EXPENSE TRACKER

22VP003 – MINI PROJECT

Submitted by

BAVYADHARSHINI R
22CSR030

MATHAV Ra
22CSR117

in partial fulfillment of the requirements for

the award of the degree
of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Estd : 1984

KONGU ENGINEERING COLLEGE

(**Autonomous)**

PERUNDURAI ERODE – 638 060

NOVEMBER 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638 060

NOVEMBER 2024

BONAFIDE CERTIFICATE

This is to certify that the Project report entitled **EXPENSE TRACKER** is the bonafide record of the project work done by **BAVYADHARSHINI R (Register No: 22CSR030), MATHAV Ra (Register No: 22CSR117),** in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Computer Science and Engineering** of Anna University Chennai during the year 2024-2025.

SUPERVISOR	**HEAD OF THE DEPARTMENT**
Date:	**(Signature with seal)**

KONGU ENGINEERING COLLEGE

(**Autonomous**)

PERUNDURAI ERODE – 638 060

NOVEMBER 2024

DECLARATION

We affirm that the Project Report titled **EXPENSE TRACKER** being submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion this or any other candidate.

Date:

BAVYADHARSHINI R
(22CSR030)

MATHAV Ra
(**22CSR117)**

I certify that the declaration made by the above candidate is true to the best of my knowledge.

Date: Name and Signature of the Supervisor

ABSTRACT

Expense trackers are invaluable digital tools that help individuals manage their finances by recording income and expenses, analyzing spending patterns, and visualizing financial trends. By keeping a close eye on finances, users can gain better control over their budgets, improve financial literacy, and make more informed decisions. These tools are particularly useful for those looking to understand their spending habits, save effectively, and plan for future financial goals.

The "Expense Tracker" project aims to provide a seamless and intuitive experience in personal finance management. Key features of this application include options to record and categorize both income and expenses, generate interactive graphs to showcase spending trends, and create comprehensive reports for in-depth financial analysis. These features allow users to monitor their finances at a glance and gain insights into their financial behavior, empowering them to make better financial decisions.

Leveraging technologies such as ReactJS for a responsive and user-friendly interface, MongoDB Atlas for secure and scalable data management, and various data visualization libraries for effective graphical insights, the "Smart Expense Tracker" aims to provide users with a seamless experience. Through robust backend management and a focus on user experience, this project strives to deliver a practical and impactful solution for individuals seeking to improve their financial awareness. The "Smart Expense Tracker" will empower users to make informed financial decisions, promoting long-term financial stability and growth.

ACKNOWLEDGEMENT

We express our sincere thanks and gratitude to **Thiru. A. K. ILANGO B.Com., M.B.A., LLB.,** our beloved Correspondent, and all other philanthropic trust members of Kongu Vellalar Institute of Technology Trust who have always encouraged us in academic and co-curricular activities.

We are extremely thankful with no words of a formal nature to the dynamic Principal **Dr. V. BALUSAMY, M.Tech., Ph.D**., for providing the necessary facilities to complete our work.

We would like to express our sincere gratitude to our respected Head of the Department **Dr. S. MALLIGA, M.E., Ph.D.**, for providing the necessary facilities.

We extend our thanks to **Ms. R. S. SHUDAPREYAA, M.Tech.,** Assistant Professor, Computer Science Engineering, Project Coordinator for his encouragement and valuable advice that made us carry out the project work successfully.

We extend our gratitude to our Supervisor **Dr. K. NIRMALA DEVI, Ph.D.,** Assistant Professor of Computer Science Engineering, for her valuable ideas and suggestions, which have been very helpful in the project. We are grateful to all the faculty members of the Computer Science and Engineering Department, for their support.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

"Smart Expense Tracker", steps into a world where financial clarity aligns with your goals, creating a seamless blend of insight and control. Explore our comprehensive features that redefine budgeting and inspire mindful spending. Embark on a journey through an interactive dashboard designed to illuminate your financial path. Experience the satisfaction of organized finances as you gain a deeper understanding of your spending and saving habits. Dive into a world where each transaction tells a story of progress and potential. Take control of your finances with ease and indulge in the peace of mind that comes with financial awareness. "Smart Expense Tracker" offers tools to boost your confidence in managing money, setting you on a path toward financial stability with grace and assurance. Join a community of financially-savvy individuals who share a vision for intentional living. Step into "Smart Expense Tracker" and transform your financial aspirations into reality.

1.1 EXISTING SYSTEM

From the requirement analysis, it is evident that there is no existing system in place for managing personal finances effectively. Currently, users often rely on manual methods, such as maintaining a physical ledger or using basic spreadsheets, to track their income and expenses. This approach can be time-consuming, prone to errors, and lacks the insights needed to make informed financial decisions.

The new "Smart Expense Tracker" will provide a digital platform to simplify and automate the process of tracking finances. It will improve efficiency by categorizing expenses, summarizing spending patterns, and offering visual insights, significantly reducing the likelihood of errors and missed entries.

1.2 SYSTEM STUDY

1.2.1 Understanding the Business Requirements

The first step in the system study is to understand the user requirements for the "Smart Expense Tracker." This involves identifying key objectives, such as enabling users to easily track their income and expenses, visualize financial patterns, and generate insightful reports. It also includes defining the target audience, which could range from individuals looking for basic budget management to those seeking more advanced financial insights.

1.2.2 Analyzing the Existing Systems and Processes

The next step is to analyze the current methods users may be using to track their finances, such as manual record-keeping, spreadsheets, or existing budget apps. This helps identify limitations, pain points, and areas where the "Smart Expense Tracker" can provide added value, such as enhanced automation, improved accuracy, and real-time insights.

1.2.3 Identifying User Needs

The system study should involve identifying the specific needs of users who will rely on the expense tracker. This can be done through surveys, interviews, or user research, uncovering requirements like ease of use, security of financial data, customizable categories, and visualization tools that make financial data easy to understand.

1.2.4 Developing Use Cases

Use cases should be developed to illustrate how the "Smart Expense Tracker" will function and how users will interact with it. These use cases help in visualizing user workflows, such as adding income and expenses, generating reports, and setting financial goals, which in turn aid in refining the design and functionality for an intuitive user experience.

1.3 OBJECTIVE

The objective of "Smart Expense Tracker" is to provide users with a comprehensive and user-friendly platform for managing their personal finances. The tool aims to empower users to track income, record expenses, and gain valuable insights into their spending habits. By offering features such as categorization, visual analysis, and report generation, "Smart Expense Tracker" enables users to make informed financial decisions, helping them plan and achieve their financial goals.

In addition, the tracker will offer seamless functionality to monitor budget trends, set savings targets, and visualize financial progress over time. Through intuitive design and ease of access, "Smart Expense Tracker" aims to support users in building financial awareness, fostering responsible spending habits, and ultimately enhancing financial well-being.

1.4 SCOPE

The scope of the "Smart Expense Tracker" encompasses a comprehensive suite of tools designed to help users effectively manage their personal finances. The platform allows users to track both income and expenses across various categories, offering a holistic view of their financial activities. From everyday expenses such as groceries, transportation, and utilities, to larger financial goals like savings, investments, and debt management, the tracker is designed to cater to a wide range of financial needs. In addition to basic expense tracking, the system includes features such as budget creation, goal setting, and financial reports to help users stay on top of their spending and savings plans. It supports multiple devices, offering users the flexibility to manage their finances at home or on the go, ensuring access to real-time data wherever they are. The tracker will also incorporate visual tools such as charts and graphs to give users an intuitive understanding of their financial patterns. The "Smart Expense Tracker" prioritizes ease of use and accessibility, allowing individuals from all walks of life—regardless of financial expertise—to manage their finances effectively. With a focus on data privacy and security, the platform will protect sensitive user information through advanced encryption methods. By offering customizable features, the tracker will support users in tailoring their financial management approach to meet their specific needs, whether they are focused on budgeting for the month or planning for long-term financial goals.

