and make informed decisions.

7. **Sustainability Initiatives**

- Promotion of sustainable financial practices, encouraging users to save and invest in environmentally-friendly companies.
- Tools to track and reduce personal carbon footprints related to spending and consumption.

8. **Mobile Application Development**

- Mobile-friendly platform for seamless access on smartphones and tablets.
- Dedicated app for users to manage budgets, track expenses, and receive financial insights on the go.

METHODOLOGY

1. Agile Development Approach:

- **Iterative Development:** Implement sprints to facilitate continuous enhancement of the platform based on user feedback and emerging requirements.
- **Scrum Framework:** Organize tasks in time-boxed sprints, incorporating regular planning sessions, daily stand-ups, and sprint reviews to maintain consistent progress and alignment.

2. Requirement Gathering and Analysis:

- **Stakeholder Interviews:** Conduct interviews with individual users, financial advisors, and industry experts to understand their needs, preferences, and challenges in personal finance management.
- **User Stories:** Develop comprehensive user stories that reflect the requirements of both novice and experienced users, ensuring a user-centric approach to feature development.
- **Prioritization:** Rank features based on user importance, focusing on enhancing the user experience and delivering maximum value through effective financial management tools.

3. Design Phase:

- **UI/UX Design:** Create wireframes and interactive prototypes that prioritize a clean, user-friendly interface, enhancing navigation and overall user experience.
- **Responsive Design:** Ensure that both the web and mobile applications are fully responsive, offering a seamless experience across all devices, including desktops, tablets, and smartphones.

4. Development:

- **Frontend Development:** Utilize HTML, CSS, and JavaScript to build interactive and dynamic elements for budgeting, expense tracking, and reporting features.
- **Backend Development:** Implement PHP for server-side logic to support core functionalities such as user management, data processing, and analytics.
- **Database Integration:** Use MySQL to store user data, transaction histories, budgeting information, and reports, ensuring data integrity and security.
- **API Integration:** Integrate third-party financial APIs for accessing real-time market data, enabling users to make informed financial decisions.

5. Testing:

- **Unit Testing:** Conduct tests on individual modules and components, such as budgeting tools and transaction logging, to verify their functionality.
- **Integration Testing:** Ensure that the frontend, backend, and database components work seamlessly together, providing a cohesive user experience.
- **Performance Testing:** Assess platform performance under different loads to identify and rectify potential bottlenecks, ensuring smooth operation.

6. Deployment:

- **Web Hosting Deployment:** Deploy the E-TRACK platform on a reliable web hosting service that supports scalability and high availability.
- **CI/CD:** Establish continuous integration and delivery pipelines to automate testing and streamline the deployment of updates and new features.

REQUIREMENT ANALYSIS

Requirement Analysis for E-TRACK

Functional Requirements:

1. User Registration and Authentication:

- Easy registration process for users, including financial advisors and individuals managing personal budgets.
- Secure authentication methods, including email verification, password hashing, and multi-factor authentication for enhanced security.

2. Budget and Expense Tracking:

- Users can create and manage budgets by setting financial goals and tracking expenses in real-time.
- Categorization of expenses (e.g., groceries, utilities, entertainment) with detailed entries including descriptions, amounts, and dates.

3. Reporting and Analytics:

- Generate detailed reports on spending habits, budget performance, and savings over specified periods.
- Visual representations of financial data (e.g., charts and graphs) to provide insights into spending patterns.

4. Goal Setting and Notifications:

- Users can set financial goals (e.g., saving for a vacation) and receive notifications or reminders related to goal progress.
- Alerts for overspending or budget limit breaches to help users stay on track.

5. User Profile Management:

- Users can manage their profiles, including updating contact information, changing passwords, and setting privacy preferences.

6. Customer Support:

- Provide a contact form or live chat feature for user inquiries and support.
- Comprehensive help section with FAQs and tutorials to assist users in navigating the platform.

Non-Functional Requirements:

1. Performance:

- Ensure fast loading times and a responsive design to deliver optimal performance across all devices (mobile, tablet, desktop).
- Scalability to accommodate a growing user base and handle peak traffic during financial planning seasons.

2. Security:

- Implement secure data transmission protocols using HTTPS.
- Protect against common vulnerabilities, including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

3. Usability:

- Develop an intuitive interface that emphasizes user-friendliness, with accessibility features for users with disabilities.
- Support multiple languages to cater to a diverse user demographic.

Integration Requirements:

1. Financial Data Integration:

- Integrate with banks and financial institutions via APIs to allow automatic transaction imports and real-time balance updates.

2. Third-party Application Integration:

- Enable integration with popular budgeting and financial tools (e.g., Mint, QuickBooks) to provide users with a comprehensive financial overview.

Regulatory Requirements:

1. Compliance:

- Adhere to local financial regulations and consumer protection laws to ensure the safety and security of user data.
- Ensure compliance with data protection standards, including GDPR for users in applicable regions.

Environmental Considerations:

1. Sustainability:

- Promote sustainable financial practices, encouraging users to adopt budgeting habits that support environmental responsibility.
- Provide educational resources on financial planning's impact on sustainability, such as budgeting for eco-friendly products and services.

TIMELINE

E-TRACK Project Timeline (August to September)

August

Week 1 (August 1-7):

- **Design Phase:**
 - Create Use Case Diagrams to visualize user interactions between individuals, financial advisors, and the E-TRACK platform.
 - Develop Activity Diagrams to outline the processes for budget creation, expense tracking, and reporting.

Week 2 (August 8-14):

- **Design Phase:**
 - Complete Data Flow Diagrams (DFDs) to depict how data flows within the system, including user data, expense entries, and budget reports.
 - Develop the Entity Relationship Diagram (ERD) to outline database relationships, including users, budgets, expenses, and reports.
-

September

Week 1 (September 1-7):

- **Development Phase - Part 1:**
 - Set up the development environment and tools for coding (HTML, CSS, JavaScript, PHP, MySQL).
 - Begin frontend development to build responsive components for budget creation, expense tracking, and reporting.

Week 2 (September 8-14):

- **Development Phase - Part 1:**
 - Continue backend development using PHP for server-side scripting and MySQL for database management, focusing on user profiles and expense tracking.
 - Implement basic navigation and user interface design to enhance user experience.

Week 3 (September 15-21):

- **Development Phase - Part 2:**
 - Integrate user registration and secure authentication features, including email verification and password recovery options.
 - Implement budgeting features, allowing users to set financial goals and track their expenses in real-time.

Week 4 (September 22-30):

- **Development Phase - Part 2:**

- Add reporting functionalities to generate visual analytics (charts and graphs) of spending habits and budget performance.
- Implement user profile management, enabling users to update personal information and preferences.

Testing Phase:

- Conduct comprehensive testing, including unit testing and integration testing for all features related to budget tracking and reporting.
- Identify and resolve any bugs or issues discovered during testing

Final Adjustments and Deployment Preparation:

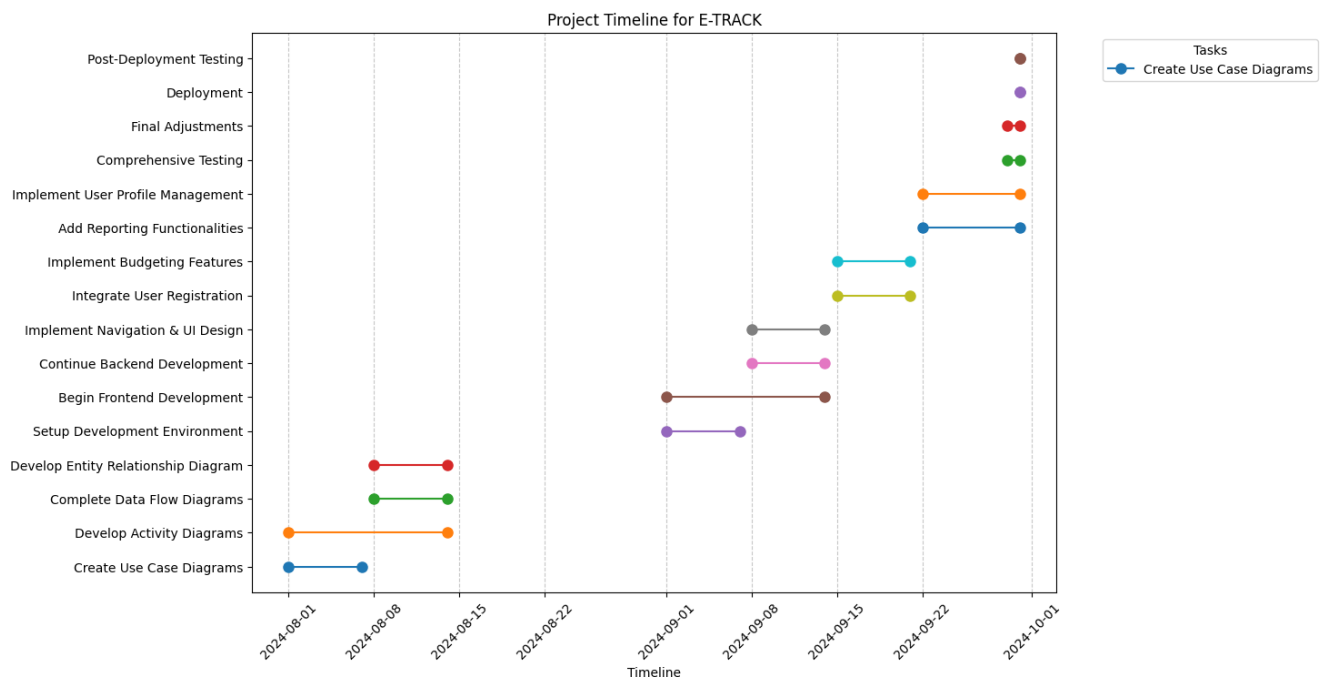
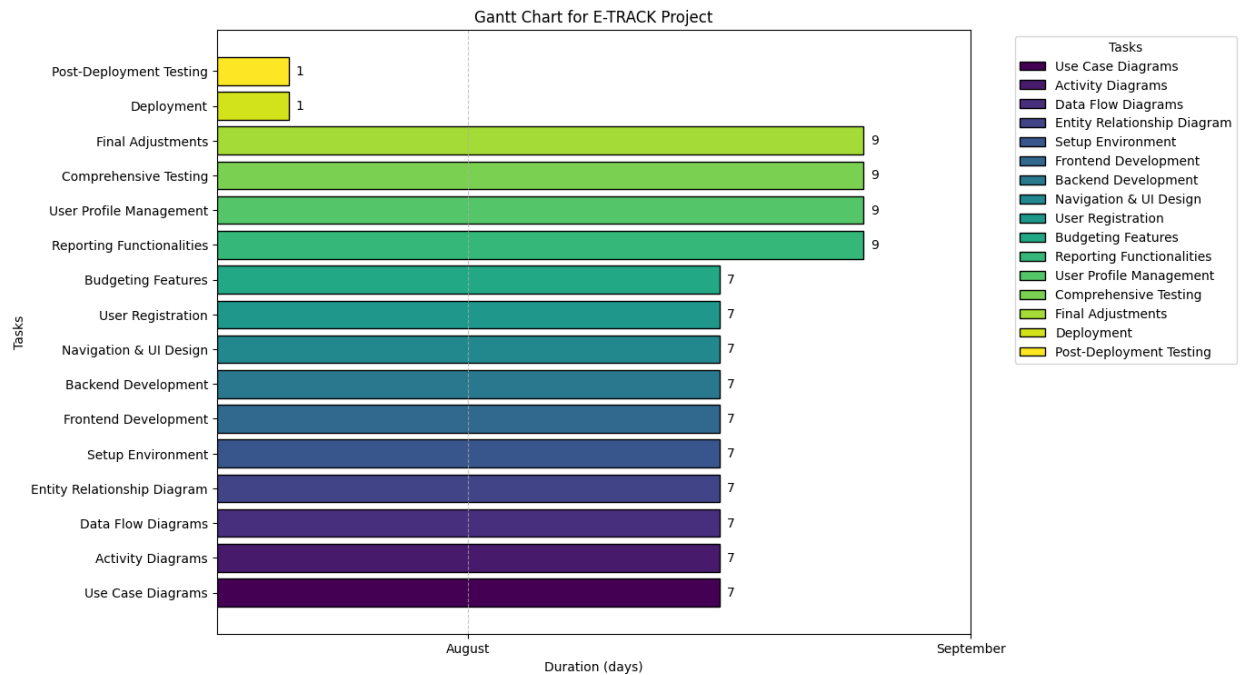
- Make necessary adjustments based on testing feedback.
- Prepare the platform for deployment by setting up the server and configuring security and performance settings.

