International Journal of Recent Engineering Research and Development (IJRERD)
ISSN: 2455-8761
www.ijrerd.com || Volume 09 – Issue 02 || Mar - Apr 2024 || PP. 78-83

Tracking the Expense using MERN Stack and Data Visualization

[1]Dr. Bindiya M K, [2]Satvik Somvanshi, [3]Shama S Kolhar, [4]Shreya Gupta, [5]Shreyas Ramchandra Haladi

[1]Professor, Department of Computer Science and Engineering & SJB Institute of Technology
[2, 3, 4,5]Student, Department of Computer Science and Engineering & SJB Institute of Technology

Abstract: The Expense Tracker emerges as an urbane and user-friendly solution for every individual seeking to enhance their personal financial stability. In the era, where financial stability is predominant, the expense tracker application is designed to streamline the process of tracking, categorizing, budgeting and analyzing expenses. The central aim of the research was to identify an efficient approach to implement budgeting, delivering personalized budget plans based on individuals' income and spending preferences. The paper makes a noteworthy contribution by exploring links between prediction, categorization, and analysis of the expenses to implement tracking and budgeting. Identifying research voids in this field is emphasized, providing direction for forthcoming investigations.

Index Terms: Financial Management, MERN Stack, Data Visualization, Budgeting, Categorization

I. Introduction

In the present landscape of financial dynamics and consumer psychology, there is an extreme need for an effective personal financial management system. Financial stress leads to a chain of challenges, significantly impacting various aspects of an individual’s life. The weight of financial strain impacts more than just our wallets, but also our physical and mental well-being. The stress and anxiety caused by financial difficulties can take a toll on our overall health, leading to elevated levels of stress, and even depression. Moreover, it can put a strain on our personal relationships, causing conflicts and differences in spending habits within families and partnerships. In the workplace, financial stress can lead to distraction and worry, resulting in a decline in job performance and productivity. Additionally, education is not spared from the impacts of financial strain, hindering both individuals and their children from pursuing academic goals. As the world deals with the complexities of various income sources, fluctuating expenses and the need to achieve the set financial goal, it becomes a salient requirement to not only streamline financial tracking but also empower users to make informed decisions.

The paper analyzes the technological foundations of expense tracker, real-time data synchronization, and user-friendly interfaces on enhancing the overall user experience. Additionally, we will examine the out-turn of these tools on individual financial literacy, adherence to budgeting, and the achievement of financial goals. Our research is aimed at determining how these technological features influence users’ compliance with budgeting practices. Through evaluating the user friendliness of interfaces and the intelligent classification enabled by machine learning, we seek to determine in how far these tools support disciplined and efficient budget control.

To conclude, this paper goes beyond the mere examination of technological components; it explores into wider issues such as financial literacy and adhering to a budget in order for long-term objective achievement. Thus, we seek to provide meaningful knowledge for both academics and practitioners regarding the multi-dimensional effect involved in modern expense tracking technologies on individual financial health.

II. Literature Review

[1]. A detailed comparison between the MEAN (MongoDB, Express.js, Angular.js, Node.js) and MERN (MongoDB, Express.js, React.js, Node.js) stacks in web development is provided. It explores the components of each stack, such as frontend and backend frameworks, runtime environments, and data storage. Benefits and drawbacks of Angular.js and React.js are discussed, along with factors to consider when choosing between the two stacks, including development speed, security, and performance. The paper concludes that both stacks are reliable for frontend development but suggests React.js as a preferred choice for smaller applications due to its security and performance benefits.

[2]. It outlines the transformations in web-based applications and data management requirements, highlighting the limitations of relational databases in handling large-scale data. MongoDB, a prominent NoSQL database, is specifically designed for web-based applications, offering exceptional scalability, performance, and availability. Its document-based data model allows for seamless adjustments to data structures without costly migrations, making it highly adaptable for managing unstructured data. The

paper emphasizes MongoDB's advantages over other NoSQL databases, such as automated sharing, replication, and indexing, which contribute to its scalability and high performance.