CHAPTER 2

GENERAL DESCRIPTION

2.1 PROJECT PERSPECTIVE

- The growing reliance on digital financial tools highlights the need for a comprehensive system to help individuals effectively manage their finances, track spending, and achieve financial goals.
- By embracing digital finance management, "Smart Expense Tracker" provides users with an efficient and convenient way to manage their finances, overcoming the limitations of manual tracking and enhancing their financial awareness.
- The platform offers unparalleled ease of use, enabling users to track income and expenses, set budgets, and generate financial reports, all from the comfort of their devices, anytime, anywhere.

2.2 USER CHARACTERISTICS

For "Smart Expense Tracker," understanding user characteristics is essential for creating a platform that meets their specific needs. Users, typically individuals looking to improve their financial management, may have varying levels of financial knowledge. They seek an easy-to-use interface, seamless navigation, and quick access to insights about their spending habits. Users value tools that help them stay on top of their budgets, track daily expenses, and visualize financial data through charts and graphs. They also appreciate features like expense categorization, automated reports, and reminders for bills or savings goals. Security and data privacy are top priorities for users, and they seek a trustworthy platform to store sensitive financial data.

2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

2.3.1 Time

The project must adhere to a defined timeline for design, development, testing, and deployment. Timely delivery is crucial for maintaining user interest and ensuring the system meets seasonal or financial-year-related deadlines. Delays could impact user adoption and hinder the overall success of the "Smart Expense Tracker."

2.3.2 Budget Limitations

There may be constraints on the budget allocated for the development, integration of technologies, and marketing of the expense tracker. This could affect the selection of advanced features, third-party services, or premium tools that might be integrated into the platform, potentially limiting some functionalities initially.

2.3.3 Security

Given the sensitive nature of financial data, the expense tracker must comply with industry standards for data security, such as encryption protocols and privacy regulations. Compliance with relevant standards is critical to ensuring the integrity and safety of user information. Security constraints may arise when balancing robust protection with a smooth user experience.

2.3.4 Scalability

The design of the "Smart Expense Tracker" must be scalable to accommodate future growth, such as an increase in the number of users, larger volumes of data, and enhanced features. The system architecture needs to be capable of handling higher user loads and expanding functionality without compromising performance. Scalability constraints could arise if the platform’s architecture is not prepared for future upgrades.

CHAPTER 3
REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

3.1.1 User Authentication

Users should be able to securely register, log in, and log out of their accounts to access the expense tracker platform. Authentication should allow users to manage their financial data, track expenses, add income, set budgets, and access reports securely.

3.1.2 Expense and Income Tracking

The platform should allow users to add, edit, and view both income and expense records. Users should be able to categorize expenses (e.g., groceries, food, entertainment) and income sources. The platform should display a summary of income versus expenses to provide users with an overview of their financial situation.

3.1.3 Budget Management

Users should be able to set and manage budgets for various expense categories. The system should allow users to monitor their spending against the set budgets and notify them if they are close to exceeding their limits.

3.1.4 Data Visualization

The platform should provide users with visual representations of their financial data, including graphs, charts, and other visual tools. These visualizations should display income versus expenses, budget progress. This feature should enhance users' ability to quickly interpret their financial data, track their goals, and make informed decisions.

3.1.5 Report Generation

The platform should allow users to generate financial reports based on specific date ranges, categories, and other parameters. Reports should be exportable in formats such as PDF of income, expenses.

3.2 NON-FUNCTIONAL REQUIREMENTS

3.2.1 Performance

The "Smart Expense Tracker" should load quickly and handle user interactions efficiently, even with a large amount of data (e.g., numerous transactions and expenses). Page load times should be optimized to ensure a smooth user experience, especially during peak usage times.

3.2.2 Usability

The platform should be intuitive and easy to navigate. The user interface should be clear, visually appealing, and designed with a focus on usability, ensuring users can track their expenses and income without confusion. The application should support features like search, filtering, and sorting for easy navigation.

3.2.3 Reliability

The expense tracker should be stable and reliable, with robust error handling and recovery mechanisms to prevent data loss or corruption in case of system failures or unexpected errors. All user data should be backed up securely.

3.2.4 Availability

The platform should have high availability, with minimal downtime for maintenance. Redundant systems and backup mechanisms should be in place to ensure users can access their data at any time, maintaining a seamless experience

CHAPTER 4
DETAILED DESIGN

4.1 ARCHITECTURAL DESIGN

The architectural design of the **Smart Expense Tracker** focuses on the user experience and interaction with the platform. It includes various diagrams such as use case diagrams, sequence diagrams, and activity diagrams, which illustrate the user's interaction flow with the system.

4.1.1 MODULE CHARACTERISTICS

User Module: This module is responsible for managing user data, including income, expenses, budget tracking, Visual Representation and report generation. The user module allows users to interact with the platform through various pages and functionalities.

4.1.1 User Module

The User Module provides access to the following pages and features:

- **Dashboard:** Offers a quick overview of total expenses, remaining budgets, and spending insights, helping users monitor their financial health at a glance.
- **Add Expense:** Allows users to log, and delete expenses with details like amount, category, and date, ensuring accurate tracking of transactions.
- **Add Income:** Allows users to log, and delete expenses with details like amount, category, and date, ensuring accurate tracking of transactions.
- **Reports:** Provides visual analytics (charts, graphs) and allows users to generate reports for deeper insights into spending patterns and budgeting.

FLOW CHART:

Illustrates the logical flow and interactions within the User Module, showing steps from user authentication shown in FIGURE 4.1 FLOWCHART.

