Expense Manager

A PROJECT REPORT

submitted by

Kamal S **(715521104020)**

Kamal Chander R **(715521104021)**

Department Computer Science and Engineering

PSG Institute of Technology and Applied Research
Coimbatore 641 062

in collaboration with

SAP Labs, Bengaluru

FEBRUARY - 2024

BONAFIDE CERTIFICATE

Certified that the mini-project titled **"Expense Manager"** is the bonafide work of **"Kamal S (715521104020), Kamal Chander R (715521104021)"** which is carried out under our mentoring.

SIGNATURE
Srivatsan Santhanam
VICE PRESIDENT
Spend Engineering, Head of Concur R&D,
SAP Labs
Bengaluru,
Karnataka, India

SIGNATURE

ACADAMIC MENTOR
Dr. Sunitha Nandhini
Assistant Professor
Computer Science and Engineering
PSG Institute of Technology and
Applied Research,
Coimbatore – 641 062

SIGNATURE
Dr. R. Manimegalai
HEAD OF THE DEPARTMENT
Professor
Computer Science and Engineering
PSG Institute of Technology and
Applied Research,
Coimbatore – 641 062

Submitted for evaluation on _____

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to our beloved Principal, **Dr. N. SARAVANA KUMAR** for his overwhelming support and encouragement on this project.

We extent our indebted to **Mr. SRIVATSAN SANTHANAM,** Vice President, Spend Engineering, Head of Concur R&D, SAP Labs India for the opportunity and support which was instrumental in the completion of this project.

We are greatly indebted to **Dr. R. MANIMEGALAI, M.E, Ph.D** Head of the Department, Computer Science and Engineering for her guidance and continuous support which was instrumental in the completion of this project.

We extend our thanks to our industry mentor, **SAP Labs,** Computer Science and Engineering for his technical support and constant supervision without which we could not have completed this project study.

KAMAL S

KAMAL CHANDER R

TABLE OF CONTENTS

ABSTRACT

Our project, titled "Expense Manager," aims to create an efficient expense tracking application with a user-friendly calendar-based interface. Utilizing React for frontend development, we aim to design a visually appealing and intuitive platform that allows users to effortlessly manage and visualize their expenses.

The core functionality of the Expense Manager revolves around providing users with tools to track their expenditures efficiently. Through the calendar interface, users can input their daily expenses, categorize them, and add relevant details such as descriptions. This approach aims to simplify the process of recording expenses while providing a clear overview of spending patterns over time.

In addition to basic expense tracking, our application offers visualizations and analytics. Users will be able to view their spending trends and compare monthly expenses. This feature empowers users to make informed financial decisions and achieve better financial health.

To ensure seamless communication between the frontend and backend, developed a robust backend API using Spring Boot. The API will facilitate smooth data retrieval, storage, and manipulation, ensuring real-time updates to the calendar interface.

Furthermore, the Expense Manager will include a secure database, powered by MongoDB, to store user expense data efficiently. This database will provide a reliable repository for storing detailed expense records, ensuring data integrity and easy retrieval whenever users need to review or analyze their spending history.

By combining the power of React for frontend development, Spring Boot for backend API creation, and MongoDB for database management, our project aims to deliver a comprehensive Expense Manager. This platform will not only simplify expense tracking but also empower users to make informed financial decisions, ultimately leading to better financial management and well-being.

Keywords: React, Spring Boot, MongoDB, Calendar Interface, Data Visualization.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROJECT MOTIVATION

Our project, "Expense Manager," is born out of a deep interest in financial management and the need for a straightforward tool to track expenses. We aim to create a platform that simplifies expense tracking, offering users a clear overview of their finances.

Inspired by the challenges faced in managing personal expenses, we saw an opportunity to develop an application that provides users with an easy-to-use interface for efficient expense monitoring. Our goal is to empower individuals to take control of their finances with a tool that is both intuitive and effective.

Furthermore, this project serves as a valuable learning experience, allowing us to enhance our skills in frontend and backend development. By utilizing modern technologies such as React.js for the frontend and Spring Boot for the backend, we aim to create a robust and user-friendly Expense Manager.

1.2 PROBLEM STATEMENT AND OBJECTIVES

Problem Statement: Existing expense tracking applications often lack simplicity and personalization, making it challenging for users to manage their finances effectively. Our objective is to address this by creating an intuitive and user-centric Expense Manager.

