

Phainance – Expense Tracker

Aashirwad Asmit Pradhan

24SUPMCAGL001

MCA – B

Sapthagiri NPS University

TABLE OF CONTENT

CHAPTER NO	DESCRIPTION	PAGE NO
1	Introduction	2
2	Problem Statement	3
3	Objectives	4
4	System Analysis	5-6
4.1	Existing System	5-6
4.2	Limitations of Existing System	5-6
4.3	Proposed System	5-6
5	System Design	7-9
5.1	Architecture	7-9
6	Database Design	10-13
7	Module Description	14-16
8	Implementation Details	17-20
9	System Workflow	21-22
10	Future Enhancements and Conclusion	23

CHAPTER 1

INTRODUCTION

Financial management plays an essential role in an individual's long-term stability and decision-making. As living expenses increase and financial responsibilities become more diversified, it has become increasingly important for individuals to track how their money is spent, monitor budget limits, and understand spending behavior. However, most people still depend on manual recording, scattered digital notes, or memory-based tracking, all of which often result in mismanagement, overspending, and poor financial planning.

Phainance is developed as a modern digital platform designed to simplify and enhance the way users handle their everyday financial activities. It serves as an intelligent expense tracking and budget management system that allows individuals to record expenses, categorize spending, monitor their monthly budget usage, and visualize financial patterns through clear analytics. The application supports users in maintaining better financial discipline and gaining deeper insights into their spending habits.

This system introduces an automated, structured, and interactive solution where users can add, modify, and delete expenses effortlessly. It also provides monthly budgeting tools that help users stay within spending limits by displaying usage statistics and remaining balance clearly. Phainance includes analytical visualizations such as pie charts and line graphs that summarize category-wise expenditure and daily trends. These features collectively provide users with a complete picture of their financial health at any given moment.

Behind the scenes, Phainance is built using a reliable and scalable full-stack architecture. The frontend is developed using React, offering a responsive, component-driven interface that enhances usability and speed. Spring Boot powers the backend, ensuring secure API communication, efficient data processing, and smooth integration with MySQL, which stores all financial records in a structured and dependable manner. JWT authentication is implemented for secure login and session handling, ensuring user data privacy.

The aim of Phainance is not only to track expenses but also to help users make informed decisions. By providing clarity, structure, and automation, Phainance enables individuals to take control of their financial habits, avoid unnecessary expenditures, and work towards better financial discipline. With future scalability and enhancement possibilities, Phainance serves as a foundation for a comprehensive personal finance management ecosystem.

CHAPTER 2

PROBLEM STATEMENT

Managing personal finances has become increasingly difficult due to fast-paced lifestyles and scattered financial information. Many individuals do not have a structured or reliable method to record and monitor their daily expenses. As a result, they struggle to maintain accurate spending records, understand their financial habits, or stay within planned budgets. Traditional tracking approaches such as notebooks, manual logs, or basic spreadsheets are prone to errors, time-consuming to maintain, and offer little analytical insight.

These limitations lead to overspending, poor planning, and a lack of financial awareness. With no automated reminders, visual dashboards, or categorized spending information, users find it difficult to make informed decisions about saving and budgeting.

To address these issues, Phainance introduces a simplified digital platform that automates expense tracking, enables budget creation, and provides users with meaningful financial insights.

Key Problems Identified:

- Expenses are recorded manually, resulting in inaccurate or incomplete data.
- No real-time visibility of monthly spending or budget usage.
- Users struggle to categorize expenses and understand spending patterns.
- Traditional methods lack visual analytics such as charts or graphs.
- No centralized system to track budgets and expenses together.
- Difficulty identifying overspending until it is too late.
- Limited ability to make informed financial decisions based on data.

CHAPTER 3

OBJECTIVES

The primary objective of the Phainance Expense Management System is to provide users with a structured, reliable, and intelligent platform to record, monitor, and analyze their financial activities. By offering automation, budget tracking, and visual insights, the system aims to simplify personal finance management and help users make informed financial decisions.