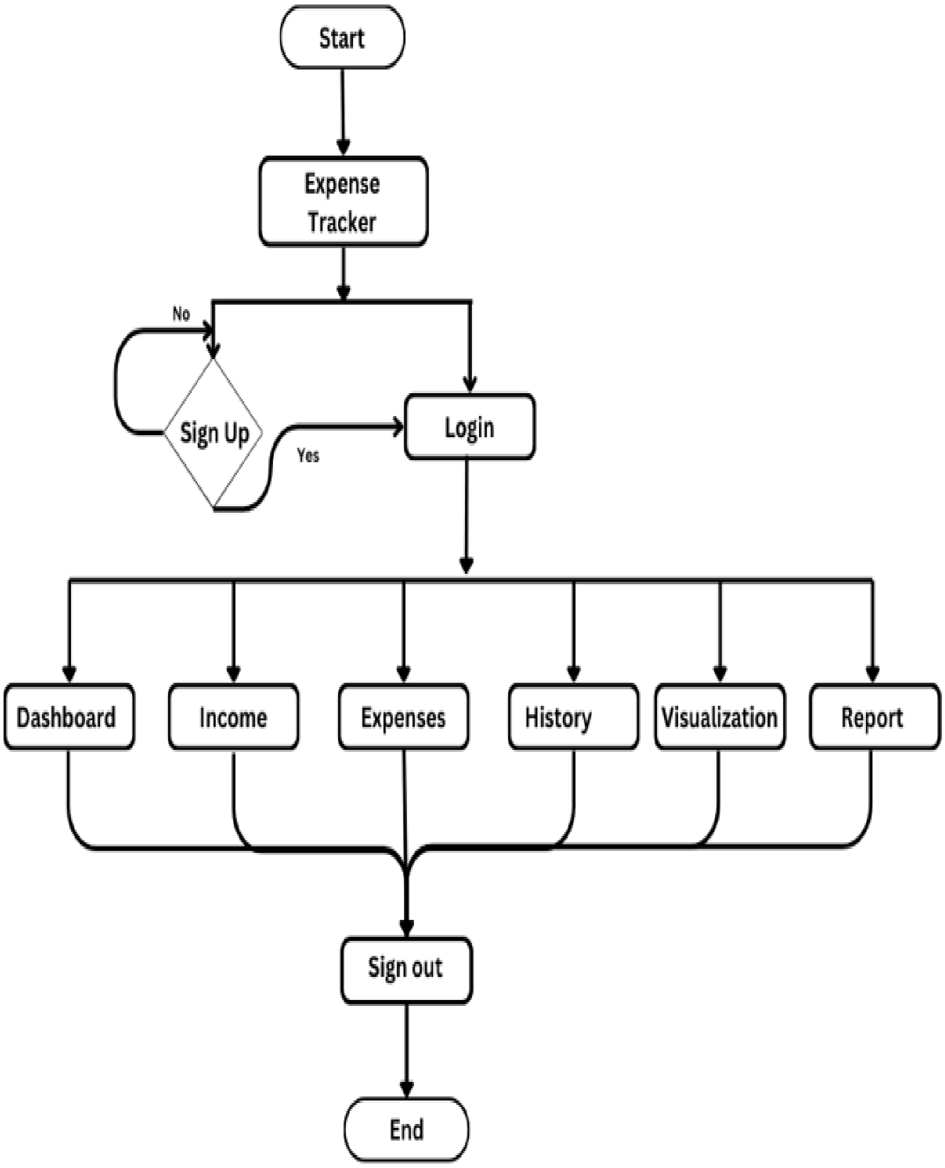


FIGURE 4.1 FLOW CHART

Illustrating the logical flow and interactions within the User Module, showing steps from user authentication.

4.1.1.2 USE CASE DIAGRAM

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are set of actions, services, and functions that the system needs to perform. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. The use cases are represented by either circles or ellipses. Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders shown in FIGURE 4.2 USE CASE DIAGRAM.

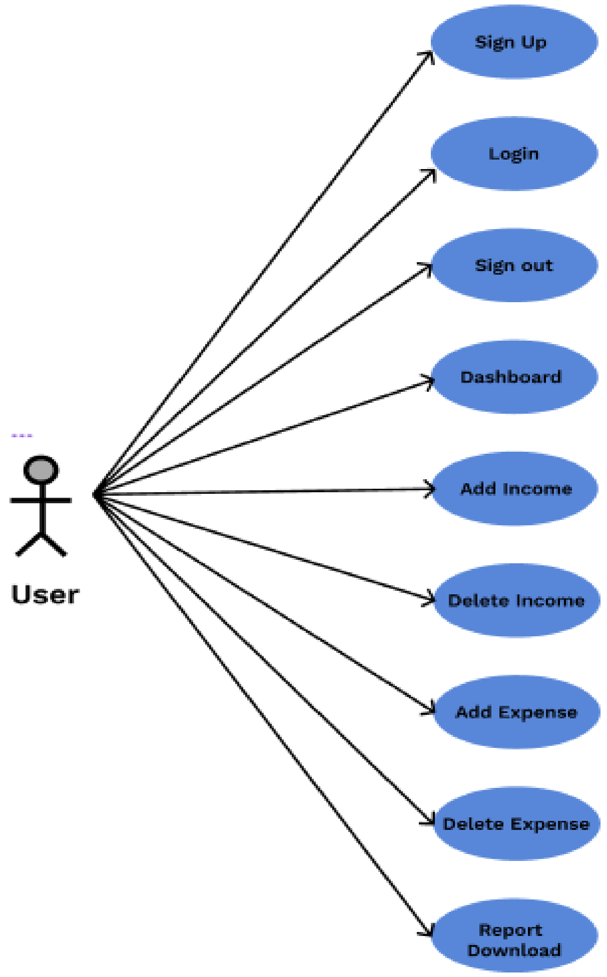


FIGURE 4.2 USE CASE DIAGRAM

Illustrating the interactions within the User Module, showcasing user actions.

4.1.1.3 SEQUENCE DIAGRAM

Sequence chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. The admin defines the sequence of an object during its lifetime shown in FIGURE 4.3 SEQUENCE DIAGRAM.

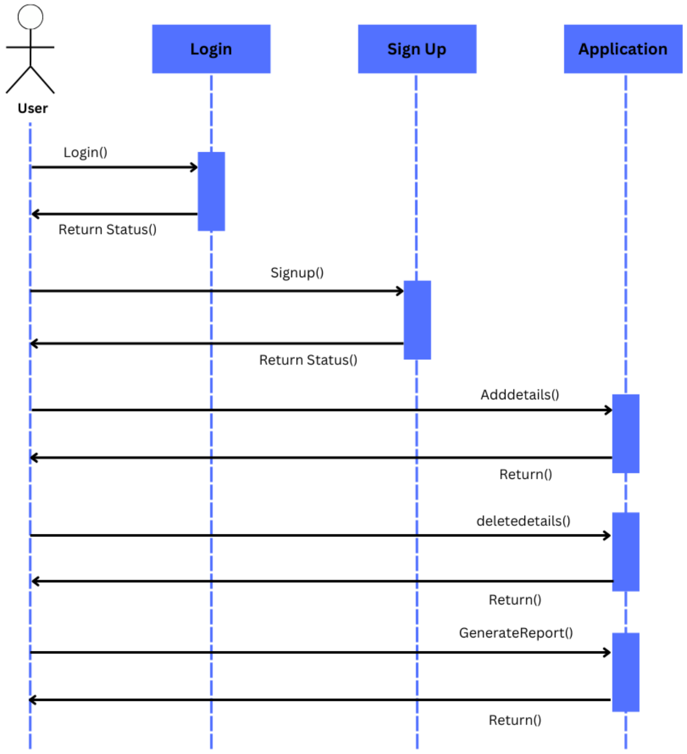


FIGURE 4.3 SEQUENCE DIAGRAM

Illustrating the interaction flow within the User Module, detailing the sequence from user login to generating reports.

4.2 INTERFACE DESIGN

The application provides various interfaces to the users that make it more convenient to manage their expenses. The user interface contains the Home page, where users can view an overview of their finances, including income, expenses, and budget usage. The application also features a section to track detailed income and expenses, making it easier for users to record and categorize transactions. Further, users can generate and view detailed reports on their financial data, helping them to monitor their spending and savings. The design is simple to use and clear to understand, making it easy for users to track their financial activities.

LOGIN PAGE:

The Login page serves as the entry point to the Expense Tracker Application shown in FIGURE 4.1 LOGIN PAGE.

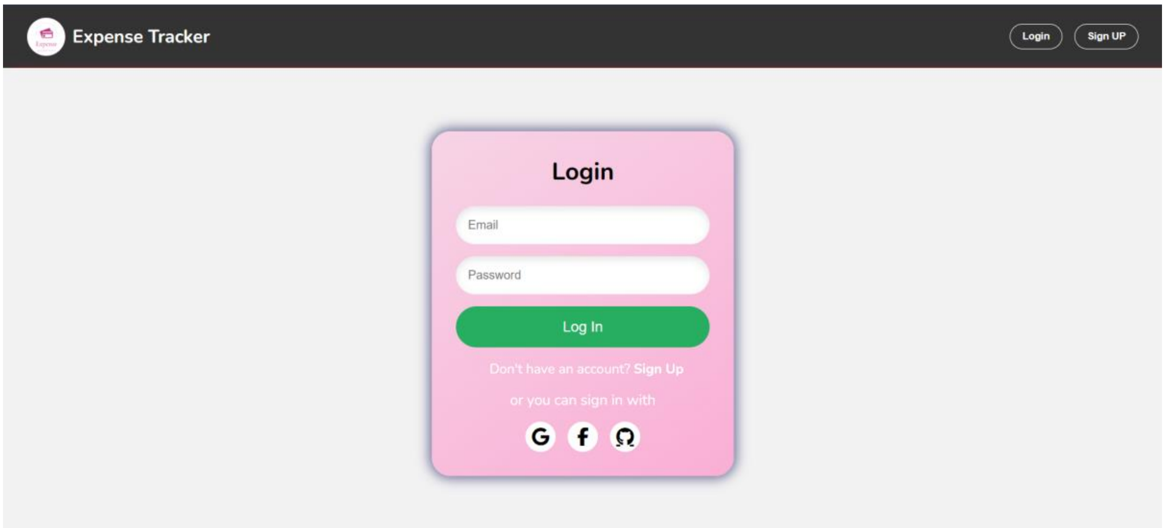


FIGURE 4.1 **LOGIN PAGE**

The Login page serves as the entry point to the Expense Tracker Application.