Deployment and Launch:

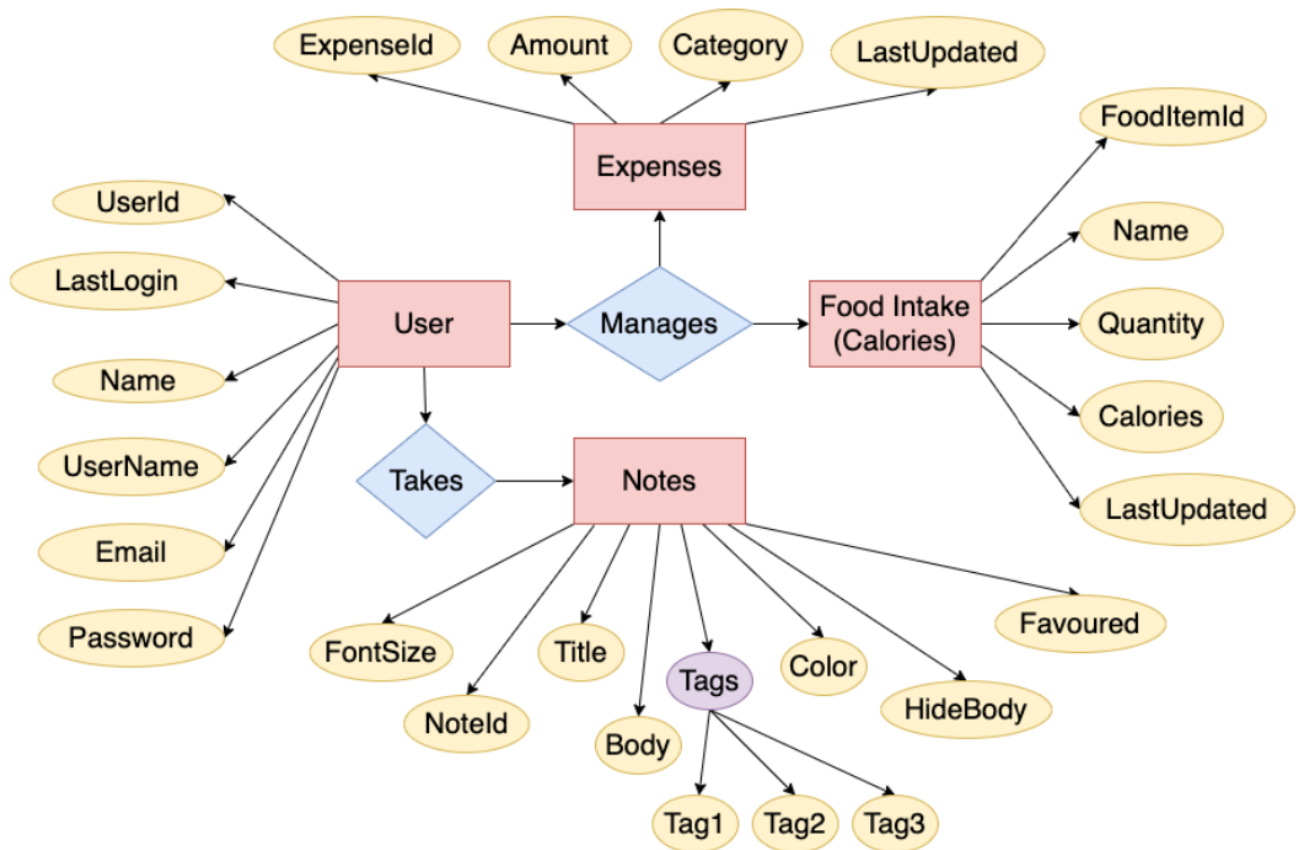
- Deploy the E-TRACK platform on a live server, making it accessible to users.
- Perform post-deployment testing to ensure that all features (budget tracking, reporting, user management) work smoothly.

**SHETH L.U.J. COLLEGE OF ARTS AND SIR M.V. COLLEGE
OF
SCIENCE AND COMMERCE**

GANTT CHART



ENTITY RELATIONSHIP DIAGRAM



ER DIAGRAM EXPLANATION

An **ER (Entity-Relationship) diagram** for the **E-TRACK** system shows the structure of how data is organized and the relationships between different entities (objects) in the system. Let's go over the key components of the ER diagram for **E-TRACK**.

Key Components:

1. **Entities:** These represent the major objects or "things" in the system that the database will store information about.
 - For **E-TRACK**, the main entities are:
 - **User:** Represents people using the system.
 - **Expense:** Represents expenses recorded by users.
 - **Category:** Represents types of expenses (e.g., food, travel).
 - **ExpenseReport:** Represents summaries of expenses over time.
 - **CurrencyConversion, Notification, BackupRestore, Feedback, SystemSettings, Admin.**
2. **Attributes:** These are the details about each entity. Attributes store the data related to the entity.
 - Example: For the **User** entity, the attributes could be UserID, Name, Email, Password.
3. **Primary Key (PK):** Each entity has a primary key, a unique identifier for every record in that entity.
 - Example: **UserID** is the primary key for the **User** entity, uniquely identifying each user.
4. **Relationships:** Entities are connected to each other through relationships. These define how the data in different entities is related.
 - **One-to-Many (1)**
): One entity can be related to multiple entities.
 - Example: A **User** can have multiple **Expenses** (one user, many expenses).
 - **Foreign Key (FK):** A key that links one entity to another by referring to its primary key.
 - Example: The UserID in the **Expense** entity is a foreign key that connects expenses to the specific user who made them.

Example of Relationships in E-TRACK:

- A **User** can create many **Expenses**, so there's a **one-to-many** relationship between **User** and **Expense**.

- Each **Expense** is assigned to a specific **Category** (e.g., "Food", "Travel"), creating a link between the **Expense** and **Category** entities.
- **ExpenseReports** summarize expenses over time (daily, monthly, yearly), so there's a relationship between **User** and **ExpenseReport**.

Diagram Symbols:

- **Rectangles:** Represent entities like **User**, **Expense**, **Category**, etc.
- **Ellipses:** Represent attributes like Name, Email, Amount, Date.
- **Diamonds:** Represent relationships between entities (e.g., **User** has many **Expenses**).
- **Lines:** Show connections between entities and relationships or attributes.

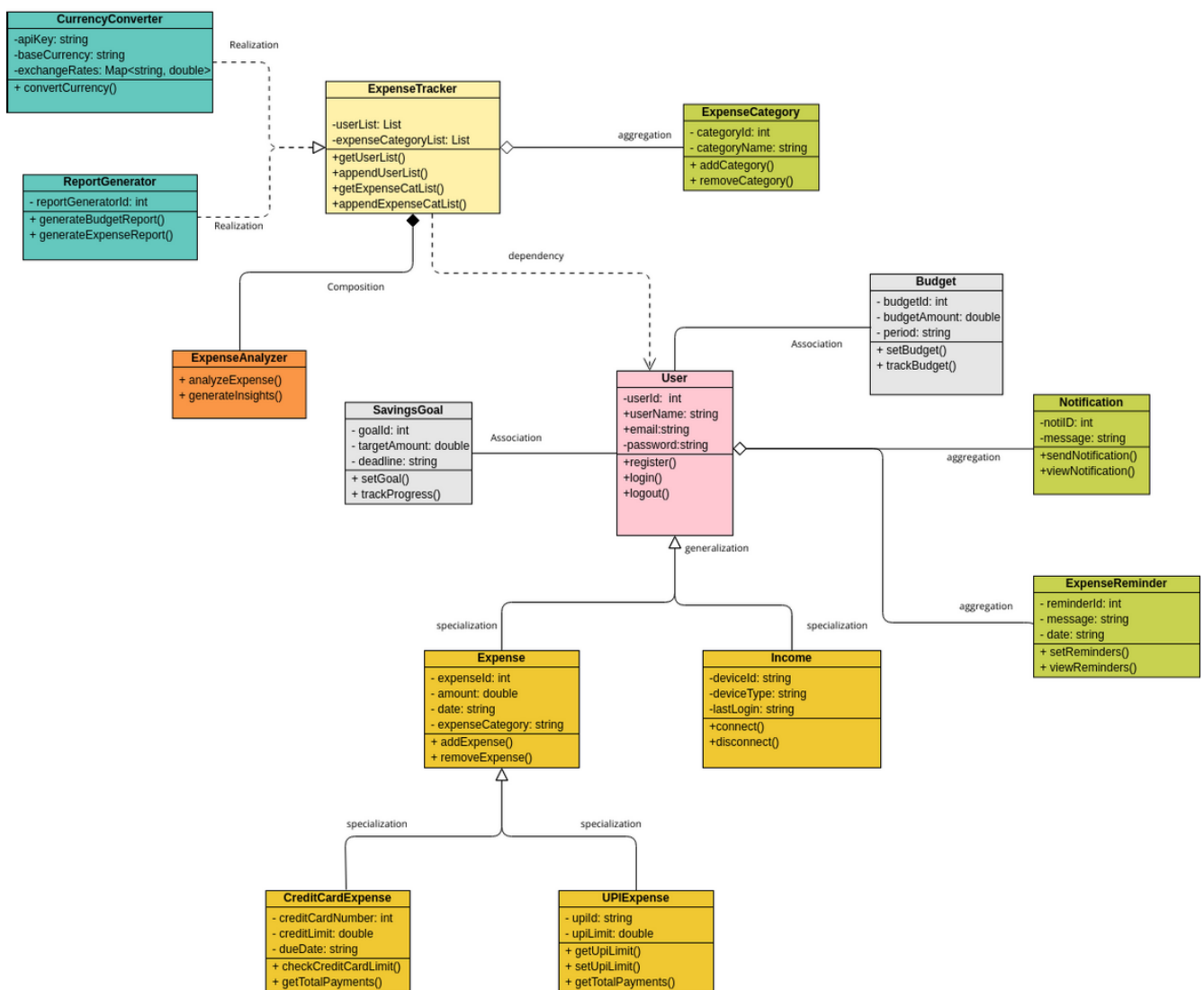
Purpose of the ER Diagram for E-TRACK:

The **ER diagram** serves several important purposes in **E-TRACK**:

- **Data Organization:** It visually shows how data is organized into different entities, making it easier to manage and understand.
- **Entity Relationships:** It clearly shows how entities are related, helping developers understand the links between users, expenses, categories, reports, and other components of the system.
- **Database Design:** The ER diagram is used as a blueprint for creating the database. It guides how tables (entities) are created, what fields (attributes) they will have, and how tables are related (through foreign keys).
- **Efficiency:** By mapping out relationships and data requirements visually, the diagram helps ensure that the database is well-structured and avoids redundancy (duplicate data) while promoting consistency.
- **System Functionality:** The diagram also helps define what functionality is supported by the system, such as user expense tracking, reporting, and category management, ensuring that all necessary data is captured.

