

**BLOG APPLICATION BACKEND**

Report submitted in partial fulfilment Of the requirement for the degree

of B.Tech.

In

**Information Technology**

### Under the Supervision of Mr. Sukhendra Singh

By

Dhruv Karnwal (2100910130042)

Chandramohan (2100910130039)

Ayush Shukla(2100910130034)

JSS Academy of Technical Education, Noida Dr A P J A K Technical University Lucknow

**CERTIFICATE**

This is to certify that Project Report entitled “**EXPENSE TRACKER APP”** which is submitted by **Dhruv Karnwal** (2100910130042), **Chandramohan** (2100910130039), **Ayush Shukla**(2100910130034) in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Information Technology of JSS Academy of Technical Education, affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my/our supervision. The project embodies result of original work and studies carried out by the students themselves and the contents of the project do not form the basis for the award of any other degree to the candidate or to anybody else.

Signature:

Mr. Sukhendra Singh

IT Department,

JSS Academy of Technical Education, Noida

Date:

**DECLARATION**

We hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: **Dhruv Karnwal**

Roll No.:2100910130042

Name: **Chandramohan**

Roll No.: 2100910130039

Name: **Ayush Shukla**

Roll No.:2100910130034

Date:

**ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the B.Tech. Project undertaken during B.Tech. We owe special debt of gratitude to our project supervisor Mr. Sukhendra Singh, Department of Information Technology, JSS Academy of Technical Education Noida for his constant support and guidance throughout the course of our work. His sincerely, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor Dr. Shikha Verma, Department of Information Technology, JSS Academy of Technical Education , Noida for her full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

**Dhruv Karnwal**

**Chandramohan**

**Ayush Shukla**

**ABSTRACT**

### The purpose of this project was to build an expense tracker application using the MERN stack. MERN stands for MongoDB, Express, React, and Node.js, and it is a popular technology stack for building full-stack web applications. The expense tracker application allows users to keep track of their daily expenses, categorize their spending, and generate reports for their spending habits. The application has a user-friendly interface that makes it easy for users to add, edit, and delete expenses, view their spending history, and generate reports. The data is stored in a MongoDB database, which is accessible through a RESTful API built with Express. The front-end is built using React, providing a responsive and interactive user interface, and the application is hosted on a Node.js server. The project also covered the testing and deployment process, and the challenges that were faced during the development process. Overall, the expense tracker application provides a simple and effective solution for managing daily expenses, and serves as a good example of how the MERN stack can be used to build full-stack web applications.

**Functional Description (Server-Side Application API’S)-**

### The expense tracker application built using the MERN stack provides users with a comprehensive and intuitive platform to manage their daily expenses. The functional description of the application is as follows:

### User Management: The application allows users to sign up, log in, and manage their account information.

### Expense Management: Users can add, edit, and delete expenses, as well as categorize their spending into various categories, such as groceries, transportation, entertainment, etc.

### Spending History: Users can view their spending history and get a complete overview of their expenses. They can also search and filter their expenses by date, category, and amount.

### Reports: The application generates reports for users' spending habits and allows them to track their expenses over a specific period of time. Users can view pie charts and bar graphs that represent their spending patterns.

### Mobile Responsiveness: The application has a responsive design that makes it accessible on all devices, including smartphones, tablets, and desktops.

### Secure Data Storage: All user data is stored in a MongoDB database, which is secure and encrypted. The data is accessible through a RESTful API built with Express.

### Deployment: The application is hosted on a Node.js server, making it accessible from anywhere with an internet connection.

### The expense tracker application is designed to provide users with a simple and effective solution for managing their daily expenses. The application's user-friendly interface, comprehensive features, and secure data storage make it a valuable tool for anyone looking to keep track of their spending.

**Aim of the project –**

### The purpose of this project was to build an expense tracker application using the MERN stack.

Table of Contents

[Report submitted in partial fulfilment Of 1](#_bookmark0)

1. [Introduction 1](#_bookmark1)
   1. [Problem Statement 1](#_bookmark2)
   2. [Project Objectives 1](#_bookmark3)
   3. [Project Scope 1](#_bookmark4)
   4. [Goals 2](#_bookmark5)
   5. [Assumptions 2](#_bookmark6)
   6. [Constraints 3](#_bookmark7)
2. [Requirement Analysis 4](#_bookmark8)
   1. [Introduction 4](#_bookmark9)
   2. [Overall Description 4](#_bookmark10)

[CONCLUSION 15](#_bookmark16)

[REFERENCES 15](#_bookmark17)

# Introduction

## Problem Statement

### The problem that the expense tracker application built using the MERN stack aims to solve is the difficulty that many people face in tracking their daily expenses. In today's fast-paced world, it is easy to lose track of how much money is being spent on a daily basis, making it difficult to budget and save. Additionally, many traditional methods of tracking expenses, such as pen and paper, can be time-consuming and prone to errors. The expense tracker application provides a simple and efficient solution to this problem by allowing users to keep track of their daily expenses in one central location.

