BudgetBuddy - MERN Stack Project

A Project Report

Submitted in partial fulfillment of the requirements

of

MERN Stack Web Development

By

Shreenivas Dudhate – shreenivas.dudhate@mitaoe.ac.in

Sumedh Meshram – sumedh.meshram@mitaoe.ac.in

Pratik Bangar – pratik.bangar@mitaoe.ac.in

Sion Mankar – sion.mankar@mitaoe.ac.in

Under the Guidance of

Mr. Kaushal Joshi

ACKNOWLEDGEMENT

We would like to take this opportunity to express our sincere gratitude to all those who have supported and guided us throughout the course of this project.

First and foremost, we extend our heartfelt thanks to our project guide, Mr. Kaushal Joshi, for his constant encouragement, valuable insights, and constructive feedback at every stage of this project. His expert guidance has been indispensable in shaping our work and ensuring its successful completion.

We are also deeply grateful to the faculty and staff of our institution for providing the resources, infrastructure, and academic support necessary for the successful execution of this project. Their unwavering belief in our abilities and continuous support has inspired us to give our best efforts.

Our sincere appreciation is extended to our fellow classmates and friends for their collaborative spirit and constructive criticism, which helped us refine and better present our ideas and strategies.

Lastly, we would like to express our profound gratitude to our families for their patience, understanding, and unconditional support. Their encouragement has been a constant source of motivation and has played a crucial role in the successful completion of this project. It is through this endeavor that we have gained invaluable learning and a deeper understanding of the field.

ABSTRACT of the Project

Personal finances can be a nightmare without a structured approach, usually resulting in some bad decisions and less than ideal awareness of spending. The Expenses Tracker , using the MERN (MongoDB, Express.js, React.js, Node.js) stack, solves this problem by creating an easy-to-use platform for recording, categorizing, and analyzing expenses in real-time.

The main goal behind this project is to propose and develop a web application that simplifies the processes of expense management and provides users with relevant insights into their spending habits. Under this application, users could register, securely log on, add or delete a record of expenses, or view their spending patterns on graphical representations.

The methodology is to create a responsive frontend with React.js, which will provide a smooth user experience, a robust backend with Express.js and Node.js to handle data operations, and MongoDB for efficient and scalable data storage. User authentication is done using JSON Web Tokens (JWT) to ensure the security of data. DFDs and systematic module design were used to make development more streamlined.

Key results are the successful implementation of CRUD operations in expense management and dynamic visualization of spending patterns, thus enhancing users' ability to make informed decisions. The application is tested for functionality, scalability, and user-friendliness.

In conclusion, the Expenses Tracker achieves its goal of providing a practical and efficient means of managing personal finances. This project clearly shows the potential of the MERN stack in developing modern full-stack applications. Budgeting tools, integration with financial APIs, and multi-user support for shared expenses would be some future enhancements to this project.

TABLE OF CONTENTS

Abstract ....

List of Figures ....

List of Tables ....

Chapter 1. Introduction

1.1 Problem Statement

1.2 Motivation

1.3 Objectives

1.4. Scope of the Project

Chapter 2. Literature Survey

Chapter 3. Proposed Methodology

Chapter 4. Implementation and Results

Chapter 5. Discussion and Conclusion

References

LIST OF FIGURES

		Page No.
Figure 1	Architecture Diagram	**14**
Figure 2	System architecture	**14**

CHAPTER 1

Introduction

1.1 Problem Statement:

Most people find it challenging to monitor their spending properly.
Inappropriate tracking of finances results in overspending, budgeting problems, and poor understanding of how money is being spent.
Tracking expenses manually is a time-consuming and error-prone process.
Significance
Proper expense tracking is essential to accomplish goals such as saving for a down payment, paying off debt, or retirement planning.
This equips individuals to make more informed financial decisions, decrease unnecessary spending, and overall, enhance their financial wellness.

1.2 Motivation:

The purpose of choosing this project was to overcome the universal challenges of people handling their personal finance.
Developing a friendly and intuitive expense tracking application will help people have better control over their expenses.
Potential Applications and Impact:
Personal Finance: The app will assist individuals to handle their personal budget, save money in some areas, and help them meet financial goals.
Small Businesses: It will assist small businesses in keeping track of expenses, handling budgets, and making financial decisions more effective.
An educational tool that could teach individuals how to budget and save as well as gain financial literacy.

