BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI

MTech. In Software Systems

Expenso : Smart Finance Planning & Management

Submitted by

Sakhee Manjari Panda

2020WA15164

Dissertation Work carried out at Wipro Technologies, Chennai

SSWTZG628T DISSERTATION

Expenso : Smart Finance Planning and Management

Outline of Dissertation submitted in partial

Fulfillment of the requirements of

M. Tech. Software Systems Degree Program

By

Sakhee Manjari Panda (2020WA15164)

Under the supervision of

Suganya C

Dissertation work carried out at

Wipro Technologies, Chennai

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Dissertation entitled **Expenso : Smart Finance Planning and Management** submitted by Sakhee Manjari Panda ID No. 2020WA15164.
In partial fulfillment of the requirements of SSWTZG628T Dissertation, embodies the work done by him/her under my supervision.

[illegible]

Signature of the Supervisor
Name: Suganya C
Designation: Sr. Project Lead

Date: 10th November 2024

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

First Semester 2024- 25

SSWT ZG628T DISSERTATION

FINAL SEMESTER EVALUATION FORM

Section I

(To be filled by the Student and returned to the Supervisor)

ID No. : 2020WA15164 **Name of Student :** Sakhee Manjari Panda

Name of Supervisor: Suganya C

Name of the Examiner(s) : Sudharsan Selvaraj & Saikiran Ananthula

Dissertation Title : “Expenso : Smart Finance Planning and Management”

Section II

(To be filled by the Supervisor in consultation with the examiner(s))

Comments on the dissertation from Examiner and Supervisor (Select Y or N)

1. **Quantum of work**

a. Justifiable as efforts for 5 weeks duration	**Y**
b. Work is in line with the commitments made in outline	**Y**

2. **Type of work**

a. Client assignment	**N**
b. Organization specific task	**N**
c. General study project such as white paper	**Y**
d. Any other (kindly elaborate below in a line or two if Y	**N**

3. **Nature of work**

a. Routine in nature	**Y**
b. Involved creativity and rational thinking	**Y**

Kindly elaborate below if answer for above is "Y" :

The application looks good and all the features included are working properly

4. Evaluation methodology

a. Evaluation done based on presentation to supervisor and examiner	**Y**
b. Evaluation done through Viva conducted by supervisor and examiner	**Y**
c. Student regularly interacted with supervisor and incorporated the suggestions made	**Y**

d. Brief description on the report submitted, quality of presentation and suggestions given for improvement :

As per the discussion I had with Sakhee, the document is properly describing the understanding of her work. But still I suggest her to work on it more to make it more user interactive, mobile app integration, and add gamification and API for Third-party Integrations.

5. Final semester evaluation matrix

Tick the appropriate box (1 is lowest and 5 is the highest)

Dimension **Rank→**	**1**	**2**	**3**	**4**	**5**
Student abilities in general					
Understanding of the subject of dissertation					*
Creative thinking ability to come up with new ideas					*
Viva / Seminar presentation					
Communication ability					*
Organization of material				*	
Response to review questions					*
Cohesive thinking ability				*	
Report submitted					
Report structure and format				*	
Technical content of the report				*	
Explanation on the significance of the assignment					*
Analysis of alternative approaches				*	

Any other comments : It is having all the features for the budget and finance management. The good thing is the addition of the EMI, SIP calculators. Based on that the user can plan for investment. In the future work the notification, add some educational reference for the user so that he/she can learn about the investment and how to manage their finances.

Date: 10th November 2024

Signature of Supervisor : [illegible]

Signature of examiner : [illegible]

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1	Introduction	9
	Overview of Expenso	9
	Purpose of the Application	10
	Project Overview	10
	Key Objectives	12
2	System Overview	13
	Key Features	13
	Technologies Used	14
3	System Architecture	16
	High-Level Architecture Diagram	16
	Explanation of Architecture Components	17
	Data Flow Diagram	18
4	Modules and Features	20
	Overview of Modules	20
	Detailed Description of Each Module	21
	- User Management	21
	- Registration, Login, and Authentication	21
	- Role-based Access Control (Admin, User)	21
	- Expense Management	21
	- Adding, Editing, Deleting, and Listing Expenses	21
	- Expense Categorization	21
	- Income Management	22
	- Income Entry and Categorization by Source	22
	- Income Tracking and Summary	22
	- Budget Management	22
	- Setting and Managing Budgets	22
	- Financial Goals	22
	- Setting, Tracking, and Completing Goals	22
	- Goal Progress Visualization	22
	- Currency Converter	23
	- Real-Time Conversion and Calculator	23
	- Loan EMI Calculator	23
	- EMI Calculation and Summary	23
	- Reports and Analytics	23
	- Export Options (PDF, CSV)	23
	- Feedback	24
	- User Feedback Submission and Management	24
5	Database Design	25
	ER Diagram	25
	Description of Key Tables and Relationships	27

6	Security Configuration	32
	Overview of Spring Security Configuration	32
	Role-Based Access Control Implementation	32
7	Frontend Overview	34
	HTML and Thymeleaf Integration	34
	CSS and JavaScript Components	34
	Responsive Design Approach	35
	UI Components and Styling	35
8	External Integrations	37
	Currency Exchange API Integration for Real-Time Data	37
	PDF/Excel Export Libraries	37
9	User Interface (UI) Guide	39
	Screenshots of Each Page	39
	Navigation Flow and Menu Descriptions	45
	Key Actions Available on Each Screen	45
10	Future Enhancements	47
11	Conclusion	49
12	References	50
13	Appendix	51
	Configuration File	51
	SQL Code	51
	Code Snippets	52
14	Detailed plan of Work	55

CHAPTER 1: INTRODUCTION

1.1 Overview of Expenso

Expenso is a robust financial management system developed to simplify and enhance the process of personal finance tracking. In today's fast-paced world, managing finances can be overwhelming, and individuals often lack insight into their spending, saving, and budgeting habits. Expenso aims to bridge this gap by providing an all-in-one solution for managing income, tracking expenses, setting financial goals, and analyzing financial habits.

The system is designed to cater to various user groups, including individuals managing personal finances, families who want to share budget information, and small teams or businesses looking for collaborative financial tracking. Expenso includes interactive visual aids, goal-setting modules, and budgeting tools, making it easy for users to gain actionable insights into their financial health. With its intuitive design and comprehensive features, Expenso empowers users to take control of their finances, make informed decisions, and achieve their financial objectives.

1.2 Purpose of the Application

The purpose of Expenso is to provide users with a comprehensive financial management tool that helps simplify and optimize personal finance tracking. Expenso is designed to address the common challenges faced by individuals in managing income, tracking expenses, setting and achieving financial goals, and making data-driven financial decisions. The application aims to serve multiple purposes:

1. **Financial Tracking and Organization**: Expenso allows users to efficiently track and categorize their expenses and income. This feature helps users stay organized and maintain a clear picture of their financial inflows and outflows, ultimately enabling them to manage their budget better.
2. **Budgeting and Spending Control**: By setting budgets for different categories and monitoring spending limits, Expenso empowers users to control their expenditures and avoid overspending. Notifications and alerts ensure that users are aware of budget limits, encouraging responsible financial behaviour.

3. **Goal Setting and Motivation**: Expenso includes goal-setting capabilities that help users plan for specific financial objectives, such as saving for a major purchase or building an emergency fund. Progress tracking and rewards, such as badges for early goal completion, motivate users to stay on track and develop positive financial habits.
4. **Data-Driven Insights and Analysis**: Expenso provides visual aids like graphs, charts, and detailed reports, enabling users to analyze their spending patterns, income trends, and budget adherence. These insights support data-driven decision-making and offer users a deeper understanding of their financial health.
5. **Real-Time Currency Conversion and Financial Calculations**: With features like a currency converter and an EMI calculator, Expenso helps users perform quick financial calculations. This is especially useful for users who manage finances across different currencies or who need to calculate loan installments.
6. **Enhanced Financial Awareness**: By providing comprehensive overviews, Expenso promotes financial awareness and helps users gain control over their finances. It enables users to set realistic financial goals, achieve budget discipline, and make informed decisions to improve their financial well-being.

1.3 Project Overview

Expenso was developed to provide a streamlined and efficient solution for personal finance management. It incorporates a range of features designed to simplify financial tracking, promote savings, and offer insights into spending habits. Below is an overview of the key modules and features:

- **Expense Tracking**: Users can log daily expenses, categorize them, and view a summary of their spending. Each expense entry can be tagged with categories like "Groceries," "Utilities," "Entertainment," etc., making it easy to identify spending patterns. The system also allows users to view and edit past entries, delete unneeded records, and generate expense summaries for analysis.
- **Income Management**: Users can record their income sources (e.g., salary, freelance, investments), categorize them by source, and track income trends. This module helps users understand their primary sources of income and compare them with their expenses, providing a clear picture of their net financial position.

- **Budget Management**: Users can create budgets for specific categories and monitor their adherence to these budgets. Expenso provides notifications and alerts when spending in a particular category approaches or exceeds the set limit. This feature encourages users to stay within their financial limits and avoid overspending.
- **Financial Goals**: Expenso allows users to set financial goals (e.g., saving for a vacation or building an emergency fund) with a target amount and a deadline. The system tracks goal progress, displays completion percentages, and provides visual aids like progress bars and charts. Additionally, users receive a badge for completing goals before the deadline, rewarding them for their financial discipline and motivating them to achieve further goals.
- **Currency Converter**: The currency converter feature allows users to input amounts in various currencies and see conversions in real-time, with updated exchange rates. This is particularly useful for users who travel frequently or manage finances across multiple currencies. It integrates with financial forms in the expense, income, and budget modules for convenient conversion.
- **Bill Split**: Users can split bills with friends, family, or roommates, and track who owes what. This feature is ideal for shared expenses, such as dining out, travel, or household costs. The bill-splitting interface is designed for ease of use, showing items, amounts, and the individual shares of each participant.
- **Loan EMI Calculator**: This tool helps users calculate monthly installments (EMI) for loans based on the loan amount, interest rate, and tenure. It provides a summary of the repayment plan, making it easier for users to understand their debt obligations and plan accordingly.
- **Reports and Analytics**: Expenso generates financial reports for different periods (monthly, quarterly, and yearly), allowing users to track their spending trends and income patterns over time. The reports are presented with charts and can be exported in formats like PDF and CSV for offline analysis or sharing.
- **Feedback System**: Users can submit feedback directly within the application, enabling developers to receive suggestions, comments, and reports of issues. This feedback loop helps improve the application based on real user experiences and needs.

1.4 Key Objectives

The main objectives of Expenso are:

1. **Simplicity and User-Friendliness**: Make financial tracking accessible and intuitive for users, regardless of their financial literacy or technical expertise.
2. **Comprehensive Financial Overview**: Provide users with a holistic view of their financial health, encompassing expenses, income, budgets, and goals.
3. **Data-Driven Insights**: Enable users to make informed financial decisions based on spending trends, budget adherence, and progress toward financial goals.
4. **Motivational Elements**: Encourage good financial habits through features like goal-setting, milestone tracking.
5. **Scalability and Extensibility**: Design the system with a modular architecture that supports easy maintenance and the addition of new features as user needs evolve.
6. **Data Security and Privacy**: Ensure that all sensitive financial data is securely managed, with role-based access control and robust authentication.

CHAPTER 2 : SYSTEM OVERVIEW

2.1 Key Features :

Expenso includes a wide range of features to meet its system objectives, providing a complete suite of tools for personal financial management:

1. **User Management**:
   - Registration, login, and role-based access control (Admin, User).
2. **Dashboard**:
   - Provides a financial overview, including balance, total income, total expenses, and budget adherence.
   - Visual elements like charts for monthly spending insights, using Chart.js.
3. **Expense Management**:
   - Allows users to add, edit, delete, and list expenses.
   - Categorization of expenses for better organization and analysis.
   - Budget tracking and notifications for spending limits.
4. **Income Management**:
   - Allows users to record and categorize income sources.
   - Income tracking and summary, showing sources of income over time.
5. **Budget Management**:
   - Allows users to set monthly budgets for different categories.
   - Provides notifications for budget adherence and spending insights.
6. **Financial Goals**:
   - Users can set and track financial goals, such as saving a target amount by a deadline.
   - Progress visualization through charts or progress bars.
   - Badge system to reward users for completing goals before the deadline.
7. **Currency Converter**:
   - Real-time currency conversion and calculator for managing finances in different currencies.
   - Integrated dropdown lists of currency types for easy selection.
   - Supports INR (Indian Rupee) and other currencies, with real-time exchange rates fetched via API.
8. **Loan EMI Calculator**:
   - Allows users to calculate their loan EMIs, showing the monthly payment and total interest.

9. **Reports and Analytics**:
    - Monthly, quarterly, and yearly reports.
    - Spending and income trend visualizations through pie charts, bar charts, and line graphs.
    - Export options to PDF or CSV.
10. **Notification System**:
    - Sends alerts for important events like nearing budget limits, goal completions, and upcoming bills.
11. **Feedback**:
    - Provides a feedback system where users can submit feedback or suggestions for improvements.