Objectives:

1. Develop a User-Friendly Expense Tracking Platform:
2. Create a straightforward calendar-based interface for easy expense entry.
3. Ensure smooth navigation and user interactions for a seamless experience.
4. Provide Comprehensive Expense Tracking Features and visualisations
5. Allow users to input, categorize, and visualize their expenses effortlessly.
6. Enable easy tracking of spending habits and trends over time.

1.3 SCOPE AND LIMITATIONS OF THE PROJECT

Scope:

Our project aims to develop a feature-rich Expense Manager with a focus on simplicity and functionality. This includes creating an intuitive frontend interface for expense tracking, implementing secure backend systems for data storage, and offering insightful analytics for financial insights. We also plan to engage with users for feedback and iterative improvements to the platform.

Limitations:

While we strive to create a user-centric Expense Manager, limitations such as time constraints and technical expertise may impact the scope of the project. Additionally, the storage capacity for user-uploaded receipts and documents may be limited initially. However, we are committed to refining and expanding the application to meet user needs and improve performance over time.

CHAPTER 2

PROJECT ARCHITECTURE, DESIGN AND IMPLEMENTATION

2.1 SYSTEM ARCHITECTURE

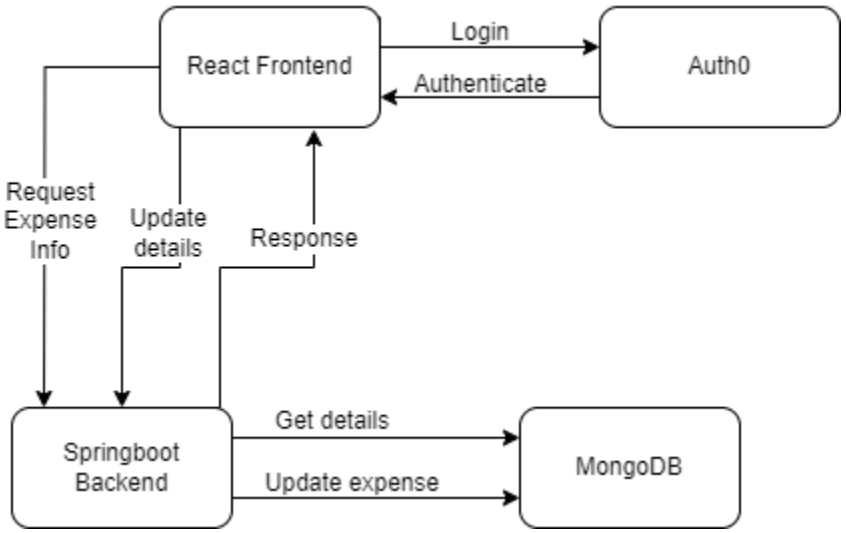


Figure 2.1 Overall Architecture

2.2 OVERVIEW OF THE DESIGN PROCESS

Identified essential features such as expense input, categorization, calendar view and visualization. Selected React for frontend, Spring Boot for backend API, and MongoDB for database management based on project requirements and team expertise. Designed a visually appealing and user-friendly interface focusing on intuitive navigation and clear layouts. Designed MongoDB database schema to store and retrieve expense data. Developed the frontend using ReactJS, implementing features such as expense input and display as a calendar and visualization of spending trends. Built backend API using Spring Boot to handle data retrieval, storage, and manipulation.

2.3 EXPLANATION OF THE ENGINEERING PRINCIPLES USED IN THE DESIGN

- **Modularity**: Breaking down the application into smaller, reusable modules promotes better organization and maintainability. In React, components encapsulate specific functionality, making it easier to understand, test, and modify. Similarly, in Spring Boot, modularization can be achieved through the use of services, controllers, and repositories, separating concerns and promoting code reusability.

- **Separation of Concerns**: Dividing the application into distinct layers, such as presentation (frontend), business logic (backend), and data access, ensures that each component has a single responsibility. React components handle rendering and user interactions, Spring Boot controllers manage HTTP requests and responses, and services handle business logic, while the repository interacts with the database.