1.3 Objective:

Develop a user-friendly and intuitive web application that simplifies the process of tracking personal or business expenses.
Enable users to easily record expenses with details such as date, amount, category, description, and payment method.
Provide a flexible categorization system allowing users to organize expenses effectively and gain insights into spending patterns.
Facilitate the creation and management of budgets by allowing users to set spending limits for different categories and track progress towards financial goals.
Generate comprehensive reports and visualizations of spending data, including charts and graphs, to help users understand their financial behavior.
Ensure the security and privacy of user data through appropriate authentication and authorization mechanisms.
Deliver a reliable and scalable application that can accommodate a growing user base and evolving financial tracking needs.

.

1.4 Scope of the Project:

Scope of the Project

- The project offers an online solution to track one's personal expenses.
- Users can register and maintain securely, individually protected accounts.
- Users can add, edit, view, and delete expense records categorized.
- The application provides graphical visualizations of expense trends and summaries.
- It supports real-time update and database synchronization using MongoDB.
- It is designed with accessibility in mind and functions on both desktop and mobile browsers.
- The system supports multi-user scalability with isolated data.

Limitations

- Features like predictive expense analysis or budgeting are not in the application.
- External APIs or financial institutions integration is not supported.
- Multi-currency tracking and calculation of exchange rates are not implemented.
- Offline usage is not supported; an internet connection is mandatory.
- Advanced user roles or administrative features are beyond the scope.

CHAPTER 2

Literature Survey

2.1 Review relevant literature or previous work in this domain.

Traditional Methods: Expense tracking was historically done using physical ledgers or spreadsheets, which are prone to human error and lack real-time accessibility.
Mobile and Web Applications: Tools like Mint, Expensify, and YNAB (You Need A Budget) have simplified expense management by providing digital platforms. However, they often lack customizability for individual needs.
MERN Stack Applications: The MERN stack (MongoDB, Express.js, React.js, Node.js) has gained popularity for developing scalable and dynamic web applications, offering robust tools for real-time data handling and visualization.
Studies on Expense Tracking: Research highlights the importance of visualization and categorization in financial tools to enhance user engagement and decision-making.

2.2 Mention any existing models, techniques, or methodologies related to the problem.

Spreadsheets: Simple yet powerful for basic tracking, but they lack automation and user-friendliness for non-technical users.
Mobile Applications: Apps like Mint and PocketGuard offer automated expense tracking but require integration with bank accounts, raising security concerns.
Standalone Web Platforms: Tools like Wave and Tiller offer browser-based tracking but are often subscription-based and lack modularity for personal or specific user needs.
MERN-Based Solutions: The MERN stack is increasingly used for financial applications due to its ability to handle complex data flow, real-time updates, and user authentication.

2.3 Highlight the gaps or limitations in existing solutions and how your project will address them.

Lack of Customization: Existing tools often provide fixed features, which might not suit all users.
Solution: Our project allows users to define and manage their expense categories based on personal requirements.
Security Concerns: Many applications integrate with financial institutions, raising data privacy issues.
Solution: Our project ensures data security by isolating the system from external bank or financial institution APIs.
High Cost: Most advanced tools are subscription-based, limiting accessibility for all users.
Solution: Our solution is open-source, offering a free and accessible expense management tool.
Limited Real-Time Interactivity: Some platforms lack real-time data updates and visualization.
Solution: Using the MERN stack, we provide a responsive and dynamic system that supports real-time updates and analytics.

CHAPTER 3

Proposed Methodology

3.1 System Design

Overview:

The system is a web-based expense tracking application developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js). It features a modular design for scalability and ease of maintenance.

Key Components:

Frontend: React.js is used to build an interactive user interface for expense input, reports, and visualizations.
Backend: Node.js and Express.js handle API requests, authentication, and business logic.

Database: MongoDB is employed for storing user data, expenses, and categories in a structured, schema-less format.