**SHETH L.U.J. COLLEGE OF ARTS AND SIR M.V. COLLEGE
OF
SCIENCE AND COMMERCE**

CLASS DIAGRAM



CLASS DIAGRAM EXPLANATION

Key Classes and Explanations:

1. User

- Attributes: userId, email, password, name, profilePicture, contactDetails, securityQuestion, securityAnswer
- Methods: register(), login(), updateProfile(), changePassword(), recoverPassword()
- Explanation: Handles user registration, login, and profile management. It interacts with user authentication, profile updates, and password recovery.

2. Dashboard

- Attributes: dailySummary, monthlySummary, yearlySummary
- Methods: displaySummary()
- Explanation: Displays financial summaries and breakdowns for the user.

3. Expense

- Attributes: expenseId, amount, category, date, description, userId
- Methods: addExpense(), editExpense(), deleteExpense()
- Explanation: Manages the entry, modification, and deletion of expenses for users.

4. Category

- Attributes: categoryId, categoryName, userId
- Methods: addCategory(), editCategory(), deleteCategory()
- Explanation: Allows users to customize their expense categories, such as adding, editing, and removing categories.

5. ExpenseReport

- Attributes: reportId, startDate, endDate, expenseData
- Methods: generateReport()
- Explanation: Generates reports based on date ranges or criteria for better analysis.

6. Search

- Attributes: keyword, dateRange, category
- Methods: searchExpenses()
- Explanation: Provides search capabilities for users to find specific expenses using filters.

7. DataVisualization

- Attributes: chartData
- Methods: displayChart()
- Explanation: Converts the expense data into visual formats like charts and graphs.

8. CurrencyConverter

- Attributes: baseCurrency, targetCurrency, conversionRate
- Methods: convertCurrency()
- Explanation: Handles currency conversion based on user preference or location.

9. Notification

- Attributes: notificationId, message, type, date, userId
- Methods: sendNotification()
- Explanation: Sends reminders or alerts to users for bills, deadlines, or unusual expenses.

10. BackupRestore

- Attributes: backupFile, backupDate, restoreFile
- Methods: backupData(), restoreData()
- Explanation: Manages the backup and restoration of user data for data integrity.

11. DataExportImport

- Attributes: exportFileFormat, importFile
- Methods: exportData(), importData()
- Explanation: Handles exporting and importing expense data for external use (CSV, Excel).

12. SystemSettings

- Attributes: currencyPreference, dateFormat, defaultSettings
- Methods: updateSettings()
- Explanation: Manages the configuration of system-wide preferences for users and administrators.

13. UserFeedback

- Attributes: feedbackId, userId, message, response
- Methods: viewFeedback(), respondToFeedback()
- Explanation: Enables administrators to handle user feedback and resolve issues.

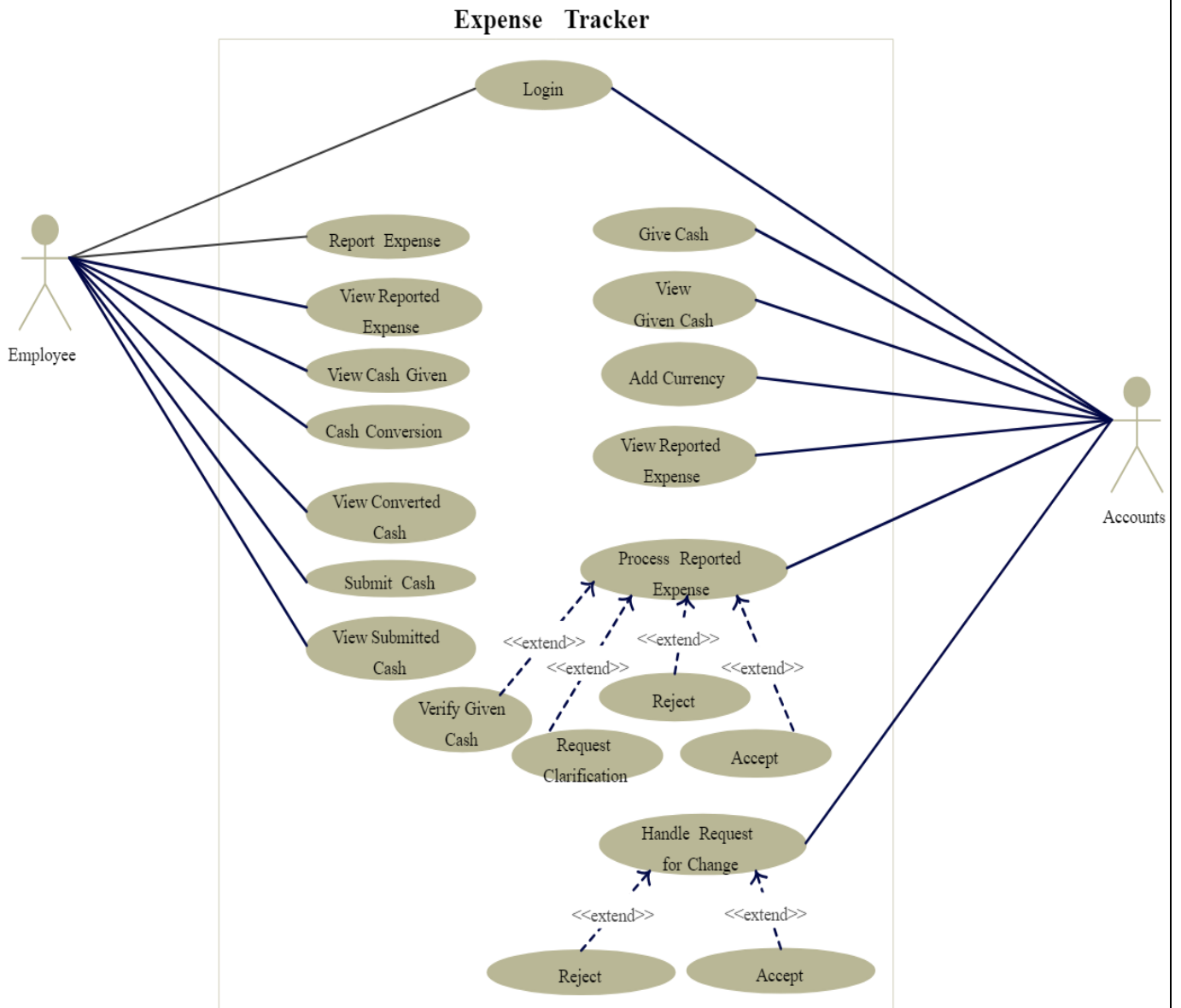
Relationships:

- **User** has a one-to-many relationship with **Expense**, **Category**, **Notification**, and **UserFeedback**.
- **Dashboard**, **Search**, **DataVisualization**, **ExpenseReport**, **CurrencyConverter**, **BackupRestore**, and **DataExportImport** rely on **Expense** data.
- **SystemSettings** affects the configuration of **Expense**, **ExpenseReport**, and other classes.

Class Diagram Structure (Description):

1. **User Class**: Central to the application, connected to **Expense**, **Category**, **Notification**, **UserFeedback**, and interacts with **Dashboard**.
2. **Expense Class**: Connected to **ExpenseReport**, **Search**, **DataVisualization**, and **Category**.
3. **SystemSettings Class**: Global settings apply to **CurrencyConverter**, **Dashboard**, and **Expense**.
4. **BackupRestore Class**: Independently connected to **User** for data management.
5. **DataExportImport Class**: Also connected to **Expense** for handling data export and import functionalities

USE CASE DIAGRAM



USE CASE DIAGRAM EXPLANATION

Actors:

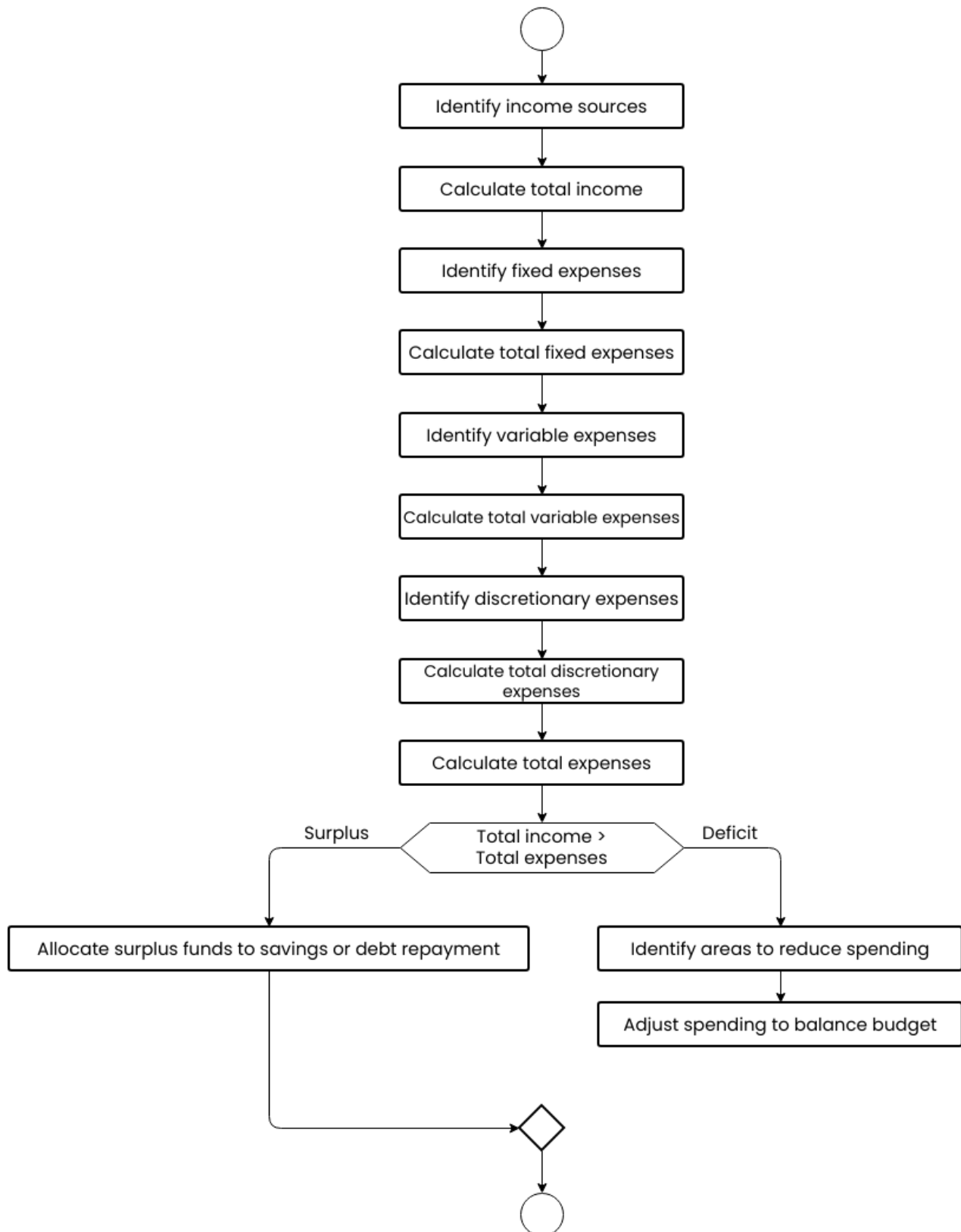
1. **User:** A registered user who interacts with the system to track and manage their expenses.
2. **Administrator:** Responsible for managing system settings, reviewing feedback, and ensuring smooth operation.