REGISTRATION PAGE:

The Registration page allows new user to create an account shown in FIGURE 4.2 REGISTRATION PAGE.

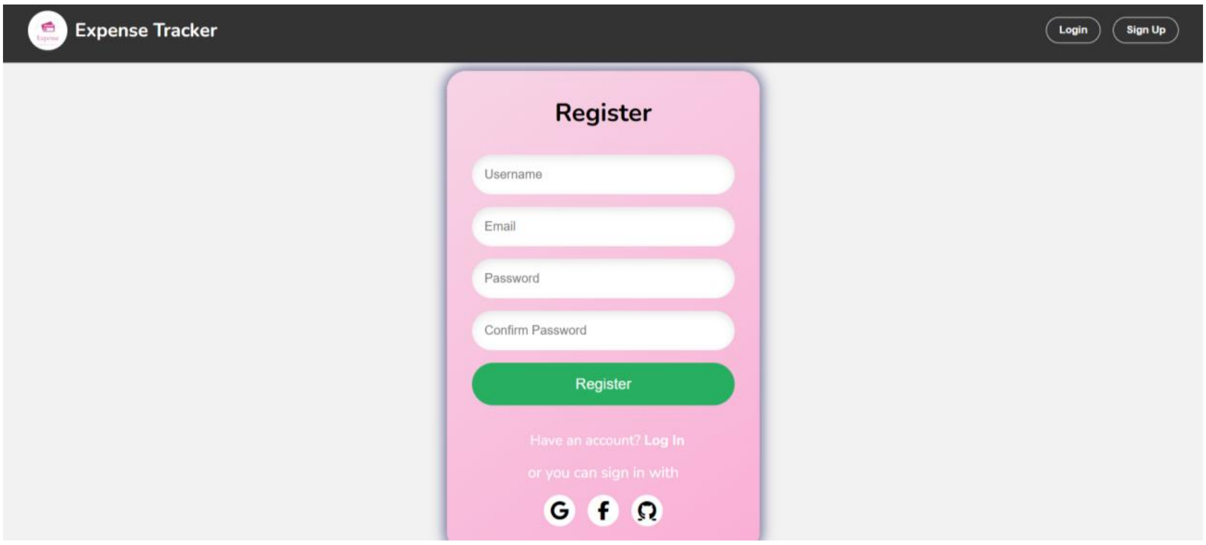
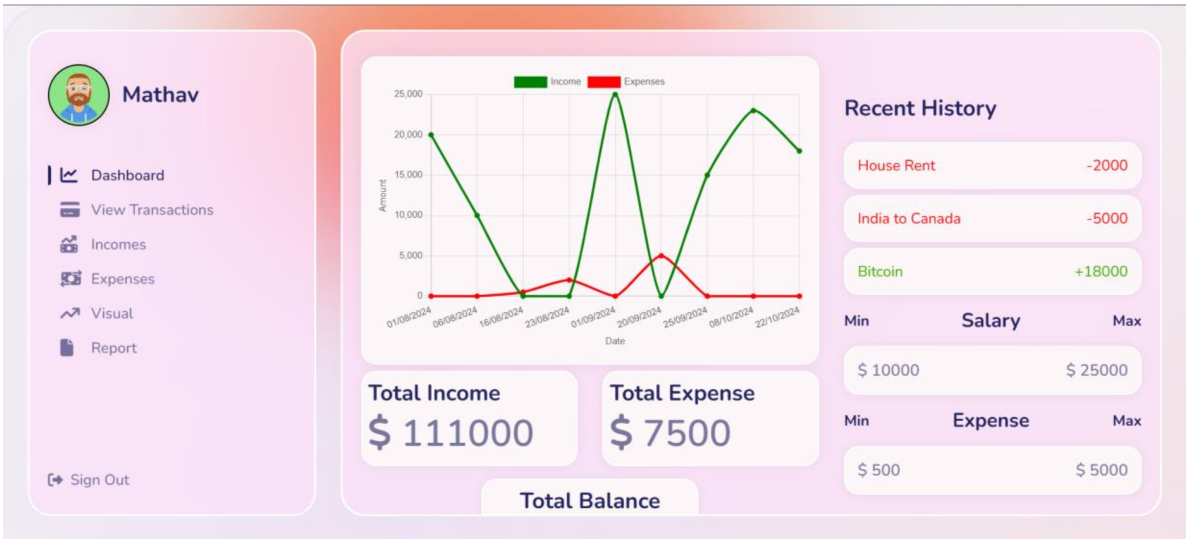


FIGURE 4.2 REGISTRATION PAGE

The Registration page allows new user to create an account.

DASHBOARD:

The Dashboard provides a comprehensive view of the user's financial data shown in FIGURE 4.3 DASHBOARD



Total Income

$ 111000

Total Expense

$ 7500

Total Balance

Recent History

House Rent	-2000
India to Canada	-5000
Bitcoin	+18000

Min	Salary	Max
$ 10000		$ 25000

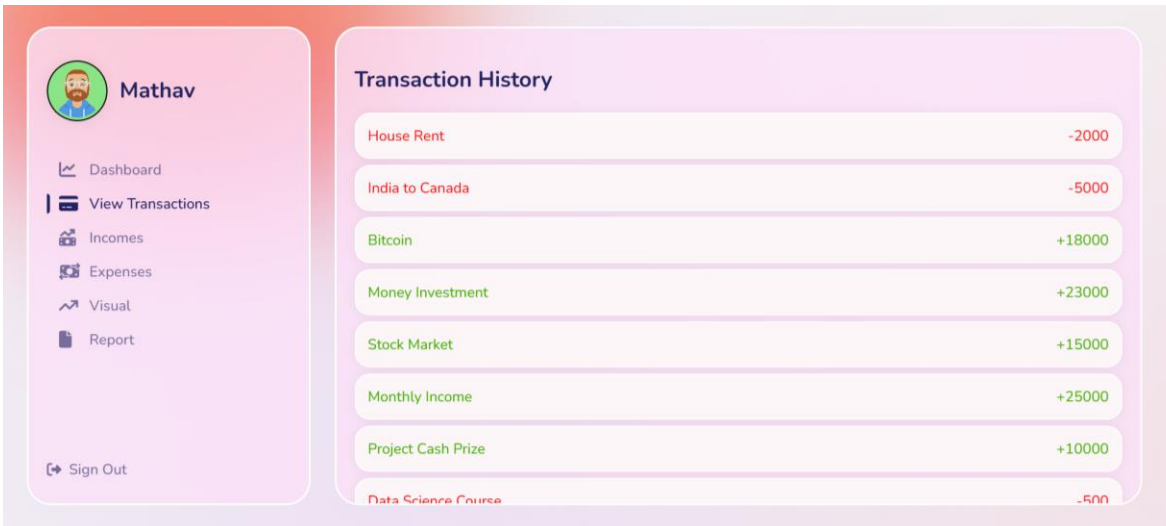
Min	Expense	Max
$ 500		$ 5000

FIGURE 4.3 DASHBOARD

The Dashboard provides a comprehensive view of the user's financial data.