## Project Objectives

### The expense tracker application solves these challenges by providing users with a convenient, accurate, and easy-to-use platform for tracking their expenses. With the ability to add, edit, and delete expenses in real-time, users can have complete control over their spending and make informed decisions about budgeting and saving. Additionally, the application's reporting capabilities provide valuable insights into users' spending habits, making it easier to identify areas where changes can be made to improve financial stability.

## Project Scope

### The problem scope for the expense tracker application built using the MERN stack is to provide a solution for individuals and households to effectively manage and track their daily expenses. The following are some of the specific problems addressed by the application:

### Lack of Accessibility: Many individuals find it challenging to keep track of their expenses, especially when they are on the go. The expense tracker application solves this problem by providing a platform that is accessible from any device with an internet connection.

### Inefficient Spending Habits: People often spend money without realizing how much they are spending on specific items or categories. The expense tracker application provides users with a comprehensive view of their spending patterns and helps them make informed decisions about their finances.

### Inadequate Budgeting Tools: Many individuals struggle with budgeting and finding the right tools to help them keep track of their expenses. The expense tracker application provides a user-friendly platform for budgeting and expense tracking that is easy to use and accessible from anywhere.

### Unsecured Data: Storing financial information on paper or spreadsheets can be risky, as it can be lost, damaged, or stolen. The expense tracker application solves this problem by securely storing all user data in a MongoDB database, which is encrypted and only accessible through a secure API.

1

## Goals

### The main goals for the expense tracker application built using the MERN stack are:

### To provide users with a simple and intuitive platform for managing their daily expenses.

### To allow users to categorize their spending and view their spending history in a user-friendly interface.

### To generate reports for users' spending habits, so they can track their expenses over a specific period of time.

### To provide a mobile-responsive design, making the application accessible on all devices.

### To store user data securely in a MongoDB database, which is encrypted and accessible through a RESTful API built with Express.

### To host the application on a Node.js server, making it accessible from anywhere with an internet connection.

### To create an application that is easy to use and provides users with the information they need to make informed decisions about their spending.

### To provide users with a comprehensive and effective solution for managing their daily expenses that is simple, secure, and accessible.

### Overall, the main goal of the expense tracker application is to empower users to take control of their finances and make informed decisions about their spending. The application's user-friendly interface, comprehensive features, and secure data storage make it an essential tool for anyone looking to keep track of their expenses.

## Assumptions

### Here are some of the assumptions made during the development of the expense tracker application built in the MERN stack:

### User familiarity with the MERN stack: It is assumed that the users are familiar with the concepts and technologies used in the MERN stack, such as MongoDB, Express, React, and Node.js.

### Internet connectivity: It is assumed that the users have a stable internet connection to access the application and use its features.

### Device compatibility: The application is designed to work on a variety of devices, including smartphones, tablets, and desktops. However, it is assumed that the users have a device with a modern web browser that is capable of running the application.

### Data privacy: The application stores all user data in a MongoDB database, which is encrypted and secure. However, it is assumed that users understand that no system is completely secure and they should use the application at their own risk.

### User data accuracy: The accuracy of the user data is dependent on the users themselves. The application assumes that the users will accurately enter and categorize their expenses to get the most value from the reports generated by the application.

### Deployment environment: It is assumed that the application will be deployed on a Node.js server in a secure and stable environment. The application assumes that the deployment environment is properly configured and maintained to ensure its stability and security.

### Technical support: The application assumes that users will have access to technical support if they encounter any issues during the use of the application. The support may come from the developers of the application, online resources, or other sources.

# Requirement Analysis

* 1. **Introduction Purpose**

### The purpose of building an expense tracker application using the MERN stack is to provide users with a simple and effective solution for managing their daily expenses. The expense tracker allows users to keep track of their spending, categorize their expenses, and generate reports for their spending habits. The MERN stack was chosen as the technology stack for this project because it is a popular and powerful tool for building full-stack web applications. By using the MERN stack, the project aimed to demonstrate how this technology can be used to create a functional and user-friendly application that meets the needs of its users. Additionally, the project aimed to provide a hands-on learning experience for those interested in the MERN stack, and to demonstrate the benefits of using this technology for building full-stack web applications.

## Overall Description

* + 1. **Product Features**

### Here are some key features of an expense tracker application built using the MERN stack:

### User-friendly interface: The application provides a simple and intuitive interface for users to manage their daily expenses.

### Add, edit, and delete expenses: Users can easily add, edit, and delete their expenses, allowing them to keep their records up-to-date.