Use Cases:

1. **User Registration:**
 - **Explanation:** Users can sign up for a new account by providing personal information such as email, password, and other necessary details.
2. **User Login:**
 - **Explanation:** Provides authentication by validating user credentials for secure access to the system.
3. **User Profile Management:**
 - **Explanation:** Users can view, update, or modify their personal information such as name, email, and profile picture.
4. **Change Password:**
 - **Explanation:** Allows users to change their current password to improve security.
5. **Password Recovery:**
 - **Explanation:** Users can recover forgotten passwords via email or security questions.
6. **Dashboard:**
 - **Explanation:** Displays a summary of the user's daily, monthly, and yearly expenses, offering a comprehensive financial overview.
7. **Expense Entry:**
 - **Explanation:** Users can add new expenses by providing the necessary details, including amount, category, date, and description.
8. **Expense Management:**
 - **Explanation:** Enables users to view, edit, or delete their existing expenses to keep track of their financial activities accurately.
9. **Category Management:**
 - **Explanation:** Users can customize their expense tracking by managing categories (add, edit, delete).
10. **Expense Reports:**
 - **Explanation:** The system generates detailed reports of expenses, which can be filtered by day, month, or year, helping users analyze their spending habits.
11. **Search Functionality:**

- **Explanation:** Users can search for specific expenses using keywords, dates, or categories for more efficient management.
- 12. **Data Visualization:**
 - **Explanation:** Uses charts and graphs to visually represent expense data, making analysis easier for users.
- 13. **Currency Conversion:**
 - **Explanation:** Allows users to convert expenses into different currencies based on preferences or location.
- 14. **Notification System:**
 - **Explanation:** Sends reminders or alerts for upcoming bills, deadlines, or unusual spending patterns.
- 15. **Backup and Restore:**
 - **Explanation:** Users can back up their expense data and restore it in case of system failure, ensuring data safety.
- 16. **Export/Import Data:**
 - **Explanation:** Provides users with the option to export data to formats like CSV or Excel and import data from other sources.
- 17. **System Settings:**
 - **Explanation:** Administrators can configure system settings such as currency, date formats, and other preferences to enhance the user experience.
- 18. **User Feedback Management:**
 - **Explanation:** Allows administrators to manage and respond to user feedback, addressing issues or improving the application.

SYSTEM FLOW CHART

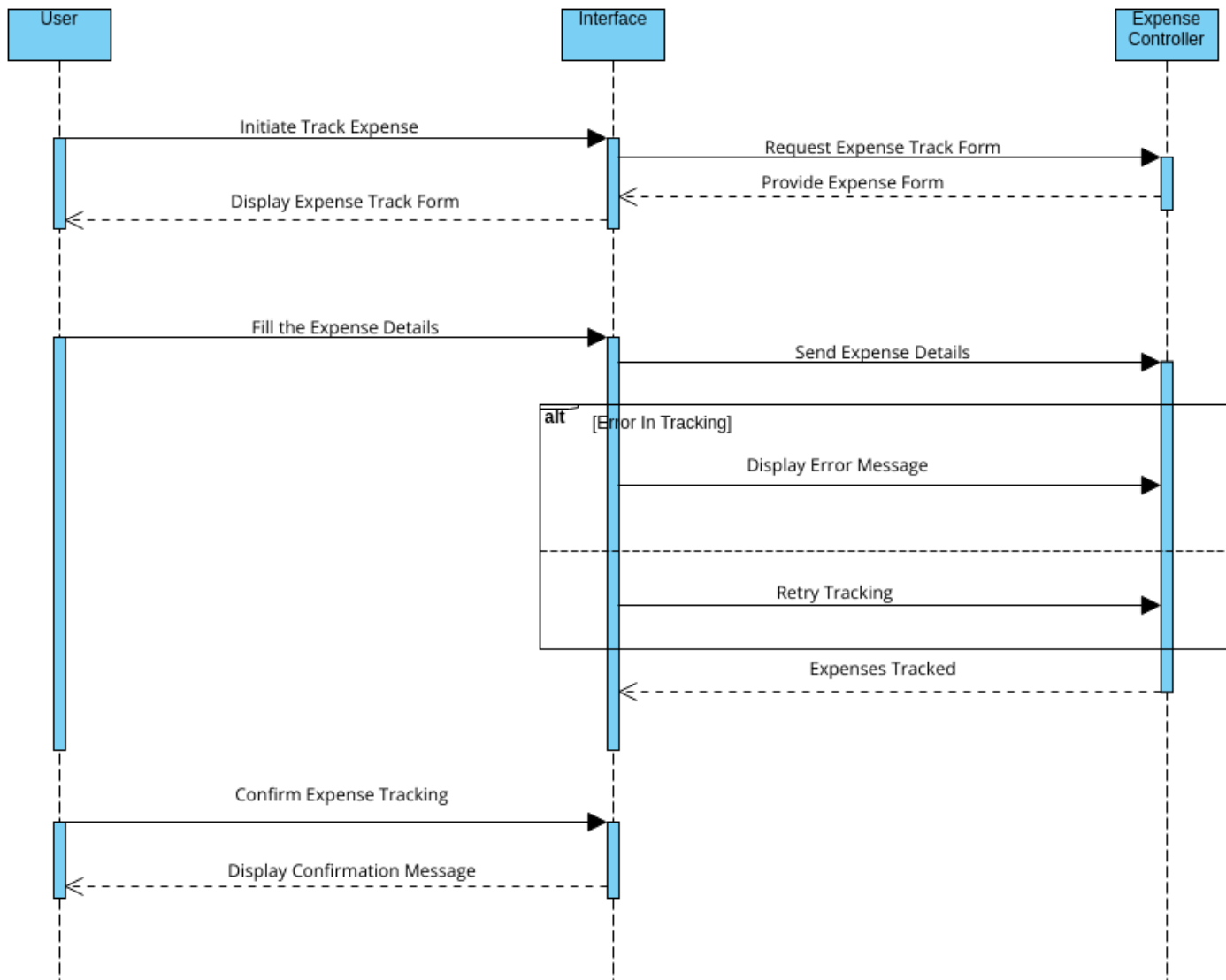


SYSTEM FLOW CHART EXPLANATION

Explanation of Flow Chart Features

1. **User Registration:** New users provide necessary details to create an account in E-TRACK.
2. **User Login:** Users authenticate themselves to access their accounts securely.
3. **User Profile Management:** Users can manage their personal information, ensuring it's up-to-date.
4. **Change Password:** Users enhance their account security by changing their passwords.
5. **Password Recovery:** Users can recover forgotten passwords through email verification or security questions.
6. **Dashboard:** A summary interface that shows users their financial status through expense breakdowns.
7. **Expense Entry:** Users can input their expenses, detailing amounts, categories, dates, and descriptions.
8. **Expense Management:** A feature to help users manage their existing expenses, providing options to edit or delete entries.
9. **Category Management:** Users can customize their experience by managing categories for expenses.
10. **Expense Reports:** Generates insightful reports to help users analyze their spending over time.
11. **Search Functionality:** Users can easily find specific expenses through targeted searches.
12. **Data Visualization:** Charts and graphs visually present the user's financial data, enhancing understanding.
13. **Currency Conversion:** Users can view their expenses in different currencies according to preferences.
14. **Notification System:** Users receive alerts for upcoming bills and spending anomalies to help manage finances.
15. **Backup and Restore:** Protects users' data by allowing backups and restoring in case of issues.
16. **Export/Import Data:** Facilitates data transfer for further analysis or integration with other tools.
17. **System Settings:** Administrators can modify system preferences to improve user experience.
18. **User Feedback Management:** Administrators address user feedback promptly, ensuring continuous improvement of the application

SEQUENCE DIAGRAM



SEQUENCE DIAGRAM EXPLANATION

Sequence Diagram Overview for E-TRACK

1. User Registration

- **Actor:** User
- **Process:** User inputs necessary details (email, password, personal information).
- **System Response:** System validates input and creates a new account.

2. User Login

- **Actor:** User
- **Process:** User enters credentials to log in.
- **System Response:** System verifies credentials and grants access.

3. User Profile Management

- **Actor:** User
- **Process:** User views or updates personal information.
- **System Response:** System saves updated information.

4. Change Password

- **Actor:** User
- **Process:** User requests to change the password.
- **System Response:** System validates the request and updates the password.

5. Password Recovery

- **Actor:** User
- **Process:** User requests password recovery.
- **System Response:** System sends recovery instructions via email or asks security questions.

6. Dashboard

- **Actor:** User
- **Process:** User accesses the dashboard.
- **System Response:** System displays a summary of expenses.

7. Expense Entry

- **Actor:** User
- **Process:** User adds new expense details (amount, category, date, description).
- **System Response:** System saves the new expense.

8. Expense Management

- **Actor:** User
- **Process:** User views, edits, or deletes existing expenses.
- **System Response:** System updates or removes expenses as requested.

9. Category Management

- **Actor:** User
- **Process:** User manages expense categories.
- **System Response:** System updates category information.

10. Expense Reports

- **Actor:** User
- **Process:** User requests expense reports.
- **System Response:** System generates reports based on selected criteria.

11. Search Functionality

- **Actor:** User
- **Process:** User searches for specific expenses.
- **System Response:** System returns relevant expenses.

12. Data Visualization

- **Actor:** User
- **Process:** User views visual representations of expenses.
- **System Response:** System generates charts and graphs.

13. Currency Conversion

- **Actor:** User
- **Process:** User selects currency conversion options.
- **System Response:** System converts expenses into the selected currency.

14. Notification System

- **Actor:** User
- **Process:** User receives alerts or reminders.
- **System Response:** System sends notifications based on criteria.

15. Backup and Restore

- **Actor:** User
- **Process:** User initiates backup or restore actions.
- **System Response:** System performs backup/restore operations.

16. Export/Import Data

- **Actor:** User
- **Process:** User exports/imports expense data.
- **System Response:** System handles the data format conversion.

17. System Settings

- **Actor:** Administrator
- **Process:** Admin configures application settings.
- **System Response:** System saves new settings.

18. User Feedback Management

- **Actor:** Administrator
- **Process:** Admin reviews user feedback.
- **System Response:** System updates status of feedback as addressed.

TOOLS & TECHNOLOGY

Frontend Development

1. HTML (HyperText Markup Language)

- **Description:** The foundational language for creating web pages. HTML structures the E-TRACK platform, organizing content such as budget details, expense entries, and reports.

2. CSS (Cascading Style Sheets)

- **Description:** A stylesheet language utilized to style web pages. CSS ensures a visually appealing and consistent layout throughout the E-TRACK application, managing elements like navigation menus, buttons, and overall aesthetics.