3.2 Modules Used

Frontend

- React.js
- Material-UI (MUI)
- React Router

Backend

- Node.js
- Express.js

Database

- MongoDB

3.3 Data Flow Diagram

3.3.1 DFD Level 0

- The user logs into the system and inputs expense details through the interface.
- The system processes the expense data and stores it in MongoDB.
- Results, including visualizations and summaries, are displayed on the user’s dashboard, allowing interaction.

3.3.2 DFD Level 1 - Expense Management Module

- Users create, edit, or delete expense records categorized by type (e.g., food, transport).
- The system dynamically updates and retrieves data from MongoDB to ensure real-time updates on the user dashboard.
- The dashboard visualizes expenses using charts and graphs for better analysis.

3.3.3 DFD Level 1 - Notifications Module

- Users set reminders for bill payments or budget limits.
- Notifications are scheduled and sent in real-time to users through the interface or email.

This structure adapts the DFD to the specific functionality and workflow of the MERN Stack Expense Tracker project.

3.4 Advantages

- Efficient management of personal finances through seamless expense tracking.
- Users can categorize and track expenses based on their preferences, enhancing financial control.
- Real-time updates and dynamic visualizations help users analyze spending patterns.
- User-friendly interface built on the MERN stack, ensuring scalability and performance.
- Cross-platform accessibility with support for both desktop and mobile browsers.
- Secure user authentication and data management using JWT and MongoDB.

3.5 Requirement Specification

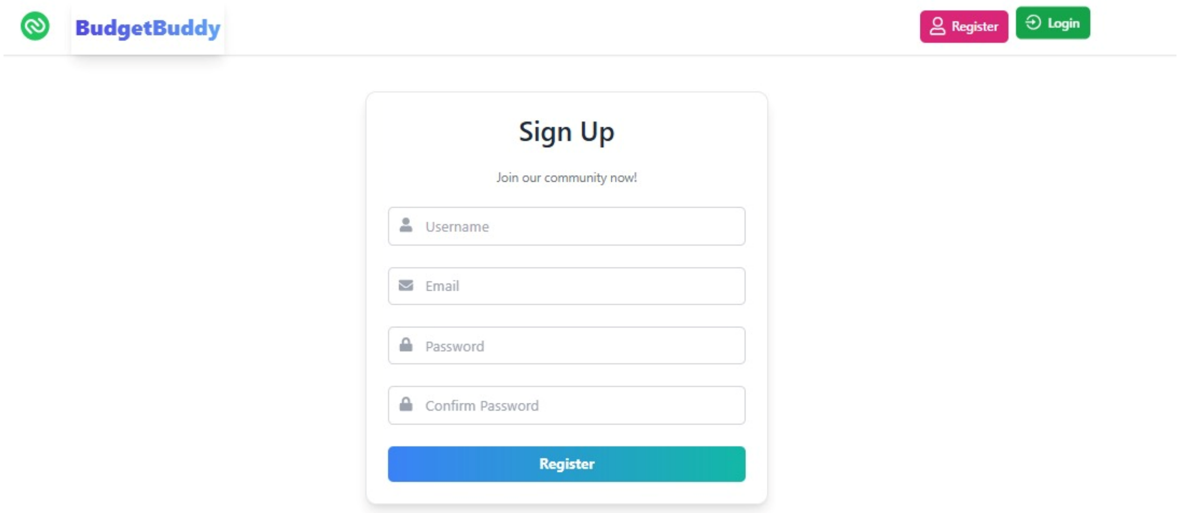
- Operating System: Windows 10/11, macOS, Linux.
- Languages & Frameworks: JavaScript (Node.js, React.js), Express.js, HTML, CSS.
- Database: MongoDB for storing user data and expense records.
- Tools: Visual Studio Code, Git, Postman for API testing.
- API Integration: None required for basic expense tracking, but Grok API (if applicable) for future recommendations or AI-based features.
- Browser: Latest Chrome, Firefox, or Edge for full functionality.

CHAPTER 4

Implementation and Result

Results :-

Sign up Page:-



Sign Up

Join our community now!