2.2 Technologies Used :

The Expenso application uses a combination of frontend, backend, and external technologies to deliver a responsive, secure, and feature-rich financial management system:

1. **Backend**:
    - **Java Spring Boot**: The main framework for building the backend, including RESTful APIs, controllers, services, and security configurations.
    - **Spring Security**: For handling authentication and role-based access control.
    - **Hibernate/JPA**: For database management and ORM (Object-Relational Mapping), making it easy to work with database entities.
    - **Currency Conversion API**: Used to fetch real-time exchange rates for the currency converter.
2. **Frontend**:
    - **Thymeleaf**: Template engine integrated with Spring Boot for rendering dynamic HTML pages.
    - **HTML, CSS, JavaScript**: Core web technologies used to design user interfaces, forms, and interactive elements.
    - **Chart.js**: JavaScript library used to render charts for visualizing financial data like spending insights and income trends.
3. **Database**:
    - **MySQL** : Relational database used for storing user data, financial records, goals, and other application data.

4. **API Integrations**:
    - **Currency Exchange API**: Provides real-time exchange rates for the currency converter.
    - **PDF/CSV Export Libraries**: For exporting reports in different file formats.
5. **Version Control**:
    - **Git**: Used for tracking changes in the source code, enabling collaboration among developers, and maintaining project history.

Technology Used :

sts	4.25.0.RELEASE
spring-boot-starter-parent	3.4.0-SNAPSHOT
itextpdf	5.5.13.1
poi	5.00.00
jfreechart	1.05.05
jdk	22.0.2
mySQL	8.00.38
maven	3.9.6

CHAPTER 3 : SYSTEM ARCHITECTURE

The System Architecture chapter provides a comprehensive overview of the architectural layout of the Expenso application, explaining its components, data flow, and interaction patterns. This section is crucial for understanding how various modules and services work together to deliver the application's functionalities.

3.1 High-Level Architecture Diagram

The high-level architecture diagram illustrates the major components and layers of the Expenso application. It typically consists of three primary layers:

1. **Presentation Layer (Frontend)**:
    - Manages user interaction and presentation logic.
    - Developed using Thymeleaf templates for dynamic HTML generation, along with CSS and JavaScript for styling and interactivity.
    - Chart.js is used for visual elements like charts and graphs to display financial data insights.
    - The frontend interacts with the backend via HTTP requests, utilizing controllers that handle user inputs and direct them to the appropriate service layer.
2. **Business Logic Layer (Backend)**:
    - Implements the core functionality and business logic of the application.
    - Built using Java Spring Boot, with modules organized by feature (e.g., User Management, Expense Tracking, Income Management).
    - Service classes handle business logic, interacting with the database layer to retrieve or store data as needed.
    - Spring Security is used for authentication and role-based access control, ensuring users access only the functionalities permitted by their role.
    - Integrates with external APIs, such as the Currency Exchange API, for real-time currency conversions.
3. **Data Access Layer (Database)**:
    - Manages data storage and retrieval using Hibernate (JPA) to communicate with the database.

- Consists of MySQL , where tables are created for each entity (e.g., User, Expense, Income, Budget, Goal).
- Stores data such as user details, expenses, income records, budget information, and goal progress.
- Implements relational models and ER (Entity-Relationship) diagrams to define the relationships between entities.

This layered structure allows the Expenso application to be modular, making it easy to extend, maintain, and scale. The components interact through defined interfaces, ensuring clear separation of concerns and enhancing flexibility.

3.2 Explanation of Architecture Components

Each component in the Expenso architecture plays a specific role, ensuring smooth and efficient handling of tasks from data processing to user interaction.

1. **Frontend (Presentation Layer)**:
   - **Thymeleaf**: Used for server-side HTML generation, allowing dynamic content based on user data and real-time input.
   - **CSS and JavaScript**: CSS is used to style the application's UI, while JavaScript enables interactivity on the client side. These ensure an engaging user experience.
   - **Chart.js**: A JavaScript library that renders visualizations (e.g., bar charts, pie charts, and line graphs) for financial insights, such as spending breakdowns and income trends.
   - **HTTP Controllers**: Intermediates between the frontend and backend, receiving user requests and directing them to the appropriate service layer in the backend.
2. **Backend (Business Logic Layer)**:
   - **Java Spring Boot**: The core framework for handling business logic, application configurations, and dependency injection.
   - **Controllers**: Manage HTTP requests, route them to the appropriate service, and return responses to the frontend.
   - **Services**: Implement the main business logic of the application, such as adding expenses, managing budgets, and calculating EMI for loans. Each service handles a specific business function.
   - **Repositories**: Use JPA (Java Persistence API) and Hibernate to interact with the database, handling CRUD (Create, Read, Update, Delete) operations for different entities.

    - **Spring Security**: Manages user authentication and role-based access control, ensuring that only authorized users can access certain resources.
    - **External Integrations**:
        - **Currency Exchange API**: Fetches real-time exchange rates to support multi-currency expense tracking and conversions.
        - **Export Libraries**: Used to create and export PDF/CSV reports, allowing users to save or share their financial data.

3. **Database (Data Access Layer)**:
    - **MySQL or PostgreSQL**: The relational database used to store and manage data, including user profiles, financial records, budget limits, goals, and notifications.
    - **ER (Entity-Relationship) Diagram**: Defines the relationships between tables, such as the one-to-many relationship between User and Expense or Budget entities.
    - **Entities and Relationships**:
        - User: Stores user data, authentication credentials, and role information.
        - Expense, Income, Budget: Track individual financial transactions, budgets, and income entries, respectively.
        - Goal: Stores user-defined financial goals, tracking progress over time.
    - **Data Security and Integrity**: Data is validated and sanitized before storing it in the database, ensuring data consistency and reliability.

3.3 Data Flow Diagram

The Data Flow Diagram (DFD) demonstrates the flow of data within the Expenso application, showing how data is processed from user input to the final output. Here’s a general description of the flow of data across different modules:

1. **User Input and Authentication**:
    - The user initiates an interaction, such as logging in or accessing the dashboard, through the frontend interface.
    - The HTTP request is sent to the backend controller, where Spring Security checks for authentication and authorizes access based on the user’s role (Admin or User).
2. **Data Processing in Business Logic Layer**:

   - Once authenticated, the request is routed to the appropriate service layer based on the user’s action (e.g., adding an expense or viewing a report).
   - The service layer processes the request, applying business logic as necessary. For example, if a user is adding an expense, the expense details are validated, categorized, and then prepared for storage.
3. **Database Interaction (Data Access Layer)**:
   - The service layer interacts with the data repository, sending the validated data to be stored in the database or retrieving data for display.
   - Using JPA and Hibernate, the data is fetched, updated, or deleted from the database tables according to the action requested.
4. **External API Interaction**:
   - For certain actions, such as currency conversion, the backend interacts with an external API (e.g., Currency Exchange API) to fetch real-time data.
   - The API data is processed within the service layer before being sent back to the frontend.
5. **Data Output and Display**:
   - Once the data is processed, it is returned as a response from the backend to the frontend controller.
   - The frontend dynamically updates the view using Thymeleaf templates, displaying results such as financial insights, budget progress, or income/expense trends.
   - Visual elements, like graphs and charts generated by Chart.js, provide users with a clear and intuitive view of their financial data.

Summary :

In summary, Expenso's architecture follows a layered, modular design that separates presentation, business logic, and data access layers. This design ensures ease of maintenance, scalability, and flexibility, enabling developers to add new features or update existing ones without affecting other parts of the system. The high-level and data flow diagrams, along with the architecture component explanations, offer a complete view of how Expenso’s system functions from end to end. This setup provides users with a reliable, secure, and interactive financial management experience.

CHAPTER 4 : MODULES AND FEATURES

This chapter provides an in-depth overview of the primary modules and features of the Expenso application. Each module is designed to help users manage different aspects of their finances, from tracking expenses to setting financial goals and generating reports. This section explains the purpose, functionality, and key features of each module in detail.

4.1 Overview of Modules

Expenso consists of several interconnected modules that together provide comprehensive financial management capabilities. Each module handles a specific aspect of the user's financial data and offers tailored functionalities to simplify tracking, analysing, and managing finances. The main modules in Expenso include:

1. **User Management**: Handles user registration, login, and authentication, and supports role-based access control.
2. **Expense Management**: Allows users to track and categorize expenses.
3. **Income Management**: Supports tracking of income sources and categorizing income entries.
4. **Budget Management**: Helps users set budgets and monitor spending against those budgets.
5. **Financial Goals**: Assists users in setting and tracking personal financial goals, with visualizations for goal progress.
6. **Currency Converter**: Enables real-time currency conversions to manage expenses or income in multiple currencies.
7. **Loan EMI Calculator**: Calculates monthly loan EMIs and provides summaries of loan payments.
8. **Reports and Analytics**: Allows users to generate reports and analyse their spending and income trends, with options to export data.
9. **Feedback**: Collects user feedback to improve the system's functionality.

Each module is designed to be intuitive and user-friendly, ensuring that users can manage their financial data efficiently.

4.2 Detailed Description of Each Module :

4.2.1 User Management

The User Management module handles essential user interactions related to account access and security.

- **Registration, Login, and Authentication**:
    - Allows users to create an account with necessary details such as email and password. The login functionality verifies user credentials to grant secure access to the system.
    - Authentication is managed by Spring Security, ensuring that only authenticated users can access sensitive information.
- **Role-based Access Control**:
    - Supports role-based access for users and administrators (Admin role to be added in future updates). Each user has limited access based on their role, ensuring security and privacy.
    - The initial implementation restricts access to user-specific data, while admin-specific access will allow for broader system control in future enhancements.

4.2.2 Expense Management

This module enables users to track their expenses, organize them by category, and manage their spending efficiently.

- **Adding, Editing, Deleting, and Listing Expenses**:
    - Users can add new expenses, update existing ones, delete incorrect entries, and view a list of all expenses.
    - Expenses are displayed in a table format, and users can filter or sort them by date, category, or amount.
- **Expense Categorization**:
    - Each expense can be categorized (e.g., Food, Transport, Entertainment), making it easy for users to see where their money goes.

4.2.3 Income Management

The Income Management module is designed to help users record and categorize their income sources.

- **Income Entry and Categorization by Source**:
    - Users can record different types of income (e.g., Salary, Freelance, Investment) and categorize them based on the source.
    - Categorization helps users track income streams and understand which sources contribute most to their finances.
- **Income Tracking and Summary**:
    - Users can view a summary of all income entries, allowing them to monitor monthly or yearly income trends.
    - This feature helps users assess their earnings and compare them against expenses.

4.2.4 Budget Management

The Budget Management module allows users to set budget limits for various categories and monitor their spending.

- **Setting and Managing Budgets**:
    - Users can set budgets for different categories, such as food, travel, and entertainment.
    - The system tracks spending against these budgets, warning users when they are close to or exceed their budget limits.

4.2.5 Financial Goals

The Financial Goals module enables users to define and monitor personal financial goals, such as saving for a trip or buying a car.

- **Setting, Tracking, and Completing Goals**:
    - Users can set goals with specific target amounts and deadlines, track their progress, and update goal completion percentages.
    - This feature motivates users to save and track progress towards financial achievements.
- **Goal Progress Visualization**:
    - Progress is displayed using visual aids, such as progress bars, making it easy for users to see how close they are to achieving their goals.

4.2.6 Currency Converter

The Currency Converter module helps users manage their finances in multiple currencies, with real-time conversion capabilities.

- **Real-Time Conversion and Calculator**:
  - The module integrates with a currency exchange API to fetch real-time exchange rates.
  - Users can enter amounts in one currency and instantly see the equivalent amount in another currency, which is useful for managing expenses or income in different currencies.

4.2.7 Loan EMI Calculator

This module calculates the Equated Monthly Installment (EMI) for loans, helping users understand their loan obligations.

- **EMI Calculation and Summary**:
  - Users can enter loan details, including principal amount, interest rate, and loan tenure, to calculate their monthly EMI.
  - The system provides a summary of the loan repayment plan, including total interest and principal amounts.

4.2.8 Reports and Analytics

The Reports and Analytics module provides users with insights into their financial behaviour over specific time periods.

- **Timely Reports**:
  - Users can generate reports for different time frames to view summaries of income, expenses, and budget adherence.
  - This feature helps users analyse spending habits and make data-driven financial decisions.
- **Export Options (PDF, Excel)**:
  - The system offers export options, allowing users to download reports in PDF or Excel format.
  - Exported reports can be shared, stored, or printed for future reference.

4.2.9 Feedback

The Feedback module provides a way for users to submit feedback on their experience with the Expenso application.

- **User Feedback Submission and Management**:
  - Users can submit feedback regarding system usability, features, and any encountered issues.
  - Feedback is recorded and accessible to the development team for future improvements, ensuring that user suggestions and concerns are addressed.

Summary :

The Modules and Features chapter provides a breakdown of Expenso’s primary functionalities. Each module is designed to address specific financial management needs, from tracking expenses and setting budgets to achieving financial goals and analysing income trends. By implementing these modules, Expenso aims to provide users with a comprehensive and user-friendly financial management solution. This modular structure allows for flexibility, enabling future updates and additions that will further enhance the user experience.