3. JavaScript

- **Description:** A versatile scripting language that adds dynamic and interactive features to the E-TRACK platform. JavaScript enhances user experience by enabling functionalities such as real-time expense tracking, form validation, and data visualizations.

Backend Development

4. PHP (Hypertext Preprocessor)

- **Description:** A server-side scripting language crucial for dynamic web applications. In E-TRACK, PHP manages essential operations like user authentication, processing expense data, generating reports, and handling budget management.

Database Management

5. MySQL

- **Description:** A powerful relational database management system (RDBMS). MySQL stores and organizes structured data for E-TRACK, including user accounts, budget information, expense records, and analytical reports, ensuring reliable data retrieval and storage.

Server & Database Management Tools

6. XAMPP

- **Description:** An open-source cross-platform server solution that integrates Apache, MySQL, PHP, and phpMyAdmin. XAMPP facilitates the local development of E-TRACK's backend and database functionalities, providing a robust environment for testing and deployment.

7. phpMyAdmin

- **Description:** A web-based application for managing MySQL databases. phpMyAdmin simplifies the management of E-TRACK's database, allowing users to create tables, run SQL queries, and handle data related to budgets, expenses, and user profiles efficiently.

DATABASE DESCRIPTION

Database Description for E-TRACK

1. feedback

- **Functionality:** Stores user feedback and reviews on budget management experiences, app features, and overall usability.
- **Key Data:** Feedback content, rating, user ID, timestamps, and associated transaction IDs.

2. profile

- **Functionality:** Manages user profile information for individuals utilizing the E-TRACK platform, including personal details and preferences.
- **Key Data:** User ID, name, contact information, profile picture, address, and user preferences.

3. transactions

- **Functionality:** Tracks financial transactions and budget entries made by users within the E-TRACK application.
- **Key Data:** Transaction ID, user ID, amount, transaction type (income or expense), category, date, and notes.

4. users

- **Functionality:** Stores user account and authentication information for customers and admin users within the E-TRACK platform.
- **Key Data:** User ID, email, hashed password, user role (admin, customer), and registration date.

5. conversions

- **Functionality:** Manages currency conversions to support users in tracking expenses in different currencies.
- **Key Data:** Conversion ID, source currency, target currency, conversion rate, and timestamps.

| Table Name | Functionality | Key Data |
|--------------|--|--|
| feedback | Stores user feedback and reviews on budget management experiences. | Feedback content, rating, user ID, timestamps, transaction IDs. |
| profile | Manages user profile information for individuals using E-TRACK. | User ID, name, contact information, profile picture, address, preferences. |
| transactions | Tracks financial transactions and budget entries made by users. | Transaction ID, user ID, amount, type (income/expense), category, date, notes. |
| users | Stores user account and authentication information. | User ID, email, hashed password, user role, registration date. |
| conversions | Manages currency conversions for tracking expenses. | Conversion ID, source currency, target currency, conversion rate, timestamps. |

**SHETH L.U.J. COLLEGE OF ARTS AND SIR M.V. COLLEGE
OF
SCIENCE AND COMMERCE**

DATABASE DESCRIPTION TABLE (ATTRIBUT, DATATYPE AND CONSTRAIN)

| Table Name | Attribute | Data Type | Constraints |
|---------------------|------------------|----------------|---|
| conversions | id | INT | PRIMARY KEY, AUTO_INCREMENT |
| | amount | DECIMAL(10, 2) | NOT NULL |
| | currency_from | VARCHAR(3) | NOT NULL |
| | currency_to | VARCHAR(3) | NOT NULL |
| | converted_amount | DECIMAL(10, 2) | NOT NULL |
| | conversion_date | DATETIME | NOT NULL |
| | user_id | INT | FOREIGN KEY REFERENCES users(id) |
| feedback | id | INT | PRIMARY KEY, AUTO_INCREMENT |
| | user_id | INT | FOREIGN KEY REFERENCES users(id) |
| | message | TEXT | NOT NULL |
| | created_at | DATETIME | DEFAULT CURRENT_TIMESTAMP |
| profile | id | INT | PRIMARY KEY, AUTO_INCREMENT |
| | user_id | INT | FOREIGN KEY REFERENCES users(id) |
| | name | VARCHAR(100) | NOT NULL |
| | email | VARCHAR(100) | NOT NULL, UNIQUE |
| | password | VARCHAR(255) | NOT NULL |
| | profile_picture | VARCHAR(255) | NULLABLE |
| | created_at | DATETIME | DEFAULT CURRENT_TIMESTAMP |
| transactions | id | INT | PRIMARY KEY, AUTO_INCREMENT |
| | user_id | INT | FOREIGN KEY REFERENCES users(id) |
| | amount | DECIMAL(10, 2) | NOT NULL |
| | category | VARCHAR(100) | NOT NULL |
| | date | DATE | NOT NULL |
| | description | TEXT | NULLABLE |
| | created_at | DATETIME | DEFAULT CURRENT_TIMESTAMP |
| users | id | INT | PRIMARY KEY, AUTO_INCREMENT |
| | username | VARCHAR(50) | NOT NULL, UNIQUE |
| | email | VARCHAR(100) | NOT NULL, UNIQUE |
| | password | VARCHAR(255) | NOT NULL |
| | created_at | DATETIME | DEFAULT CURRENT_TIMESTAMP |
| | updated_at | DATETIME | DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP |

MODULE DESCRIPTION

1. User Management Module

- **Functionality:** Manages user accounts for both customers and administrators.
- **Features:**
 - User registration and authentication (login/signup).
 - Profile management (updating personal information).
 - Role management (customer or admin).
 - Password recovery and change functionality.

2. Profile Management Module

- **Functionality:** Allows users to manage their personal profiles.
- **Features:**
 - View and edit personal information (name, email, contact details).
 - Update profile picture and preferences.
 - View transaction history and account activity.

3. Transaction Management Module

- **Functionality:** Tracks and manages financial transactions and budgeting.
- **Features:**
 - Record income and expenses with categorization.
 - Edit and delete transactions.
 - View transaction history with filtering options (date, type, category).
 - Generate reports on income and expenses.

4. Feedback and Review Module

- **Functionality:** Collects user feedback on their experience with the platform.
- **Features:**
 - Submit feedback on transactions and overall user experience.
 - Rate services based on user experience.
 - View feedback history and responses from administrators.

5. Conversion Management Module

- **Functionality:** Manages currency conversions for tracking expenses.
- **Features:**
 - Enter and manage currency conversion rates.
 - Convert amounts between different currencies.
 - Historical tracking of conversion rates.

6. Reports and Insights Module

- **Functionality:** Provides insights into financial health and spending patterns.
- **Features:**
 - Generate detailed reports on expenses, income, and budget balances.
 - Visualize data using charts and graphs.

- Export reports in various formats (CSV, PDF).

7. Notifications and Reminders Module

- **Functionality:** Sends alerts and reminders to users regarding their financial activities.
- **Features:**
 - Set up reminders for upcoming bills or expenses.
 - Notifications for significant transactions or budget limits.
 - Customizable reminder settings.

8. Admin Management Module

- **Functionality:** Provides administrative tools for managing the platform.
- **Features:**
 - View and manage all user accounts and transactions.
 - Monitor feedback and address user concerns.
 - Generate admin-specific reports for overall platform performance.

Integration Features

- **Payment Gateway Integration:** Secure payment processing for transactions and budget management.
- **Data Backup and Recovery:** Ensure data security and recovery options for user data and transactions.

CODE (SNIPPET)

- **Login Page**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Form</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
<div class="container">
  <!-- Login Form -->
  <div id="login-form" class="form-box">
    <h1>Login</h1>
    <form id="loginForm" action="login.php" method="POST">
      <input type="text" id="login-username" name="username" placeholder="Username" required>
      <input type="password" id="login-password" name="password" placeholder="Password" required>
      <button type="submit">Login</button>
      <p id="login-error" class="error-message"></p>    </form>
    </div>
  </div>
  <script src="login.js"></script>
</body>
</html>
```

- **Signup Page**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Signup Form</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
<div class="container">
  <!-- Signup Form -->
  <div id="signup-form" class="form-box">
    <h1>Signup</h1>
    <form id="signupForm" action="signup.php" method="POST">
      <input type="text" id="signup-username" name="username" placeholder="Username" required>
      <input type="password" id="signup-password" name="password" placeholder="Password" required>
      <button type="submit">Signup</button>
      <p>Already have an account? <a href="login.html">Login here</a></p>
    </form>
  </div>
</div>
<script src="signup.js"></script>
</body>
</html>
```

- **Home Page**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Expense Calculator</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <!-- Navigation Bar -->
  <div class="navbar">
    <a href="home.html">Home</a>
    <a href="profile.html">Profile Settings</a>
    <a href="convert.html">Convert Currency</a>
    <a href="feedback.html">Feedback & Suggestions</a>
    <a href="visualize.html">Data Visualization</a>
    <button id="logoutButton">Logout</button </div>
  <!-- Container for Main Content -->
  <div class="container">
    <!-- Form Box for Adding Transactions -->    <div class="form-box">
      <h1>Expense Calculator</h1>
      <form id="transactionForm">
        <input type="text" id="description" name="description" placeholder="Description" required>
        <input type="number" id="amount" name="amount" placeholder="Amount" step="0.01"
requirmin="0">
        <select id="type" name="type" required>
          <option value="" disabled selected>Type</option>
          <option value="income">Income</option>
          <option value="expense">Expense</option>
        </select>
        <button type="submit">Add Transaction</button>
        <p id="form-error" class="error-message"></p>    </form>    </div>
    <!-- Summary Box for displaying Income, Expenses, and Remaining Amount -->
    <div class="summary-box">
      <h2>Summary</h2>
      <div class="summary-item" id="incomeTotal">
        <span>Income:</span> <span>₹0.00</span>
      </div>
      <div class="summary-item" id="expenseTotal">
        <thead>
          <tr>
            <th>Description</th>
            <th>Amount</th>
            <th>Type</th>
            <th>Action</th>
          </tr>
        </thead>
        <tbody>
          <tbody>
        </tbody>
      </table>    </div>
    </div>
    <script src="home.js"></script>
    <script src="logout.js"></script>
  </body>
</html>
```

- **Currency convert page**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Currency Converter</title>
</head>
<body>
  <div class="container">
    <h2>Currency Converter</h2>
    <form id="convertForm" class="grid-layout" action="process_conversion.php" method="post">
      <div>
        <label for="amount">Amount</label>
        <input type="number" id="amount" name="amount" placeholder="Enter amount" required>
      </div>
      <div>
        <label for="fromCurrency">From</label>
        <select id="fromCurrency" name="fromCurrency" required>
          <option value="USD">USD - US Dollar</option>
          <option value="EUR">EUR - Euro</option>
          <option value="INR">INR - Indian Rupee</option>
          <option value="GBP">GBP - British Pound</option>
        </select>
      </div>
      <div>
        <label for="toCurrency">To</label>
        <select id="toCurrency" name="toCurrency" required>
          <option value="INR">INR - Indian Rupee</option>
          <option value="USD">USD - US Dollar</option>
          <option value="EUR">EUR - Euro</option>
          <option value="GBP">GBP - British Pound</option>
        </select>
      </div>
      <div class="full-width">
        <button type="submit">Convert</button>
      </div>
      <div class="full-width">
        <button type="button" class="back-btn" onclick="window.location.href='home.html'">Back to
Home</button>
      </div>
    </form>

    <div id="result" class="result"></div>

    <!-- Displaying conversion records -->
  </body>
</html>
```