Username

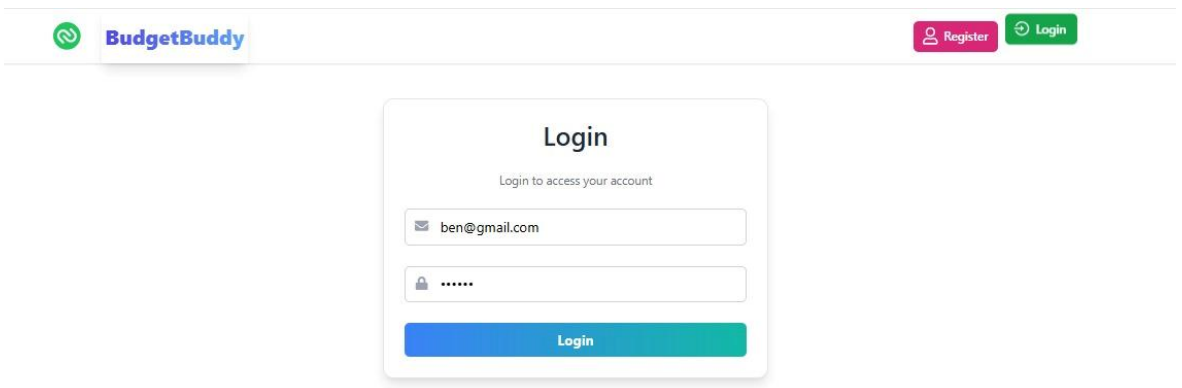
Email

Password

Confirm Password

Register

Login Page:-



Login

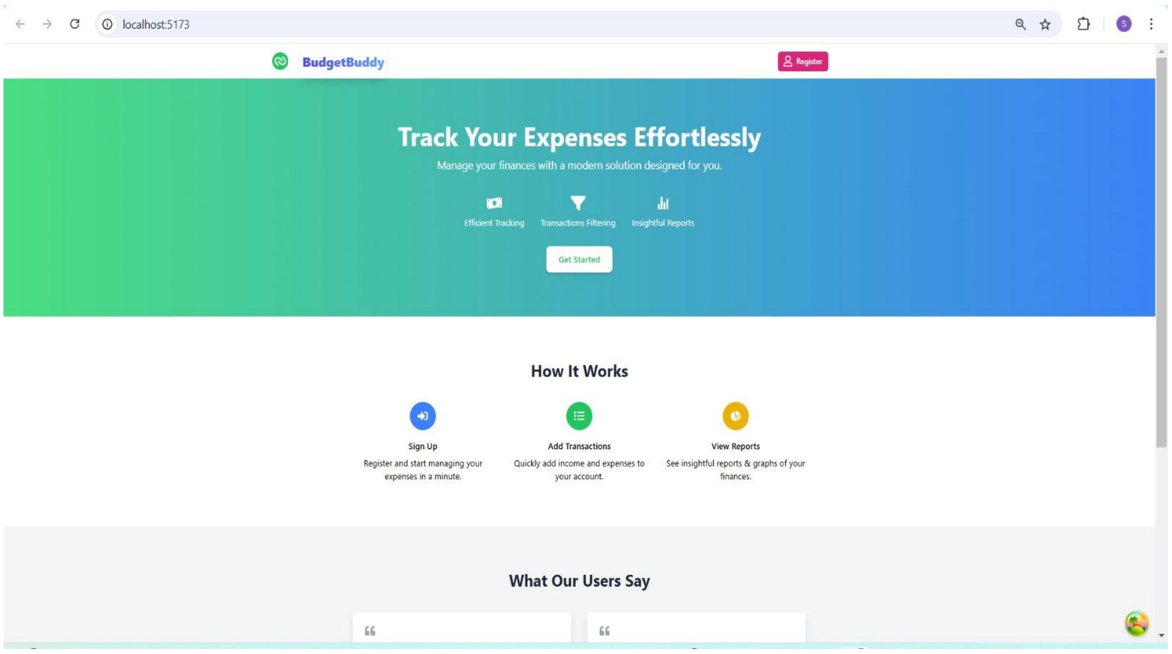
Login to access your account

ben@gmail.com

••••••

Login

Home Page :-



Track Your Expenses Effortlessly

Manage your finances with a modern solution designed for you.

How It Works

Sign Up

Register and start managing your expenses in a minute.