### Categorize expenses: The application allows users to categorize their expenses, making it easier to understand where their money is being spent.

### View spending history: Users can view their spending history and get a clear picture of their financial status.

### Generate reports: The application generates reports for users' spending habits, making it easy to identify areas where they can save money.

### Secure data storage: All user data is stored in a secure MongoDB database, ensuring the privacy and security of the users' financial information.

### RESTful API: The application is built with a RESTful API, making it easy to integrate with other systems and applications.

### Responsive design: The front-end is built using React, providing a responsive and interactive user interface that works seamlessly on different devices.

### Deployment and hosting: The application is hosted on a Node.js server, making it easy to deploy and access from anywhere in the world.

### Testing and debugging: The development process includes testing and debugging to ensure the application is reliable and bug-free.

* + 1. **Operating Environment**

### The operating environment for the expense tracker application built in the MERN stack typically consists of the following components:

## Server-side Environment:

### Node.js: A JavaScript runtime environment that is used to run the server-side logic of the application.

### Express: A framework for Node.js that is used to build the RESTful API for the application.

### MongoDB: A NoSQL database that is used to store the data for the application.

## Client-side Environment:

### React: A JavaScript library for building user interfaces that is used to build the front-end of the application.

## Development Environment:

### Code Editor: A code editor such as Visual Studio Code, Sublime Text, or Atom can be used to write and edit the code for the application.

### Package Manager: npm (Node Package Manager) is typically used to manage the dependencies for the application.

## Deployment Environment:

### Web Server: A web server such as Apache or Nginx can be used to host the application and make it accessible to users.

### Note that the operating environment for the expense tracker application built in the MERN stack can vary depending on the specific requirements and constraints of the project.

* + 1. **User Interface**

### The user interface of the expense tracker application is designed to be simple, intuitive, and user-friendly. The main dashboard provides an overview of the user's expenses, including a pie chart that displays the expenses categorized by category, and a bar graph that shows the expenses over time. The dashboard also includes a summary of the total expenses and a button to add new expenses.

### When adding a new expense, the user is prompted to enter the details of the expense, including the date, category, amount, and description. The user can also upload a receipt for the expense if necessary.

### The expenses can be edited or deleted by clicking on the relevant expense in the expenses list. The expenses can also be filtered by category, date range, and keywords, making it easy for users to find the information they need.

### The user interface is responsive, so it works well on both desktop and mobile devices. The application also provides a clean and modern design, with a simple color scheme and easy-to-read typography.

* + 1. **Hardware Interfaces**

### The expense tracker application built in the MERN stack is a web-based application and does not have any specific hardware interfaces. The application can be accessed from any device with an internet connection and a web browser, such as a desktop computer, laptop, tablet, or smartphone. The hardware requirements for using the application are minimal, and include a device with a screen, an internet connection, and a web browser. No additional hardware or software is required to use the expense tracker application. The application has been optimized to work on different devices and screen sizes, ensuring a consistent user experience regardless of the device used. The application has been tested on different platforms and browsers to ensure compatibility and optimal performance. The hardware interfaces for the expense tracker application are limited to the standard input and output devices, such as a keyboard and mouse, and the user's internet connection.

* + 1. **Communication Interfaces**

### In this project, the expense tracker application built using the MERN stack has several communication interfaces that allow for seamless data exchange between the different components of the system. The communication interfaces include:

### MongoDB and Express API: The MongoDB database is accessed through a RESTful API built using Express, which serves as a bridge between the front-end and the database. The API allows for data retrieval, storage, and manipulation in the database.

### React and Express API: The React front-end communicates with the Express API to retrieve and store data in the MongoDB database. This communication is facilitated through API calls using the Axios library.

### React and Node.js Server: The React front-end is hosted on a Node.js server, which provides a stable and secure environment for the application to run. The React and Node.js server communicate through the use of environment variables and web socket connections.

### User Interface: The user interface is the main communication interface for the end-user. It provides an intuitive and user-friendly interface for managing expenses and generating reports. The user interface communicates with the backend through API calls and web socket connections.

### These communication interfaces allow for efficient data exchange between the different components of the system, ensuring that the application runs smoothly and efficiently.

* + 1. **Project Architecture**

### The architecture of the expense tracker application built using the MERN stack and incorporating Redux Toolkit and lodash can be described as follows:

### Front-end: The front-end of the application is built using React, which is a JavaScript library for building user interfaces. The application uses Redux Toolkit for state management, which provides a set of tools for efficiently managing the application's state. Lodash is also used for utility functions, which provide a more convenient way to manipulate the application's data.