- **Feedback Page**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Feedback</title>
</head>
<body>
  <div class="container">
    <h2>We Value Your Feedback</h2>
    <form id="feedbackForm" action="process_feedback.php" method="post">
      <div>
        <label for="name">Name</label>
        <input type="text" id="name" name="name" placeholder="Your Name" required>
      </div>
      <div>
        <label for="email">Email</label>
        <input type="email" id="email" name="email" placeholder="Your Email" required>
      </div>
      <div>
        <label for="feedback">Feedback</label>
        <textarea id="feedback" name="feedback" placeholder="Your Feedback" required></textarea>
      </div>
      <button type="submit">Submit Feedback</button>
    </form>
    <div id="responseMessage" class="message"></div>
    <!-- Popup for thank you message -->
    <div id="thankYouPopup" class="popup">
      <h3>Your feedback has been submitted. Thank you!</h3>
      <button onclick="closePopup()">Close</button>
    </div>
    <!-- Back to Home Button -->
    <div class="back-to-home">
      <button onclick="window.location.href='home.html'">Back to Home</button> </div> </div>
  </body>
</html>
```

- **Profile Page**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Profile Settings</title>
</head>
<body>
  <div class="container">
    <a href="home.html" class="back-button">Back to Home</a>
    <div class="form-card">
      <h2>Update Your Profile</h2>
      <form id="profileForm" action="save_profile.php" method="post">
        <div class="grid-container">
          <div class="form-group">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" required>
          </div>
          <div class="form-group">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
          </div>
          <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
          </div>
          <div class="form-group">
            <label for="gender">Gender:</label>
            <select id="gender" name="gender" required>
              <option value="">Select Gender</option>
              <option value="male">Male</option>
              <option value="female">Female</option>
              <option value="others">Others</option>
            </select>
          </div>
          <div class="form-group">
            <label for="contact">Contact:</label>
            <input type="text" id="contact" name="contact">
          </div>
          <div class="form-group">
            <label for="address">Address:</label>
            <input type="text" id="address" name="address">
          </div>
          <div class="form-group">
            <label for="occupation">Occupation:</label>
            <input type="text" id="occupation" name="occupation">
          </div>
          <div class="form-group">
            <label for="age">Age:</label>
            <input type="number" id="age" name="age">
          </div>
          <div class="form-group">
```

**SHETH L.U.J. COLLEGE OF ARTS AND SIR M.V. COLLEGE
OF
SCIENCE AND COMMERCE**

```
<label for="dob">Date of Birth:</label>
<input type="date" id="dob" name="dob">
</div>
</div>
<div class="form-group">
  <button type="submit">Save</button>
</div>
</form>
<div id="profileData" class="profile-data" style="display: none;">
  <h3>Profile Details</h3>
  <p><strong>Username:</strong> <span id="displayUsername"></span></p>
  <p><strong>Email:</strong> <span id="displayEmail"></span></p>
  <p><strong>Gender:</strong> <span id="displayGender"></span></p>
  <p><strong>Contact:</strong> <span id="displayContact"></span></p>
  <p><strong>Address:</strong> <span id="displayAddress"></span></p>
  <p><strong>Occupation:</strong> <span id="displayOccupation"></span></p>
  <p><strong>Age:</strong> <span id="displayAge"></span></p>
  <p><strong>Date of Birth:</strong> <span id="displayDob"></span></p>
  <div class="action-buttons">
    <button id="editButton" class="edit-button">Edit</button>
    <button id="deleteButton" class="delete-button">Delete</button>
  </div>
</div>
</div>
</div>
</body>
</html>
```

- **Visualization Page**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Data Visualization</title>
  <link rel="stylesheet" href="styles.css">
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
  <!-- Navigation Bar -->
  <div class="navbar">
    <a href="home.html">Home</a>
    <a href="profile.html">Profile Settings</a>
    <a href="convert.html">Convert Currency</a>
    <a href="feedback.html">Feedback & Suggestions</a>
    <a href="visualize.html">Data Visualization</a>
  </div>

  <!-- Container for Main Content -->
  <div class="container">
    <h1>Data Visualization</h1>

    <!-- Dropdown for Chart Type Selection -->
    <div class="dropdown">
      <label for="chartType">Select Chart Type:</label>
      <select id="chartType">
        <option value="bar">Bar Chart</option>
        <option value="line">Line Chart</option>
        <option value="pie">Pie Chart</option>
      </select>
    </div>

    <!-- Canvas for Chart -->
    <canvas id="summaryChart" width="400" height="200"></canvas>

    <!-- Buttons -->
    <div class="buttons">
      <button id="downloadChart">Download Chart</button>
      <a href="home.html" id="backButton">Back to Home</a>
    </div>
  </div>

  <script src="visualize.js"></script>
</body>
</html>
```

- **Home.js**

```
document.addEventListener('DOMContentLoaded', function() {
  const form = document.getElementById('transactionForm');
  const descriptionInput = document.getElementById('description');
  const amountInput = document.getElementById('amount');
  const typeSelect = document.getElementById('type');
  const transactionTable =
document.getElementById('transactionTable').getElementsByTagName('tbody')[0];
  // Load transactions and summary from the database on page load
  loadTransactions();
  updateSummary();
  form.addEventListener('submit', function(e) {
    e.preventDefault();
    const description = descriptionInput.value.trim();
    const amount = parseFloat(amountInput.value).toFixed(2);
    const type = typeSelect.value;

    if (description && !isNaN(amount) && type) {
      addTransaction(description, amount, type);
    } else {
      showError('form-error', 'Please fill in all fields correctly.');
```

```
}

function deleteTransaction(id, row) {
    const xhr = new XMLHttpRequest();
    xhr.open('POST', 'delete_transaction.php', true);
    xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
    xhr.onreadystatechange = function() {
        if (xhr.readyState === XMLHttpRequest.DONE && xhr.status === 200) {
function updateSummary() {
    const xhr = new XMLHttpRequest();
    xhr.open('GET', 'get_summary.php', true);
    xhr.onreadystatechange = function() {
        if (xhr.readyState === XMLHttpRequest.DONE && xhr.status === 200) {
            try {
                const summary = JSON.parse(xhr.responseText);
                console.log('Summary data:', summary); // Debugging line
                if (summary.status === 'success') {
                    const income = parseFloat(summary.income).toFixed(2);
                    const expense = parseFloat(summary.expense).toFixed(2);
                    const remaining = parseFloat(summary.remaining).toFixed(2);

                    document.getElementById('incomeTotal').innerHTML = `<span>Income:</span>
<span>₹${income}</span>`;
                    document.getElementById('expenseTotal').innerHTML = `<span>Expenses:</span>
<span>₹${expense}</span>`;
                    document.getElementById('remainingAmount').innerHTML = `<span>Remaining:</span>
<span>₹${remaining}</span>`;
                } else {
                    showError('form-error', 'Error fetching summary.');
```

- **Visualize.js**

```
document.addEventListener('DOMContentLoaded', function() {
  const chartTypeSelect = document.getElementById('chartType');
  const downloadButton = document.getElementById('downloadChart');
  let chart = null;
  // Fetch summary data and generate chart
  fetchSummaryData();
  chartTypeSelect.addEventListener('change', function() {
    fetchSummaryData();
  });
  downloadButton.addEventListener('click', function() {
    if (chart) {
      const link = document.createElement('a');
      link.href = chart.toBase64Image();
      link.download = 'chart.png';
      link.click();
    }
  });
  function fetchSummaryData() {
    const xhr = new XMLHttpRequest();
    xhr.open('GET', 'get_summary.php', true);
    xhr.onreadystatechange = function() {
      if (xhr.readyState === XMLHttpRequest.DONE && xhr.status === 200) {
        try {
          const summary = JSON.parse(xhr.responseText);
          if (summary.status === 'success') {
            const income = parseFloat(summary.income).toFixed(2);
            const expense = parseFloat(summary.expense).toFixed(2);
            const remaining = parseFloat(summary.remaining).toFixed(2);
            generateChart(income, expense, remaining);
          } else {
            console.error('Error fetching summary.');
```

- **Db.php**

```
<?php
// Database configuration
$servername = "localhost";
$username = "root"; // Use your DB username
$password = ""; // Use your DB password
$dbname = "E-TRACK";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

- **add_transcation.php**

```
<?php
// Database connection
$conn = new mysqli('localhost', 'root', '', 'E-TRACK');
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$description = $conn->real_escape_string($_POST['description']);
$amount = $conn->real_escape_string($_POST['amount']);
$type = $conn->real_escape_string($_POST['type']);
// Insert transaction
$sql = "INSERT INTO transactions (description, amount, type) VALUES ('$description', '$amount', '$type')";
if ($conn->query($sql) === TRUE) {
    $transaction_id = $conn->insert_id;
    echo json_encode(['status' => 'success', 'transaction' => ['id' => $transaction_id, 'description' =>
$description, 'amount' => $amount, 'type' => $type]]);
} else {
    echo json_encode(['status' => 'error']);
}

$conn->close();
?>
```

- **delete_transcation.php**

```
<?php
// Database connection
$conn = new mysqli('localhost', 'root', '', 'E-TRACK');
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$id = intval($_POST['id']);
// Delete transaction
$sql = "DELETE FROM transactions WHERE id = $id";
if ($conn->query($sql) === TRUE) {
    echo json_encode(['status' => 'success']);
} else {
    echo json_encode(['status' => 'error']);
}
$conn->close();
?>
```

- **currency_conversion.php**

```
<?php
$servername = "localhost";
$username = "root"; // Change as needed
$password = ""; // Change as needed
$dbname = "E-TRACK";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $amount = $_POST['amount'];
    $fromCurrency = $_POST['fromCurrency'];
    $toCurrency = $_POST['toCurrency'];
    $convertedAmount = $_POST['convertedAmount'];
    $description = $_POST['description'];
    // Insert data into the table
    $stmt = $conn->prepare("INSERT INTO conversions (amount, from_currency, to_currency,
converted_amount, description) VALUES (?, ?, ?, ?, ?)");
    $stmt->bind_param("sssss", $amount, $fromCurrency, $toCurrency, $convertedAmount, $description);

    if ($stmt->execute()) {
        echo "Data stored successfully.";
    } else {
        echo "Error: " . $stmt->error;
    }
}

$stmt->close();
}
// Retrieve latest conversion records
$sql = "SELECT * FROM conversions ORDER BY timestamp DESC LIMIT 5";
$result = $conn->query($sql);

$conversionRecords = [];
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $conversionRecords[] = $row;
    }
}

$conn->close();

?>

<!-- Displaying the converted amounts -->
<div class="conversion-records">
    <h3>Recent Conversions</h3>
    <ul>
        <?php foreach ($conversionRecords as $record): ?>
            <li>
```

```
Amount: <?= htmlspecialchars($record['amount']) ?> <?=
htmlspecialchars($record['from_currency']) ?> to <?= htmlspecialchars($record['converted_amount']) ?> <?=
htmlspecialchars($record['to_currency']) ?> - <?= htmlspecialchars($record['description']) ?> (<?=
$record['timestamp'] ?>)
</li>
<?php endforeach; ?>
</ul>
</div>
```

- **feedback.php**

```
<?php
$servername = "localhost";
$username = "root"; // Change as needed
$password = ""; // Change as needed
$dbname = "E-TRACK";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $feedback = $_POST['feedback'];
    // Insert data into the table
    $stmt = $conn->prepare("INSERT INTO feedback (name, email, feedback) VALUES (?, ?, ?)");
    $stmt->bind_param("sss", $name, $email, $feedback);

    if ($stmt->execute()) {
        echo "success";
    } else {
        echo "error";
    }

    $stmt->close();
}

$conn->close();
?>
```

EXPECTED OUTCOME

Enhanced Accessibility to Financial Management:

- Users will have easy access to comprehensive tools for tracking expenses, income, and budgeting, enabling them to manage their finances effectively all in one platform.