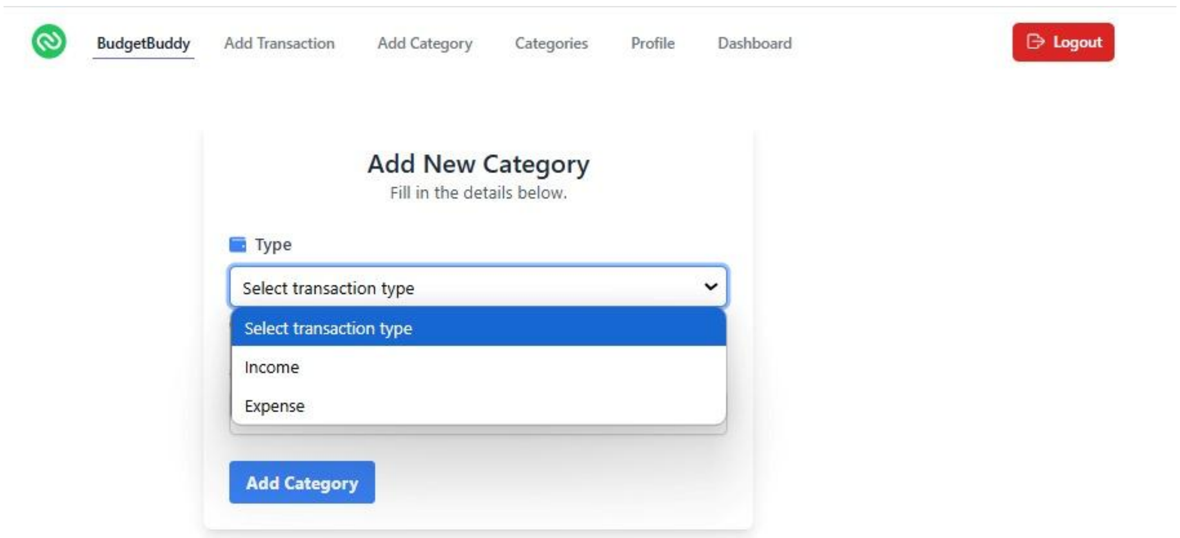
Add Transactions

Quickly add income and expenses to your account.

View Reports

See insightful reports & graphs of your finances.