[3]. The literature survey provides information on the field of data visualization. After the discussion one could argue that data visualization is the method to represent data in a pictorially, clearly and effectively. It emphasizes not only how to visualize but also how visualization can be helpful. It has appeared as a very powerful, widely acceptable and applicable tool to analyze and understand huge and complex data. Data visualization has come up as a rapid, easy method of conveying ideas in a universally accepted format. It communicates complex ideas with ease, clarity, accurately, and effectively. These benefits make data visualization to be useful in various fields of study. Data visualization is the need of future to reduce the report generation effort.

[4]. It streamlines the process of estimating monthly income and expenses by removing the necessity for manual calculations. Instead of users having to manually input and compute their financial data, our system automates this process, saving time and effort. This automation not just diminishes the chances of errors but also furnishes users with precise and current financial insights. It features modules that are meticulously designed to prioritize efficiency and user experience. These modules are crafted to perform tasks swiftly and seamlessly, ensuring a smooth user interaction. Additionally, a significant emphasis was placed on visual aesthetics, ensuring that the interface is visually appealing and user-friendly. By incorporating intuitive design elements and attractive visuals, the aim was to enhance user engagement and satisfaction while using our platform.

III. Existing System

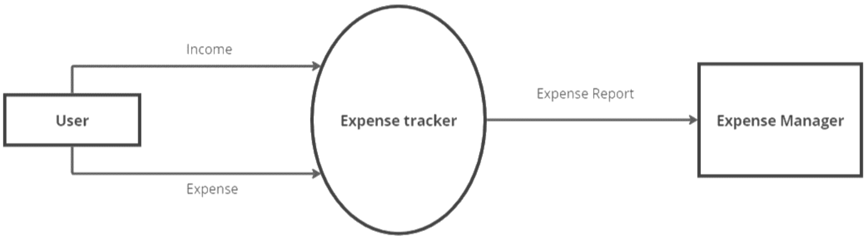


Fig.1 Existing System for Expense Tracker

The primary purpose of the existing expense tracker is to record and categorize expenses, Furnishing users with insights regarding their financial habits and helping them make decisions. The expenses are categorized into predefined or customizable categories such as groceries, transportation, utilities, and entertainment. Overall, expense trackers serve as valuable tools for promoting financial awarenessand fostering responsible spending habits.

IV. Proposed System

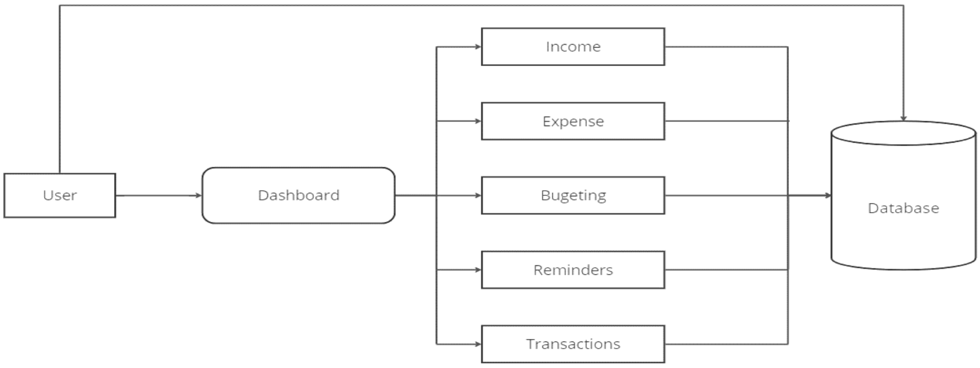


Fig.2 Proposed System for Expense Tracker

The proposed expense tracker system offers an extensive solution enabling individuals to effectively manage their finances. At the core of the system is a user-friendly dashboard that presents users with a graphical representation of their monthly expenses, upcoming reminders, recent transactions, and an option to generate and send email reports analyzing their expenditures. This dashboard serves as a center point for users to gain swift insights into their financial status and activities.