- **RESTful API Design:** Designing the backend as a RESTful API follows the principles of statelessness, uniform interface, and resource-based interactions. Spring Boot makes it easy to create RESTful endpoints using annotations like @RestController and @RequestMapping, enabling communication between the frontend and backend.

2.4 DESCRIPTION OF THE STEPS TAKEN TO IMPLEMENT THE PROJECT DESIGN

- The format for the expense data was defined.
- MongoDB collection was created with the chosen format as document structure.
- A SpringBoot entity “Expense” created to be used for data retrieval and storage.
- Calendar interface made using React to view and enter expense details in an easy and intuitive way.
- The interface allows users to easily navigate through dates in a visually appealing and user-friendly manner.
- User authentication performed with the help of Auth0.
- SpringBoot APIs developed to serve the requests of the user.
- Monthly and daily expense provided as requested by the frontend.
- Mongo Repository in SpringBoot used to update and query the database.
- Visualization for the expense data made in React and data for it provided via API
- All API functionalities tested using Postman.
- Separate collection maintained for each user to ensure data privacy.
- Both the React frontend and SpringBoot integrated to give the working webapp.

CHAPTER 3

RESULTS AND ANALYSIS

3.1 TEST RESULTS

- Login page for the expense manager with options to sign in with Google,etc.

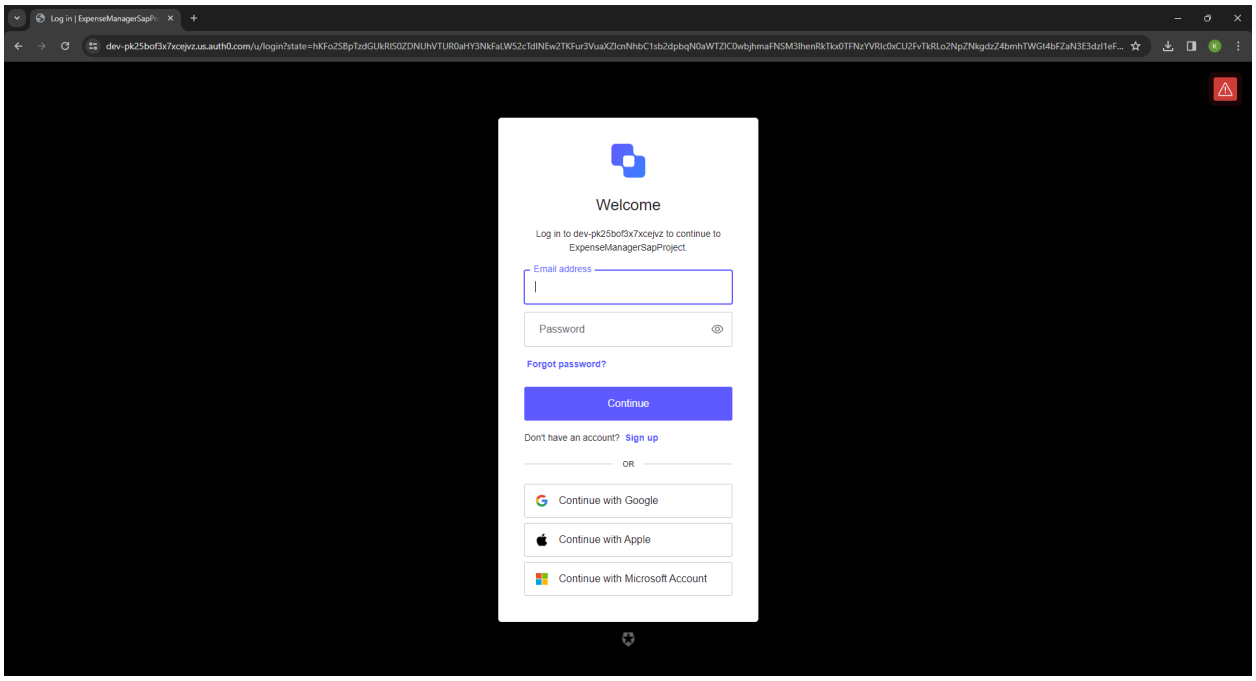


Figure 3.1 Login Page

- Homepage showing the calendar to view and input the expense of the user
- The month, date can be selected according to the users' requirements
- Current date is taken during the startup.