CHAPTER 5 : DATABASE DESIGN

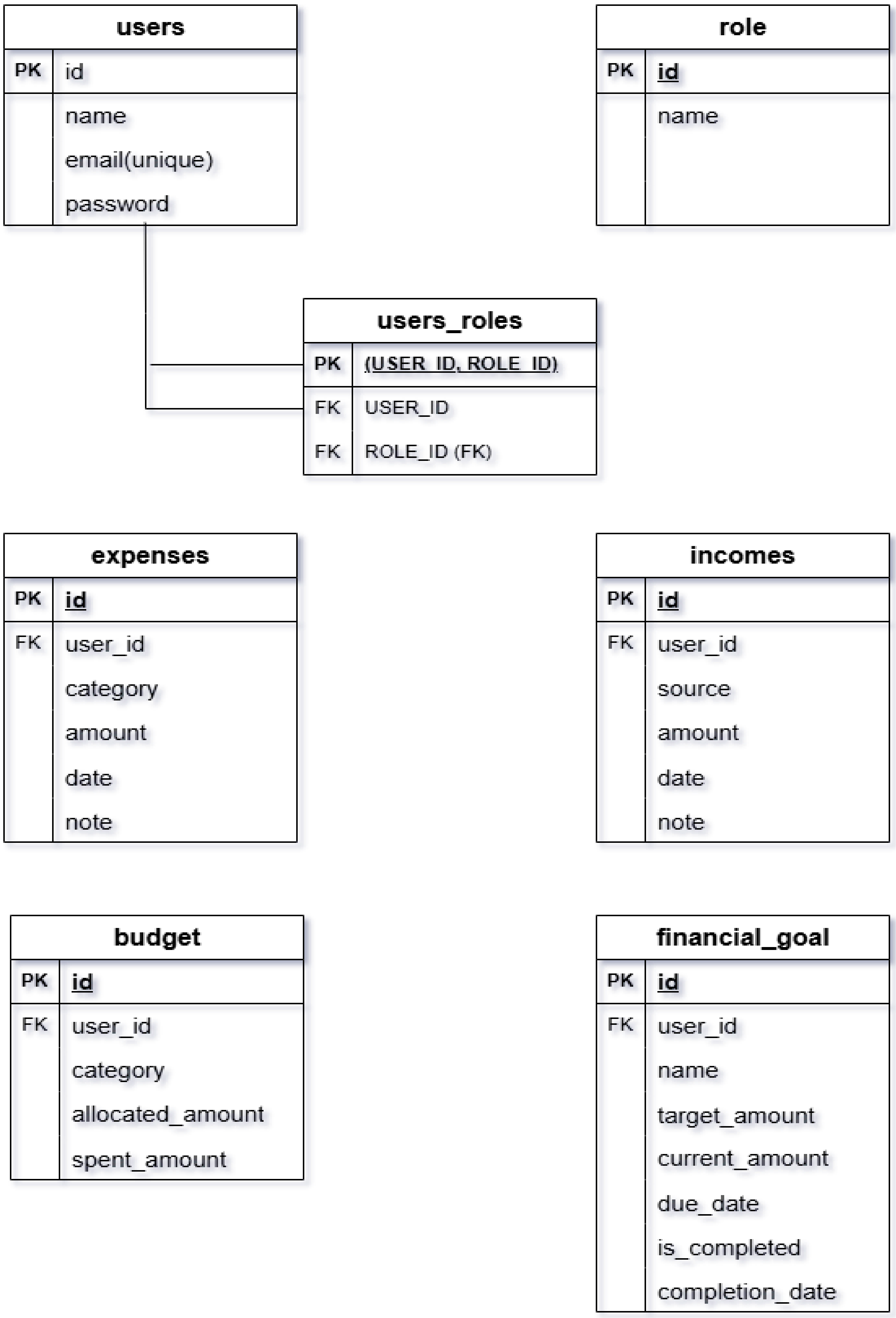
The database design chapter outlines the structure of the Expenso application's data storage, detailing how various data entities are organized, related, and stored. This chapter includes the Entity-Relationship (ER) diagram and descriptions of key tables and relationships within the database.

5.1 ER Diagram :

The ER Diagram provides a visual representation of the relationships between different entities in the Expenso application. It highlights the key entities, such as Users, Roles, Expenses, Incomes, Budgets, and Financial Goals, and their associations, enabling a clear understanding of how data is structured.

Key Entities and Relationships:

1. **User**: Represents each user in the system, storing essential information like name, email, and password. It has a one-to-many relationship with Expenses, Incomes, Budgets, and Financial Goals.
2. **Role**: Defines the different access roles within the system. Users are associated with roles through a many-to-many relationship, stored in the users_roles table.
3. **Expense**: Represents individual expense records for each user. Each expense entry is linked to a specific user through a foreign key relationship.
4. **Income**: Captures income details for each user, categorized by source. Each income entry is associated with a specific user.
5. **Budget**: Holds budget information for each category of spending, helping users track allocated and spent amounts. Each budget is linked to a user.
6. **Financial Goal**: Stores information about each user's financial goals, including target amounts, current savings, due dates, and completion status.


users
PK id
name
email(unique)
password
role
PK id
name
users_roles
PK (USER_ID, ROLE_ID)
FK USER_ID
FK ROLE_ID (FK)
expenses
PK id
FK user_id
category
amount
date
note
incomes
PK id
FK user_id
source
amount
date
note
budget
PK id
FK user_id
category
allocated_amount
spent_amount
financial_goal
PK id
FK user_id
name
target_amount
current_amount
due_date
is_completed
completion_date

5.2 Description of Key Tables and Relationships

Below are detailed descriptions of the primary tables in the Expenso database, along with their structure and relationships.

5.2.1 Users Table

The users table stores essential information for each user, such as name, email, and password.

```
sql
Copy code
CREATE TABLE users (
   id BIGINT NOT NULL AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL,
   email VARCHAR(255) NOT NULL UNIQUE,
   password VARCHAR(255) NOT NULL,
   PRIMARY KEY (id)
);
```

- **id**: Unique identifier for each user.
- **name**: Stores the user's full name.
- **email**: Stores the user's email address, which must be unique.
- **password**: Stores the user's password, securely hashed.

This table serves as the primary reference for other tables such as expenses, incomes, budget, and financial_goal, linking user data across different modules.

5.2.2 Roles Table

The roles table defines different roles within the system, such as User and Admin.

```
sql
Copy code
CREATE TABLE roles (
   id BIGINT NOT NULL AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL,
   PRIMARY KEY (id)
);
```

- **id**: Unique identifier for each role.
- **name**: Stores the name of the role (e.g., "User", "Admin").

Roles are assigned to users through a many-to-many relationship, implemented in the users_roles table.

5.2.3 Users_Roles Table

The users_roles table manages the many-to-many relationship between users and roles.

sql
Copy code

```
CREATE TABLE users_roles (
   USER_ID BIGINT NOT NULL,
   ROLE_ID BIGINT NOT NULL,
   PRIMARY KEY (USER_ID, ROLE_ID),
   FOREIGN KEY (USER_ID) REFERENCES users(id) ON DELETE CASCADE,
   FOREIGN KEY (ROLE_ID) REFERENCES roles(id) ON DELETE CASCADE
);
```

- **USER_ID**: Foreign key referencing the user's ID in the users table.
- **ROLE_ID**: Foreign key referencing the role's ID in the roles table.

Each entry in this table assigns a specific role to a user. The cascade delete ensures that if a user is deleted, all associated roles are removed.

5.2.4 Expenses Table

The expenses table stores individual expense records for users, capturing details about each expense.

sql
Copy code

```
CREATE TABLE expenses (
   id BIGINT AUTO_INCREMENT PRIMARY KEY,
   category VARCHAR(255) NOT NULL,
   amount DOUBLE NOT NULL,
   date DATE NOT NULL,
   note VARCHAR(255),
   user_id BIGINT NOT NULL,
   CONSTRAINT fk_user FOREIGN KEY (user_id) REFERENCES users(id)
);
```

- **id**: Unique identifier for each expense entry.
- **category**: Stores the category of the expense (e.g., "Food", "Transport").
- **amount**: Stores the amount spent.
- **date**: Stores the date of the expense.
- **note**: Optional field for additional notes.
- **user_id**: Foreign key referencing the user associated with the expense.

The user_id field links each expense to a user, ensuring that expenses are tracked per individual.

5.2.5 Incomes Table

The incomes table tracks income records for users, allowing them to manage different income sources.

sql
Copy code
```
CREATE TABLE incomes (
   id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   source VARCHAR(255) NOT NULL,
   amount DOUBLE NOT NULL,
   date DATE NOT NULL,
   note VARCHAR(255),
   user_id BIGINT NOT NULL,
   CONSTRAINT fk_user_income FOREIGN KEY (user_id) REFERENCES users(id)
ON DELETE CASCADE
);
```

- **id**: Unique identifier for each income entry.
- **source**: Specifies the income source (e.g., "Salary", "Freelance").
- **amount**: Stores the amount received.
- **date**: Records the date of the income.

- **note**: Optional field for additional notes.
- **user_id**: Foreign key referencing the user associated with the income.

This table enables users to categorize income sources and view income trends.

5.2.6 Budget Table

The budget table stores budget information for various expense categories, allowing users to manage allocated and spent amounts.

sql
Copy code
```
CREATE TABLE budget (
   id BIGINT PRIMARY KEY AUTO_INCREMENT,
   category VARCHAR(255) NOT NULL,
   allocated_amount DECIMAL(15, 2) NOT NULL,
   spent_amount DECIMAL(15, 2) NOT NULL,
   user_id BIGINT,
   CONSTRAINT fk_users FOREIGN KEY (user_id) REFERENCES users(id) ON
DELETE CASCADE
);
```

- **id**: Unique identifier for each budget entry.
- **category**: Category for which the budget is set (e.g., "Groceries", "Entertainment").
- **allocated_amount**: The budgeted amount allocated to this category.
- **spent_amount**: The amount spent within this category so far.
- **user_id**: Foreign key linking the budget to a user.

This table helps users monitor their spending relative to predefined budget limits.

5.2.7 Financial Goal Table

The financial_goal table records individual financial goals for each user, allowing them to track progress towards each goal.

sql
Copy code

```
CREATE TABLE financial_goal (
   id BIGINT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(255) NOT NULL,
   target_amount DECIMAL(19, 2) NOT NULL,
   current_amount DECIMAL(19, 2) DEFAULT 0.00 NOT NULL,
   due_date DATE NOT NULL,
   is_completed BOOLEAN DEFAULT FALSE,
   completion_date DATE,
   user_id BIGINT,
   CONSTRAINT fk_user_goal FOREIGN KEY (user_id) REFERENCES users(id)
);
```

- **id**: Unique identifier for each financial goal.
- **name**: The name of the goal (e.g., "Vacation Fund", "New Car").
- **target_amount**: The amount needed to achieve the goal.
- **current_amount**: Tracks the current savings towards the goal.
- **due_date**: The target date for achieving the goal.
- **is_completed**: Indicates whether the goal is completed.
- **completion_date**: Records the date of goal completion, if applicable.
- **user_id**: Foreign key linking the goal to a user.

This table supports user-driven goal tracking, allowing users to save and visualize progress toward specific financial targets.

Summary :

The Expenso database design is built around a relational structure that effectively supports financial management functionalities. Each key table serves a specific purpose, with carefully designed relationships between entities, ensuring data integrity and efficient access. Through primary keys, foreign keys, and cascade deletion options, Expenso maintains a clean and secure database structure that supports the system’s core features, including expense tracking, income management, budgeting, and financial goal setting.

CHAPTER 6 : SECURITY CONFIGURATION

6.1 Overview of Spring Security Configuration

The security configuration in Expenso is implemented using **Spring Security**, a powerful and customizable authentication and access-control framework widely used in Java applications. The primary goal of this configuration is to ensure that only authorized users have access to specific parts of the application and that sensitive financial data remains protected.

Key aspects of Spring Security implemented in Expenso include:

- **User Authentication**: Ensures that only registered users with valid credentials can access the system.
- **Authorization and Access Control**: Enforces role-based access, restricting certain features to specific user roles.
- **Session Management**: Manages user sessions and prevents unauthorized access once a session is terminated.
- **Password Encryption**: Uses BCrypt password hashing to securely store user passwords in the database.

The security configuration is defined in a SecurityConfig class, which extends WebSecurityConfigurerAdapter and customizes settings for authentication, authorization, and security policies for the entire application.

6.2 Role-Based Access Control Implementation

Expenso uses **Role-Based Access Control (RBAC)** to grant or restrict access to different parts of the system based on user roles. Currently, Expenso includes two primary roles: USER and ADMIN (with ADMIN functionality planned for future development). RBAC allows for easy expansion, enabling additional roles or permissions as the system grows.

Implementation Details

1. **User Roles and Authorities**:
    - The application has a roles table in the database, where each role (e.g., USER, ADMIN) is defined.
    - Users and roles are connected through a users_roles join table, which enables each user to have one or more roles.
    - The role assigned to each user determines what resources or features they can access.

2. **Access Control in Security Configuration**:
   - The SecurityConfig class defines specific access permissions for each URL path, using the authorizeRequests method to allow or restrict access based on roles.
   - For example, paths related to expense management, budget, and financial goals are restricted to authenticated users, while administrative paths (once developed) will be restricted to users with the ADMIN role.
   - Access control annotations (such as @PreAuthorize or @Secured) are used within controller methods to further specify role-based access at a granular level.
3. **Custom Authentication Handling**:
   - A UserDetailsService is configured to load user information from the database, verifying credentials at login and assigning roles based on the user's profile.
   - Upon successful authentication, users are redirected to the appropriate dashboard based on their role, providing a customized experience.
4. **Password Security and Session Management**:
   - Passwords are encrypted using BCrypt to enhance security.
   - Session management settings are configured to prevent session fixation and unauthorized access, enhancing the overall security of user sessions within the application.
5. **Error Handling and Feedback**:
   - Expenso's security configuration includes custom handling of unauthorized access attempts, providing feedback through custom error pages or messages to help users understand access restrictions and enhance usability.

Summary :

The security configuration in Expenso ensures that user data is protected, with access restrictions in place based on user roles. It's a scalable setup that enables new roles and permissions to be added as the system grows, ensuring flexibility and robust security for financial data management. This security layer is essential to maintain user trust and data integrity across all modules in Expenso.