The system includes dedicated pages for managing expenses, income, budgets, reminders, and transactions. Users can provide their expenses and income, set budgets for different categories, and receive notifications if they exceed their predefined budgets. Additionally, the system offers a reminder feature to notify users of recurring EMIs and bills payments, ensuring they never miss a payment deadline.

The transaction page provides a detailed overview of all debited and credited amounts, allowing users to track financial transactions efficiently. Users can sort transactions by category, date, or other relevant criteria, making it easy to analyze their expenditure habits and pinpoint areas that could be enhanced. Overall, the proposed expense tracker system offers a set of features to help users to keep up with their finances, make informed decisions, and achieve financial goals effectively.

V. Implementation

NextJS, the react framework, while not part of the traditional MERN stack, can still be used as a frontend framework alongside the MERN stack to strengthen the performance and user experience of an expense tracker application. NextJS is used to build the front-end part of the website. All interactive elements, whether it's filling out forms, interacting with pie charts, buttons, or any other components, are powered by NextJS. NextJS was opted as it was a better fit for the project, especially considering the numerous pages involved. A higher number of pages can often slow down a web application, but Next.js addresses this issue through features like Hot Reloading and Server-Side Rendering. Additionally, its built-in routing support and efficient handling of images to assure that the web application is always production-ready.

Node.Js permits the execution of JavaScript code outside of a browser. Developers leverage Node.Js to creator command-line equipment and carry out server-side scripting, allowing the generation of dynamic web pages via server-side execution of scripts. Powered through Chrome's V8 JavaScript engine, Node.Js boasts terrific velocity and performance. V8, an open-supply JavaScript engine evolved by way of Google and written in C++, serves as the basis for each Google Chrome and the Node.Js runtime surroundings.

Express.js simplifies the process of building robust and scalable server-side applications by providing a minimalistic and adaptable web development toolkit for Node.js. With reference to the expense tracker, Express.js facilitates the creation of RESTful APIs to handle various functionalities such as user authentication, managing expenses, retrieving data from the database, and serving static files. It serves as the intermediary between the frontend and the MongoDB database, enabling seamless communication between the client-side React components and the server-side logic. Additionally, Express.js offers middleware capabilities that allow for easy integration of additional functionality such as authentication middleware, error handling middleware, and request logging. Overall, Express.js empowers developers to efficiently build and deploy the backend infrastructure of the expense tracker application, enabling smooth interaction between the frontend and the database while ensuring robustness and scalability.

MongoDB stores a variety of crucial information, including user account details such as usernames, email addresses, and hashed passwords for secure authentication. Additionally, MongoDB records income sources, expenses, budgets, and transaction histories, each organized into separate collections within the database. Income sources and expenses are logged with relevant details such as amounts, categories, and dates, while budgets are stored with user-defined categories and allocated amounts. The transaction history provides users with a detailed audit trail of their financial activities, facilitating analysis and decision-making. MongoDB's flexible querying capabilities allow for efficient retrieval and manipulation of financial data, empowering users to track their expenses, manage budgets, and make smarter financial decisions with instant spending insights.

Authentication API encompasses a registration page allowing users to authenticate via email and password. Additionally, it integrates an external authentication API, facilitating user login through a third-party service, such as Google Mail.