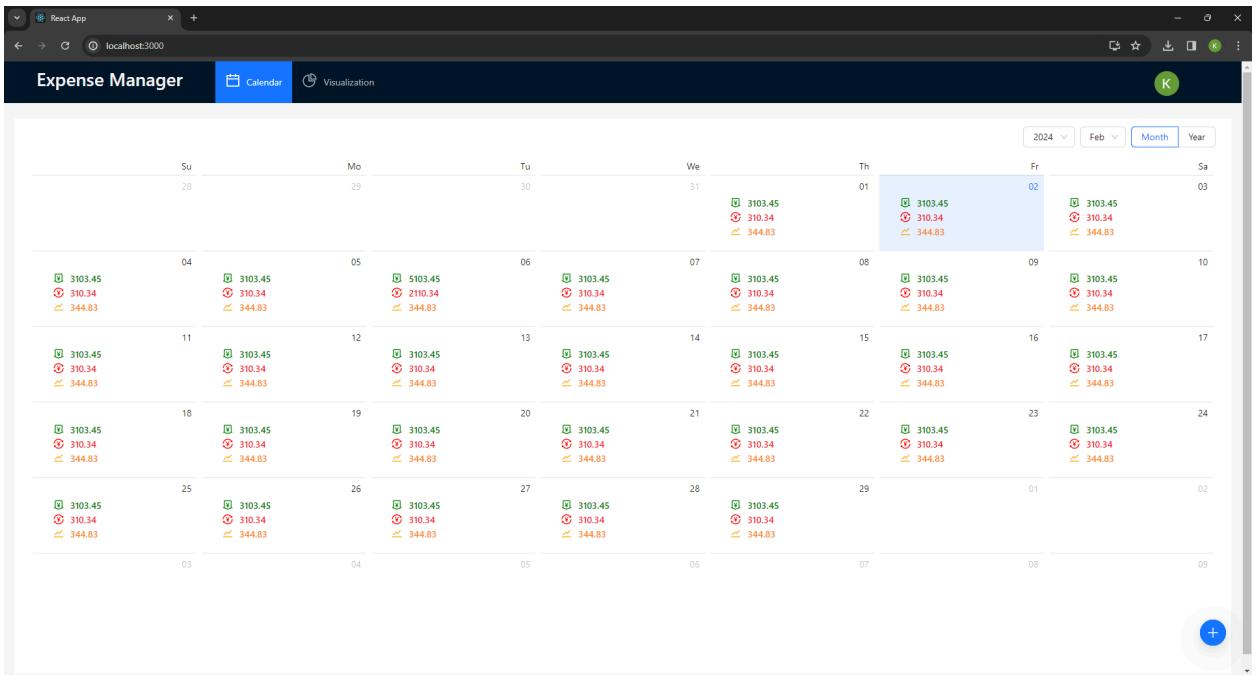


Figure 3.2 Calendar View

- Monthly expense can be added which splits it equally among all the days of the month.