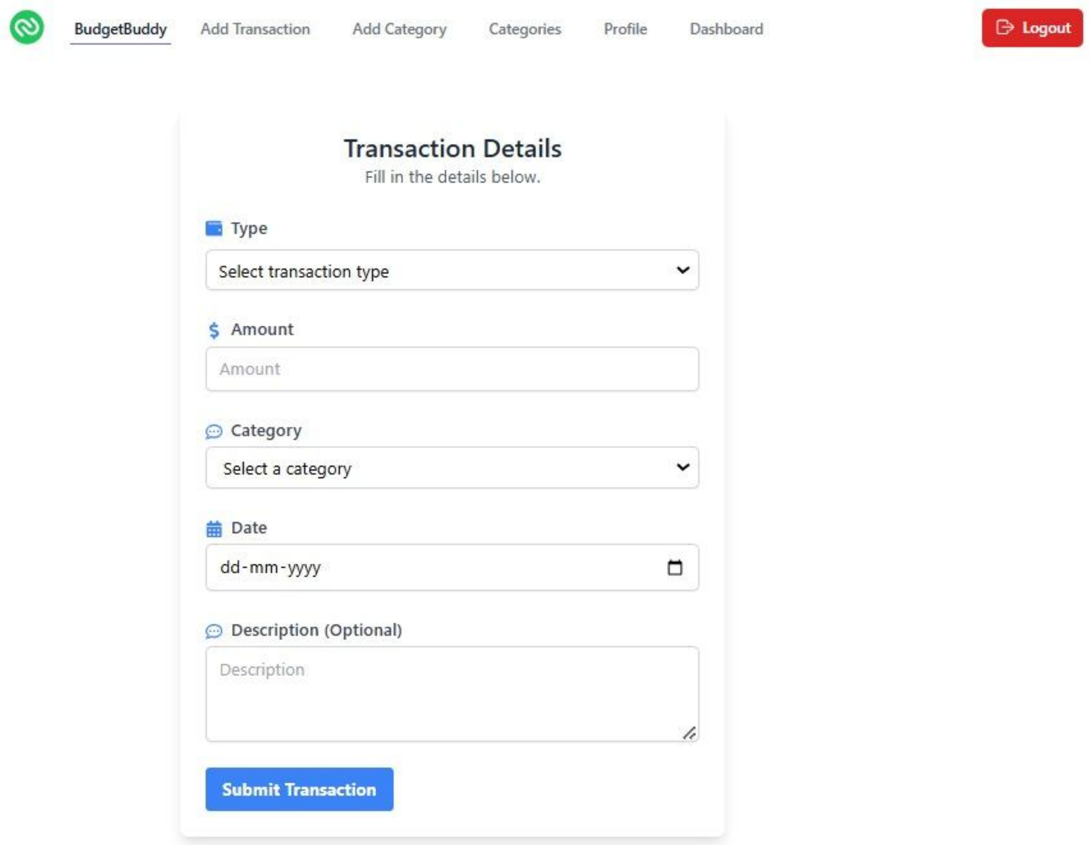
What Our Users Say



Add New Category

Fill in the details below.

Type



BudgetBuddy Add Transaction Add Category Categories Profile Dashboard Logout

Transaction Details

Fill in the details below.

Type

Select transaction type

Amount

Amount

Category

Select a category

Date

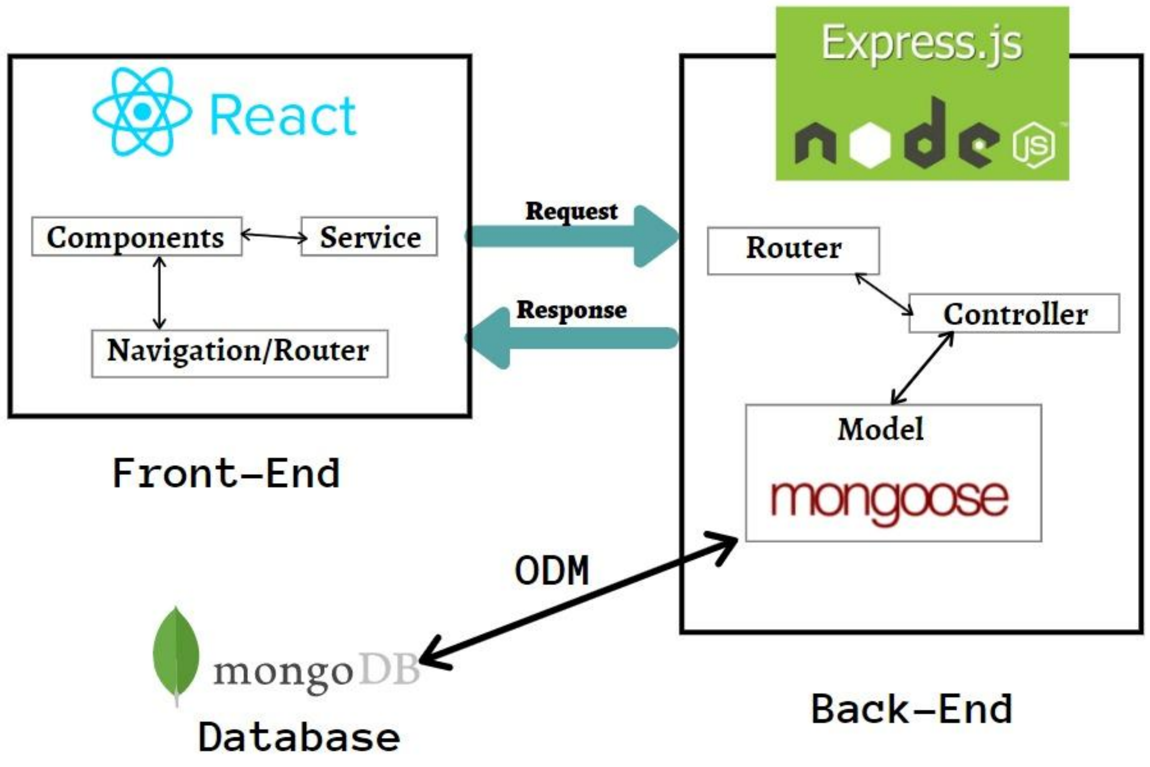
dd-mm-yyyy

Description (Optional)