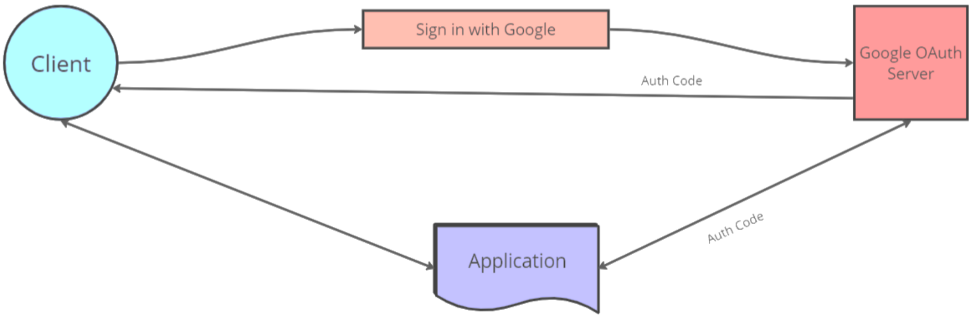
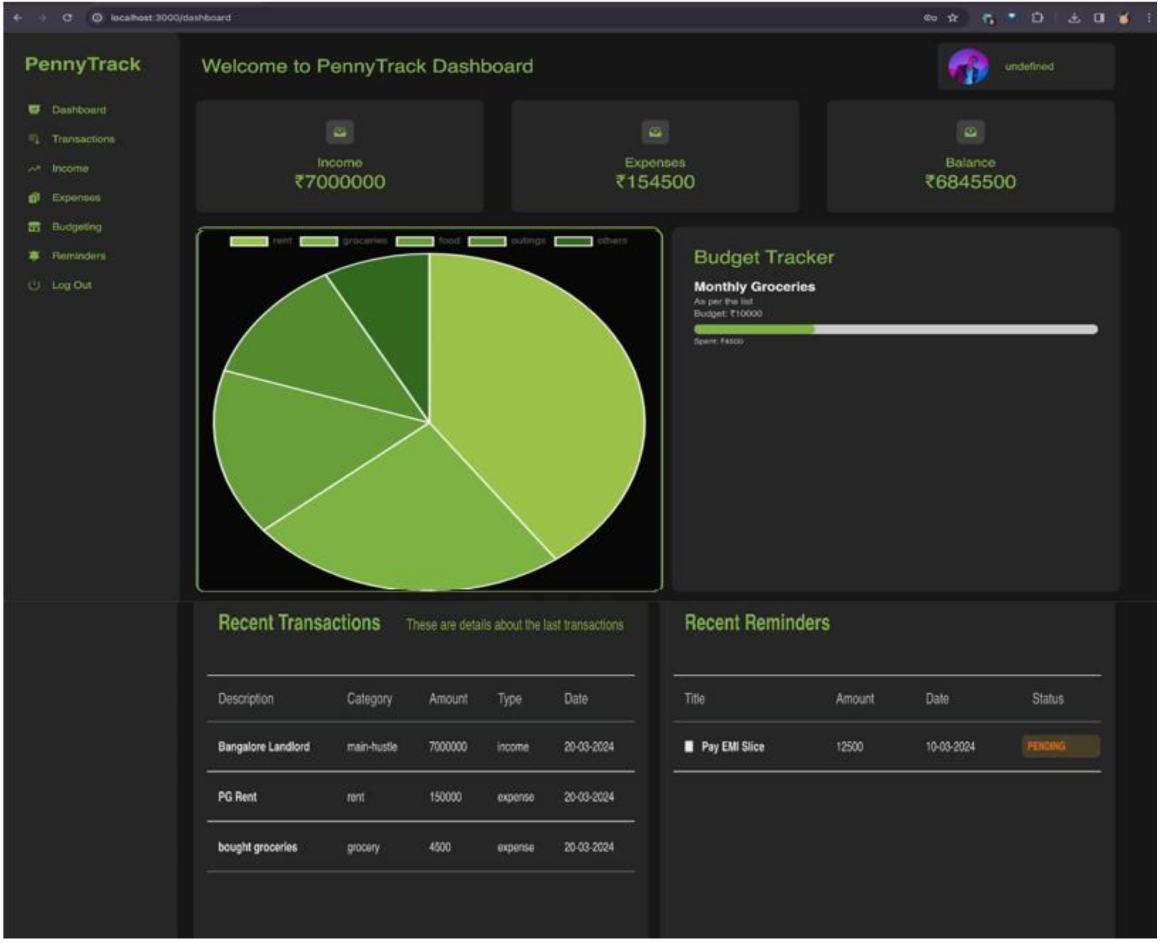


Fig 3. Authentication API

The flowchart outlines the sequence of steps involved in authentication. The process starts when a user clicks on the "Login with Gmail" button on the website. The server receives the OAuth request from the client. The server handles the OAuth redirect by redirecting the user to the Google OAuth endpoint. The user logs in with their Gmail credentials on the Google OAuth page. Google responds to the server with an OAuth token. The server requests the OAuth token and the user's profile information from Google. Google sends the OAuth token and user profile information to the server. The server creates or retrieves the user's account based on the received profile information. The server verifies the user's identity and authenticates the user's session. Once authenticated, the server redirects the user to the dashboard or another authorized page. Error handling is included to find solutions to any issues that might arise during the authentication process.