Core Objectives:

- Develop a user-friendly digital system for recording daily expenses with accuracy and consistency.
- Enable users to create, update, and monitor monthly budgets effectively.
- Provide real-time visual analytics such as pie charts and line charts for better financial understanding.
- Implement secure user authentication using JWT-based login and token validation.
- Ensure that every expense is categorized for clearer spending insights.
- Offer a centralized dashboard where users can view budgets, recent expenses, and analytical summaries.
- Improve decision-making by highlighting spending trends, category-wise distribution, and daily patterns.
- Maintain a secure backend for storing user data using Spring Boot, MySQL, and REST APIs.
- Deliver a responsive and visually modern interface developed with React JS and Recharts.
- Support easy scalability for future features such as filters, reports, or export options.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Existing System

In a typical personal finance management approach, individuals rely on manual methods to record their expenses. These include physical notebooks, basic mobile notes, or simple spreadsheets. While these methods may work for occasional tracking, they lack consistency, accuracy, and analytical depth.

Users often fail to maintain regular entries, cannot track budgets effectively, and struggle to understand their long-term spending patterns. This leads to poor financial planning, overspending, and limited control over personal finances.

Characteristics of Existing System:

- Expense tracking is mostly manual and prone to errors.
- No automation or reminders for timely updates.
- Budget monitoring is inconsistent and not centralized.
- No graphical insights or trend analysis.
- Difficult to extract meaningful observations from raw data.
- No secure login or user-specific financial history.

4.2 Limitations of the Existing System

The manual and unstructured systems currently used for tracking expenses present several challenges:

Key Limitations:

- Lack of real-time expense monitoring.
- No category-wise summary or financial breakdown.
- Inability to compare monthly spending patterns.
- No intelligent insights into budget usage or overspending.
- No centralized storage for easy access across devices.
- Limited security for personal financial data.

These limitations highlight the need for a modern digital solution that streamlines tracking, improves budgeting, and enhances financial clarity.

4.3 Proposed System

The proposed system, Phainance, provides an intuitive digital platform for managing personal finances with automation, analytics, and security at its core. It allows users to add and categorize expenses, set monthly budgets, and view visual insights — all from a unified dashboard. The aim is to replace manual tracking with an efficient and user-friendly system that supports informed financial decision-making.

Features of the Proposed System:

- User authentication using secure JWT tokens.
- Easy-to-use interface for adding, editing, and deleting expenses.
- Category-wise and daily analytics using visual charts.
- Monthly budget creation and automatic usage tracking.
- Centralized dashboard showing key financial metrics.
- Secure backend with Spring Boot and MySQL.
- Seamless communication between frontend and backend via REST APIs.
- Responsive and modern UI built using React JS.

CHAPTER 5

SYSTEM DESIGN

System design defines the architecture, structure, and data flow of the Phainance Expense Management System. It ensures that all components interact efficiently and securely, providing users with a seamless experience for managing expenses and budgets. The design follows established software engineering principles such as modularity, scalability, and ease of maintenance.

5.1 Architecture

Phainance follows a **three-tier architecture** consisting of:

1. Presentation Layer (Frontend - React JS)
2. Application Layer (Backend - Spring Boot REST APIs)
3. Data Layer (MySQL Database)

This structured approach separates responsibilities, improves maintainability, and supports scalability for future enhancements.

1. Presentation Layer — React JS

The frontend is built using React JS for modern, responsive, and component-based UI. Key responsibilities include:

- Handling user interactions (Login, Add Expense, Edit Expense, budgeting)
- Displaying dashboards, tables, and charts
- Communicating with backend APIs through Axios
- Managing user session through JWT token stored in localStorage
- Rendering Recharts visual analytics (Pie charts, Line charts, Bar charts)

The UI is designed with a clean, modern layout and theme system (Light/Dark mode) to enhance usability.

2. Application Layer — Spring Boot

The backend handles the business logic and acts as the bridge between the frontend and the database. Responsibilities include:

- Authenticating users via JWT tokens
- CRUD operations for expenses and budgets
- Category management for classification of expenses
- Generating monthly analytics (category-wise, daily trends)
- Ensuring data validation and error handling
- Providing secure REST API endpoints for communication

The layered backend architecture includes:

- Controller Layer
- Service Layer
- Repository Layer
- Entity Layer

This structure ensures clean separation of concerns.

3. Data Layer — MySQL

MySQL stores all persistent data required by the application. This includes:

- User credentials
- Expense records
- Budget details
- Categories

The database schema ensures referential integrity using foreign keys and unique constraints to avoid duplication of categories.

