

Document Information

Analyzed document	BT2177.docx (D169725352)
Submitted	2023-06-05 09:22:00
Submitted by	Janarthanan.S
Submitter email	janarthanan@galgotiasuniversity.edu.in
Similarity	0%
Analysis address	janarthanan.galgot@analysis.arkund.com

Sources included in the report

Entire Document

A Project Report on EXPENSE TRACKER APP

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

Title Page No. Candidates Declaration I Acknowledgement II Abstract III Functional Description IV Chapter 1 Introduction 1

1.1 Introduction 2

1.2 Formulation of Problem 3

1.2.1 Tool and Technology Used Chapter 2 Literature Survey/Project Design 5

Chapter 3 Functionality/Working of Project 9

Chapter 4 Results and Discussion 11

Chapter 5 Conclusion and Future Scope 41

5.1 Conclusion 41

5.2 Future Scope 42

Reference 43

Publication/Copyright/Product 45

CHAPTER-1 INTRODUCTION

1.1 Introduction:

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.

1.2 Formulation of problem:

The problem is to develop an expense tracker app that allows users to effectively manage their finances by tracking and categorizing their expenses, setting budgets, and gaining insights into their spending habits. The app should provide a user-friendly interface, secure data storage, and functionalities for expense tracking, budget management, data visualization, and integration with financial institutions.

Key Objectives:

Expense Tracking: The app should allow users to record their expenses, including the amount, date, category, and description of each transaction. It should provide options for manual entry or automatic import of transaction data from financial institutions.

Budget Management: Users should be able to set budgets for different expense categories and track their spending against these budgets. The app should provide alerts or notifications when users are nearing or exceeding their budget limits. Data Visualization and Analysis: The app should present users with visual representations, such as charts and graphs, to help them understand and analyze their spending patterns. It should generate reports and provide insights into their financial habits. Integration with Financial Institutions: The app should allow users to securely connect their bank or credit card accounts, importing transaction data and categorizing expenses automatically. It should ensure the privacy and security of users' financial information. User Interface and Experience: The app should have an intuitive and user-friendly interface, making it easy for users to navigate, record expenses, set budgets, and view reports. It should prioritize usability, responsiveness, and a visually appealing design. Data Storage and Security: The app should implement secure data storage mechanisms to protect users' personal and financial information. It should follow industry-standard security practices and comply with relevant privacy regulations. User Engagement and Motivation: To encourage regular usage, the app should incorporate features that keep users engaged and motivated. This may include personalized insights, goal setting, gamification elements, or reminders for expense tracking and budget adherence.

1.2.1 Tool and Technology Used:

The tools and technology 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

CHAPTER :2

2.1 Literature Survey/Project Design:

we have come across all the finance manageable apps in the market available, specifically we are going to target those people, who need more specific and user friendly interface. we let the people choose their specification that is more flexible than all the apps available. we will try to put some intelligence in our app. we will try to accommodate in minimum size of app as that possible in the market. Cost: While there are free expense tracker apps available, some may require a subscription fee or in-app purchases to access all features. Limited functionality: Some expense tracker apps may not have all the features users need, which can lead to frustration and the need to use multiple apps.

Existing apps like: 1. Money Manager Expense & Budget 2. Wallet – Daily Budget & Profit 3. Buddy: Budget & Save Money 4. Spending Tracker 5. Rydoo All have some specialties but every app like these falls back in either open sourcing or functionality. so we are going to make an app defines the optimality between these features and open source free software

CHAPTER: 3

3.1 Functionality/Working of Project:

IMPLEMENTATION:

Create wireframes and design mockups to visualize the apps layout, screens and navigation flow. Focus on creating an intuitive and user-friendly interface with clear labeling and easy access to essential features. functions of an expense tracker is helping you to maintain a well-organized and consolidated record of different expenses. For managing transaction records and arranging them nearly through the tracker app, you need to take images and screenshots of all receipts and keep them saved through the app. The receipts and records being saved in the cloud allow you to retrieve and check them whenever needed.

Fig 1. Overall workflow

3.1 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.

3.2 Client-side Environment:

- React: A JavaScript library for building user interfaces that is used to build the front-end of the application.[2]

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.

3.3 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.[3]

3.4 Backend Development:

- Develop the backend infrastructure to support data storage, user authentication, and integration with financial institutions (if required).
- Implement APIs to fetch and update data, ensuring secure transmission and storage of sensitive information.

3.5 Frontend Development:

- Implement the user interface based on the design mockups using the chosen development platform.
- Develop screens and functionalities for expense tracking, budget management, data analysis, and other features identified in the requirements.

3.6 Data Visualization and Analysis:

- Create visual representations, such as charts and graphs, to present users with a clear overview of their spending patterns and financial health.
- Implement data analysis features to generate reports, identify trends, and provide insights into user spending habits.

3.7 Testing and Quality Assurance:

- Conduct thorough testing to ensure the app functions correctly, is stable, and performs well across different devices and platforms.
- Test various scenarios, including different expense types, budgets, and user interactions, to identify and fix any bugs or issues.[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.

Fig. 2 Deployment model CHAPTER 4

RESULTS AND DISCUSSION:

Expense tracker apps offer a range of benefits to users, including improved financial management, budgeting capabilities, and data visualization. They provide functionalities for expense tracking, budget management, and data analysis. However, challenges exist in terms of user experience, usability, privacy, and maintaining user engagement. Future research could focus on addressing these challenges and further enhancing the effectiveness and user experience of expense tracker apps. 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.

Hit and source - focused comparison, Side by Side

Submitted text	As student entered the text in the submitted document.
Matching text	As the text appears in the source.