Welcome to PennyTrack Dashboard

Income
₹7000000

Expenses
₹154500

Balance
₹6845500

Budget Tracker

Monthly Groceries
As per the list
Budget: ₹10000
Spent: ₹4500

Recent Transactions These are details about the last transactions

Description	Category	Amount	Type	Date
Bangalore Landlord	main-hustle	7000000	income	20-03-2024
PG Rent	rent	150000	expense	20-03-2024
bought groceries	grocery	4500	expense	20-03-2024

Recent Reminders

Title	Amount	Date	Status
Pay EMI Slice	12500	10-03-2024	PENDING

Fig 4. Dashboard Page

The creation of interactive and visually appealing charts, including pie charts, can be done using a charting library Chart.js. When the frontend receives expense data, it is sent to the pie chart component. This chart uses this data to create a graph that slices expenses into various categories. Each category would then be represented by a slice of the pie chart and its magnitude would depend on the amount of money spent in that particular category. This interface usually houses the pie-chart-component within the expenditure tracking app, appearing in the dashboard or another analytics page separately. With this pie-chart users can view their expenditures grouped into different categories.

With a spend tracker application built using the MERN stack, the trio of Nodemailer, SendGrid, and Vercel work together seamlessly to facilitate efficient reporting to users Nodemailer, a Node.js module is the backbone for configuring and sending emails systematically to the internal application backbone. Using the versatile Nodemailer API, the application can run notifications, such as reminders or financial alerts, tailored to users' financial activities. SendGrid, a cloud-based email service steps as a delivery engine to assure that notifications are delivered to users' email addresses. With its robust APIs and features such as email templates and tracking, SendGrid improves the effectiveness of email communications to a higher level, providing insights into delivery status and engagement metrics through backend API endpoints responsible for sending notifications. Using Vercel's serverless functionality, the application executes simple backend logic without the cost of maintaining the server infrastructure. This collaborative process streamlines the processing of information, enabling users to submit timely and relevant information about their spending, and ultimately improving their financial management experience further.

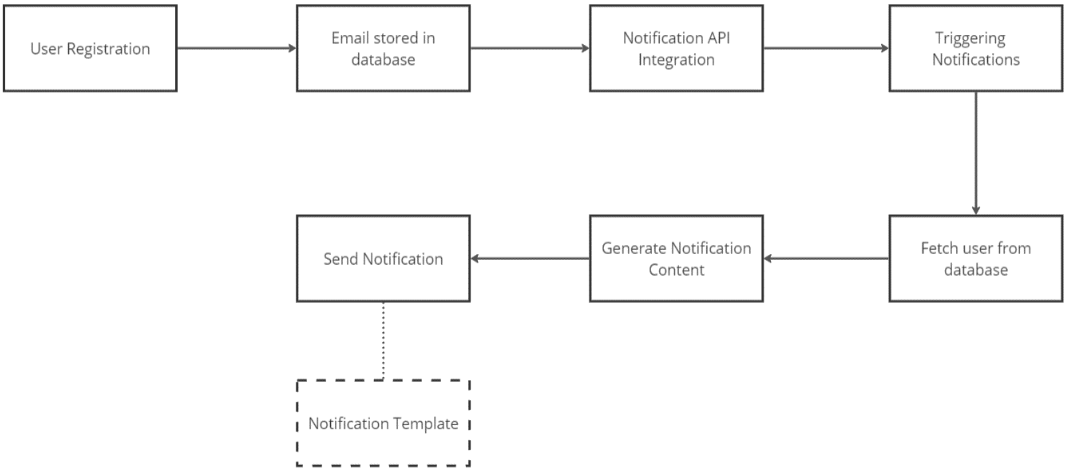


Fig 5. Notification API

Users register on the website and provide their email addresses. When user registers, the email address is securely stored along with other account information in our database. The Notification API service is integrated into the backend of the website. Notifications are triggered based on certain events, such as reminders or exceeding budget thresholds. The backend fetches the email address of the user from the database. The backend generates the content of the notification, including the message and details. The backend sends the notification using the Notification API service. The process ends after the notification is successfully sent.