TRANSACTION PAGE:

The Transactions page offers a detailed view of all financial transactions shown in FIGURE 4.3 TRANSACTION PAGE



Transaction History

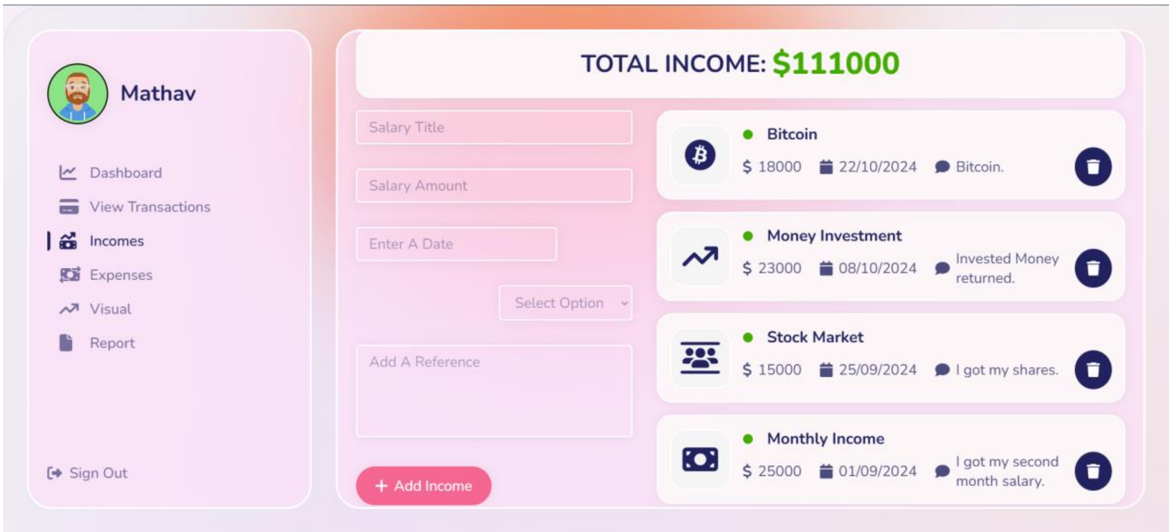
House Rent	-2000
India to Canada	-5000
Bitcoin	+18000
Money Investment	+23000
Stock Market	+15000
Monthly Income	+25000
Project Cash Prize	+10000
Data Science Course	-500

FIGURE 4.3 TRANSACTION PAGE

The Transactions page offers a detailed view of all financial transactions.

INCOME PAGE:

The Income page allows users to record new income sources and amounts shown in FIGURE 4.4 INCOME PAGE



TOTAL INCOME: $111000

Bitcoin
$ 18000 22/10/2024 Bitcoin.

Money Investment
$ 23000 08/10/2024 Invested Money returned.

Stock Market
$ 15000 25/09/2024 I got my shares.

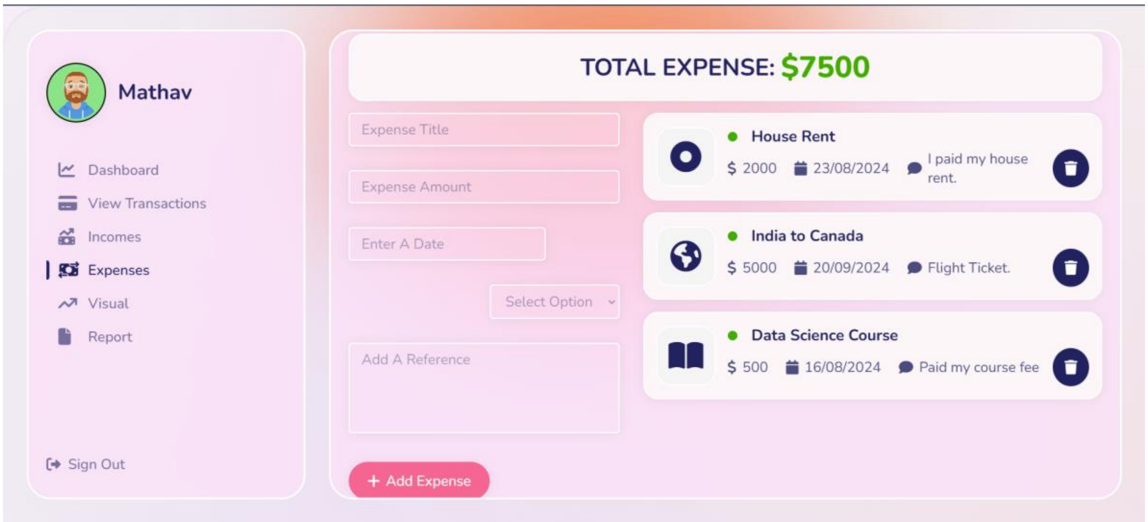
Monthly Income
$ 25000 01/09/2024 I got my second month salary.

FIGURE 4.4 INCOME PAGE

The Income page allows users to record new income sources and amounts.

EXPENSES PAGE:

The Expense page enables users to log new expenses shown in FIGURE 4.5 EXPENSE PAGE



TOTAL EXPENSE: $7500

House Rent
$ 2000 23/08/2024 I paid my house rent.

India to Canada
$ 5000 20/09/2024 Flight Ticket.

Data Science Course
$ 500 16/08/2024 Paid my course fee

FIGURE 4.5 EXPENSES PAGE

The Expense page enables users to log new expenses.

4.3 DATABASE DESIGN

The database used in this system is **MongoDB**. In MongoDB, the data is stored in the form of collections and documents. The **Income** collection stores the income details entered by users, allowing for operations such as storing, updating, deleting, and retrieving income records, which include attributes like amount, source, date, and category. Similarly, the **Expense** collection is dedicated to managing the expense records, supporting the same operations for expense details like amount, category, date, and payment method. The **User** collection stores personal information such as name, email, and password, which are essential for user authentication and account management shown in **FIGURE 4.4** DATABASE DESIGN.