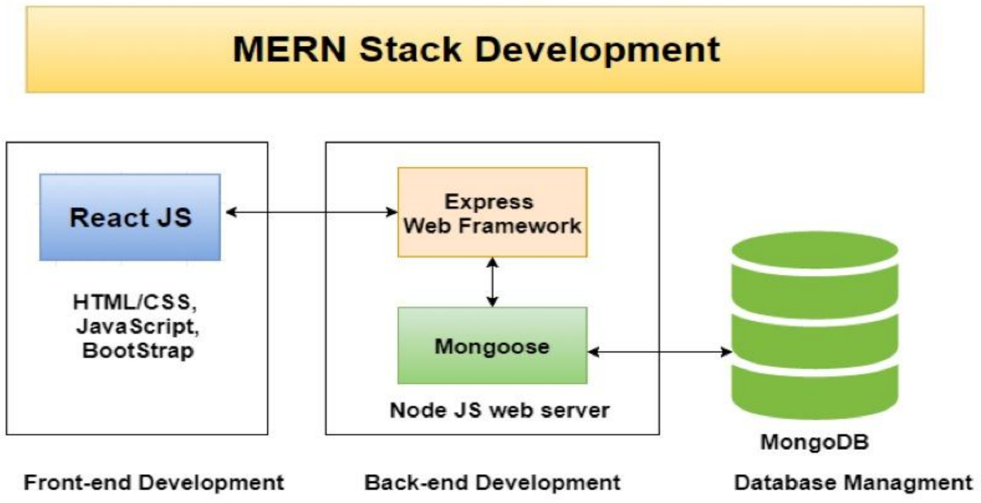
Description

Submit Transaction

Architecture Diagram :



System Architecture :



CHAPTER 5

Discussion and Conclusion

5.1 Key Findings:

1. Good Expense Management: The app has implemented CRUD operations such that users can add, update, view, and delete expenses with ease.
2. Real-Time Insights: Dynamic graphical visualizations assist users in tracking spending patterns to make informed financial decisions.
3. Scalable Data Storage: MongoDB is used to efficiently store user data and expenses scalable enough for future growth.
4. Easy to Use Interface: Front-end responsive using React.js allows for a seamless user experience across devices.

5.2 Git Hub Link of the Project:

https://github.com/Shree740/BudgetBuddy

5.3 Video Recording of Project :

https://drive.google.com/drive/folders/1TXF4laWEkTlPw97HLcTZ1wW5TtL37CLR

5.4 Limitations:

Here are the limitations of the "Expenses Tracker - MERN Stack Project" in 3-4 points:
1. Limited Budgeting Tools: The current version lacks features like budgeting and financial planning tools, which would help users manage their finances more effectively.
2. No Multi-user Support: The application does not support multiple users or shared accounts, limiting its use for household or group expense tracking.
3. Manual Input of Expensed: The users have to input their expenses manually, where the records might not be totally complete or consistent in data entries, which might affect how the financials are represented.
4. No Integration to Financial APIs: The app did not integrate with external platforms or APIs that connect their bank accounts or payment service to automate expense tracking.

5.5 Future Work:

Multi-user and Shared Accounts: Implement multi-user support for families or groups, allowing multiple users to share and track common expenses, enhancing collaboration and transparency.

Mobile Application Development: Develop a mobile app version of the platform to expand accessibility and improve user experience on smartphones, which is increasingly preferred for financial management.