VI. Challenges

The task of convincing users to consistently input their expenses, income, and other financial data can be challenging, as it requires active participation to ensure the system's effectiveness in providing accurate insights. However, striking a balance between user engagement with the system's simplicity poses another hurdle. Offering too many features risks overwhelming users, while the lack of options may limit the system's usability for diverse user needs. Moreover, the frequency and content of notifications, such as budget limits, reminders, and transaction updates, must be carefully managed to prevent notification fatigue and maintain user engagement. Striking the correct balance between user involvement, customization, and notification frequency is important for the success of the expense tracker system.

VII. Results

A website for keeping track of expenses, developed with MERN stack, features a strong set of functionalities aimed at giving users the ability to effectively manage their funds. An interactive Dashboard forms the very core of this site and it shows people where they stand financially at a glance. Individuals can get real time updates on their total earnings, monitor the money they are spending and keep tabs on their present budget status. Illustrative visuals like pie charts showing expense segregation make it easier to understand personal expenditures.

The page named Income functions as the central place where all the revenue streams owned by a user can be managed from. On this web page, users have all the privileges of adding, editing or deleting incomes sources so that their financial inflows are correctly represented. The Expense web page equally provides a perfect tool for users who want to record and arrange their expenses in order. By breaking down costs into different categories such as groceries/food items, utilities/bills or transport fares customers can maintain accurate records about what they usually spend money on and gain deeper insights into their own spending patterns.

The page deals with Budgeting so as to enable people to put their money into the right purposes. This is where users can set budgets for individual expense groups and therefore allocate available resources wisely and follow up their expenditures against the previously defined limits. As for Bill Payment Reminder feature, users receive a reminder just in case they forget it through email or within an application. They will get reminders either by email or from the app in case of any important deadlines such as payment dates. Additionally, apart from having all-inclusive tracking capabilities, Transaction pages on this website provide an overview on all financial activities. This will allow them access income, spending and transfer history over time thus making it possible for accounting books that are verifiable and transparently maintained. By implementing advanced search, filter and sort functionalities across your transaction history; these features can enhance organizational efficiency while enabling advanced financial analysis.

Overall, the expense tracker website harnesses the capabilities of the MERN stack to deliver a user-centric platform tailored to meet the various needs of modern financial management. With intuitive tools for financial inflows and outflows management, budgeting, reminders, and transaction tracking, the website empowers users to simplify your finances, make confident decisions, and effortlessly reach your financial goals.

VIII. References

[1]. “Data Visualization: A Study of Tools and Challenges”, Ahmad Tasnim Siddiqui Associate Professor - Visiting at BGI, Faizabad Assistant Professor - Department of Computer Science & Applications Sherwood College of Professional Management Lucknow, India, 2021

[2]. “Expenditure Management System” Dr. V. Geetha, G. Nikhitha, H. Sri Lasya Dr. C. K. Gomathy Assistant Professor and Students, Dept. of CSE, SCSVMV(Deemed to be University), Kancheepuram ,TamilNadu, India, March 2022

[3]. “MongoDB - a comparison with NoSQL databases” by Hema Krishnan, M. Sudheep Elayidom and T. Santhanakrishnan, May 2016

[4]. “Comparative analysis of MEAN stack and MERN stack” by Prati Jain and Manjunath C R, May 2021

[5]. “New technologies for web development” by Grega Jakus, Matija Jekovec, Saso Tomazic and J. Sodnik, 2014

[6]. “Comprehensive Study of MERN Stack - Architecture, Popularity and Future Scope” by Bhavyaa, Suhani Gupta and Ms. Vaishali, December 2021