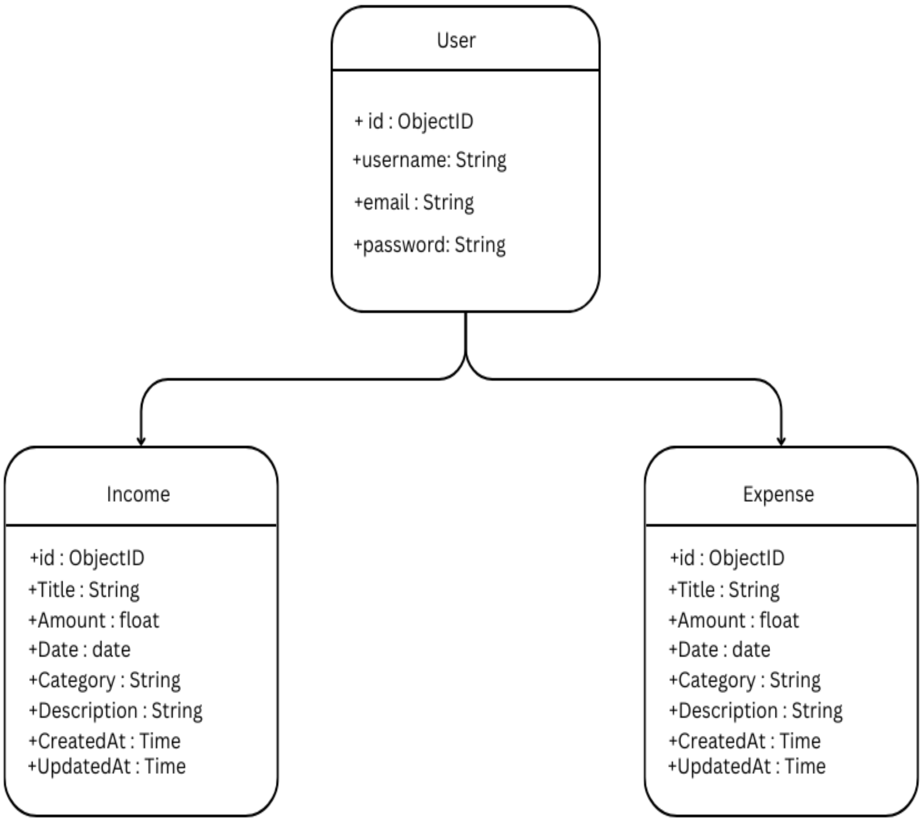
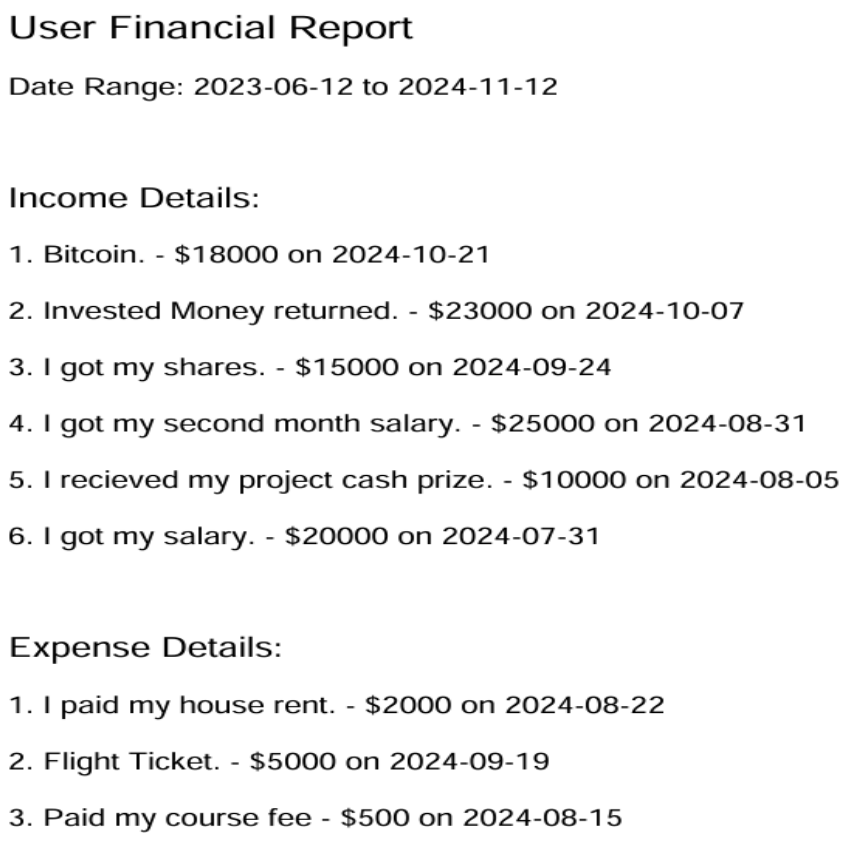


FIGURE 4.4 DATABASE DESIGN

Illustrating the structure and relationships between data.

4.4 OUTPUT DESIGN

Output design refers to the process of creating the results and information generated by the system for end-users. It focuses on presenting the data in a clear, understandable, and visually appealing format. In the case of the expense tracker, output design plays a crucial role in ensuring that the financial reports, such as income vs. expenses, budget status, and expense categories, are easily comprehensible and actionable. These reports are generated based on the user’s input and are presented in a user-friendly manner, which allows the user to make informed decisions regarding their finances. Previewing the output reports before finalizing or exporting them is an important feature, as the user is the ultimate judge of the quality and usefulness of the output. A well-designed output ensures that users have a positive experience and that the system’s functionality is successfully aligned with their needs.



User Financial Report

Date Range: 2023-06-12 to 2024-11-12

Income Details:

1. Bitcoin. - $18000 on 2024-10-21
2. Invested Money returned. - $23000 on 2024-10-07
3. I got my shares. - $15000 on 2024-09-24
4. I got my second month salary. - $25000 on 2024-08-31
5. I recieved my project cash prize. - $10000 on 2024-08-05
6. I got my salary. - $20000 on 2024-07-31

Expense Details:

1. I paid my house rent. - $2000 on 2024-08-22
2. Flight Ticket. - $5000 on 2024-09-19
3. Paid my course fee - $500 on 2024-08-15

CHAPTER 5
TESTING

5.1 UNIT TESTING

Unit testing is a fundamental software development process where the smallest testable components, known as units, are individually assessed to verify their proper functioning. This testing can be performed manually, but it is often automated to ensure accuracy and efficiency. In the context of the Smart Expense Tracker, each module, such as the user authentication, income/expense management, and budget tracking features, will be tested to ensure they meet the specified requirements. If the units perform as expected, they will proceed to integration testing, where multiple modules are combined and tested together for their compatibility and functionality.

5.2 REGRESSION TESTING

Regression testing is critical for identifying any unexpected side effects or issues arising from updates to the application code, modifications in dependencies, or infrastructure changes. This testing involves rerunning existing test cases to ensure that previously working features have not been negatively impacted by the recent changes. For the Smart Expense Tracker, regression testing ensures that features like expense tracking, budget management, and report generation continue to function seamlessly, even after system updates. The primary goal is to maintain the website's functionality and reliability, ensuring no unintended disruptions are introduced during enhancements or bug fixes.

5.3 VALIDATION TESTING

Validation testing is performed to verify that the software meets the business requirements and fulfills the needs of the end users. This process ensures that the application accurately captures user data, provides insightful visualizations, and allows users to manage their finances effectively. Validation testing confirms that the platform is deployed in the correct environment and performs as expected, aligning with the client's specifications and user expectations.

5.4 VERIFICATION TESTING

Verification testing is aimed at ensuring that each development phase or module meets the defined requirements. At each stage of the Smart Expense Tracker's development, individual modules (such as income input, expense categorization, and reporting features) will be verified to ensure they function as intended. This process helps detect issues early and ensures that each component is consistent with its design specifications before proceeding to the next stage.

5.5 INTEGRATION TESTING

Integration testing is conducted to test how various software components work together as a whole. In the case of the Smart Expense Tracker, this involves testing the interaction between different modules like user login, income tracking, expense management, and report generation. Integration testing aims to identify issues that may arise when these modules are combined, ensuring that the integrated system works as expected. The testing will begin after the individual modules are completed and integrated, ensuring a smooth experience for users interacting with multiple features of the platform.

CHAPTER 6

CONCLUSION AND FUTURE WORK

CONCLUSION:

The development of the Smart Expense Tracker represents a significant advancement in helping individuals manage their finances and achieve their financial goals. By offering an intuitive platform where users can track income, expenses, set budgets, and generate insightful reports, this project meets the growing demand for digital personal finance management. With seamless data input, real-time visualization, and user-friendly features, the Smart Expense Tracker enables users to monitor their financial health, make informed decisions, and control their spending effectively.

FUTURE WORK:

Looking ahead, there are numerous opportunities to expand and enhance the platform. Future work will focus on adding advanced features, such as:

1. AI-Driven Insights: Providing personalized budgeting advice using artificial intelligence to help users make smarter financial decisions.
2. Bank Account Synchronization: Integrating with banks to enable automatic transaction tracking, reducing the need for manual data entry.
3. Machine Learning Algorithms: Implementing algorithms to detect unusual spending patterns and identify potential savings opportunities.
4. Mobile Application Development: Expanding accessibility through mobile applications and multi-device synchronization for improved convenience.
5. Financial Goal Tracking and Integration: Adding features for financial goal setting, along with integration options with other financial tools, and possibly expanding the platform to support business expense management.