CHAPTER 7 : FRONTEND OVERVIEW

The frontend of Expenso is designed with an emphasis on usability, responsiveness, and visual appeal. Built with **HTML**, **Thymeleaf**, **CSS**, and **JavaScript**, the frontend offers an intuitive experience for users managing finances. The use of **Thymeleaf** allows seamless integration with the backend, dynamically rendering data directly within HTML templates. **CSS** and **JavaScript** are used extensively for styling, layout control, and interactive components, contributing to a visually engaging, responsive interface.

7.1 HTML and Thymeleaf Integration :

Thymeleaf is integrated into each HTML page to allow real-time data population, making the frontend interactive and data-driven. The pages utilize Thymeleaf's templating capabilities, enabling:

- **Dynamic Data Binding**: Values from backend entities (such as user data, budgets, expenses, etc.) populate specific fields and tables within the templates.
- **Conditional Rendering**: Controls which components or messages display based on user roles or status (e.g., logged-in status, specific roles, error messages).
- **Reusable Components**: Elements like headers, footers, and sidebars are organized as fragments, promoting a consistent layout across all pages and enabling easy updates.
- **Form Handling**: Forms across the application, from login to expense entry, are managed with Thymeleaf for data binding and error handling.

7.2 CSS and JavaScript Components :

CSS and **JavaScript** handle the styling, layout, and interactivity in Expenso's UI. CSS controls the overall look and feel, while JavaScript enhances user interactions, such as:

- **Animations**: Subtle animations improve user experience on forms, buttons, and notifications, providing feedback for user actions.
- **Form Validation**: JavaScript performs client-side validation, ensuring fields are correctly filled out before submission.
- **Charts and Graphs**: JavaScript libraries, such as **Chart.js**, visualize data on various financial reports and dashboards, helping users understand their finances.

- **Responsive Navigation**: JavaScript and CSS create a sidebar menu that adapts between desktop and mobile views, allowing users to easily navigate the application on any device.

7.3 Responsive Design Approach :

The frontend is designed to be fully responsive, offering an optimized experience across various screen sizes. **CSS media queries** adjust layouts, font sizes, and components based on screen width, ensuring:

- **Mobile Accessibility**: Key actions are reachable, with collapsible menus and icons to enhance ease of navigation on small screens.
- **Adaptive Layouts**: Pages like the dashboard and reports restructure their layouts for readability on tablets and mobile devices.

7.4 UI Components and Styling :

Expenso's UI components are crafted with simplicity, clarity, and function in mind. Each page contains specific components tailored to its functionality, providing users with a straightforward workflow.

Individual Page Descriptions :

1. **about-expenso.html**: This page provides an overview of Expenso's purpose, features, and benefits, giving users insight into the application's value.
2. **add-expense.html**: A form page where users can add new expense entries. Includes fields for category, amount, date, and an optional note. Uses real-time validation and links to relevant budget categories.
3. **budget.html**: Displays a summary of the user's budget categories, including allocated and spent amounts. Users can adjust budgets here, with visual indicators showing budget adherence.
4. **currency-converter.html**: Allows users to convert between currencies in real-time. Integrates with a currency API, providing up-to-date conversion rates and a calculator for ease of use.
5. **error-page.html**: A simple error message page that notifies users of access or server issues. This page is displayed when an error is encountered within the application.

6. **expenses.html**: Displays a list of all recorded expenses, with options to filter, edit, and delete entries. Includes a breakdown by category and links to summary charts.
7. **feedback.html**: Provides a form for users to submit feedback or report issues with the application. Feedback is stored for admin review and future improvements.
8. **future-investment.html**: A page for users to explore future investment options and projected returns. Users can experiment with hypothetical investments to see possible growth.
9. **goals.html**: Allows users to set and track financial goals. Includes visual progress indicators and options to edit, complete, or delete goals, rewarding badges upon goal completion.
10. **income.html**: Displays income entries, categorized by source (salary, freelance, etc.). Users can add, edit, or delete income records, helping them track earnings effectively.
11. **index.html**: The homepage, introducing users to Expenso with an overview of key features and links to login or register.
12. **loan-calculator.html**: Allows users to calculate loan EMI (Equated Monthly Installment) based on loan amount, interest rate, and tenure. Displays a detailed summary of repayment schedules.
13. **login.html**: User login page, requiring credentials to access the dashboard. Includes error messages for failed login attempts and links to registration for new users.
14. **register.html**: User registration page with fields for name, email, password, and role selection. Validates data before submission to ensure accurate account creation.
15. **reports.html**: Generates visual and numerical reports on user expenses, income, and budget adherence. Options are available to export reports in formats like PDF and Excel.
16. **user-dashboard.html**: The main user dashboard, summarizing financial status, budget adherence, and recent expenses. Interactive charts display income and expense trends.
17. **users.html**: For future use with the Admin role. This page will list all users with options to edit roles, deactivate accounts, and manage user permissions.

The frontend of Expenso has been built to provide users with a clean, accessible, and intuitive experience. With a focus on interactivity and responsiveness, each page and component is crafted to streamline financial management tasks, making Expenso a practical and user-friendly application.

CHAPTER 8 : EXTERNAL INTEGRATIONS

Expenso integrates with several external libraries and APIs to provide real-time functionality and enhanced usability. Key integrations include a **Currency Exchange API** for real-time currency conversion rates and libraries for **PDF and Excel export**, enabling users to export reports and data effortlessly.

8.1 Currency Exchange API Integration for Real-Time Data

The Currency Converter feature in Expenso is powered by an external API that provides up-to-date exchange rates for multiple currencies. This integration allows users to:

- **Convert Between Currencies**: Users can input an amount in one currency and instantly see its equivalent in another currency.
- **Real-Time Updates**: By pulling real-time data from the API, Expenso ensures that users always have access to the most current exchange rates.
- **Currency Calculator**: The currency calculator page uses the API to compute conversions seamlessly, enabling users to use accurate data in their financial planning.

Implementation Details:

- **API Key Security**: The API key for accessing currency data is securely stored in the application's configuration, preventing unauthorized access.
- **Error Handling**: If the API is unavailable, Expenso displays an error message and uses fallback data or alerts the user to try again later.
- **Update Frequency**: The system can be configured to update exchange rates at specific intervals or on-demand, ensuring balance between performance and data freshness.

8.2 PDF/Excel Export Libraries

To provide flexible data export options, Expenso incorporates libraries for generating PDF and Excel files. This feature is particularly useful in the **Reports** module, where users can export detailed records of their income, expenses, budgets, and financial goals.

- **PDF Export**: Allows users to create a snapshot of financial reports with charts, tables, and summaries. PDFs are commonly used for easy sharing and printing, enabling users to keep offline records of their finances.
- **Excel Export**: For users who want to further analyze or manipulate their data, the Excel export feature provides a format that's compatible with spreadsheet software. Excel files retain all numeric data and category details, allowing users to sort, filter, and perform custom calculations.

Implementation Details:

- **Library Choice**: Common libraries like **Apache POI** (for Excel) and **iText** or **PDFBox** (for PDFs) are used due to their robustness and compatibility with Java applications.
- **Dynamic Content**: Data for the PDF or Excel files is dynamically generated based on user selections (e.g., date ranges, categories).
- **Customization**: Users can select specific data points to include, ensuring the exported file matches their needs.
- **Error Handling**: If an error occurs during export, the system provides clear feedback, and users are prompted to try again.

Together, these integrations enhance the functionality and flexibility of Expenso, making it a reliable and versatile tool for users who require real-time data and easy access to detailed financial records.

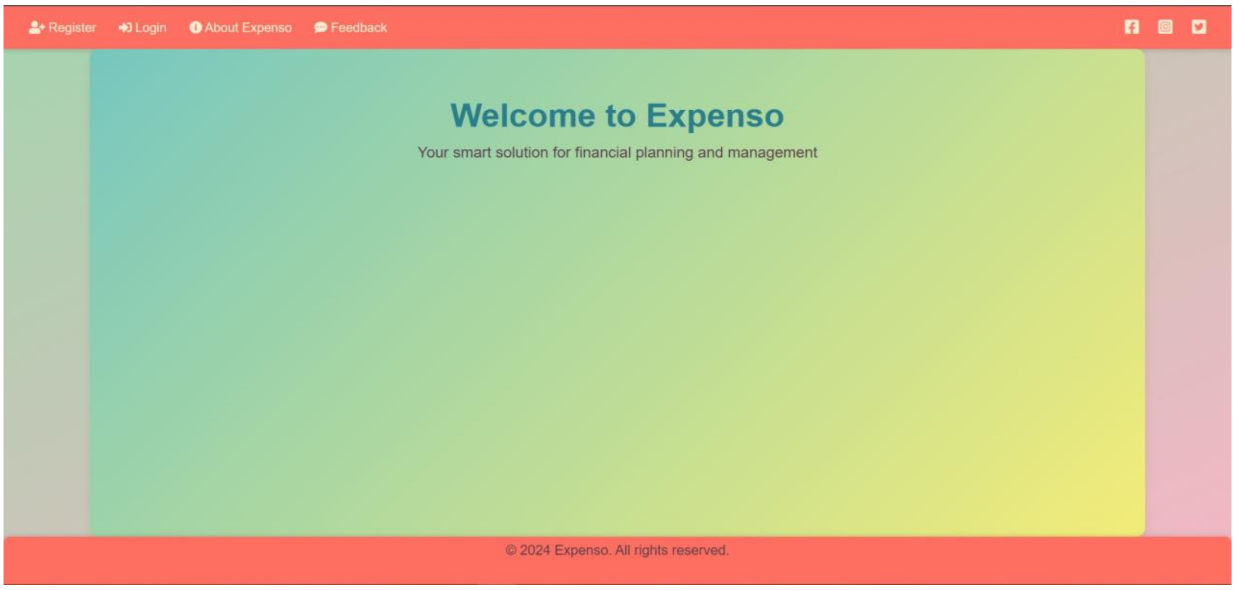
CHAPTER 9 : USER INTERFACE (UI) GUIDE

This chapter provides a comprehensive guide to Expenso's user interface, covering page screenshots, navigation flow, menu descriptions, and key actions. The goal of this guide is to help users understand and navigate each screen efficiently, enabling them to maximize Expenso's features and functionality.

9.1 Screenshots of Each Page :

Each screen in Expenso is designed with a user-friendly layout and functionality to support efficient financial management. Here are key pages with brief descriptions:

1. **Home/Index Page**: The landing page, providing an overview of Expenso's features and quick access to the login and register pages.

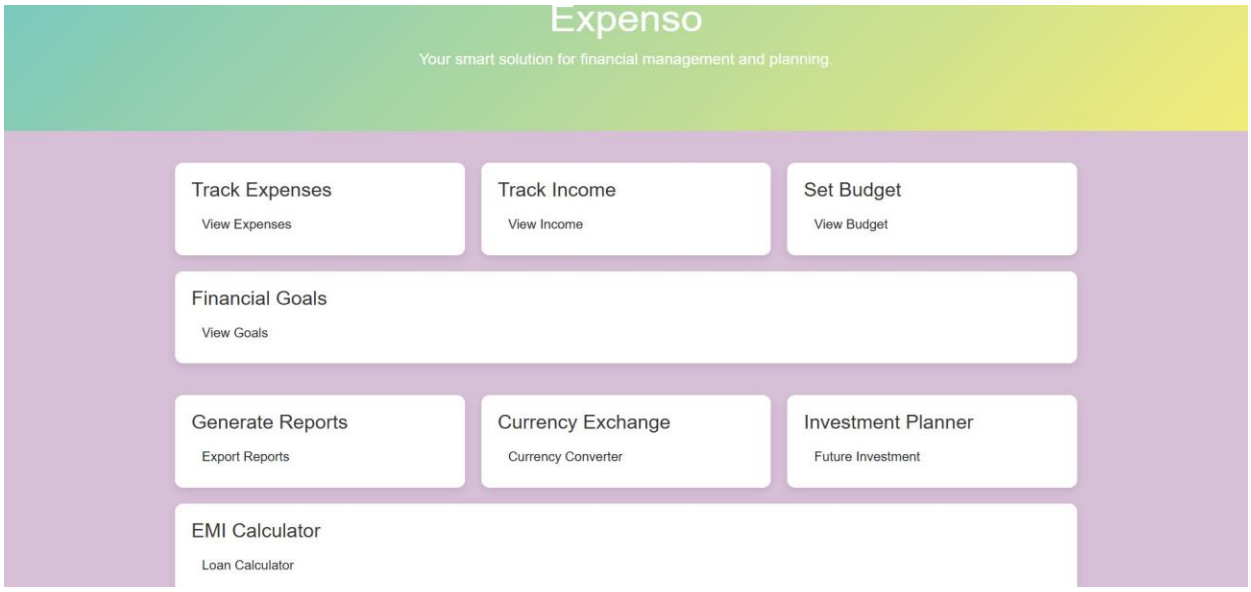


Welcome to Expenso

Your smart solution for financial planning and management

© 2024 Expenso. All rights reserved.

2. **User Dashboard**: A personalized page showing financial insights, such as recent expenses, income, and budget summaries. It includes graphs and charts for a quick visual overview.


Expenso

Your smart solution for financial management and planning.