5.2 System Flow Overview

Below is the high-level workflow of user operations within Phainance:

- User logs in → JWT generated → stored in frontend
- Dashboard loads → sends token to backend for verification
- Month selected → frontend requests expenses, budget, and analytics
- Backend fetches data → sends as JSON response
- User adds/edits/deletes an expense → update reflected immediately
- User sets budget → backend stores the limit and recalculates spending
- Charts refresh automatically based on updated data

5.3 Benefits of the Architecture

- High scalability for future enhancements
- Secure handling of data and authentication
- Faster performance through efficient REST API communication
- Cleaner, modular code structure
- Easy maintainability and debugging

CHAPTER 6

DATABASE DESIGN

The database design of Phainance plays a crucial role in managing user information, expenses, categories, and monthly budgets. The schema is optimized for accuracy, fast retrieval, and maintaining referential integrity. MySQL is used as the database due to its reliability, performance, and compatibility with Spring Boot.

6.1 Overview of Database Structure

The Phainance database contains the following core tables:

1. users – Stores user authentication details
2. categories – Stores unique expense categories
3. expenses – Records all expense entries
4. budgets – Stores monthly budget limits for users

Each table is normalized to avoid redundancy and ensure smooth CRUD operations from the backend.

6.2 Database Name

`expense_tracker`

6.3 Table Descriptions

1. users Table

Stores credentials and user identity used during authentication.

Column	Type	Description
id	BIGINT (PK)	Unique identifier for each user
username	VARCHAR(100)	Username (unique)
email	VARCHAR(150)	User email (unique)
password	VARCHAR(255)	Hashed password

2. categories Table

Stores predefined and user-specific categories for expense grouping. Includes a uniqueness constraint to prevent duplicates.

Column	Type	Description
id	BIGINT (PK)	Unique identifier
name	VARCHAR(255)	Category name (unique)

Default categories inserted:

Food, Rent, Utilities, Entertainment, Other

3. expenses Table

Stores records of user expenses with foreign key linking to categories.

Column	Type	Description
id	BIGINT (PK)	Expense entry ID
amount	DOUBLE	Amount spent
description	VARCHAR(255)	Additional details of expense
date	DATE	Date of expense
category_id	BIGINT (FK)	Links to categories(id)
user_id	BIGINT (FK)*	(If implemented) Links expense to specific user

4. budgets Table

Stores budget limits based on month.

Column	Type	Description
id	BIGINT (PK)	Budget entry ID
month	VARCHAR(20)	Month in YYYY-MM format
limit_amount	DOUBLE	Budget limit set by the user

6.4 SQL Schema (already implemented)

```
CREATE DATABASE IF NOT EXISTS expense_tracker;
```

```
USE expense_tracker;
```

```
CREATE TABLE IF NOT EXISTS categories (
```

```
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
```

```
    name VARCHAR(100) NOT NULL,
```

```
    UNIQUE KEY ux_categories_name (name)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS expenses (
```

```
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
```

```
    amount DOUBLE,
```

```
    description VARCHAR(255),
```

```
    date DATE,
```

```
    category_id BIGINT,
```

```
    FOREIGN KEY (category_id) REFERENCES categories(id)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS budgets (
```

```
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
```

```
    month VARCHAR(20) NOT NULL,
```

```
    limit_amount DOUBLE
```

```
);
```

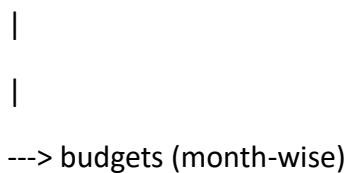
```
CREATE TABLE IF NOT EXISTS users (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(100) NOT NULL UNIQUE,
    email VARCHAR(150) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL
);
```

6.5 ER Diagram (Description)

This section can hold a screenshot in your final report.

