# Mini Project Report

### Full Stack Development Lab

## 1. Title Page

**Project Title:** Expense Tracker (MERN Stack)

**Student Names + PRN :**

1. Vishwesh Bhat - 1032232281

2.Kartik Kaushik - 1032231250

3.Akshat Kolluru – 1032231219

**Course and Panel:** TYCSF – Panel A

**Course Name:** Full Stack Development

**Course Code:** CSF3PM01A

**Submission Date:** 28-10-2025

## 2. Abstract

The Expense Tracker project is a full-stack web application developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack. It allows users to manage their daily expenses efficiently by recording, categorizing, and analyzing spending patterns. The main objective is to help users maintain financial discipline and gain insights into their income and expenditure. The system provides user authentication, CRUD operations for transactions, and visual representations of expense data. The project demonstrates a

practical implementation of full-stack development concepts and modern web technologies.

# 3. Introduction

## Background and Motivation

In today’s digital world, personal finance management has become increasingly essential. Traditional methods of tracking expenses using spreadsheets or notebooks are inefficient and prone to errors. Therefore, a modern, web-based expense tracking system helps users conveniently manage their financial activities.

## Problem Statement

Users lack a simple, secure, and visually interactive platform to monitor their income and expenses effectively.

## Objectives

- To develop a responsive web application using the MERN stack.
- To implement secure user authentication and authorization.
- To enable users to add, view, edit, and delete transactions.
- To provide data visualization for better expense analysis.

## Scope and Limitations

The project focuses on personal expense management for individual users. It does not handle multi-user collaboration or advanced financial forecasting. Future extensions may include multi-user access, data export, and AI-based budgeting suggestions.

# 4. Literature Review

Several expense management tools exist, such as Mint, YNAB (You Need A Budget), and PocketGuard. These applications offer comprehensive budgeting features but often require subscriptions or collect user data for analytics.

This project takes inspiration from such tools but emphasizes simplicity, data privacy, and educational value. It implements fundamental CRUD operations and RESTful API communication using the MERN stack, showcasing practical use of modern web development paradigms.

# 5. Methodology

## Technologies Used

- **Frontend:** React.js, HTML5, CSS3, JavaScript
- **Backend:** Node.js, Express.js
- **Database:** MongoDB (NoSQL)
- **Version Control:** Git & GitHub
- **Other Tools:** Postman (API Testing), VS Code (Development Environment)

## System Architecture

The system follows a **client-server architecture**:

- **Frontend (React.js):** Handles user interaction, displays forms, tables, and charts.
- **Backend (Node + Express):** Provides RESTful APIs for CRUD operations.
- **Database (MongoDB):** Stores user data and transaction details.

## Components

1. **User Authentication Module** – Sign Up, Login, JWT-based token authentication.
2. **Transaction Management Module** – Create, Read, Update, Delete expenses and income records.
3. **Dashboard Module** – Displays categorized expenses using charts or summary lists.

### Development Methodology

The project followed an **Agile Development** approach with iterative sprints:

- **Sprint 1:** Project setup and environment configuration
- **Sprint 2:** Backend API development
- **Sprint 3:** Frontend integration
- **Sprint 4:** Testing and deployment

## 6. Results and Discussion

### Results

- Successfully developed a fully functional Expense Tracker web app.
- User authentication and session management implemented using JWT.
- CRUD operations for transaction management integrated with MongoDB.
- Responsive dashboard displaying categorized expense summaries.

### Discussion

The project met its key objectives and provided hands-on experience with full-stack development. Challenges included managing asynchronous data flow between React components and API endpoints, configuring MongoDB connections, and ensuring proper error handling. The experience provided valuable insight into backend and frontend integration.

### Evaluation

The project effectively demonstrates understanding of:

- REST API design and implementation
- React component architecture
- State management using hooks
- MongoDB CRUD operations
- Deployment considerations for full-stack apps

## 7. Conclusion

The Expense Tracker (MERN) project successfully showcases a practical application of full-stack web development concepts. It provides a secure and efficient system for managing personal expenses, improving financial awareness.

The system can be further enhanced with features like:

- Exporting reports to PDF/Excel
- Multi-user account support
- Integration with third-party APIs for real-time analytics

This project strengthened understanding of the MERN stack and modern web technologies, aligning well with the objectives of the Full Stack Development course.