Track Expenses
View Expenses

Track Income
View Income

Set Budget
View Budget

Financial Goals
View Goals

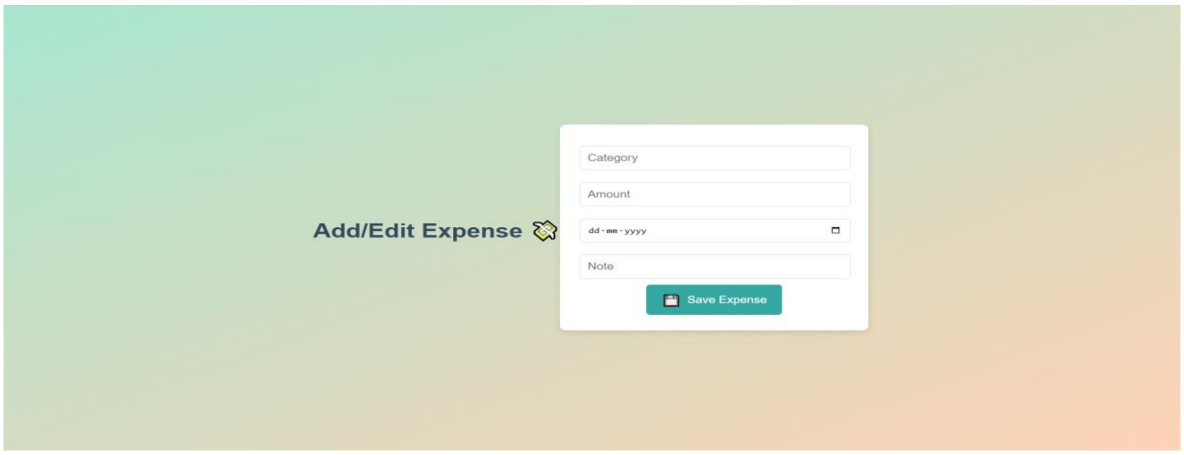
Generate Reports
Export Reports

Currency Exchange
Currency Converter

Investment Planner
Future Investment

EMI Calculator
Loan Calculator

3. **Add Expense/Income**: Forms for entering new expenses or income, categorized by type, with options for adding notes and setting dates.


Add/Edit Expense

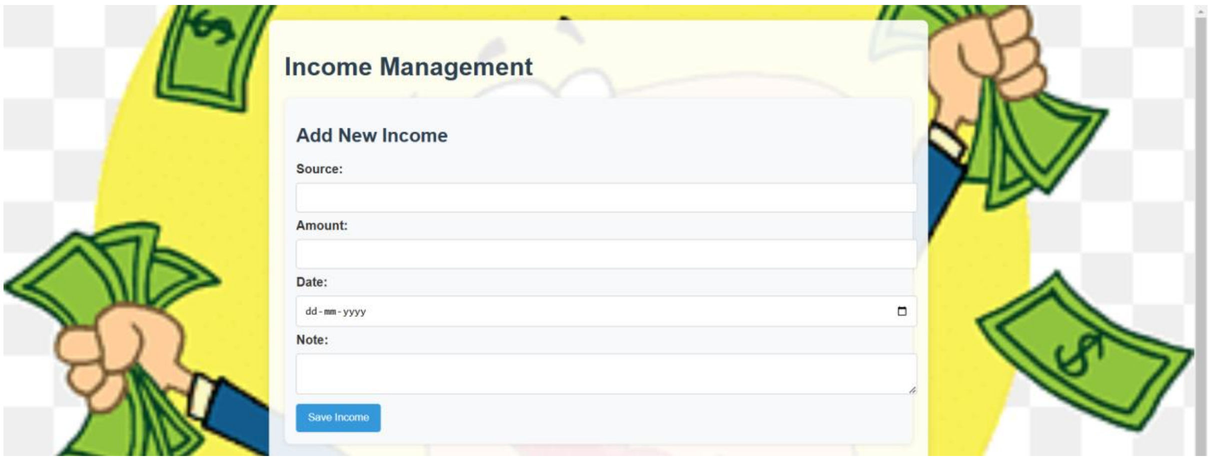
Category

Amount

dd-mm-yyyy

Note

Save Expense


Income Management

Add New Income

Source:

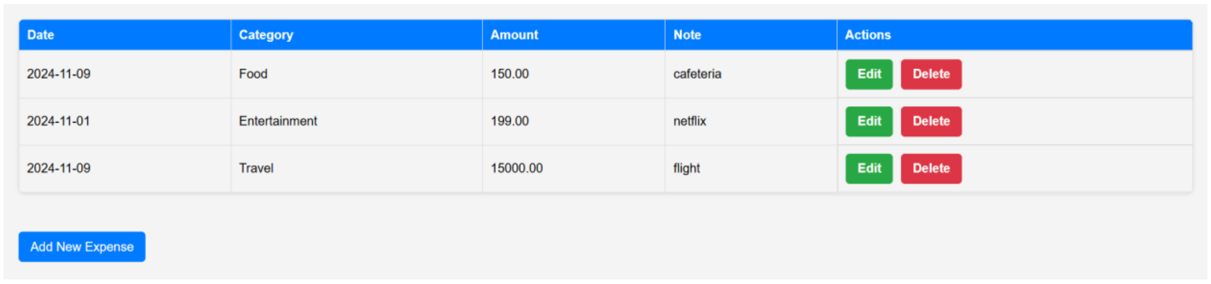
Amount:

Date:
dd-mm-yyyy

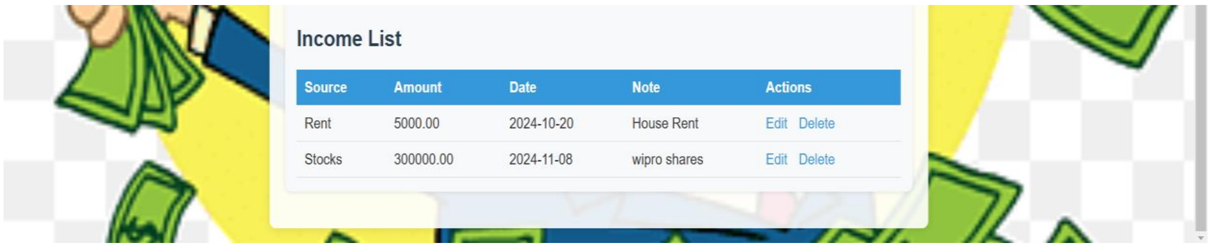
Note:

Save Income

4. **Expense/Income Listing**: Lists all recorded expenses or income, sortable by date or category. Users can edit or delete entries from this page.



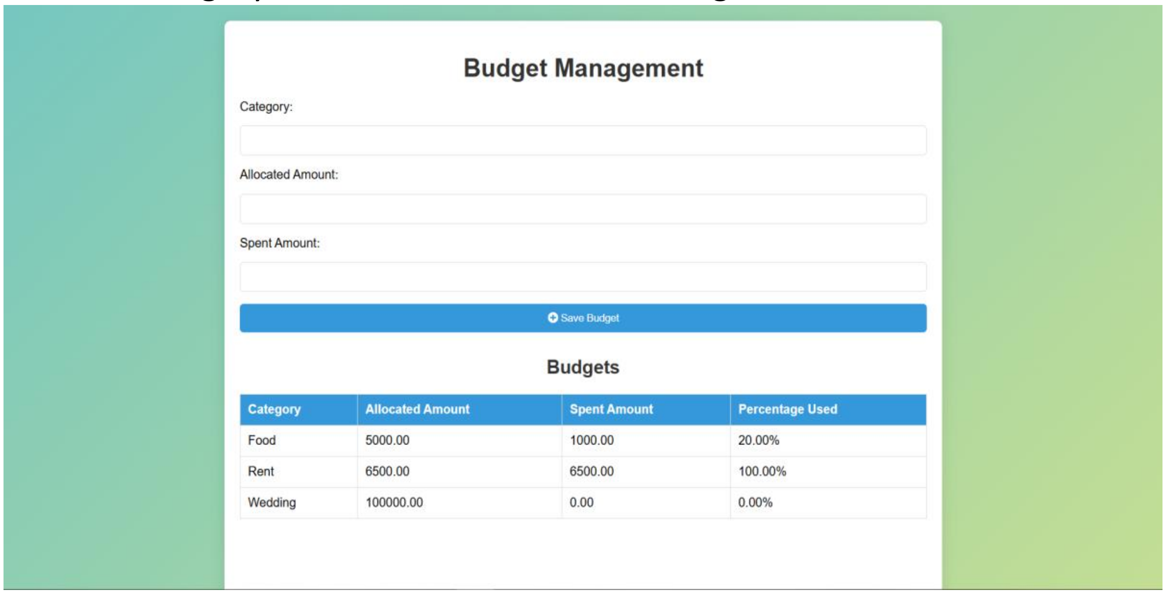
Date	Category	Amount	Note	Actions
2024-11-09	Food	150.00	cafeteria	Edit Delete
2024-11-01	Entertainment	199.00	netflix	Edit Delete
2024-11-09	Travel	15000.00	flight	Edit Delete



Income List

Source	Amount	Date	Note	Actions
Rent	5000.00	2024-10-20	House Rent	Edit Delete
Stocks	300000.00	2024-11-08	wipro shares	Edit Delete

5. **Budget Page**: A budget overview, showing allocated and spent amounts for each category, with visual indicators of budget adherence.



Budget Management

Category:

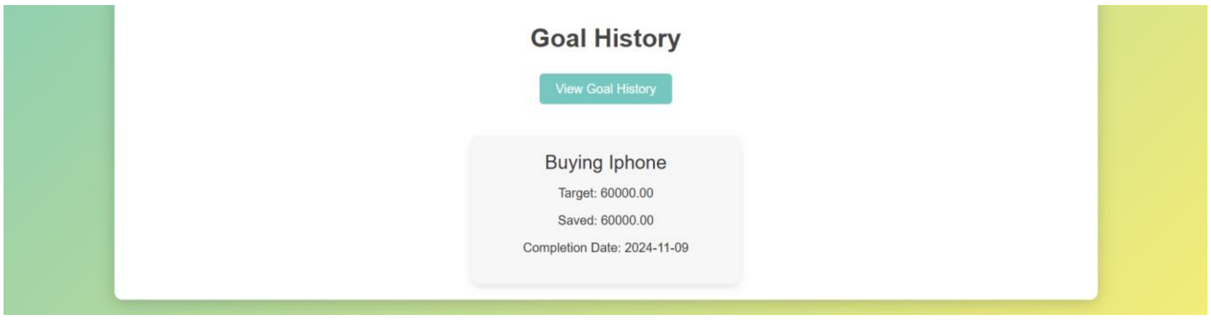
Allocated Amount:

Spent Amount:

Budgets

Category	Allocated Amount	Spent Amount	Percentage Used
Food	5000.00	1000.00	20.00%
Rent	6500.00	6500.00	100.00%
Wedding	100000.00	0.00	0.00%

6. **Financial Goals**: A page where users can set, track, and manage their financial goals, with a progress bar and completion badges.



Goal History

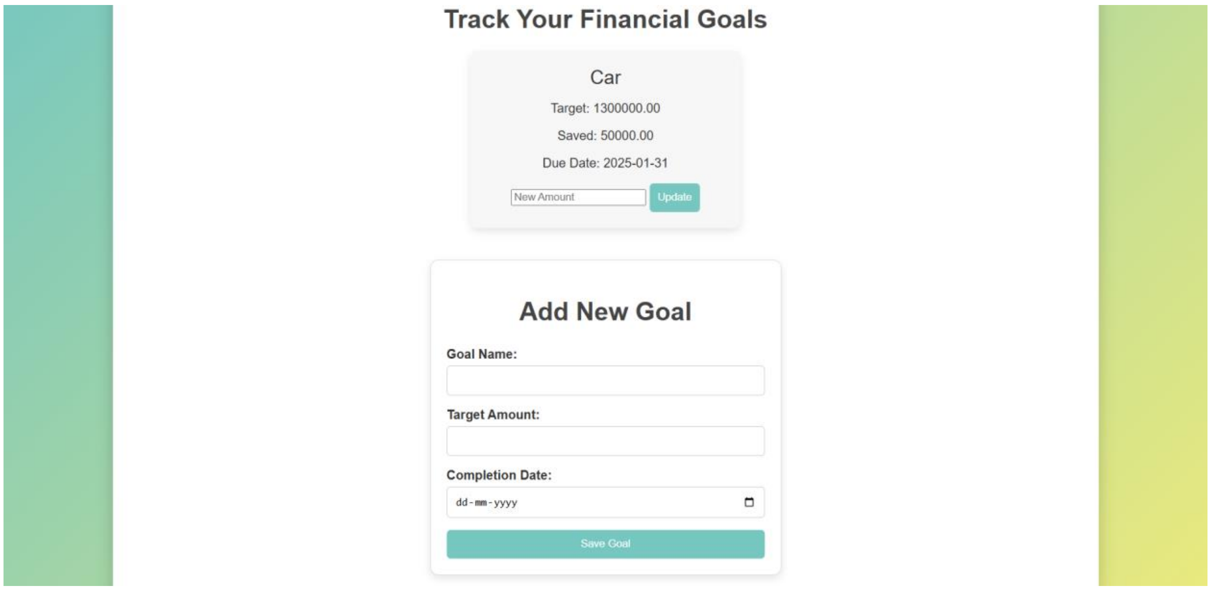
View Goal History

Buying Iphone

Target: 60000.00

Saved: 60000.00

Completion Date: 2024-11-09



Track Your Financial Goals

Car

Target: 1300000.00

Saved: 50000.00

Due Date: 2025-01-31

New Amount | Update

Add New Goal

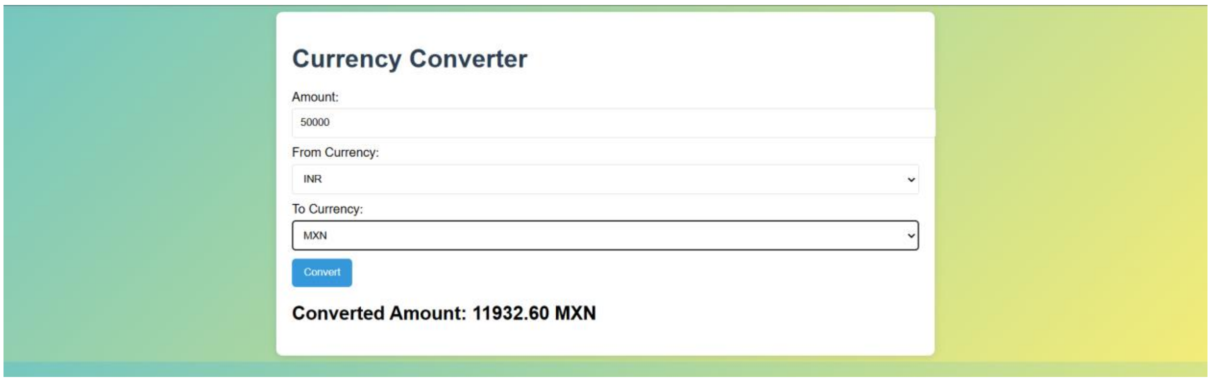
Goal Name:

Target Amount:

Completion Date: dd-mm-yyyy

Save Goal

7. **Currency Converter**: Provides real-time currency conversion and calculation, integrating with financial forms.