APPENDIX 1

CODING:

App.js:

```
import styled from "styled-components";

import { BrowserRouter, Routes, Route } from
"react-router-dom";

import bg from "./img/bg.png";

import { MainLayout } from "./styles/Layouts";

import Orb from "./Components/Orb/Orb";

import Navigation from
"./Components/Navigation/Navigation";

import React, { useState, useMemo } from "react";

// import Dashboard from
"./Components/Dashboard/Dashboard";

import Income from
"./Components/Incomes/Incomes";

import Expenses from
"./Components/Expenses/Expenses";

import { useGlobalContext } from
"./context/globalContext";

import ViewTransaction from
"./Components/ViewTransaction/ViewTransaction
";

import Login from "./Components/Auth/Login";

import Dashboard from
"./Components/Dashboard/Dashboard";
```

```
import Main from "./Main";

function App() {

  return (

    <>

      <BrowserRouter>

        <Routes>

          <Route path="/" element={<Login />} />

          <Route path="/dashboard" element={<Main />} />

          <Route path="/Register" element={<Register />} />

        </Routes>

      </BrowserRouter>

    </>

  );

}

const AppStyled = styled.div`

  height: 100vh;

  background-image: url(${(props) => props.bg});

  position: relative;

  main {

    flex: 1;

    background: rgba(252, 226, 249, 0.78);
    backdrop-filter: blur(4.5px);
    border-radius: 32px;
    overflow-x: hidden;
    &::-webkit-scrollbar {
      width: 0;
    }
  }
`;
export default App;
```

Dashboard:

```
import React, { useEffect } from "react";
import styled from "styled-components";
import { useGlobalContext } from "../../context/globalContext";
import History from "../../History/History";
import { InnerLayout } from "../../styles/Layouts";
import { dollar } from "../../utils/Icons";
import Chart from "../Chart/Chart";
function Dashboard() {
  const {
    totalExpenses,
    incomes,
    expenses,
    totalIncome,
    totalBalance,
  } = useGlobalContext();
```

```
useEffect(() => {
  getIncomes();
  getExpenses();
}, []);
return (
  <DashboardStyled>
    <InnerLayout>
      <div className="stats-con">
        <div className="chart-con">
          <Chart />
          <div className="amount-con">
            <div className="income">
              <h2>Total Income</h2>
              <p>
                {dollar} {totalIncome()}
              </p>
            </div>
            <div className="expense">
              <h2>Total Expense</h2>
              <p>
                {dollar} {totalExpenses()}
              </p>
            </div>
            <div className="balance">
              <h2>Total Balance</h2>
              <p>
                {dollar} {totalBalance()}
              </p>
            </div>
```

```
</div>
<div className="history-con">
  <History />
  <h2 className="salary-title">
    Min <span>Salary</span>Max
  </h2>
  <div className="salary-item">
    <p>
      $
      {incomes.length > 0
        ? Math.min(...incomes.map((item) => item.amount))
        : 0}
    </p>
    <p>
      $
      {incomes.length > 0
        ? Math.max(...incomes.map((item) => item.amount))
        : 0}
    </p>
  </div>
  <h2 className="salary-title">
    Min <span>Expense</span>Max
  </h2>
  <div className="salary-item">
    <p>
      $
      {expenses.length > 0
        ? Math.min(...expenses.map((item) => item.amount))
```

```
              <p>
                $
                {expenses.length > 0
                  ? Math.max(...expenses.map((item) => item.amount))
                  : 0}
              </p>
            </div>
          </div>
        </div>
      </InnerLayout>
    </DashboardStyled>
  );
}
const DashboardStyled = styled.div`
  .stats-con {
    display: grid;
    grid-template-columns: repeat(5, 1fr);
    gap: 2rem;
    .chart-con {
      grid-column: 1 / 4;
      height: 400px;
      .amount-con {
        display: grid;
        grid-template-columns: repeat(4, 1fr);
        gap: 2rem;
        margin-top: 0.5rem;
        .expense {
```

```
    .income,
    .expense,
    .balance {
      background: #fcf6f9;
      border: 2px solid #ffffff;
      box-shadow: 0px 1px 15px rgba(0, 0, 0, 0.06);
      border-radius: 20px;
      padding: 0.5rem;
      p {
        font-size: 3rem;
        font-weight: 600;
      }
    }
    .balance {
      grid-column: 2 / 4;
      display: flex;
      flex-direction: column;
      justify-content: center;
      margin-top: -1rem;
      align-items: center;
      p {
        color: var(--color-green);
        opacity: 0.6;
        font-size: 3rem;
      }
}
.history-con {
  grid-column: 4 / -1;
```

```
   h2 {
     margin: 1rem 0;
     display: flex;
     align-items: center;
     justify-content: space-between;
   }
   .salary-title {
     font-size: 1.2rem;
     span {
       font-size: 1.6rem;
     }
   }
   .salary-item {
     background: #fcf6f9;
     border: 2px solid #ffffff;
     box-shadow: 0px 1px 15px rgba(0, 0, 0, 0.06);
     padding: 1rem;
     border-radius: 20px;
     display: flex;
     justify-content: space-between;
     align-items: center;
     p {
       font-weight: 600;
       font-size: 1.3rem;
     }
  }
`;
export default Dashboard;
```

Route.js:

```
const {
  addExpense,
  getExpense,
  deleteExpense,
} = require("../controllers/expense");
const {
  addIncome,
  getIncomes,
  deleteIncome,
} = require("../controllers/income");
const { authdetails } =
require("../controllers/auth.control");
const { authlogin } =
require("../controllers/auth.control");
const router = require("express").Router();
router
  .post("/add-income", addIncome)
  .get("/get-incomes", getIncomes)
  .delete("/delete-income/:id", deleteIncome)
  .post("/add-expense", addExpense)
  .get("/get-expense", getExpense)
  .delete("/delete-expense/:id", deleteExpense)
```

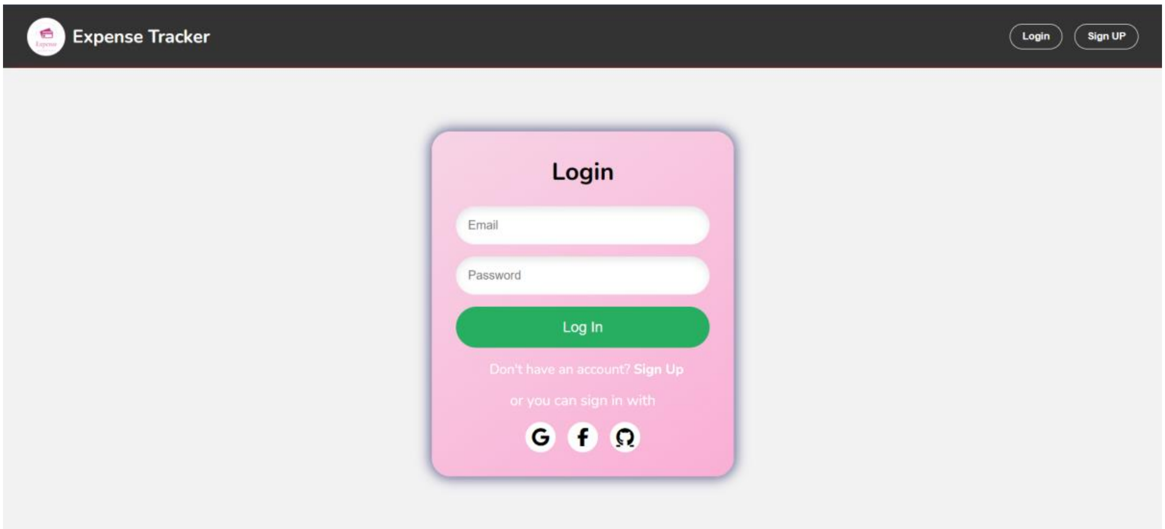
Expense.js:

```
const ExpenseSchema = require("../models/expenseModel")
exports.addExpense = async (req,res) => {
    const { title, amount, category, description, date } = req.body
    const income = ExpenseSchema({
        title,
        amount,
        category,
        description,
        date
    })

    try
    {
        if(!title || !category || !description || !date)
        {
            return res.status(400).json({Message: 'All fields are required'})
        }
        if(amount <=0 || !amount === 'number')
        {
            return res.status(400).json({Message: 'Amount must be a positive number'})
        }
        await income.save()
        res.status(200).json({message: 'Expense Added'})
    }catch(error)
    {
    console.log(income)
};
```

```
exports.getExpense = async (req,res) =>
{
   try {
      const incomes = await ExpenseSchema.find().sort({createdAt: -1})
      res.status(200).json(incomes)
   } catch (error)
   {
      res.status(500).json({message: 'Server Error' })
   }
}
exports.deleteExpense = async (req,res) =>
{
   const {id} = req.params;
   ExpenseSchema.findByIdAndDelete(id)
   .then((income) =>
   {
      res.status(200).json({message: 'Expense Deleted'})
   })
   .catch((err) =>
   {
      res.status(500).json({message: 'Server Error' })
   })
}
```