Textual representation:

users (1) ----- (many) expenses ----- (1) categories



6.6 Database Features

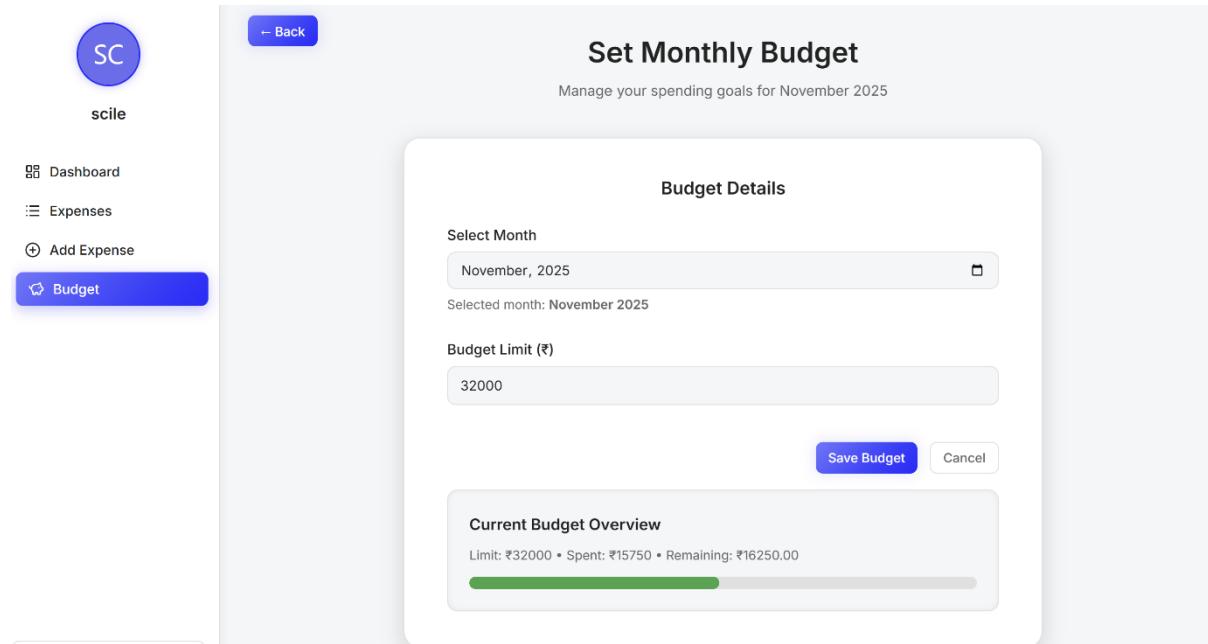
- Fully normalized structure
- Strong referential integrity using foreign keys
- Unique category constraint to prevent duplicates
- Efficient indexing for faster queries
- Easily scalable for multi-user support

CHAPTER 7

MODULE DESCRIPTION

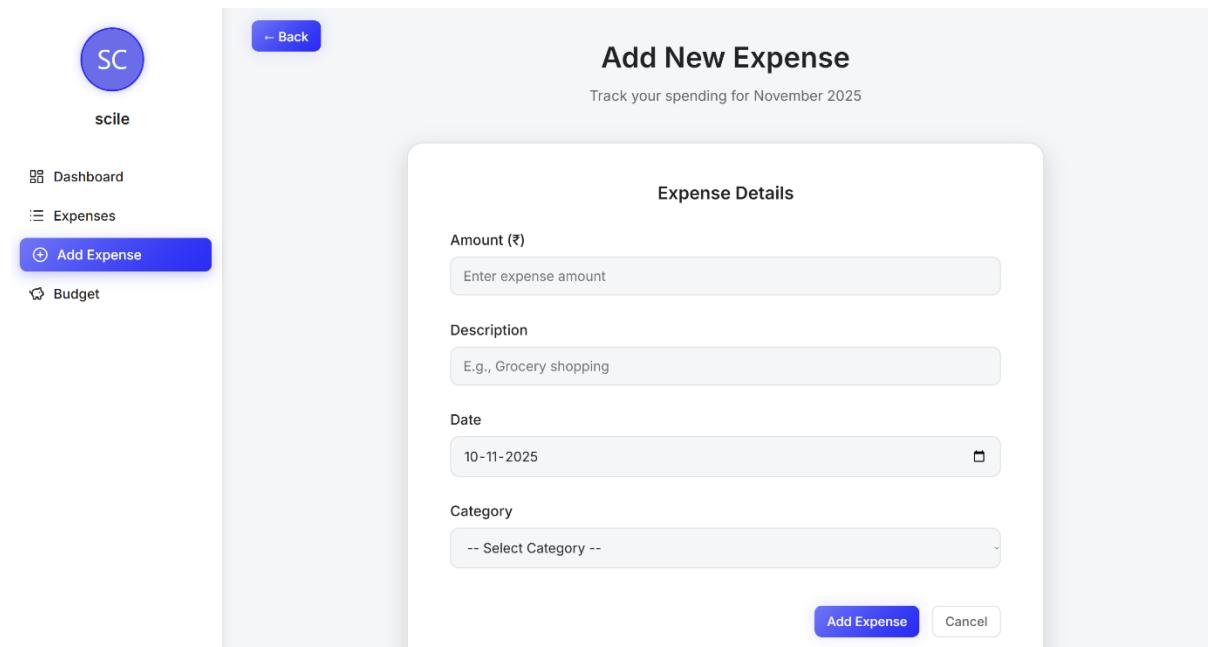
Expense Module:

Allows users to add, edit, delete, and categorize expenses. Also displays recent expenses.



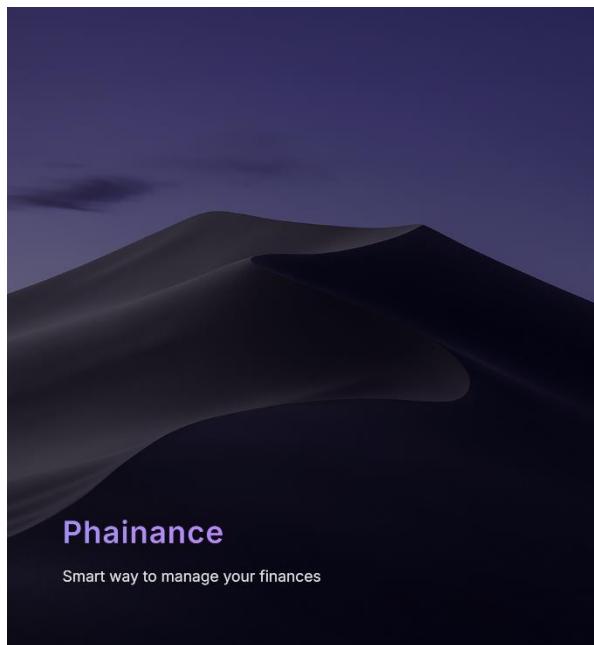
Budget Module:

Allows the user to set a monthly budget and view the remaining amount. Tracks usage percentage automatically.



Authentication Module:

Handles user registration, login, and secure access using JWT tokens.

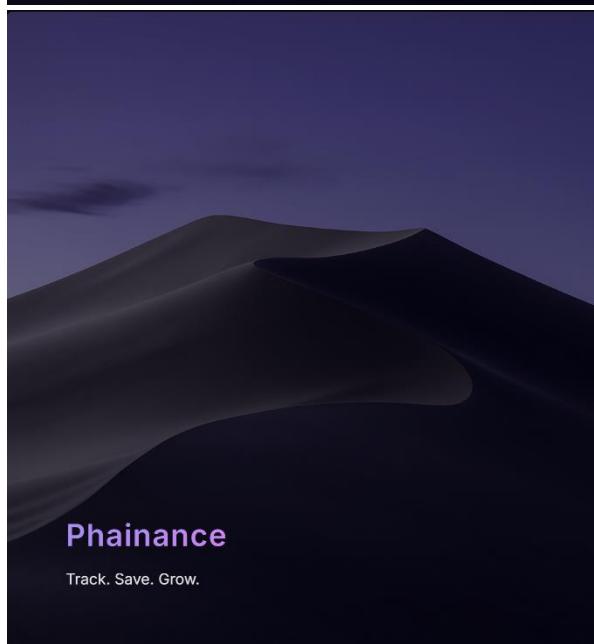


Create Account

Join us to start tracking your expenses

Sign Up

Already have an account? [Login](#)



Welcome Back

Login to continue managing your expenses

Sign In

Don't have an account? [Sign up](#)

Dashboard Module:

The screenshot shows the Dashboard Overview page. On the left is a sidebar with a user icon 'scile' and navigation links: Dashboard (highlighted in blue), Expenses, Add Expense, and Budget. The main area has a header 'Dashboard Overview' and a welcome message 'Welcome back, scile'. It displays three key metrics: Spent (November 2025) at ₹15750.00, Budget at ₹32000.00 (Remaining: ₹16250.00), and Usage at 49%. Below these are two charts: a pie chart titled 'Analytics — November 2025' showing spending by category, and a line graph showing expense trends over time. A table titled 'Recent Expenses — November 2025' lists recent transactions with columns for ID, Amount, Category, Date, and Description. At the bottom left is a 'Logout' button.