Currency Converter

Amount: 50000

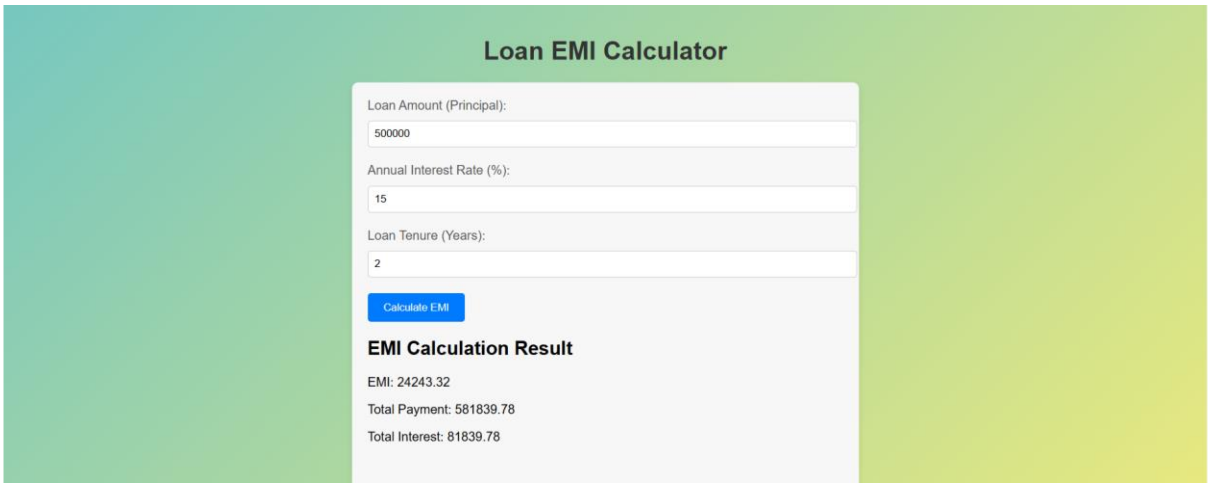
From Currency: INR

To Currency: MXN

Convert

Converted Amount: 11932.60 MXN

8. **Loan EMI Calculator**: Users can calculate monthly loan payments, with detailed summaries based on principal, interest rate, and loan tenure.



Loan EMI Calculator

Loan Amount (Principal): 500000

Annual Interest Rate (%): 15

Loan Tenure (Years): 2

Calculate EMI

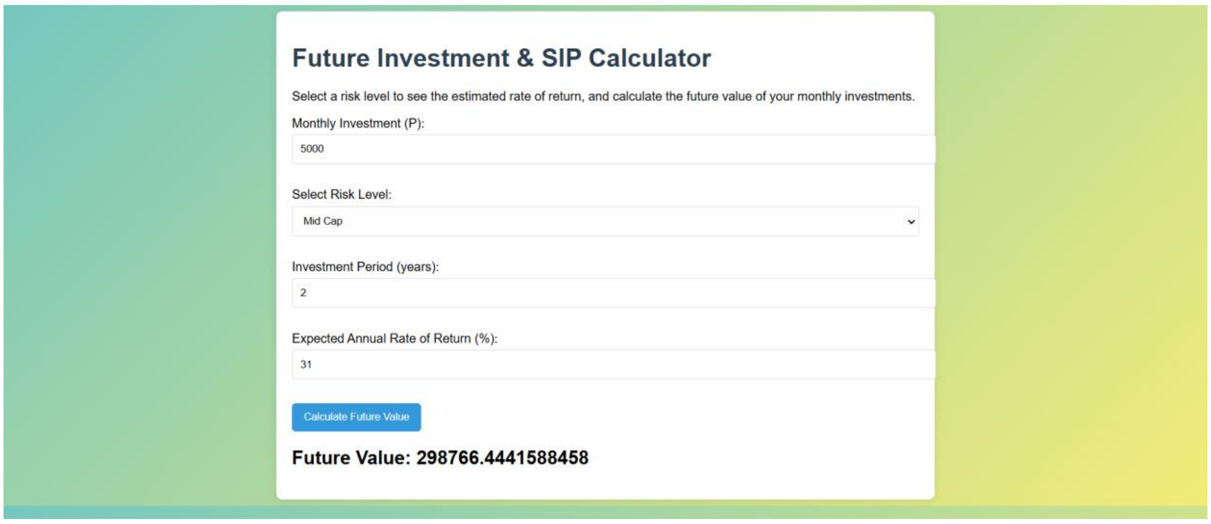
EMI Calculation Result

EMI: 24243.32

Total Payment: 581839.78

Total Interest: 81839.78

9. **SIP Calculator** :



Future Investment & SIP Calculator

Select a risk level to see the estimated rate of return, and calculate the future value of your monthly investments.

Monthly Investment (P): 5000

Select Risk Level: Mid Cap

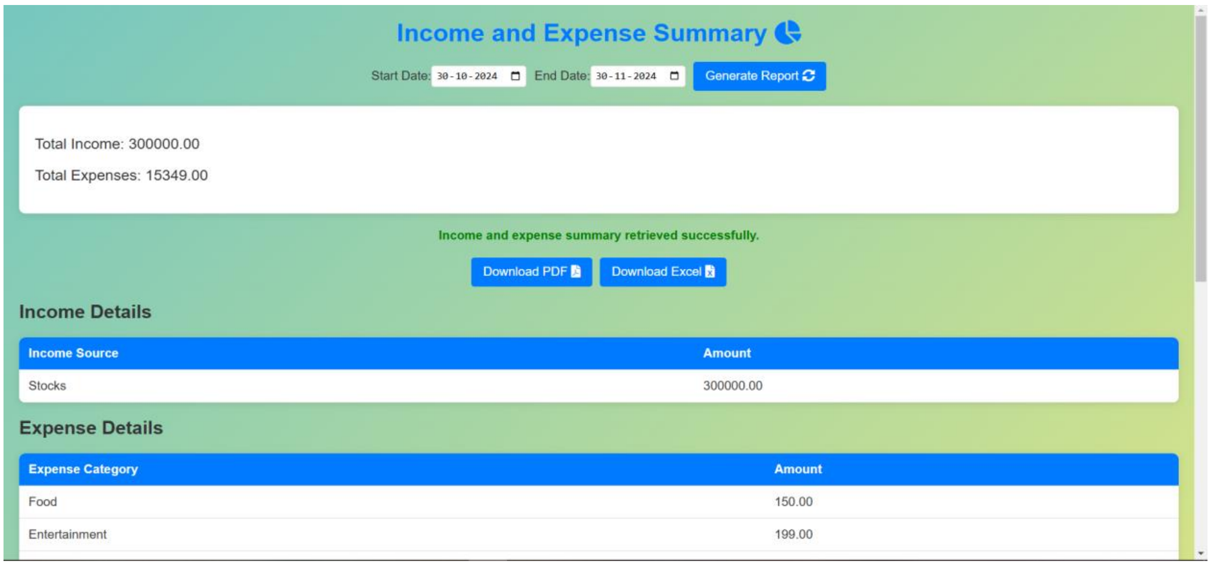
Investment Period (years): 2

Expected Annual Rate of Return (%): 31

Calculate Future Value

Future Value: 298766.4441588458

10. **Reports**: Offers options to generate and export financial reports in PDF or Excel format, based on specified date ranges and categories.


Income and Expense Summary

Start Date: 30-10-2024 End Date: 30-11-2024 Generate Report

Total Income: 300000.00

Total Expenses: 15349.00

Income and expense summary retrieved successfully.

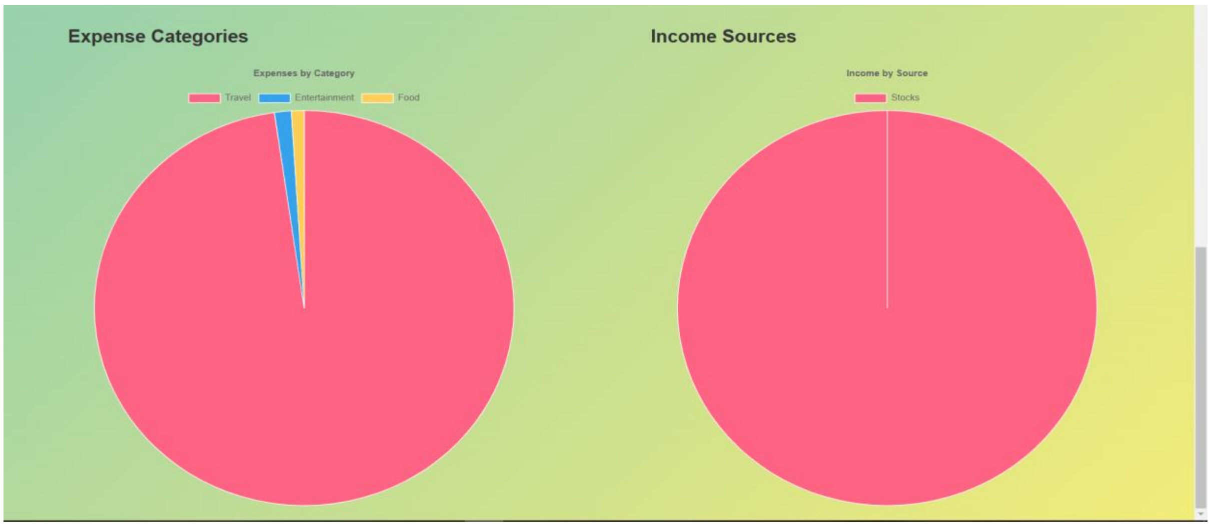
Download PDF Download Excel

Income Details

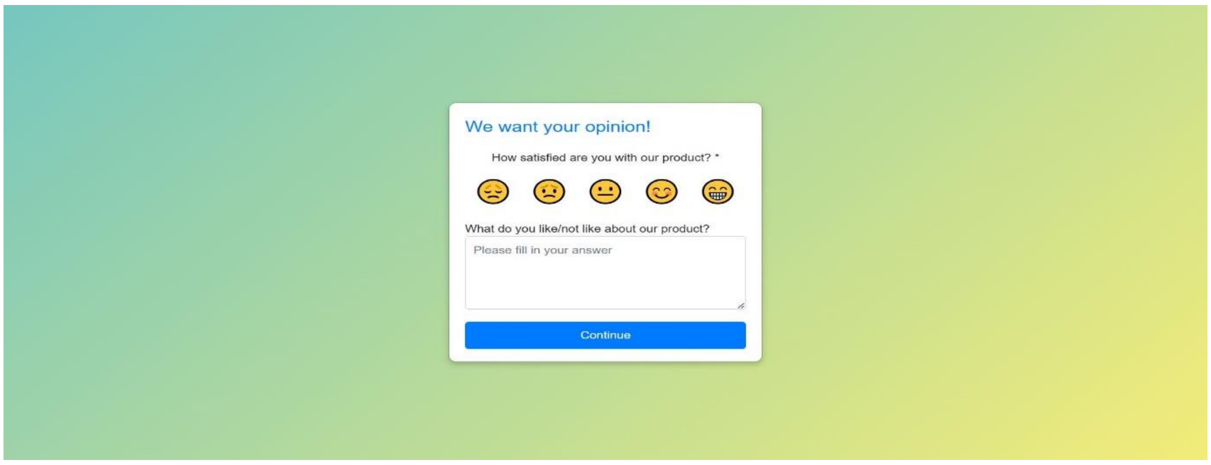
Income Source	Amount
Stocks	300000.00

Expense Details

Expense Category	Amount
Food	150.00
Entertainment	199.00



11. **Feedback Page**: A page where users can submit feedback or feature suggestions.