Convenience and Time Savings:

- By using E-TRACK, users can quickly log transactions, generate reports, and analyze their financial status, streamlining the financial management process and saving valuable time.

Improved Financial Awareness and Planning:

- E-TRACK will provide users with insights and visualizations of their spending habits, ensuring they have a clear understanding of their financial health and can make informed decisions.

Support and Recommendations:

- The platform will offer resources such as budgeting tips, financial advice, and personalized recommendations, helping users to enhance their financial literacy and make smarter choices.

Promotions and Special Offers:

- Users will have access to exclusive deals and discounts related to financial services, enhancing value for money and encouraging long-term usage of the platform.

Community Engagement:

- E-TRACK will foster a sense of community through user-generated content, allowing users to share their financial experiences, tips, and insights with fellow users.

Real-Time Updates and Notifications:

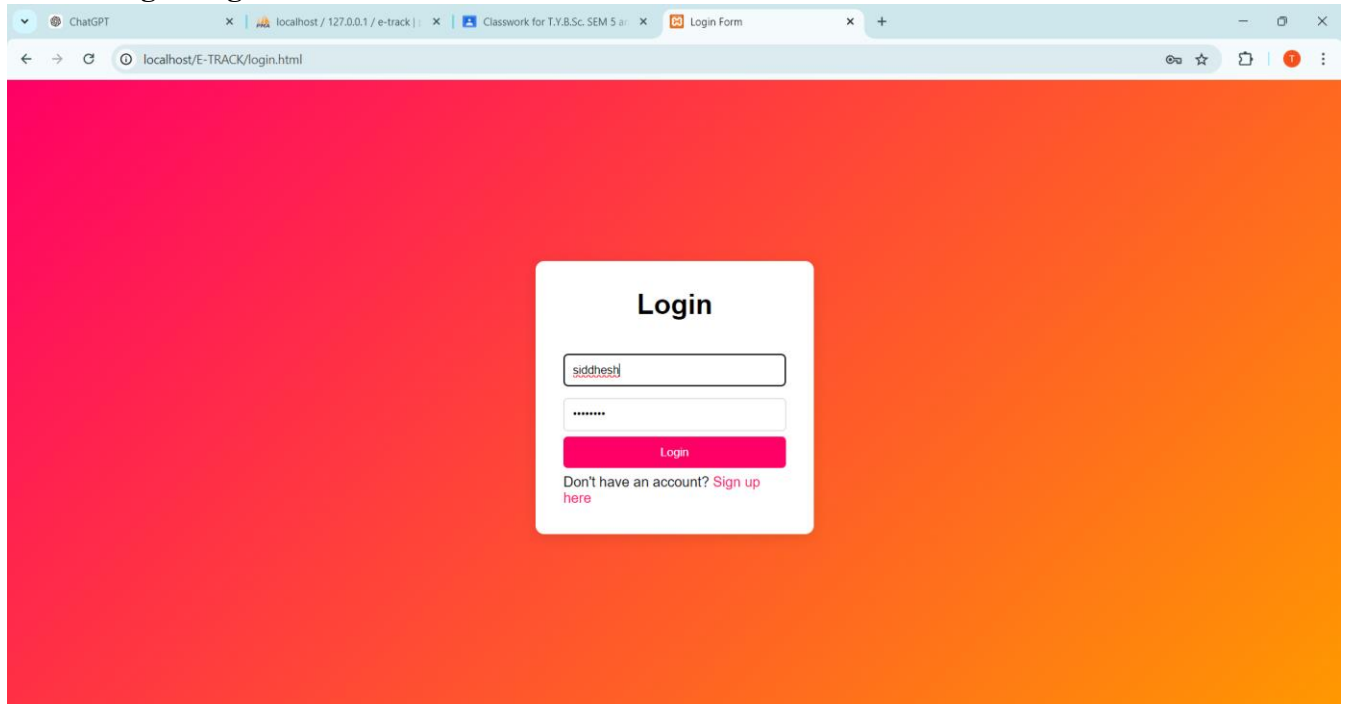
- Customers will receive real-time updates on their transactions, budget alerts, and reminders, ensuring they remain informed about their financial activities.

Streamlined Communication with Financial Advisors:

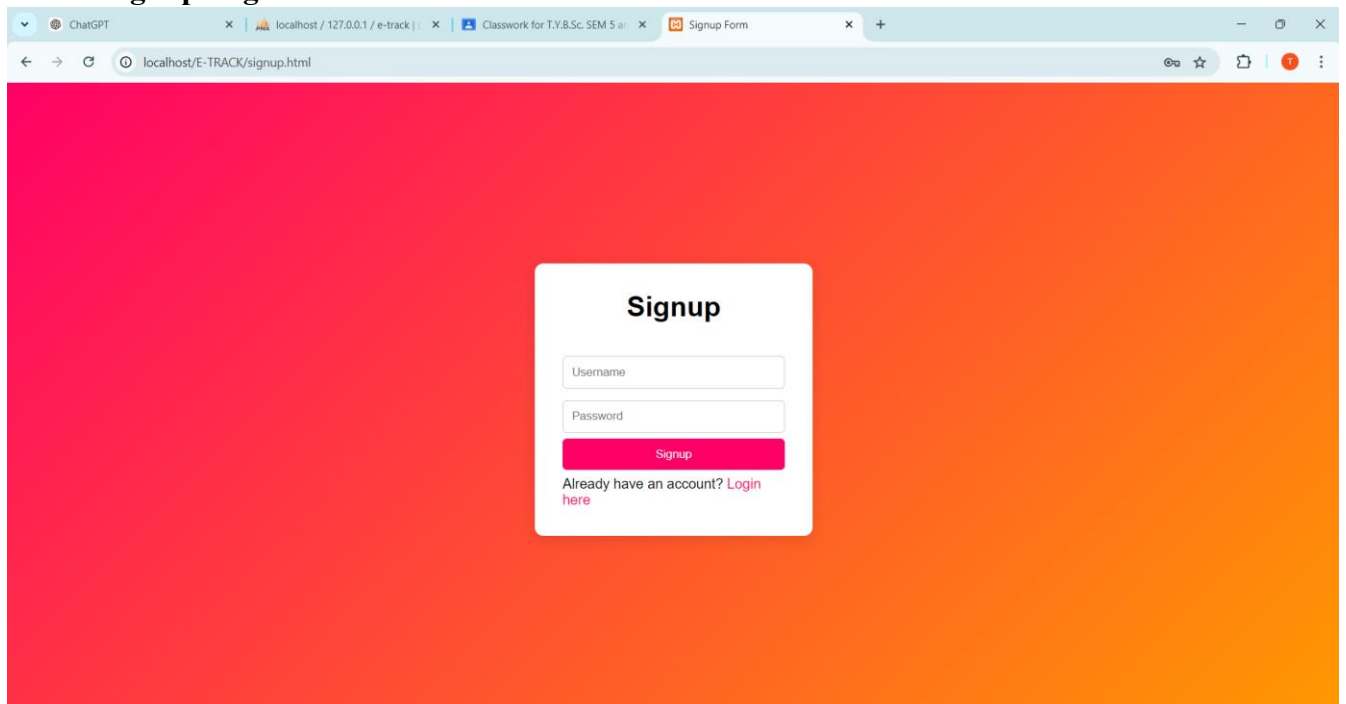
- The platform will facilitate direct communication between users and financial advisors, allowing for personalized guidance and support to address specific financial concerns efficiently.

SCREENSHOTS

- **Login Page**



- **Signup Page**



**SHETH L.U.J. COLLEGE OF ARTS AND SIR M.V. COLLEGE
OF
SCIENCE AND COMMERCE**

- **Home Page**

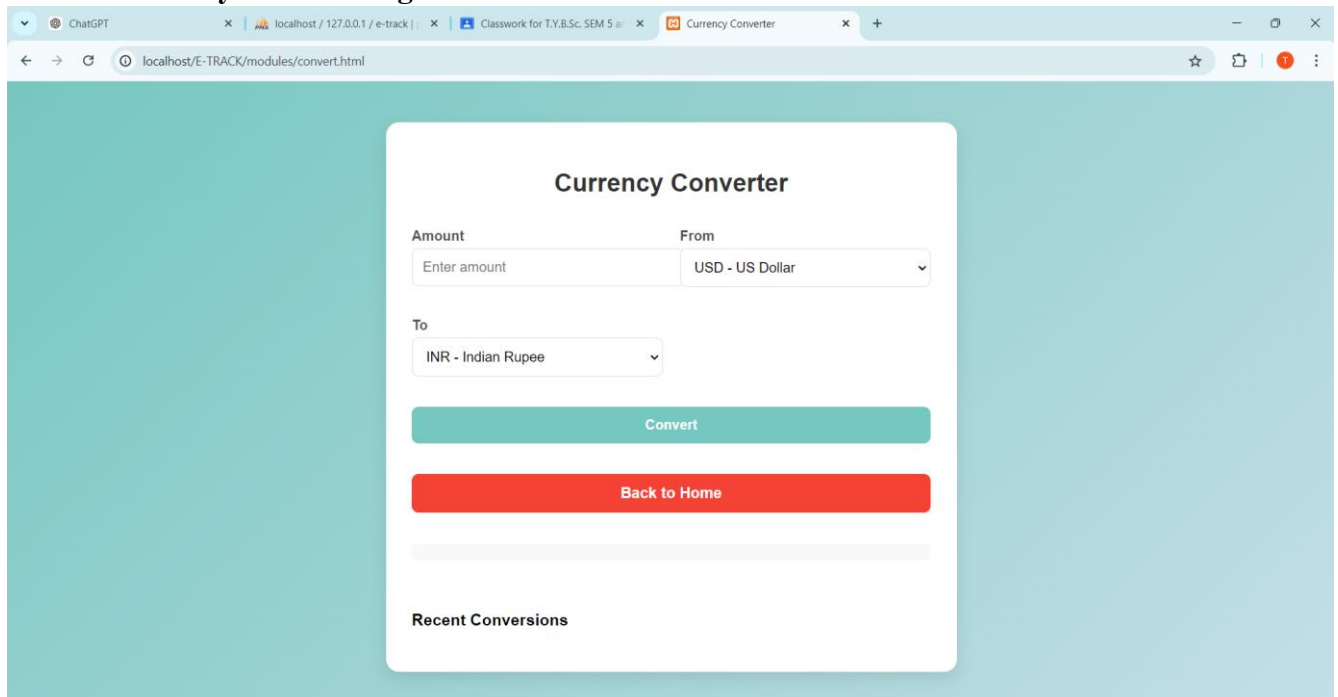
The screenshot shows a web browser window with the URL `localhost/E-TRACK/modules/home.html`. The application has a dark navigation bar with links: Home, Profile Settings, Convert Currency, Feedback & Suggestions, and Data Visualization. A red bar below the navigation bar contains a "Logout" button. The main content area is titled "Expense Calculator" and contains a form with fields for "Description", "Amount", and "Type" (a dropdown menu). Below the form is a blue "Add Transaction" button. Underneath the form is a "Summary" section with three rows: "Income:" with a value of ₹NaN, "Expenses:" with a value of ₹NaN, and "Remaining:" with a value of ₹0.00. At the bottom is a "Transaction History" section with a table that has columns for Description, Amount, Type, and Action.