ID	Amount	Category	Date	Description
26	₹2000	Entertainment	2025-11-03	primos
27	₹1000	Other	2025-11-03	lol
114	₹2800	Food	2025-11-04	Groceries
115	₹5000	Rent	2025-11-01	Rent
116	₹950	Utilities	2025-11-12	Water + Electricity
117	₹1900	Entertainment	2025-11-17	Netflix, outings, etc.
119	₹2100	Health	2025-11-19	Doctor and medicines

Displays financial summaries, analytics charts, expense trends, and quick actions.

The screenshot shows the Expense List page. On the left is a sidebar with a user icon 'scile' and navigation links: Dashboard (highlighted in blue), Expenses (highlighted in blue), Add Expense, and Budget. The main area has a header 'Expense List' and a message 'Showing expenses for November 2025'. It displays a table of recent expenses with columns for ID, Amount (₹), Category, Date, Description, and Actions (Edit and Delete buttons). At the bottom left is a 'Logout' button.

ID	Amount (₹)	Category	Date	Description	Actions
26	₹2000	Entertainment	2025-11-03	primos	<button>Edit</button> <button>Delete</button>
27	₹1000	Other	2025-11-03	lol	<button>Edit</button> <button>Delete</button>
114	₹2800	Food	2025-11-04	Groceries	<button>Edit</button> <button>Delete</button>
115	₹5000	Rent	2025-11-01	Rent	<button>Edit</button> <button>Delete</button>
116	₹950	Utilities	2025-11-12	Water + Electricity	<button>Edit</button> <button>Delete</button>
117	₹1900	Entertainment	2025-11-17	Netflix, outings, etc.	<button>Edit</button> <button>Delete</button>
119	₹2100	Health	2025-11-19	Doctor and medicines	<button>Edit</button> <button>Delete</button>

CHAPTER 8

IMPLEMENTATION DETAILS

The implementation of the Phainance Expense Tracking System follows a structured full-stack development approach using React.js for the frontend, Spring Boot for the backend, and MySQL for database storage. Each component is designed to be modular, scalable, and secure.

8.1 Frontend Implementation

The frontend of Phainance is developed using React.js with a modular component-based architecture. The interface is styled using custom CSS along with responsive layout structures to ensure compatibility across devices.

Key Technologies Used:

- React.js
- React Router DOM
- Axios for API communication
- Recharts for visual analytics
- React Toastify for notifications
- LocalStorage for theme and session persistence

Important Frontend Components:

- Login and Signup Pages
- Dashboard Page
- Add Expense Form
- Expense List with inline edit
- Budget Management Page
- Sidebar Navigation Component
- Theme Toggle System

Frontend Workflow:

1. User interacts with UI components
 2. Axios sends REST API requests to backend
 3. Responses update state variables
 4. UI re-renders dynamically
 5. Toast notifications appear for feedback
-

8.2 Backend Implementation

The backend is implemented using Spring Boot with a layered architecture for better maintainability and separation of concerns.

Layers Used:

- Controller Layer – handles user requests
- Service Layer – contains business logic
- Repository Layer – interacts with the database
- Model Layer – defines entities (User, Expense, Category, Budget)

Key Technologies Used:

- Spring Boot
- Spring Web
- Spring Data JPA
- MySQL Connector
- JWT for authentication
- HikariCP for connection pooling

Major Backend Functionalities:

- User authentication with JWT
 - CRUD operations for expenses
 - Category management with duplicate prevention
 - Budget creation and monthly filtering
 - Analytics endpoints for category and daily trends
-

8.3 REST API Endpoints

The system exposes RESTful APIs for all major operations.

Authentication Routes:

- POST /api/auth/signup
- POST /api/auth/login

Expense Routes:

- GET /api/expenses?month=YYYY-MM
- POST /api/expenses
- PUT /api/expenses/{id}
- DELETE /api/expenses/{id}

Category Routes:

- GET /api/categories

Budget Routes:

- GET /api/budget?month=YYYY-MM
- POST /api/budget

Analytics Routes:

- GET /api/analytics/category
 - GET /api/analytics/daily
 - GET /api/analytics/monthly
-

8.4 Database Interaction

Data is persisted using Spring Data JPA, which simplifies the CRUD process by using repository interfaces. The database structure includes tables for users, expenses, categories, and budgets.