We want your opinion!
How satisfied are you with our product? *
What do you like/not like about our product?
Please fill in your answer
Continue

9.2 Navigation Flow and Menu Descriptions

Expenso's navigation flow is designed for ease of use, with a sidebar or top menu that offers quick access to all main sections. Each page in the menu serves a specific purpose, as outlined below:

- **Home**: Introduction to Expenso, with options for logging in or signing up.
- **Dashboard**: Central hub displaying key financial insights and shortcuts to common actions like adding expenses or managing budgets.
- **Expenses/Income**: Access pages for managing expenses and income, with options to view, add, edit, and delete entries.
- **Budget**: Navigate to budget management, where users can view, set, or adjust their budgets by category.
- **Goals**: Access to financial goals, with options to create new goals, track progress, and view badges for completed goals.
- **Currency Converter**: Direct access to real-time currency exchange rates and conversions.
- **Loan Calculator**: Quick navigation to the EMI calculator for managing loan calculations.
- **Reports**: Link to the reports module, where users can select date ranges, generate reports, and download them in PDF or Excel format.
- **Feedback**: Allows users to submit feedback or suggest new features.

9.3 Key Actions Available on Each Screen :

1. **Dashboard**:
    - View recent expenses, income, and budget summaries.
    - Navigate to add expenses, manage budgets, and set goals.
2. **Add Expense/Income**:
    - Input details for new expenses or income, such as amount, category, date, and notes.
    - Save, cancel, or reset the form for a new entry.
3. **Expense/Income Listing**:
    - View all entries with options to filter or sort by date, category, or amount.
    - Edit or delete individual entries.
4. **Budget**:
    - Set new budget categories or adjust existing ones.
    - View progress indicators showing adherence to set budgets.
5. **Financial Goals**:
    - Create new financial goals with a target amount and due date.

    - Track progress with a visual indicator and earn completion badges.

6. **Currency Converter**:
    - Enter amounts and convert them in real-time between selected currencies.
    - Use the integrated currency calculator to add conversions directly to expenses or income.
7. **Loan EMI Calculator**:
    - Input loan amount, interest rate, and tenure to calculate monthly payments.
    - View an amortization schedule and save or reset calculations.
8. **Reports**:
    - Select date ranges and categories for generating financial reports.
    - Export reports in PDF or Excel format for offline access.
9. **Feedback**:
    - Submit feedback or feature suggestions directly to the Expenso team.
    - Provide ratings and comments for specific features.
10. **Settings**:
    - Update account information, such as email or password.
    - Customize notification settings for goals, budget limits, and income/expense alerts.

This guide covers the major user interface components in Expenso, enabling users to understand each page's purpose and functionality. The seamless navigation flow, consistent styling, and intuitive menu layout enhance the overall user experience, making Expenso accessible and effective for personal financial management.

CHAPTER 10 : FUTURE ENHANCEMENTS

The Expenso financial management system is designed to provide comprehensive financial tracking and management capabilities. To further improve user experience, usability, and system management, several additional features and enhancements are planned for future development. These additions aim to increase the value of Expenso for both users and administrators.

10.1 Planned Features and Improvements :

1. **Notification and Report Export Module**

   The notification and report export module is designed to improve user engagement and information accessibility. Currently, Expenso allows users to view their financial summaries and reports within the app. In the planned enhancement, Expenso will add email-based notifications and report distribution, enhancing the system's communication capabilities.

   - **Email Notifications**: Users will receive automated notifications sent directly to their registered email addresses. These notifications may include budget reminders, goal achievements, and spending alerts to help users stay on track with their financial goals.
   - **Exported Reports**: Users will have the option to export their financial reports in PDF or Excel formats. These reports will be generated within Expenso and sent directly to the user's registered email address. This feature will allow users to archive their financial data and share it externally if needed.
   - **Scheduled Notifications**: Users will have the option to set preferred times for receiving notifications, allowing for more personalized reminders and updates.

2. **Admin Module**

   To support user account management and system oversight, Expenso will incorporate an admin module. This module will allow administrators to monitor and manage user accounts, ensuring system integrity and providing support for account-related issues.

   - **User Management**: The admin will have a dashboard displaying a list of all users registered within the Expenso system. This includes access to basic user details, which will be used solely for administrative purposes.
   - **Account Deletion**: If necessary, the admin will have the ability to delete user accounts, for instance, in cases of account inactivity, non-compliance, or upon user request. This feature will help maintain an active user base and manage storage more effectively.
   - **User Details and Activity Monitoring**: Administrators will have access to user activity logs and financial records, enabling them to provide support and resolve any issues users may face with their accounts or usage.

10.2 Future Considerations :

In addition to these planned features, future updates may include the integration of additional services such as multi-currency support, more advanced budgeting tools, enhanced data visualization, and potential integration with external financial applications.

These enhancements will allow Expenso to become a more comprehensive financial management solution, supporting a broader set of needs and improving the overall user experience.

CHAPTER 11 : CONCLUSION

The conclusion chapter summarizes the Expenso project's achievements, reflecting on the project's objectives, challenges, and overall impact. This section highlights key accomplishments, such as:

- **Achievement of Project Goals**: Successfully developed Expenso as a comprehensive financial management system, providing users with expense tracking, budgeting, income management, goal setting, and more.
- **User Benefits**: Enhanced financial awareness and control for users through insightful analytics, visual reports, and a user-friendly interface.
- **Challenges and Solutions**: Overview of any technical or design challenges encountered during the project, such as integrating real-time currency conversion or implementing security features, along with the approaches used to overcome these obstacles.
- **Future Improvements**: Summary of potential enhancements, including additional modules, scalability improvements, or advanced features that could be introduced in subsequent versions.

This chapter aims to encapsulate the project's journey and provide insights into the benefits and value Expenso brings to users and stakeholders.

CHAPTER 12 : REFERENCES

In this section, list all resources, libraries, and frameworks referenced throughout the Expenso project, including:

- **Frameworks and Libraries**:
  - Spring Boot (for backend application development)
  - Thymeleaf (for server-side templating)
  - Chart.js (for graphs and visualizations)
  - Bootstrap (for responsive frontend design)
  - Hibernate (for ORM mapping)
- **APIs and External Integrations**:
  - Currency Exchange API (for real-time currency conversion data)
  - PDF/Excel Libraries (for report export functionality)
- **Documentation and Learning Resources**:
  - Official Spring and Thymeleaf documentation
  - Tutorials, technical blogs, and online courses that aided in the development process.

This section not only acknowledges all sources but also serves as a valuable reference for anyone seeking to understand or replicate certain aspects of the project.

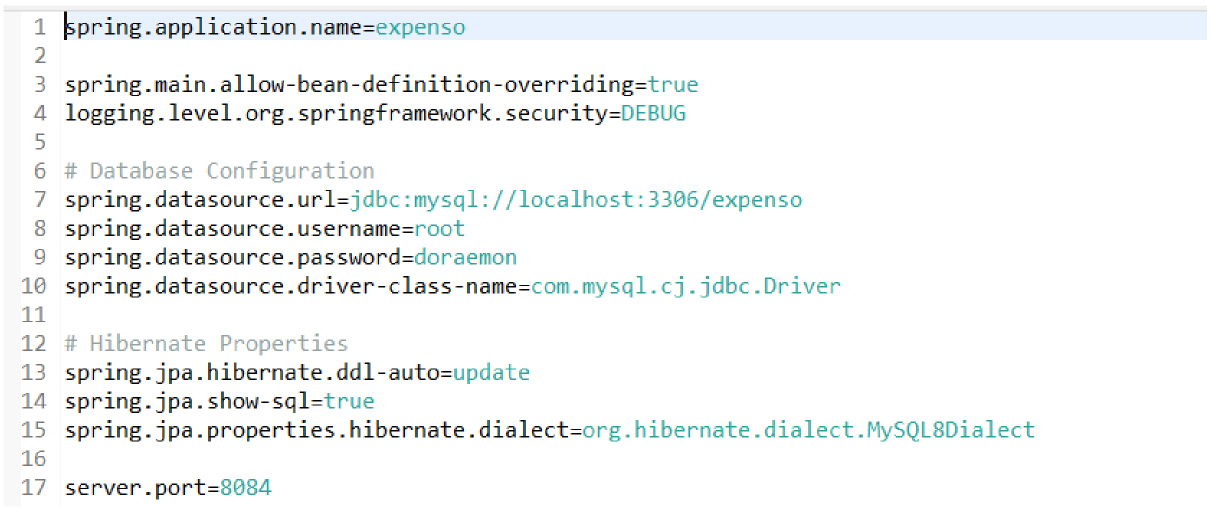
GIT Repository of the Application :

https://github.com/sakheep07/Expenso

CHAPTER 13 : APPENDIX

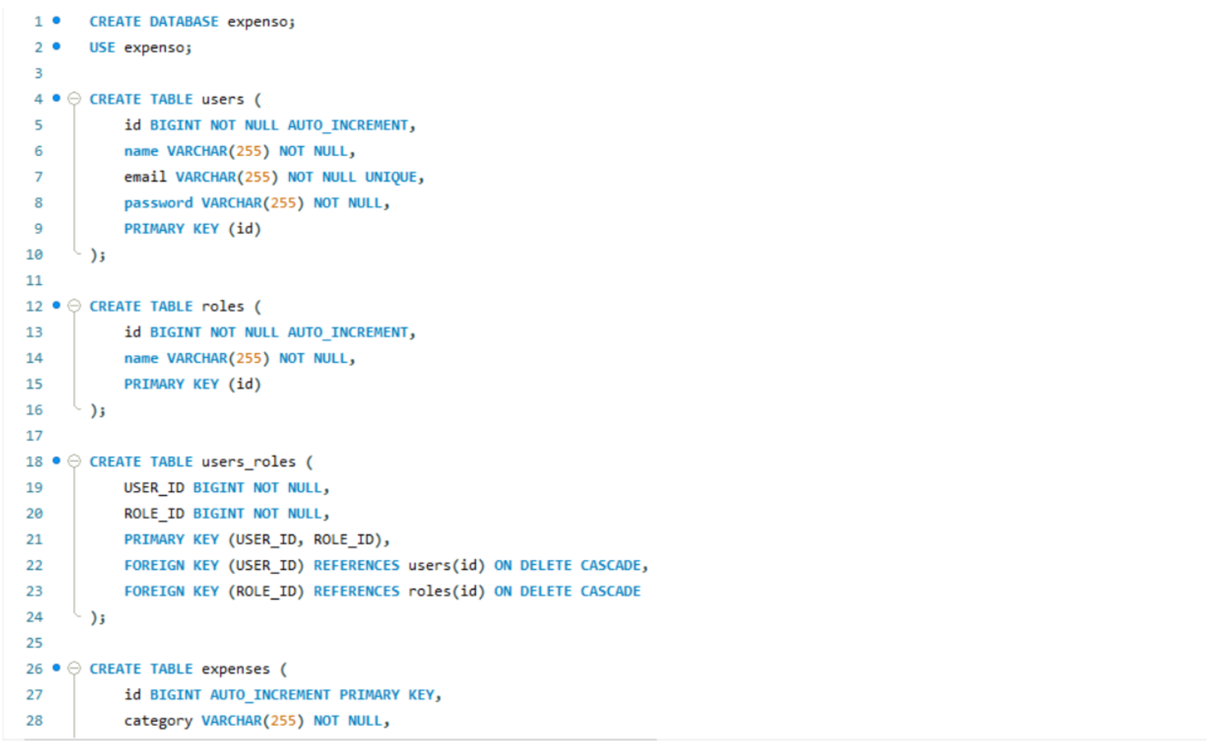
The appendices contain supplementary documentation, technical notes, and code snippets that are useful but not central to the main body of the project. This section might include:

- **Configuration Files**: Sample configurations for deploying Expenso, such as application properties for database, security, and API integration.

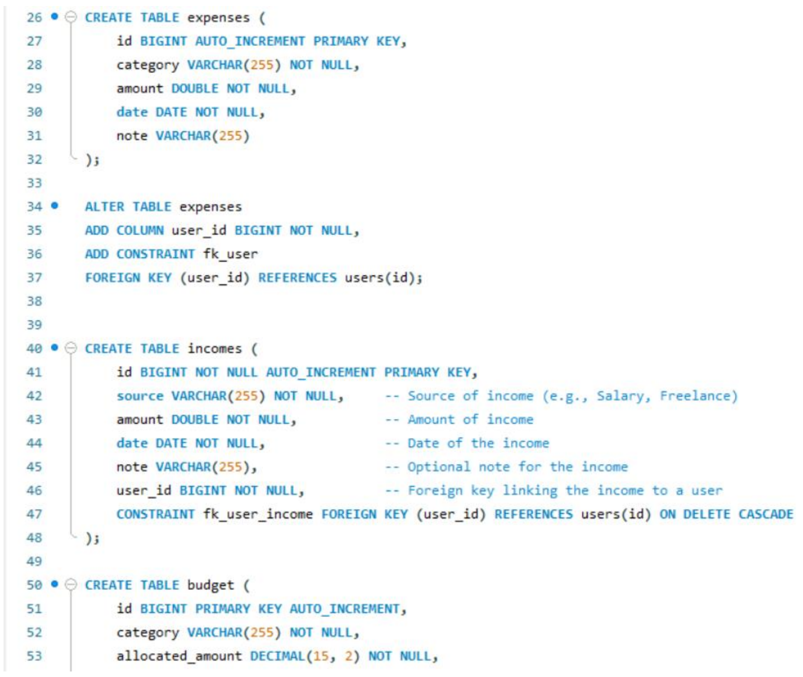


```
spring.application.name=expenso

spring.main.allow-bean-definition-overriding=true
logging.level.org.springframework.security=DEBUG

# Database Configuration
spring.datasource.url=jdbc:mysql://localhost:3306/expenso
spring.datasource.username=root
spring.datasource.password=doraemon
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# Hibernate Properties
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

server.port=8084
```

- **SQL Code**: The full SQL code used to set up Expenso's database, including table creation and sample data inserts.