- **Profile Page**

The screenshot shows a web browser window with the URL `localhost/E-TRACK/modules/profile.html`. The application has a dark navigation bar with links: Home, Profile Settings, Convert Currency, Feedback & Suggestions, and Data Visualization. A red bar below the navigation bar contains a "Logout" button. The main content area is titled "Update Your Profile" and contains a form with fields for Username, Email, Password, Gender, Contact, Address, Occupation, Age, and Date of Birth. Below the form is a green "Save" button. A blue "Back to Home" button is located at the top of the form.

**SHETH L.U.J. COLLEGE OF ARTS AND SIR M.V. COLLEGE
OF
SCIENCE AND COMMERCE**

- **Currency Convert Page**



Currency Converter

Amount From

USD - US Dollar

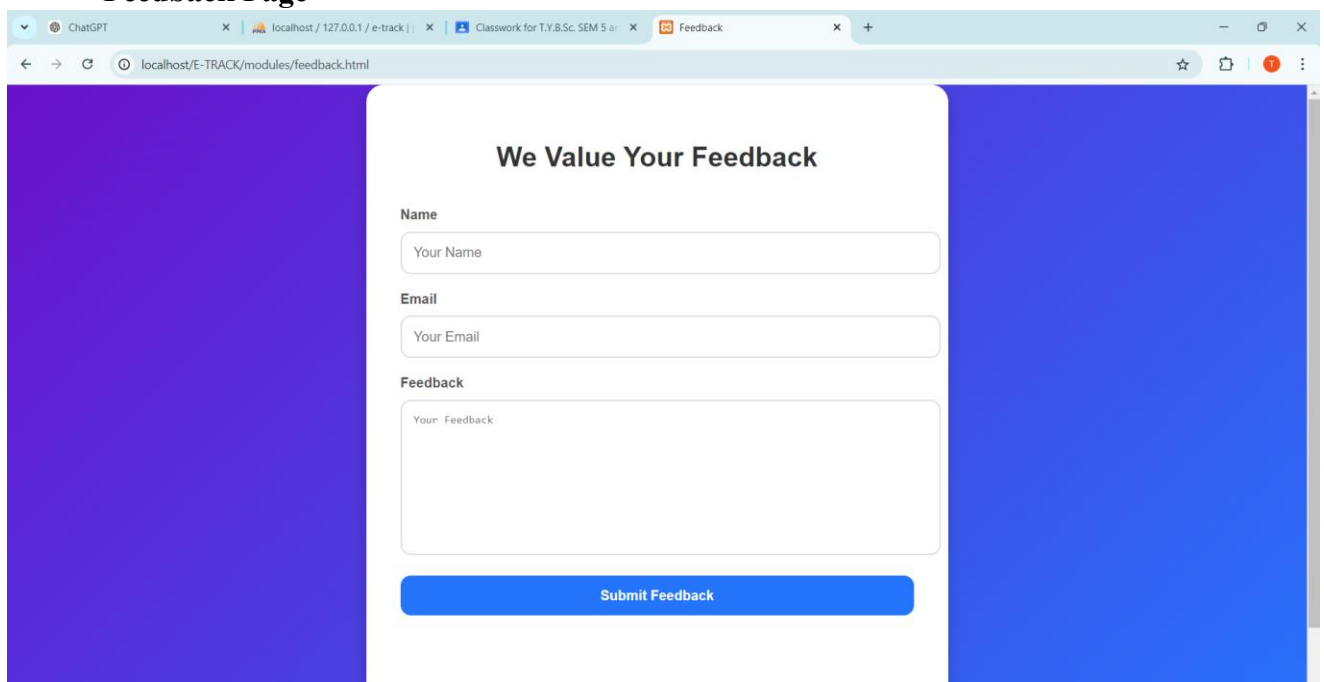
To

Convert

Back to Home

Recent Conversions

- **Feedback Page**



We Value Your Feedback

Name

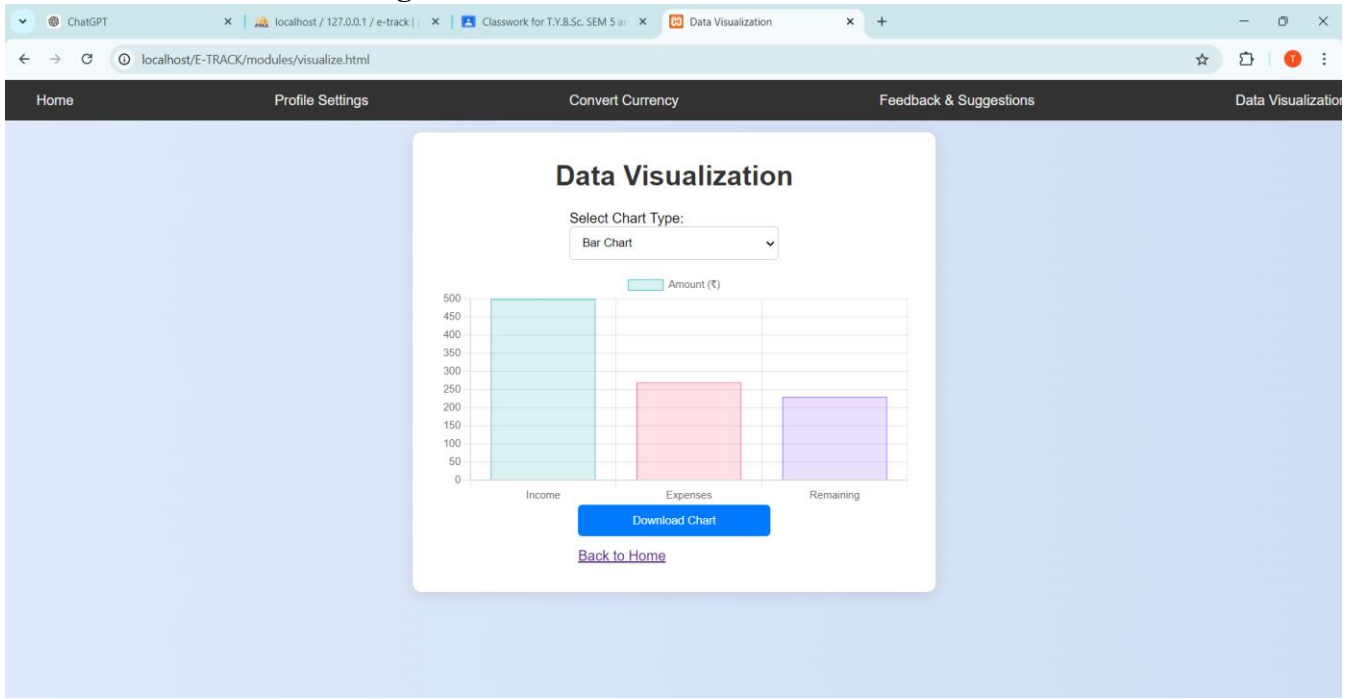
Email

Feedback

Submit Feedback

**SHETH L.U.J. COLLEGE OF ARTS AND SIR M.V. COLLEGE
OF
SCIENCE AND COMMERCE**

- **Data Visualization Page**



- **Transcation History**

The screenshot shows a web application interface for transaction history. The browser tabs include 'ChatGPT', 'localhost / 127.0.0.1 / e-track |', 'Classwork for T.Y.B.Sc. SEM 5 a:', and 'Expense Calculator'. The address bar shows 'localhost/E-TRACK/modules/home.html'. The navigation bar has links for 'Home', 'Profile Settings', 'Convert Currency', 'Feedback & Suggestions', and 'Data Visualization'. A red 'Logout' button is visible. The main content area displays a summary of financial data: 'Income: ₹500.00', 'Expenses: ₹270.00', and 'Remaining: ₹230.00'. Below this is a 'Transaction History' table with columns for 'Description', 'Amount', 'Type', and 'Action'.

| Description | Amount | Type | Action |
|-------------|---------|---------|--------|
| salary | ₹500.00 | income | Delete |
| food | ₹250.00 | expense | Delete |
| travel | ₹20.00 | expense | Delete |

CONCLUSION

E-TRACK signifies a revolutionary advancement in personal financial management, creating a seamless connection between users and their financial goals through a comprehensive digital platform. By providing an array of features such as intuitive expense tracking, insightful budgeting tools, and robust reporting capabilities, E-TRACK empowers users to take control of their finances while promoting efficiency, clarity, and informed decision-making. The platform's future prospects include the integration of innovative technologies and personalized financial advice to continuously meet the evolving needs of its users. With its commitment to excellence and user-centric design, E-TRACK is set to become a pivotal player in the financial management landscape, fostering positive financial habits and enhancing overall financial well-being.

RESOURCES

1. Human Resources

- **Project Manager:**
Responsible for overseeing the planning, execution, and closure of the E-TRACK project, ensuring that objectives, timelines, and budgets are met effectively.
- **Frontend Developers:**
Proficient in HTML, CSS, and JavaScript to create a responsive and user-friendly interface for E-TRACK, enhancing user experience and engagement.
- **Backend Developers:**
Skilled in PHP for server-side scripting and MySQL for database management, utilizing tools like XAMPP for local development and testing.
- **UI/UX Designers:**
Focus on designing an intuitive and engaging user interface that caters to the needs of users managing their expenses and finances.

2. Software Resources

- **Development Tools:**
 - **Visual Studio Code:** An integrated development environment (IDE) for writing, debugging, and managing the E-TRACK codebase efficiently.
 - **XAMPP:** A local development environment that includes Apache, MySQL, PHP, and phpMyAdmin, facilitating smooth development and testing.
- **Frameworks and Libraries:**
 - **jQuery:** A library that simplifies JavaScript operations, enhancing dynamic features on the E-TRACK platform.
- **Database:**
 - **MySQL:** A robust relational database management system used to store and manage user accounts, transactions, expense categories, and other relevant data, easily managed via phpMyAdmin.
- **APIs:**
 - **Currency Conversion API:** Provides real-time currency exchange rates for users to manage expenses in multiple currencies.
- **Development Machines:**
Computers equipped with adequate processing power, memory, and storage to develop, test, and debug the E-TRACK platform efficiently.

REFERENCES

- www.chatgpt.com
- www.youtube.com (Tutorials for web development and best practices)
- www.google.com (Search engine for research and additional resources)
- www.geeksforgeeks.com (Articles and guides on web technologies and programming)
- www.w3schools.com (Web development tutorials for HTML, CSS, JavaScript, and PHP)
- www.stackoverflow.com (Community for troubleshooting and programming questions)

UNDERTAKING

I, **Siddhesh Chalke**, a student of Computer Science at Sheth L.U.J. College of Arts And Sir M.V. College of Science and Commerce, hereby declare that the project titled “**E-TRACK**” Website of Expanse Management System is my own original work. I affirm that this project has not been copied, purchased, or plagiarised from any other source. All the ideas, code, and design elements contained within this project are the result of my own efforts and understanding. I understand the importance of academic integrity and assure that this work is my own contribution to my learning and development in the field of computer science.

Date:

Place:

Signature:

Name: Siddhesh Chalke

Roll Number: T074