

### PROJECT REPORT

**On**

# Expense Tracker

Submitted in partial fulfilment of the requirement for the Course BEE (22CS026) of

**COMPUTER SCIENCE AND ENGINEERING**

#### B.E. Batch-2022

**in**

**Jan -2025**



**Under the Guidance of**

Mrs. Aditi Sharma

**Submitted By:**Piyush Gaur (2210992042)

Piyush Bhardwaj (2210992041)

Pranjal (2210992064)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# CHITKARA UNIVERSITY

**PUNJAB**



**Sign. of Project Guide: Mrs. Aditi Sharma**

**(Mentor)**

This is to be certified that the project entitled “ Expense Tracker App” has been submitted for the Bachelor of Computer Science Engineering at Chitkara University, Punjab during the academic semester January 2024- May-2024 is a bonafide piece of project work carried out by **PiyushGaur(2210992042), PiyushBhardwaj(2210992041) and Pranjal(2210992064)** towards the partial fulfillment for the award of the course Integrated Project (CS 203) under the guidance of **Mrs. Aditi Sharma** and supervision.

### CERTIFICATE



We, **PiyushGaur(2210992042), PiyushBhardwaj(2210992041) and Pranjal(2210992064)** B.E.-2021 of the Chitkara University, Punjab hereby declare that the Integrated Project Report entitled **“Expense Tracker”** is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other course.

**CANDIDATE’S DECLARATION**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  | 2210991659 |

|  |  |  |
| --- | --- | --- |
| **Sign. of Student** |  |  |
| Pranjal  **2210992064** |  |  |
| **2210991659** | 2210991610 | 2210991659 |

|  |  |  |
| --- | --- | --- |
| **Sign. of Student** |  |  |
| Piyush Bhardwaj  **2210992041** |  |  |
| **2210992041** | 2210991610 | 2210991659 |

|  |  |  |
| --- | --- | --- |
| **Sign. of Student** |  |  |
| Piyush Gaur  **2210992042** |  |  |
| 2210991609 | 2210991610 | 2210991659 |



### ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior and acts during the course of study.

We express our sincere gratitude to all for providing me an opportunity to undergo Integrated Project as the part of the curriculum.

We are thankful to Mrs. Aditi Sharma for her support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We also extend our sincere appreciation to Mrs. Aditi Sharma who provided his valuable suggestions and precious time in accomplishing our Integrated project report.

Lastly, We would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to day experience and received lots of suggestions that improve our quality of work.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  | 2210991610 | 2210991659 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Pranjal  **2210992064** |  |  |
| **2210991659** | 2210991610 | 2210991659 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Piyush Bhardwaj  **2210992041** |  |  |
| **2210991610** | 2210991610 | 2210991659 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Piyush Gaur  **2210992042** |  |  |
| 2210991609 | 2210991610 | 2210991659 |

### INDEX

1. **Abstract/Keywords 1**
2. **Introduction to the project 2**
   1. **Background**
   2. **Problem Statement**
3. **Software and Hardware Requirement Specification 3**
   1. **Methods**
   2. **Programming/Working Environment**
   3. **Requirements to run the application**
4. **Database Analyzing, design and implementation 4**
5. **Code-Implementation and Database Connections 5**
6. **Future Scope 6**
7. **Conclusion 7**
8. **Bibliography/References 8**

### ABSTRACT

Expense Tracker is a user-friendly web application designed to help individuals manage their finances effectively by tracking income and expenses. This report outlines the development process of the Expense Tracker, highlighting its research, design, and implementation phases.The application is built using modern web technologies, including HTML, CSS, and JavaScript, ensuring a responsive and interactive user experience. Users can easily add new transactions by entering details such as the transaction name and amount. The interface provides real-time updates on balances, allowing users to visualize their financial health effectively.Key features of the Expense Tracker include:

* **Transaction Management:** Users can add or remove transactions effortlessly.
* **Dynamic Updates:** The application displays real-time updates of total income and expenses.
* **Data Storage:** Transactions are stored locally to maintain data persistence across sessions.

The report includes a detailed analysis of the software requirements and system architecture, ensuring that the application can handle multiple users without performance issues. It also covers essential implementation details such as secure data handling and user interface design.System testing was conducted to validate the application's performance and usability. While the Expense Tracker demonstrates strong functionality in managing personal finances, it acknowledges limitations such as the lack of offline capabilities and potential latency in data processing.In conclusion, the report discusses future enhancements for the Expense Tracker, including features like offline access and integration with accounting software. By showcasing that effective financial management can be achieved through a secure and intuitive platform, Expense Tracker stands as a valuable tool for users looking to gain control over their spending habits.