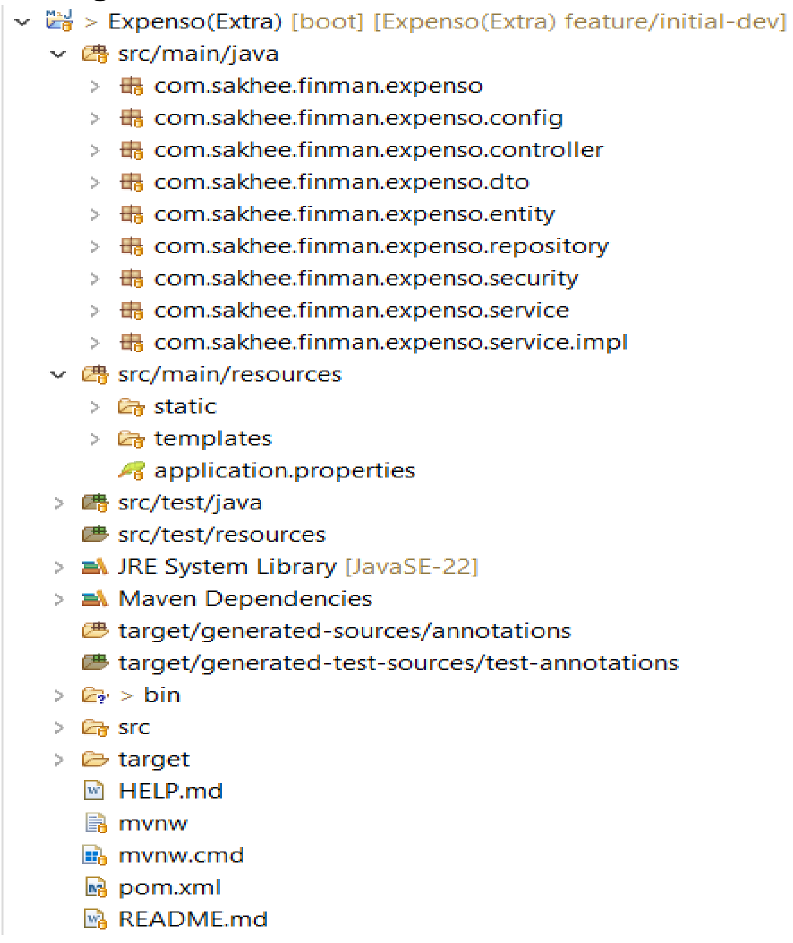


```
CREATE DATABASE expenso;
USE expenso;

CREATE TABLE users (
    id BIGINT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE roles (
    id BIGINT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE users_roles (
    USER_ID BIGINT NOT NULL,
    ROLE_ID BIGINT NOT NULL,
    PRIMARY KEY (USER_ID, ROLE_ID),
    FOREIGN KEY (USER_ID) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (ROLE_ID) REFERENCES roles(id) ON DELETE CASCADE
);

CREATE TABLE expenses (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    category VARCHAR(255) NOT NULL,
```



```sql
CREATE TABLE expenses (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    category VARCHAR(255) NOT NULL,
    amount DOUBLE NOT NULL,
    date DATE NOT NULL,
    note VARCHAR(255)
);

ALTER TABLE expenses
ADD COLUMN user_id BIGINT NOT NULL,
ADD CONSTRAINT fk_user
FOREIGN KEY (user_id) REFERENCES users(id);


CREATE TABLE incomes (
    id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    source VARCHAR(255) NOT NULL,     -- Source of income (e.g., Salary, Freelance)
    amount DOUBLE NOT NULL,           -- Amount of income
    date DATE NOT NULL,               -- Date of the income
    note VARCHAR(255),                -- Optional note for the income
    user_id BIGINT NOT NULL,          -- Foreign key linking the income to a user
    CONSTRAINT fk_user_income FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);

CREATE TABLE budget (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    category VARCHAR(255) NOT NULL,
    allocated_amount DECIMAL(15, 2) NOT NULL,
```

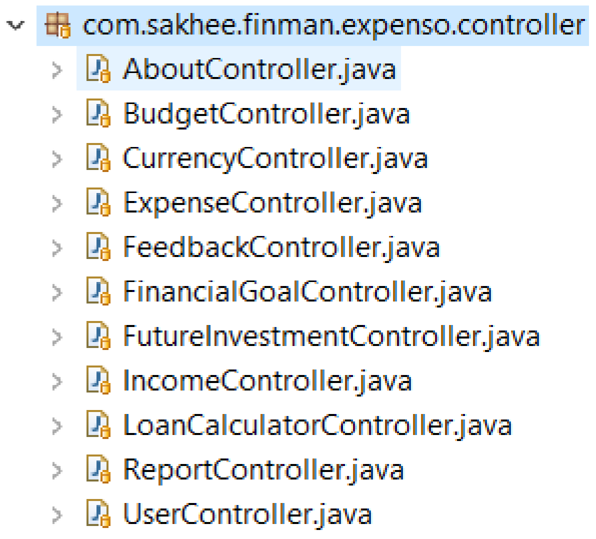
- **Code Snippets**: Key code samples that illustrate complex implementations, such as badge awarding logic, role-based access control configuration, or the currency conversion utility.
- **Package Structure :**



```
> Expenso(Extra) [boot] [Expenso(Extra) feature/initial-dev]
    src/main/java
        com.sakhee.finman.expenso
        com.sakhee.finman.expenso.config
        com.sakhee.finman.expenso.controller
        com.sakhee.finman.expenso.dto
        com.sakhee.finman.expenso.entity
        com.sakhee.finman.expenso.repository
        com.sakhee.finman.expenso.security
        com.sakhee.finman.expenso.service
        com.sakhee.finman.expenso.service.impl
    src/main/resources
        static
        templates
        application.properties
    src/test/java
    src/test/resources
    JRE System Library [JavaSE-22]
    Maven Dependencies
    target/generated-sources/annotations
    target/generated-test-sources/test-annotations
    > bin
    src
    target
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
    README.md
```

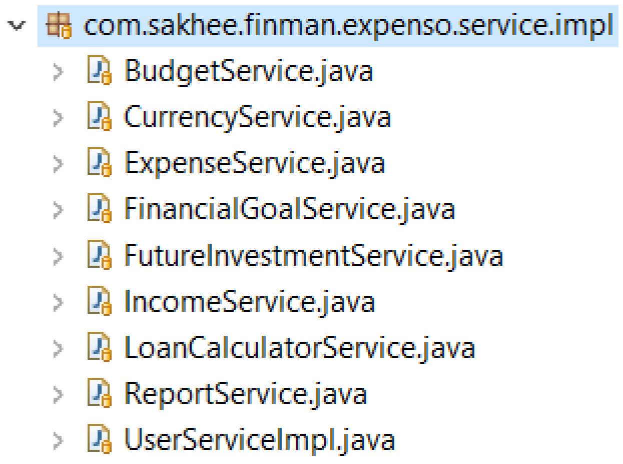
-

- Controller Classes :



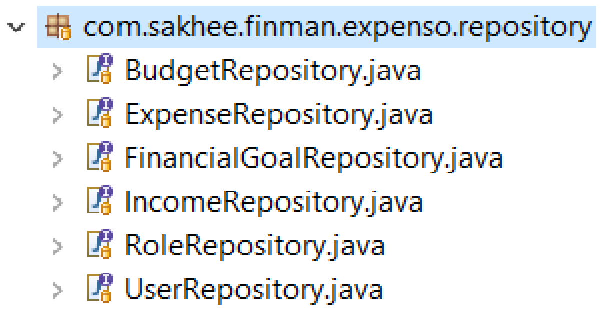
com.sakhee.finman.expenso.controller
AboutController.java
BudgetController.java
CurrencyController.java
ExpenseController.java
FeedbackController.java
FinancialGoalController.java
FutureInvestmentController.java
IncomeController.java
LoanCalculatorController.java
ReportController.java
UserController.java

- Service Classes :



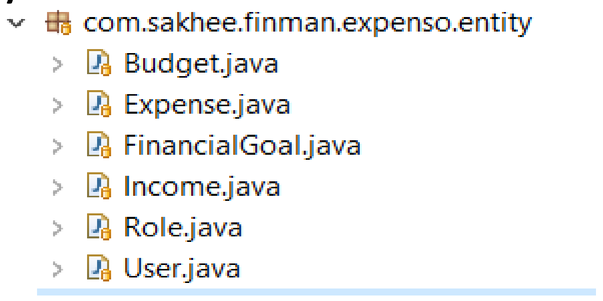
com.sakhee.finman.expenso.service.impl
BudgetService.java
CurrencyService.java
ExpenseService.java
FinancialGoalService.java
FutureInvestmentService.java
IncomeService.java
LoanCalculatorService.java
ReportService.java
UserServiceImpl.java

-
- **Repositories :**



com.sakhee.finman.expenso.repository
BudgetRepository.java
ExpenseRepository.java
FinancialGoalRepository.java
IncomeRepository.java
RoleRepository.java
UserRepository.java

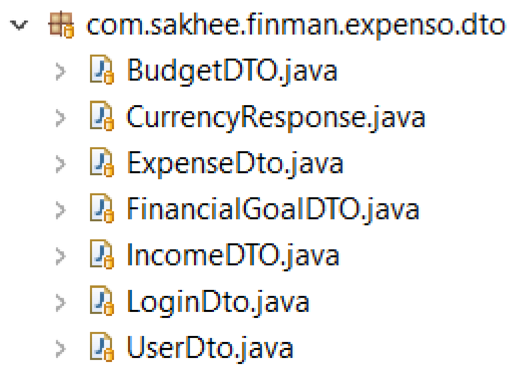
-
- **Entity :**



com.sakhee.finman.expenso.entity
Budget.java
Expense.java
FinancialGoal.java
Income.java
Role.java
User.java

-

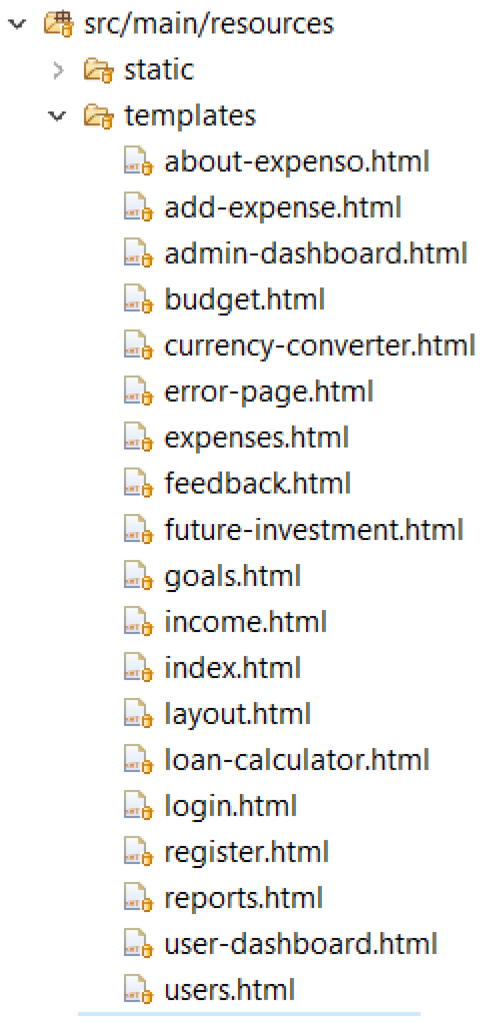
- **DTO :**



```
com.sakhee.finman.expenso.dto
    BudgetDTO.java
    CurrencyResponse.java
    ExpenseDto.java
    FinancialGoalDTO.java
    IncomeDTO.java
    LoginDto.java
    UserDto.java
```

-

- **HTML Pages :**



```
src/main/resources
    static
    templates
        about-expenso.html
        add-expense.html
        admin-dashboard.html
        budget.html
        currency-converter.html
        error-page.html
        expenses.html
        feedback.html
        future-investment.html
        goals.html
        income.html
        index.html
        layout.html
        loan-calculator.html
        login.html
        register.html
        reports.html
        user-dashboard.html
        users.html
```

Chapter 14 : Detailed Plan of Work (for 16 weeks)

The following table summarizes the detail plan of work against the planned duration in weeks.

Serial No.	Tasks or Subtasks	Planned Duration (Weeks)	Specific Deliverable	Status
1	Requirement Gathering & Analysis	1st and 2nd Week	The detailed requirements as agreed upon by the users and supervisor.	Completed
2	Setting up Eclipse IDE, SQL server and configuration Setup	3rd and 4th Week	Installing Eclipse, MS-SQL,Tomcat Server, Maven, Postman etc.	Completed
3	Creating database Table	5th, 6th and 7th Week	Creating database and SQL queries to store data and for fetching the data from DB.	Completed
Mid-Semester Report Submission				
4	Developing the Services	8th, 9th, 10th and 11th Week	Writing code using Java, Spring boot for Rest-API and Integrating the services.	Completed
5	Debugging and testing the application	12th,13th and 14th Week	End to end testing of the Application and debugging	Completed
6	Finalize the document	15th Week	Get the document reviewed from the Supervisor	Completed
7	Final Demo on the Application	16th Week	Give the demo to the supervisor and note down the feedback if any.	Started
Final Report Submission				