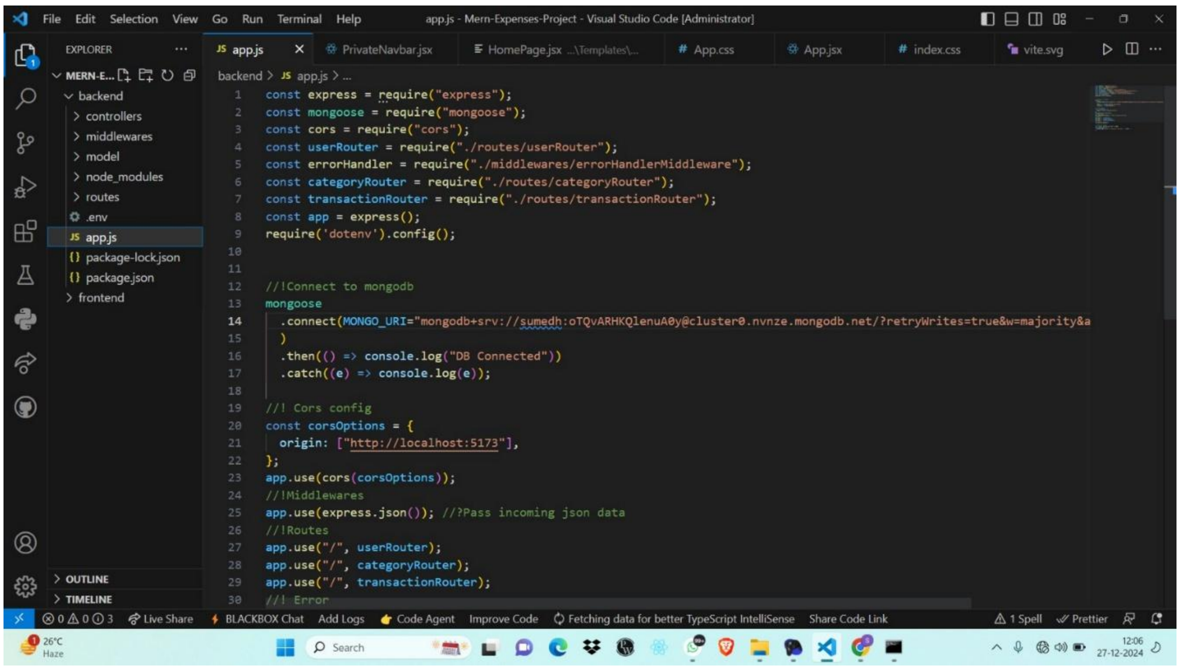
5.6 Conclusion:

The **Expenses Tracker - MERN Stack** Project simplifies personal finance management for users through an intuitive platform to track, categorize, and analyze expenses. Utilizing the MERN stack, the project ensures efficient data handling, secure user authentication, and a responsive interface. Key features such as real-time visualizations and seamless CRUD operations provide users with valuable insights into their financial habits, thus promoting informed decision-making. This project represents how full-stack development is empowered to solve real-world problems practically. Future enhancements with budgeting tools and API integration will allow the application's utility and scalability to become even greater. Overall, the project contributes to personal finance management becoming accessible and efficient for users.

REFERENCES

1. **Express.js Documentation**
   Express.js (n.d.). Express - Node.js Web Application Framework. Retrieved from https://expressjs.com
2. **React.js Documentation**
   React (n.d.). React – A JavaScript Library for Building User Interfaces. Retrieved from https://reactjs.org
3. **MongoDB Documentation**
   MongoDB, Inc. (n.d.). MongoDB Manual. Retrieved from https://docs.mongodb.com
4. **Node.js Documentation**
   Node.js (n.d.). Node.js Documentation. Retrieved from https://nodejs.org/en/docs/

Appendices (if applicable)



app.js - Mern-Expenses-Project - Visual Studio Code [Administrator]
backend > app.js > ...

```javascript
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
const userRouter = require("./routes/userRouter");
const errorHandler = require("./middlewares/errorHandlerMiddleware");
const categoryRouter = require("./routes/categoryRouter");
const transactionRouter = require("./routes/transactionRouter");
const app = express();
require('dotenv').config();


//!Connect to mongodb
mongoose
  .connect(MONGO_URI="mongodb+srv://sumedh:oTQvARHKQlenuA0y@cluster0.nvnze.mongodb.net/?retryWrites=true&w=majority&a
  )
  .then(() => console.log("DB Connected"))
  .catch((e) => console.log(e));

//! Cors config
const corsOptions = {
  origin: ["http://localhost:5173"],
};
app.use(cors(corsOptions));
//!Middlewares
app.use(express.json()); //?Pass incoming json data
//!Routes
app.use("/", userRouter);
app.use("/", categoryRouter);
app.use("/", transactionRouter);
//! Error
```