**Introduction to the project**

The Expense Tracker project aims to create a dynamic and user-friendly web application that helps users manage their finances effectively by tracking income and expenses. Utilizing the MERN stack (MongoDB, Express.js, React.js, Node.js), the platform allows users to easily log, categorize, and visualize their financial activities while offering essential features such as secure user authentication, personalized profiles, and a well-structured database for transactions. The focus on responsive design ensures accessibility across various devices, from desktops to mobile phones.

**Background**

The concept for the Expense Tracker project emerged from the increasing need for convenient financial management tools in a fast-paced digital world. As personal finance management becomes more critical, providing users with a platform that combines ease of use with secure transactions has become a priority. By leveraging the MERN stack, the project aims to deliver a comprehensive solution that integrates backend robustness with front-end interactivity.

**Problem Statement**

The current landscape of personal finance applications faces several challenges that hinder user satisfaction and overall experience:

* **Lack of Responsive Design:** Many existing platforms do not provide a satisfactory experience on non-desktop devices, leading to frustration among users.
* **Inadequate User Authentication:** Security concerns arise from insufficient user authentication systems, necessitating seamless sign-up and login functionalities.
* **Disorganized Transaction Databases:** Users often struggle to find and categorize their expenses due to poorly structured databases.
* **Absence of Personalized Profiles:** Most platforms lack features that allow users to track favourite categories or view their transaction history conveniently.

**Software and Hardware Requirement Specification**

**Software Requirements:**

* **Operating System**: Windows, macOS, or Linux.
* **Code Editor**: Visual Studio Code (VS Code).
* **Backend**: Node.js, Express.js.
* **Frontend**: React.js, HTML, CSS, JavaScript.
* **Database**: MongoDB, Mongoose.
* **Version Control**: Git.
* **Browser**: Chrome, Firefox, or any modern browser.

**Hardware Requirements:**

* **Processor**: 2 GHz or faster.
* **RAM**: Minimum 4 GB.
* **Storage**: 1 GB of free space.
* **Network**: Stable internet connection.

**Methods**

* **Agile Development**: Iterative cycles for flexibility and user feedback.
* **Modular Design**: Separation of frontend, backend, and database for easier maintenance.
* **API-Driven Development**: REST APIs for client-server communication.

**Programming/Working Environment**

* **VS Code**: Main environment for writing and managing code.
* **Terminal**: For running Node.js and React.js commands.
* **Browser DevTools**: For inspecting and debugging frontend code.

**Requirements to Run the Application**

* Install **Node.js** and use **npm** for dependencies.
* Set up **MongoDB** for database management.
* Run the backend with npm run start and the frontend with npm start in VS Code.

**Database Analysing, Design, and Implementation for Expense Tracker**

**Database Analysing**

1. **Data Requirements:**  
   The Expense Tracker identifies key data entities essential for financial management, including:

* **Users:** Information about individuals using the application.
* **Expenses:** Details of each expense recorded by users.
* **Categories:** Classifications for expenses.
* **Budgets:** User-defined spending limits for categories.

1. **Relationships:**  
   Understanding the relationships is crucial:

* A **user** can log multiple **expenses**.
* Each **expense** belongs to a specific **category**.
* Users can set various **budgets** for their categories.

**Database Design**

1. **Database Model:**  
   The Expense Tracker utilizes MongoDB, a NoSQL database that stores data in a flexible, JSON-like format.

* **Users Collection:** Stores user details such as username, email, hashed password, and profile information.
* **Expenses Collection:** Records each expense with fields like description, amount, category reference, and date.
* **Categories Collection:** Manages expense categories linked to users.
* **Budgets Collection:** Tracks user-defined budgets per category.

**Database Implementation**

1. **Mongoose Integration:**  
   Mongoose ORM is used to manage MongoDB collections effectively.

* Create models for each entity (User, Expense, Category, Budget) to ensure efficient interactions.
* Implement data validation and indexing for fast queries.

1. **CRUD Operations:**

* **Create:** Users can create accounts and add expenses or categories.
* **Read:** Users can retrieve expense records and view budgets.
* **Update:** Users can edit profiles and modify expenses or budgets.
* **Delete:** Users can remove expenses or categories as needed.