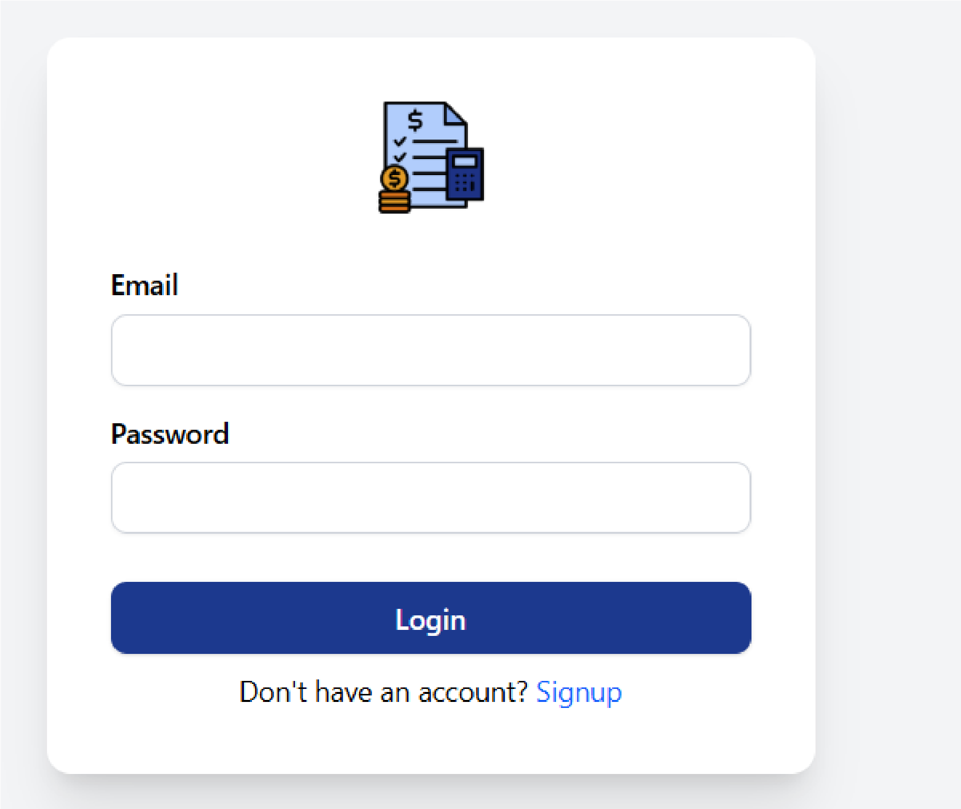
## 8. References

1. React.js Official Documentation – https://react.dev
2. Node.js Documentation – https://nodejs.org
3. Express.js Guide – https://expressjs.com
4. MongoDB Documentation – https://www.mongodb.com/docs
5. Mozilla Developer Network (MDN) – https://developer.mozilla.org