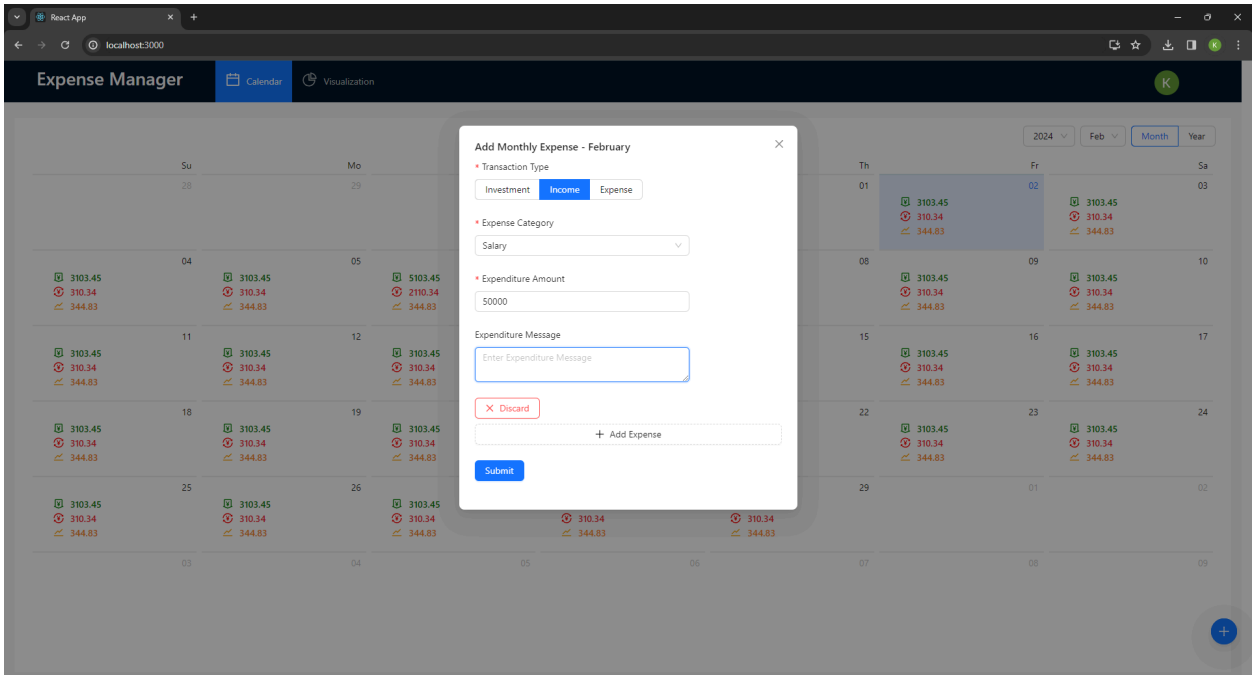


Figure 3.3 Adding Expense

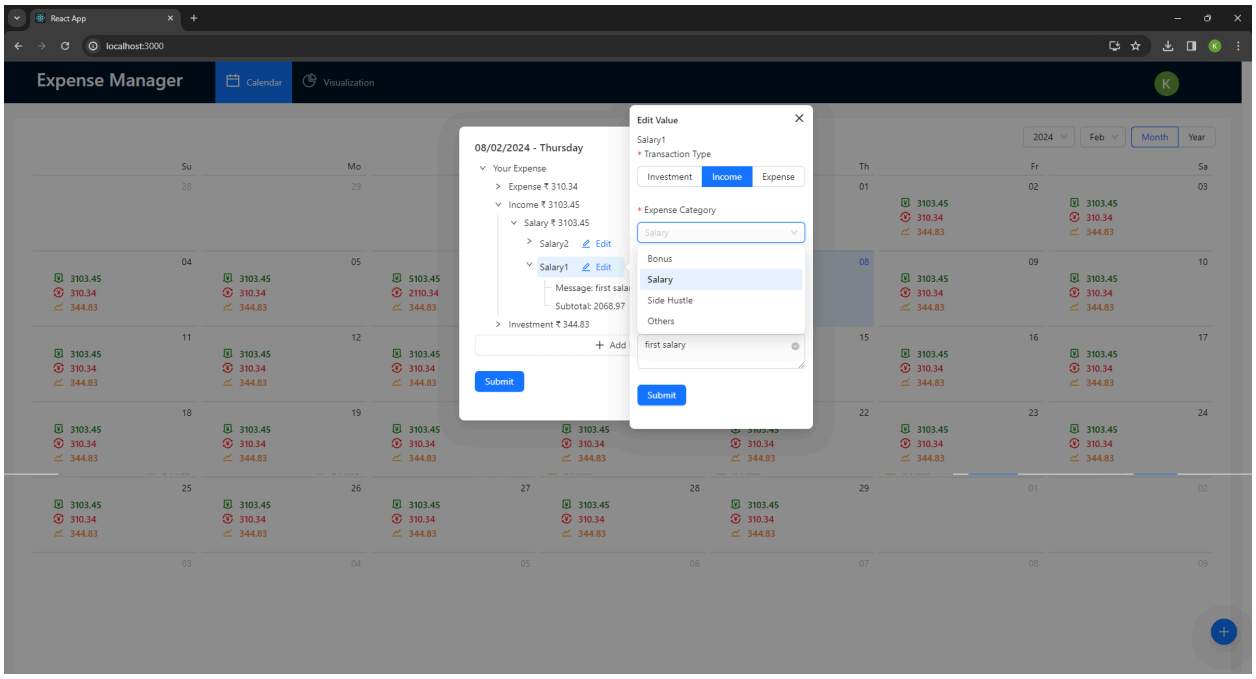