This structured approach to database design ensures that the Expense Tracker provides users with an efficient tool for managing their finances while maintaining security and performance.ng security and performance.

**Code-Implementation and Database Connections**

**Code Implementation:**

* **Frontend Development:** The frontend is implemented using React.js, where core components such as Navbar, User Profile, Expense Tracking are developed as separate reusable components.
* **Styling:** CSS and Tailwind CSS are utilized for styling the application and creating a responsive layout that adjusts ensuring a visually appealing experience across all devices.
* **Backend Development:** Node.js with Express handles server-side logic. API routes for essential features are defined in the Express.js routes, allowing for organized and efficient handling of requests.
* **User Authentication:** JWT (JSON Web Token) is employed for user authentication, ensuring secure logins and maintaining user sessions. This method protects sensitive user information and secures the application's endpoints.
* **Database Connection:** MongoDB is integrated into the application using Mongoose, an Object Data Modeling (ODM) library for Node.js. Mongoose models are created for key entities, including Users, Books, Orders, and Reviews, defining schemas that specify how data is structured and stored in the database.
* **CRUD Operations:** Comprehensive CRUD (Create, Read, Update, Delete) operations are implemented for managing user-generated content, such as adding books, placing orders, writing reviews, and updating user profiles. These operations are essential for dynamic content management, ensuring that users can interact with the platform effectively.

**Future Scope**

**Scalability Enhancements:**

* **Database Optimization:** Implement advanced database optimization techniques, such as sharding in MongoDB, to effectively manage larger datasets as the user base expands.
* **Performance Improvements:** Introduce load balancing and server clustering to enhance the platform's performance and ensure a smooth user experience during peak traffic times.

**Advanced Security Features:**

* **User Protection:** Add multi-factor authentication (MFA) and role-based access control (RBAC) to bolster user account security and ensure that sensitive data is accessible only to authorized users.
* **Privacy Assurance:** Implement end-to-end encryption for private messaging, guaranteeing user privacy and secure communication within the platform.

**Mobile Application Development:**

* **Native App Development:** Expand the House of Books to mobile platforms by developing native applications for iOS and Android using technologies like React Native, allowing users to shop on the go.

**Real-Time Communication:**

* **Instant Notifications:** Enhance real-time features by integrating WebSockets for instant notifications, live chat, and video streaming, providing users with immediate updates and interaction options.
* **Collaborative Features:** Expand real-time capabilities to include group chats, events, and collaborative features, fostering a community atmosphere within the platform.

**Integration with Third-Party APIs:**

* **Cross-Platform Connectivity:** Allow users to connect their House of Books accounts with other platforms such as Instagram, Twitter, or Facebook for a more integrated online shopping experience.

**Conclusion**

The expense tracker website exemplifies the transformative potential of contemporary web technologies in delivering a comprehensive financial management solution. By utilizing React.js for a responsive and engaging user interface, alongside Node.js and Express.js for a resilient backend infrastructure, and MongoDB for efficient data storage, the platform ensures users can effortlessly track their expenses.

Key functionalities such as secure user authentication, customizable budget categories, and insightful analytics effectively address common challenges faced by individuals managing their finances. The platform is designed with mobile responsiveness in mind, enabling users to monitor their spending habits seamlessly across various devices.

Moreover, by adopting best practices for data security and employing scalable technologies, the expense tracker is well-equipped to handle an expanding user base while maintaining high performance. Future enhancements could include features like automated expense categorization through machine learning, integration with banking APIs for real-time tracking, and advanced reporting tools.

This positions the expense tracker to evolve into a vital resource for users seeking to gain control over their financial health, ultimately enhancing their overall experience in personal finance management.

**Bibliography/References**

* Mead, A. (2018). Full-Stack React Projects. Packt Publishing.
* Murray, M. P., et al. (2020). Pro MERN Stack. Apress.
* Casciaro, M., & Mammino, L. (2020). Node.js Design Patterns (3rd ed.). Packt Publishing.
* Mead, A. (2018). Learning Node.js Development. Packt Publishing.
* Wieruch, R. (2020). The Road to React (2nd ed.). Robin Wieruch.
* Node.js, MongoDB and Angular Web Development". Author: Brad Dayley.(2017).
* Dayley, B. (2017). Node.js, MongoDB and Angular Web Development. Addison- Wesley.