WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

EXPENSE TRACKING APPLICATION USING MERN

Sakshi Tapkir[1], Aishwarya Pathak[2]

Department of Computer Science/ IT

ABSTRACT

Expense Tracker offers users a convenient digital solution for tracking their expenses on their devices, eliminating the need for traditional paper diaries. Users can categorize their income and expenses across a wide range, including but not limited to clothing, food, entertainment such as movies, investments, rent, salary, business expenses, and beyond.

Keywords: Budget Management, Financial Tracking, Personal Finance, Expense Monitoring, Financial Planning

Introduction –

In a world that is progressively digital and data-oriented, the demand for efficient personal financial management solutions is on the rise. Our project is designed to address precisely these needs. The necessity of having an intuitive and secure platform for financial analysis, planning, and spending tracking is underscored. The project's technological architecture is elaborated upon, emphasizing the utilization of Express.js and Node.js for constructing dependable storage of financial data. MongoDB is selected as the database system to manage the data efficiently.

Technologies used –

1.1. MongoDB -

MongoDB is a NoSQL database. It designed for storing and managing unstructured data, making it highly flexible and scalable for various types of applications. It excels in scalability, performance, and adaptability. It's perfect for applications with dynamic data models, storing data in JSON documents with easily customizable schemas. MongoDB Atlas provides users with a convenient GUI for data access, eliminating the need for application logins. In addition to auto-generated IDs, MongoDB offers a rich set of query and manipulation methods like "find" and "updateOne". Its user-friendly approach, coupled with a strong community, makes it particularly appealing for newcomers to database development.

1.2. Express.js -

Express.js is a lightweight and flexible web framework for Node.js, perfect for building web applications and APIs. It simplifies the process of handling HTTP requests, routing, middleware integration, and more, allowing developers to focus on building features rather than dealing with low-level details. With its minimalist design and powerful capabilities, Express.js is widely used in the Node.js ecosystem for building scalable and efficient web applications.

React.js

React.js, also known as React, is a JavaScript library primarily used for constructing user interfaces, especially in scenarios with frequent UI updates. Developed and maintained by Facebook, it allows developers to create reusable UI components that manage their own state, making it easier to build complex UIs.React.js excels in expense tracker apps because of its component-based structure, clear syntax, and adept state handling. It capitalizes on a Virtual DOM to enhance speed and boasts a vast ecosystem supported by an active community. This synergy enables the creation of adaptable, well-organized, and swift applications customized to unique requirements.

The explanation is as follows:

a) **Component-Based Architecture-** In an expense tracker app, various elements like input forms, expense lists, and charts can each be encapsulated within separate React components.

b) **Declarative Syntax–**React uses a declarative syntax to describe how the UI should look based on the application's state. In simpler terms, React refreshes only the parts of the app that change when a user adds or updates an expense, making the user experience

smoother.

c) **State Management-**In expense tracker apps where data fluctuates frequently, React empowers components to handle their own state, ensuring each component can manage pertinent information like expense lists or filter settings independently. This approach streamlines the management of application state, guaranteeing seamless synchronization between the UI and underlying data.

d) **Virtual DOM-** React's Virtual DOM is a lightweight representation of the actual DOM. When the state of a React component changes, React compares the virtual DOM with the previous version to determine the minimal set of DOM updates needed. This optimization leads to better performance, especially in applications with dynamic data like expense trackers.

e) **Rich Ecosystem –**

The term "React.js rich ecosystem" denotes the broad array of libraries, tools, and resources at the disposal of React.js developers. This ecosystem encompasses diverse technologies that augment development processes, cater to specific requirements, and expand the capabilities of React-based applications. It encompasses well-known libraries such as Redux for state management, React Router for routing, and Axios for handling HTTP requests. Moreover, it encompasses various UI component libraries, development aids, testing frameworks, and community-contributed resources, all of which collectively enhance the React.js development experience. This multifaceted ecosystem offers developers versatility, productivity, and assistance, empowering them to create robust, feature-rich applications with greater efficacy.

f) **Community Support -**Community support for React.js expense tracker apps come in various forms:

1. **Online Forums and Communities**: Platforms like Stack Overflow, Reddit's r/reactjs community, and GitHub discussions offer spaces for developers to exchange ideas and get help with React.js app development.
2. **Documentation and Tutorials**: The React.js community provides extensive documentation and tutorials covering everything from basic concepts to advanced techniques, making it easier for developers to learn and implement features in their expense tracker apps.
3. **Open-Source Projects**: Developers can find valuable resources and reusable components in open-source React.js projects on platforms like GitHub, speeding up the development process and ensuring best practices are followed.
4. **Meetups and Events**: Local or virtual React.js meetups, conferences, and workshops enable developers to connect with peers, share experiences, and gain insights into building effective expense tracker apps.
5. **Community-contributed Libraries**: The React.js ecosystem offers a wide range of community-contributed libraries and components tailored for expense tracking, such as date pickers and charting tools, helping developers enhance their apps with minimal effort.
6. **Social Media Engagement**: Platforms like Twitter, LinkedIn, and Discord provide channels for developers to interact with fellow React.js enthusiasts, exchange tips, and find support for their expense tracker projects.