Overall Database Flow:

1. API request reaches controller
 2. Controller calls service method
 3. Service interacts with repository
 4. Repository performs SQL operations
 5. Response returns to frontend
-

8.5 Security Implementation

Security is implemented using JSON Web Tokens (JWT).

Security Features:

- Token-based authentication
- Validation of token on every request
- Encrypted password storage
- Unauthorized request handling

Authentication Flow:

1. User logs in
 2. Backend generates JWT
 3. JWT stored in LocalStorage
 4. All future requests include Authorization header
-

8.6 Notification System

Notifications are implemented using React Toastify.

Features:

- Rounded corner designs
- Success and error alerts
- Auto dismiss after a few seconds
- Displayed at top-right corner

Used for:

- Adding expenses
 - Editing expenses
 - Deleting expenses
 - Saving budget
-

8.7 Theming System

Phainance supports both Light and Dark modes.

How it works:

- Theme preference stored in LocalStorage
 - CSS variables dynamically applied
 - Theme toggled using a simple button
-

CHAPTER 9

SYSTEM WORKFLOW

The workflow of the Phainance Expense Tracking System describes how data flows across the frontend, backend, and database during different operations. This ensures that the system remains structured, predictable, and easy to maintain.

9.1 User Authentication Workflow

1. User attempts login through the React UI.
2. Credentials are sent to the backend via POST /login.
3. Backend verifies credentials and generates a JWT token.
4. Token is returned and stored in the browser's LocalStorage.
5. Token is attached to all future API requests for authorization.

9.2 Expense Addition Workflow

1. User opens the Add Expense form.
2. Fills details and submits the form.
3. Frontend sends POST /expenses with token.
4. Backend validates token, maps the expense to the user, and saves it.
5. Database stores the expense in the expenses table.
6. Success notification is displayed.

9.3 Budget Management Workflow

1. User selects a month and enters a limit.
2. Request is sent to POST /budget.
3. Backend updates or creates the monthly budget for the logged-in user.
4. Database stores the limit in the budgets table.
5. Budget instantly appears on the dashboard.

9.4 Analytics Workflow

1. User opens dashboard.
2. Frontend sends parallel GET requests for:
 - o Category-wise spending
 - o Daily trend spending
 - o Monthly overview
3. Backend aggregates expense data using SQL queries.
4. Responses update charts (PieChart, LineChart).
5. Dashboard displays visual analytics.

9.5 Expense Editing Workflow (Inline Editing)

1. User clicks Edit on the expense list.
2. Fields transform into editable inputs.
3. On save, PUT /expenses/{id} is triggered.
4. Backend validates the updated data and stores changes.
5. Data refreshes and a success toast is shown.

9.6 Theme Handling Workflow

1. User clicks the theme toggle button.
 2. Theme preference updated in LocalStorage.
 3. CSS variables update dynamically to switch styles.
 4. Theme remains consistent on reload or login.
-

CHAPTER 11

FUTURE ENHANCEMENT AND CONCLUSION

11.1 Future Enhancements

The Phainance system is built with modularity and scalability in mind. Several improvements can be implemented in future versions:

- Export expenses to PDF/Excel for reporting
- Add recurring expense reminders
- Introduce multi-user family-level budgeting
- Add graphs comparing multiple months
- Integrate voice input for quick expense logging
- Add AI-powered expense categorization
- Enable syncing with bank SMS for auto-expense logging
- Add mobile app using React Native

These enhancements can elevate Phainance into a more intelligent and automated financial management solution.

11.2 Conclusion

The Phainance Expense Tracking System successfully delivers an efficient and user-friendly platform for managing personal finances. Through its secure authentication, intuitive design, and insightful analytics, the system simplifies the process of tracking expenses and maintaining budgets.

The use of React ensures a smooth and responsive UI, while Spring Boot and MySQL provide a reliable and scalable backend foundation. Features such as category-wise analytics, daily trend charts, and inline editing motivate users to better understand their spending habits.

Phainance proves to be a practical, modern, and extensible solution capable of evolving with user needs. It combines simplicity with power, making financial management accessible, visual, and organized for users of all backgrounds.