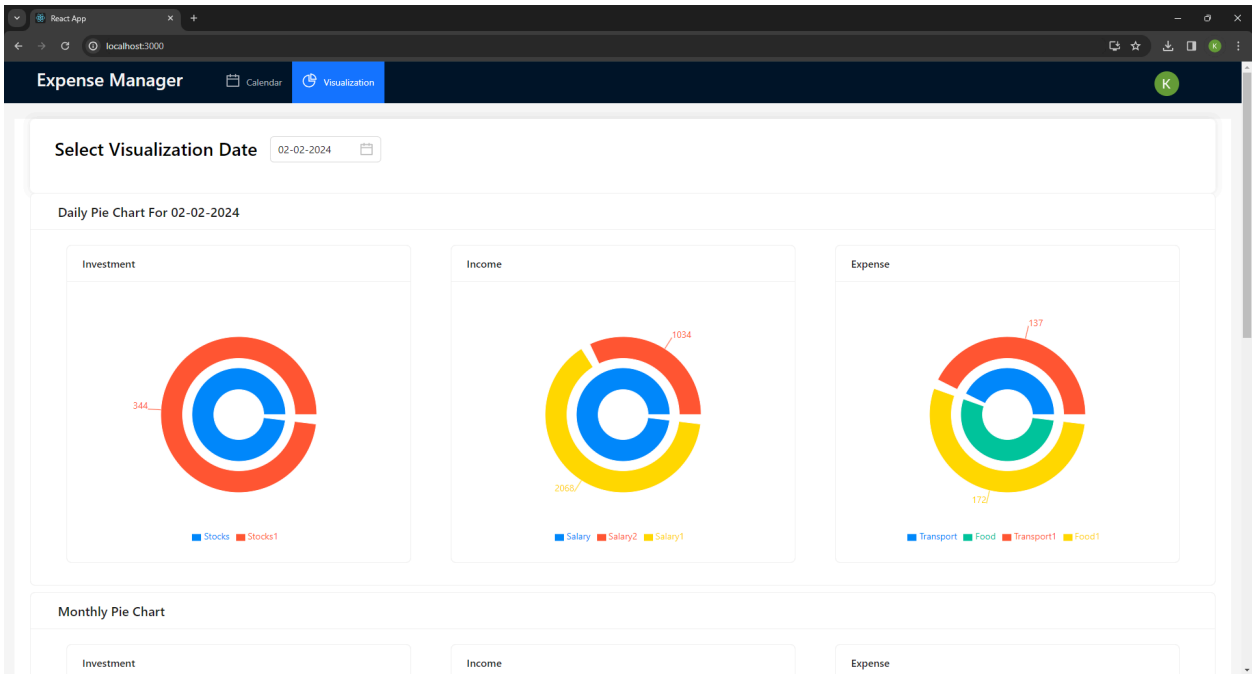


Figure 3.4 Editing expense

- Editing the expense to change the type, amount, message, etc.