React.js syntax is generally considered accessible for developers of various skill levels due to its simplicity and consistency.

Node.js –

It is a runtime environment that's used to run JavaScript outside the browser. As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. You can think of it like a place where you can run JavaScript without inserting the code into HTML. Benefits of Node is With Node.js, businesses are able to develop a prototype, MVP or a final product faster than with other programming languages, and do so while cutting development costs. Its event-driven nature, microservices architecture, and MongoDB support allows businesses to grow and scale their products easily. Due to the event-driven, non-blocking I/O mechanism on which Node.js is based, it can effectively handle numerous concurrent requests.

Advantages-

Users can also expect a monthly report that consolidates their financial data, drawing from their daily entries of expenses and income. This feature assists in maintaining thorough and organized records of their finances.

Due to component-based architecture modular approach react promotes code organization, reusability, and easier maintenance.

Literature Review -

The landscape of expense tracking systems is rich with diverse options that cater to various needs and preferences.

These systems excel in offering:

a. **Comprehensive Solutions:**

Existing expense tracking systems offer a wide range of features, including budget management, transaction categorization, and data visualization,

providing users with comprehensive tools to manage their finances effectively.

b. **Convenience:**

Mobile applications enable users to track expenses on-the-go, while web interfaces offer detailed analytics and reporting functionalities, providing convenience and accessibility across different devices.

c. **Automation:**

Integration with banking and financial institutions allows for automatic transaction syncing, streamlining data entry processes and reducing manual effort for users.

d. **Data Security:**

Existing systems prioritize data security, utilizing encryption and other measures to protect user information, ensuring peace of mind for users regarding the safety of their financial data.

e. **User Adoption:**

Despite challenges, existing systems have achieved significant user adoption, indicating the value they provide in helping individuals and businesses manage their expenses efficiently.

f. **Continuous Improvement:**

Developers of existing systems actively seek feedback and make continuous improvements to enhance user experience and add new features, demonstrating a commitment to meeting user needs and preferences.

Proposed System Scope:

The proposed expense tracker application aims to build upon the strengths of existing systems while addressing any gaps or areas for improvement. The scope of the system encompasses:

1.3. **Module Requirement Analysis:**

Conducting a thorough analysis of user requirements to identify essential features and functionalities required for effective expense tracking.

1.4. **System Architecture Design:**

Designing a flexible and scalable system architecture that accommodates future enhancements and integrations with third-party services.

1.5. **User Interface Design:**

Developing an intuitive and visually appealing user interface across web and mobile platforms, ensuring ease of navigation and accessibility for all users.

1.6. **Expense Tracking Functionality:**

Implementing core features such as transaction recording, categorization, and budget management, with options for manual entry or automatic syncing with bank accounts and credit cards.

1.7. **Reporting and Analytics:**

Providing robust reporting tools and data visualization features to enable users to gain insights into their spending habits and financial health.

1.8. **Security and Privacy:**

Implementing robust security measures to protect user data and privacy, including encryption, two-factor authentication, and compliance with data protection regulations.

Customization and Personalization:

Offering customization options to tailor the application to individual preferences, including customizable categories, budgeting goals, and notification settings.

Application Workflow:

1. **Register Page:** New users create an account here. After successful registration, they are redirected to the login page to enter their credentials.
2. **Login Page:** Users start here. If they have an existing account, they can log in and proceed to the home page. If not, they are directed to the register page.
3. **Home Page:** Upon successful login, users land here. They can navigate between the home page and categories. Logging out from either page sends them back to the login page.
4. **Category Page:** Users can access and navigate this page from the home page. It's also a "protected route" accessible only to logged-in users.
5. **Protected Routes:** Both the home and category pages are protected routes, accessible only to logged-in users.
6. **Guest Routes:** The sign-in and register pages are guest routes, accessible to anyone.

7. **Single Page Application (SPA) Trait:** Despite navigating through multiple pages, users technically stay on one page throughout their session. This SPA trait, facilitated by React, enables dynamic content updates without page reloads, providing a faster and seamless experience while conserving server resources.

Sequence Diagrams:

1. User Login Sequence Diagram:

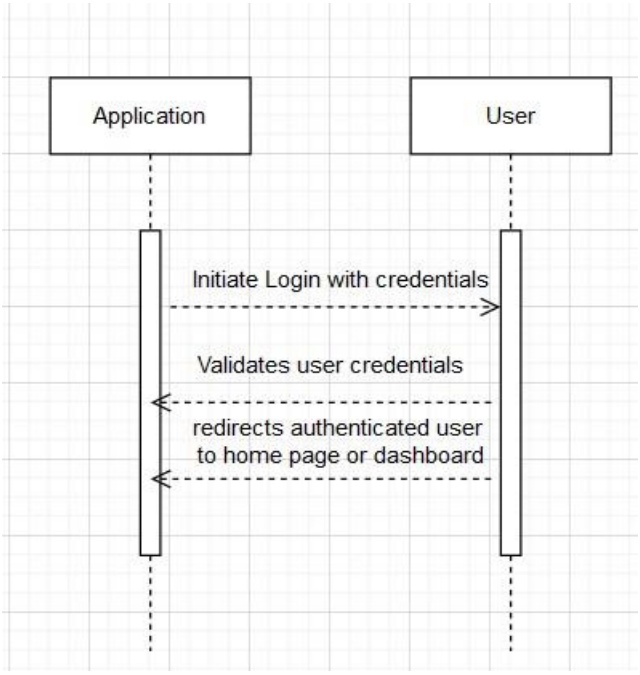

Fig 1.