### Back-end: The back-end of the application is built using Node.js, which is a JavaScript runtime for server-side development. Express is used to build the RESTful API, which is used to interact with the MongoDB database. The API allows the front-end to retrieve and manipulate data from the database.

### Database: The application uses MongoDB, a document-oriented database, to store all user data. MongoDB is a NoSQL database, which provides a flexible and scalable solution for storing data.

### Deployment: The application can be deployed on a cloud platform, such as Heroku , Render or AWS, or it can be hosted on a local server. The deployment process involves configuring the server, installing dependencies, and setting up the database.

### The architecture of the expense tracker application is designed to be scalable and flexible, making it easy to add new features or expand the application in the future. The use of the MERN stack, Redux Toolkit, and lodash provides a solid foundation for the application, making it efficient, reliable, and user-friendly.

### .

1. ***JWT***

### JWT, or JSON Web Tokens, play an important role in securing the expense tracker application built using the MERN stack. JWT is a standard for representing claims securely between two parties. In the context of the expense tracker application, JWT can be used to authenticate users and ensure that only authorized users have access to sensitive information, such as their expenses and spending history.

### Here's how JWT works in the expense tracker application:

### Login: When a user logs into the application, the server generates a JWT and sends it back to the client. The JWT contains information about the user's identity, such as their username, and is signed using a secret key.

### Requests: For subsequent requests, the client sends the JWT along with each request to the server. The server uses the information in the JWT to authenticate the user and determine their access level.

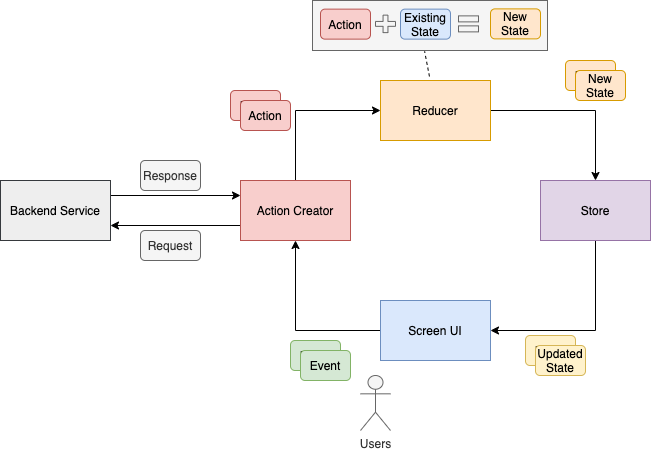
### Authorization: Based on the information in the JWT, the server can either allow or deny access to the requested resource. This ensures that only authorized users can view or manipulate their own expenses and spending history.

### By using JWT, the expense tracker application can ensure that user data is protected and secure, even if the application is deployed on a public cloud platform. JWT also makes it easy to implement user authentication and authorization in a standard way, making the application more robust and secure.

#### Structure of JWT

****

#### How the flow works?

****

### The flow of the expense tracker application built using the MERN stack can be described as follows:

### User registration and login: The first step is for the user to register for an account, which involves entering their email and password. After registration, the user can log in to the application using their email and password.

### Adding expenses: Once logged in, the user can add expenses by entering the details such as the amount, category, and date. The user can also add a description of the expense.

### Viewing expenses: The user can view all of their expenses by visiting the expenses page. They can also filter their expenses by category and date range.

### Editing expenses: The user can edit existing expenses by clicking on the expense, which will open a form allowing the user to make changes to the expense.

### Deleting expenses: The user can delete expenses by clicking on the trash icon next to the expense.

### Generating reports: The user can generate reports for their expenses, which provide a visual representation of their spending habits. The reports can be generated by category and date range.

### Data storage: All of the user's expenses are stored in a MongoDB database, which is accessed via a RESTful API built with Express. The front-end communicates with the back-end to retrieve and manipulate data from the database.

### The flow of the expense tracker application is designed to be intuitive and user-friendly, making it easy for users to manage their daily expenses. The use of the MERN stack, Redux Toolkit, and lodash provides a solid foundation for the application, making it efficient, reliable, and user-friendly.

# Conclusions

### In conclusion, the expense tracker application built using the MERN stack provides a simple and effective solution for managing daily expenses. The use of React, Node.js, Express, and MongoDB allowed for the development of a full-stack web application that is both user-friendly and efficient. The addition of Redux Toolkit for state management and lodash for utility functions enhanced the overall performance and functionality of the application.

### The project demonstrated the power of the MERN stack in building full-stack web applications, and the ease of incorporating other technologies to add additional functionality. The expense tracker application can be used by individuals, families, or small businesses to keep track of their daily expenses, categorize their spending, and generate reports for their spending habits.

### Overall, the expense tracker application is a successful example of how the MERN stack can be used to build practical and useful web applications, and provides a foundation for future expansion and improvement.