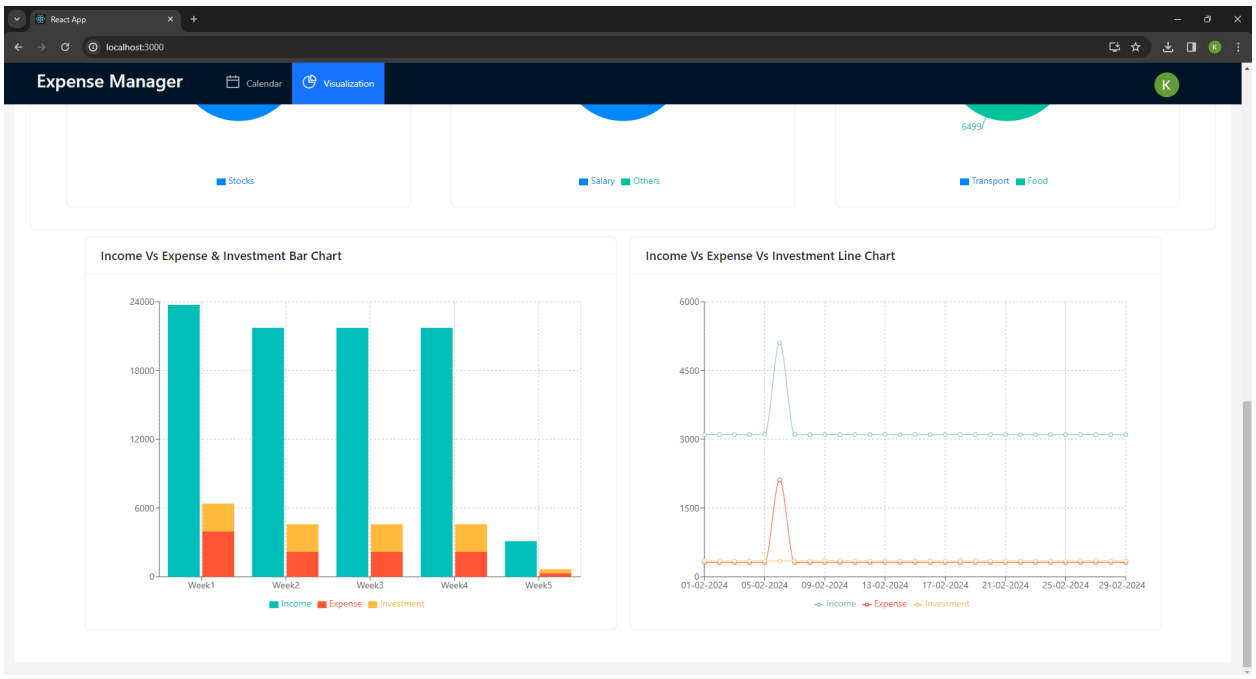


Figure 3.5 Visualization

- The expense for a date and month visualized using charts to help analyse the expenses of the user easily.

CHAPTER 4

LEARNING OUTCOMES

- **Proficiency in Full-stack Development:**

Ability to develop a complete Expense Manager application, integrating frontend (ReactJS) and backend (Spring Boot) components.

- **Database Management Skills:**

Competence in utilizing MongoDB for storing and retrieving expense-related data, including categories, transactions, and user preferences.

- **RESTful API Development**:

Mastery of building RESTful APIs with Spring Boot, facilitating communication between the frontend and backend layers of the application.

- **User Interface Design**:

Understanding of designing intuitive user interfaces with ReactJS, focusing on usability and user experience enhancements for an expense tracking application.

- **Project Management and Collaboration:**

Experience in collaborative development using version control systems like Git, with proficiency in project organization, dependency management, and deployment of the application.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

➢ PROBLEM OBJECTIVES AND ACHIEVEMENTS:

Our project aimed to create a user-friendly Expense Manager application using React.js and Spring Boot, and we successfully achieved our objectives by prioritizing user simplicity with a straightforward expense entry interface, ensuring platform stability and performance, protecting user privacy, and actively engaging with our user community for feedback and improvement. Through our efforts, we delivered a feature-rich application that meets the needs of individuals seeking efficient expense tracking, with a commitment to ongoing innovation of adding new features to enhance user satisfaction.

➢ LIMITATIONS:

While our project achieved its objectives, there are areas of improvement identified. Firstly, scalability could be enhanced to accommodate a growing user base by implementing better architectural solutions. Secondly, additional testing and user feedback incorporation are necessary to optimize the user experience. Thirdly, feature enhancements such as budget forecasting and expense category insights could be added. Fourthly, continuous security monitoring and updates are crucial to mitigate potential risks. Lastly, performance optimization, documentation improvements, and support resources should be prioritized for an enhanced user experience.

➢ FUTURE RESEARCH OR DEVELOPMENT:

For future advancements, focusing on enhancing scalability, improving user experience through innovative features, strengthening security measures, and optimizing performance should be prioritized.

Additionally, exploring emerging technologies and trends in financial management applications could lead to valuable insights and opportunities for further innovation. Implementing a more efficient storage solution for larger files or receipts uploaded by users is also a potential area for future development.

CHAPTER 6

REFERENCES

[1} Spring Boot Documentation: https://spring.io/projects/spring-boot
[2] React Documentation: https://react.dev/reference/react
[3] MongoDB Documentation. https://docs.mongodb.com/
[4] Ant Design Documentation: https://ant.design/components/overview/