2. Admin Login Sequence Diagram:

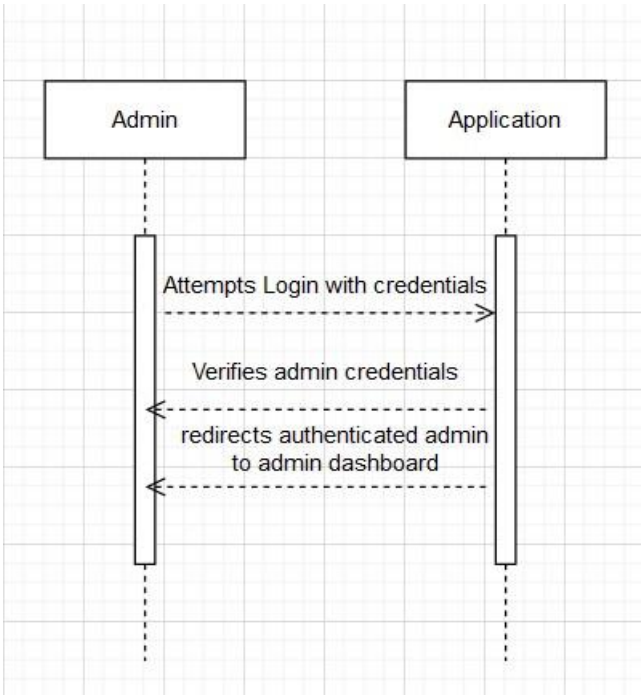

Fig 2.

Conclusion:

In conclusion, the proposed expense tracker application represents a promising solution for individuals and businesses seeking efficient and user-friendly tools for managing their finances. Drawing upon the strengths of existing systems and addressing potential areas for improvement, the proposed system aims to deliver a comprehensive and customizable expense tracking experience.

By prioritizing features such as intuitive user interfaces, robust security measures, and flexible customization options, the application seeks to empower users to gain better control over their spending habits, improve financial planning, and achieve their budgeting goals. The emphasis on continuous improvement and user feedback ensures that the application remains responsive to evolving user needs and preferences.

Recommendations:

a. **User Engagement Strategies:**

Implement strategies to encourage active user engagement, such as personalized notifications, educational resources on financial literacy, and gamification elements to incentivize consistent usage.

b. **Integration with Third-Party Services:**

Explore opportunities for integration with popular financial management tools, banking platforms, and expense reporting software to enhance interoperability and streamline user workflows.

c. **Accessibility and Inclusivity:**

Ensure the application is accessible to users with diverse needs, including those with disabilities or language barriers, by adhering to accessibility standards and offering multilingual support.

d. **Community Building:**

Foster a sense of community among users through forums, discussion boards, and social media channels, providing opportunities for peer support, knowledge sharing, and collaborative problem-solving.

e. **Scalability and Future Expansion:**

Design the application architecture with scalability in mind to accommodate future growth and expansion, including support for new features, integrations, and user base scaling.

f. **Data Privacy and Compliance:**

Stay abreast of evolving data privacy regulations and best practices, ensuring compliance with applicable laws and standards to protect user data and maintain trust.

REFERENCES:

[1] P. Thanapal, Y. Mohammed, R. T. Lokesh and S. J. Kumar, "Income and Expense Tracker", Indian Journal of Science and Technology, vol. 8, pp. 118-122, January 2015, ISSN 0974-6846.

[2] S. A. Sabab, S. S. Islam, R. M. Jewel and M. Hossain, "eExpense: A Smart Approach to Track Everyday Expense", 4th International conference on Electrical Engineering and Information & Communication Technology, 2018.

[3] M. Vanitha, K. Alekhya and A. Sai Gowthami, "Expenses Management System", Int. Jnl. Of Advanced Networking & Applications (IJANA), 2018.

[4] T. M, P. Lekshmi and M. T, "Expense Tracker", International Journal of Advanced Research in Science Communication and Technology (IJARSCT), vol. 9, no. 4, September 2020.

[5] S. Adamya and M. Nitin, "A Django Based Educational Resource Sharing", Journal of Scientific Research (JSR), vol. 64, no. 1, 2020.

[6] T. Sawant, A. Satwilkar, V. Shirke and S. V. Jadhav, "Django Based Web Application to Empower Skilled", IRE Journals, vol. 4, no. 11, MAY 2021, ISSN 2456-8880.