## 9. Appendices

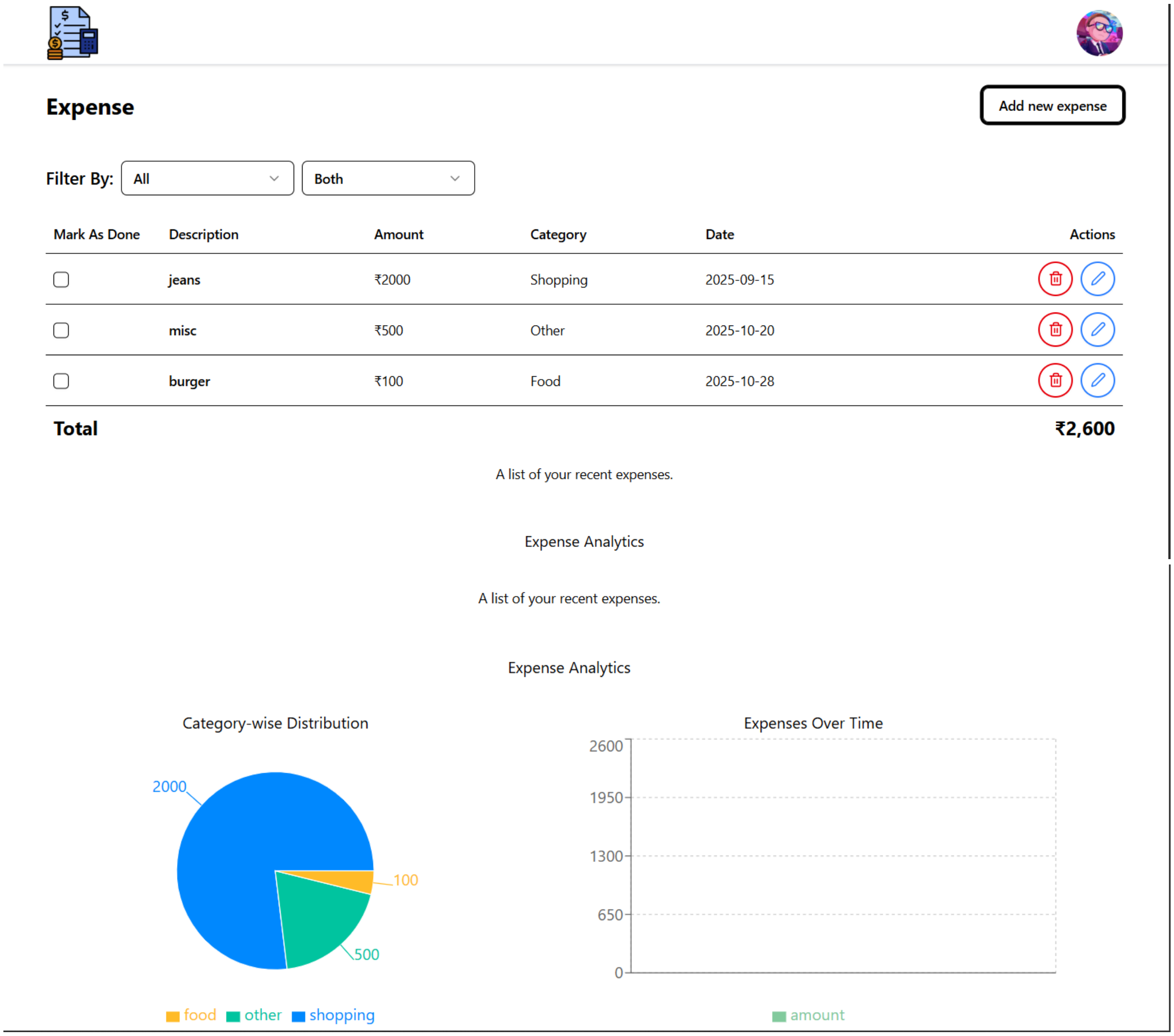
### A. Screenshot Examples

- Login Page

- 

Don't have an account? Signup

- Dashboard Page with Charts



## Expense

Filter By: All Both

| Mark As Done | Description | Amount | Category | Date | Actions |
|---|---|---|---|---|---|
| □ | jeans | ₹2000 | Shopping | 2025-09-15 | |
| □ | misc | ₹500 | Other | 2025-10-20 | |
| □ | burger | ₹100 | Food | 2025-10-28 | |
| **Total** | | | | | **₹2,600** |

A list of your recent expenses.

Expense Analytics

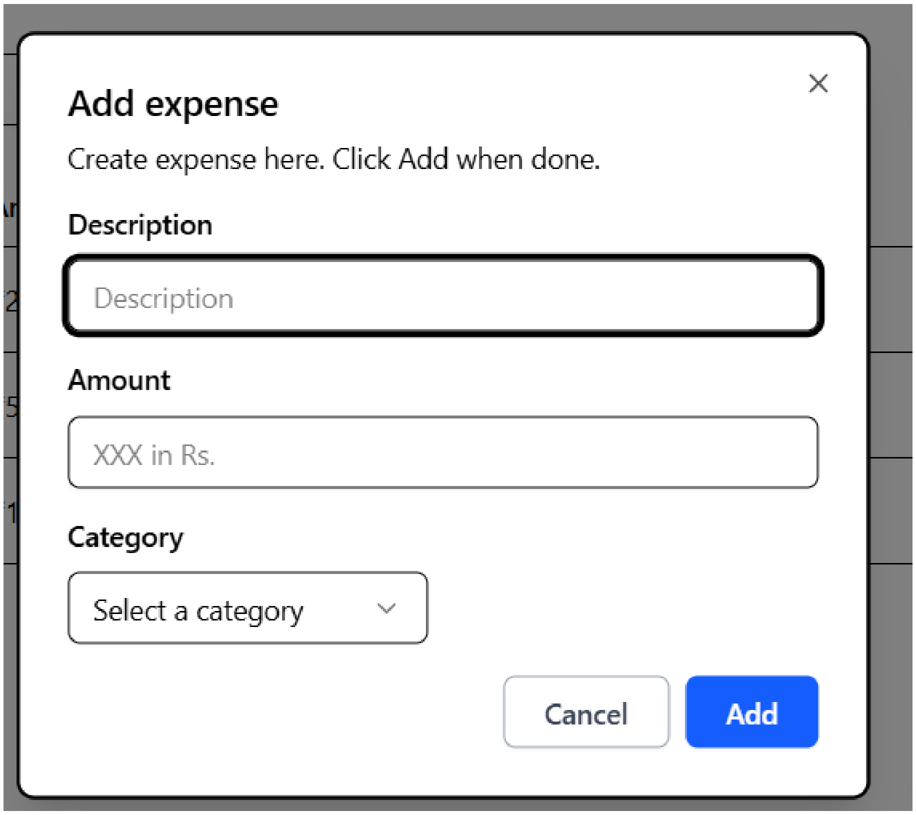
A list of your recent expenses.

Expense Analytics

Category-wise Distribution

Expenses Over Time

- Add expense

- 

Add expense

Create expense here. Click Add when done.

**End of Report**