APPENDIX 2
SNAPSHOTS



Expense Tracker

Login

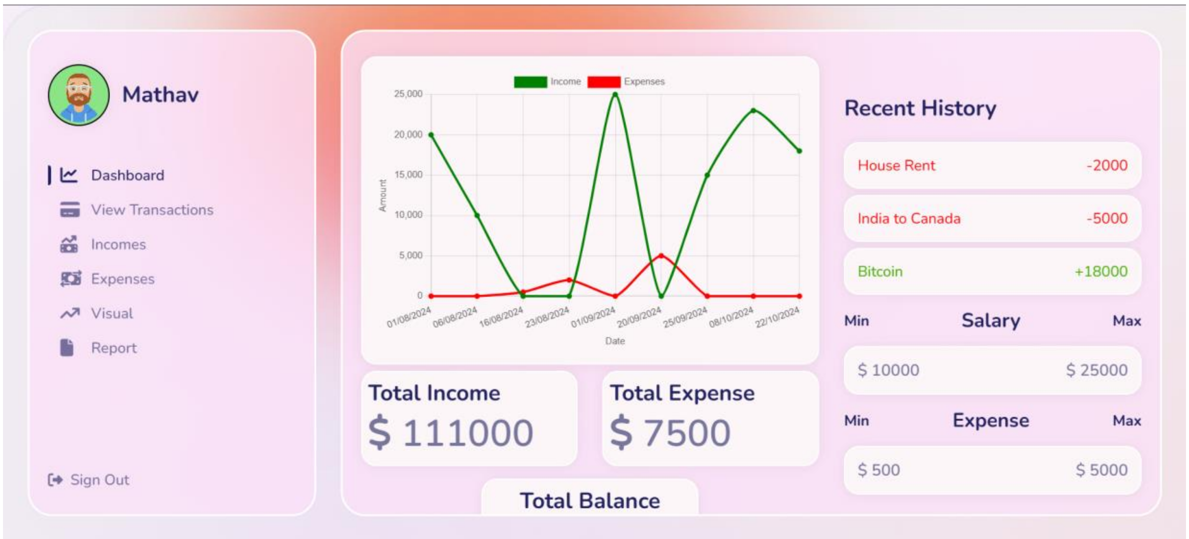
Email

Password

Don't have an account? Sign Up

or you can sign in with

FIGURE A2.0 LOGIN PAGE



Mathav

Total Income

$ 111000

Total Expense

$ 7500

Total Balance

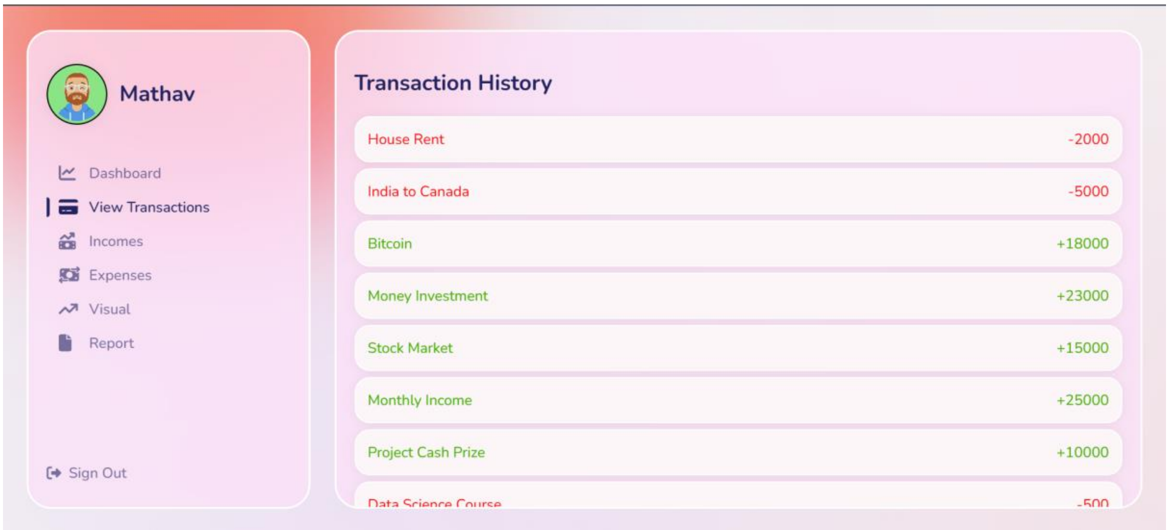
Recent History

House Rent	-2000
India to Canada	-5000
Bitcoin	+18000

Min	Salary	Max
$ 10000		$ 25000

Min	Expense	Max
$ 500		$ 5000

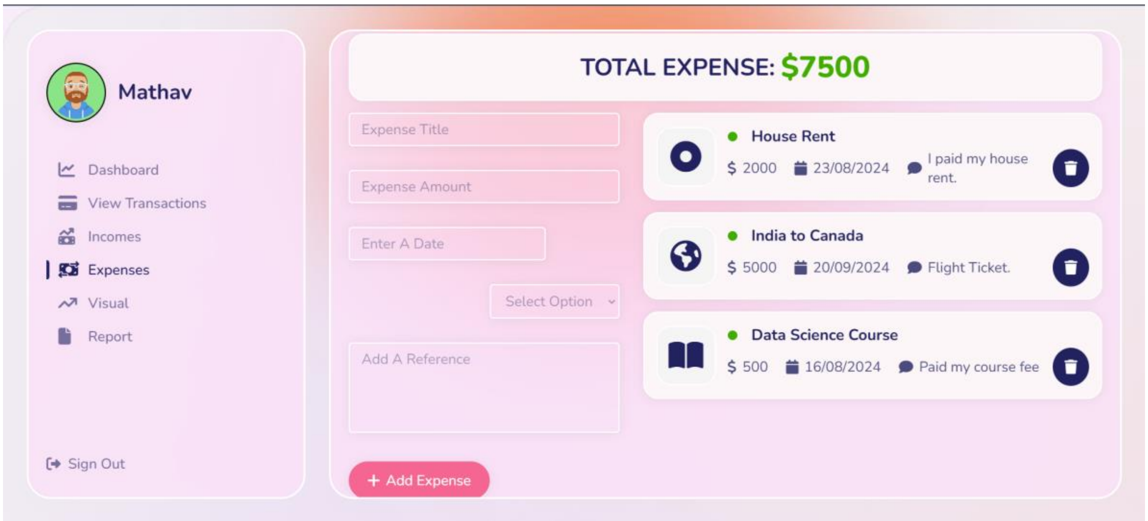
FIGURE A2.1 DASHBOARD



Transaction History

House Rent	-2000
India to Canada	-5000
Bitcoin	+18000
Money Investment	+23000
Stock Market	+15000
Monthly Income	+25000
Project Cash Prize	+10000
Data Science Course	-500

FIGURE A2.2 TRANSACTION PAGE



TOTAL EXPENSE: $7500

- House Rent — $ 2000 — 23/08/2024 — I paid my house rent.
- India to Canada — $ 5000 — 20/09/2024 — Flight Ticket.
- Data Science Course — $ 500 — 16/08/2024 — Paid my course fee

FIGURE A2.3 EXPENSE PAGE

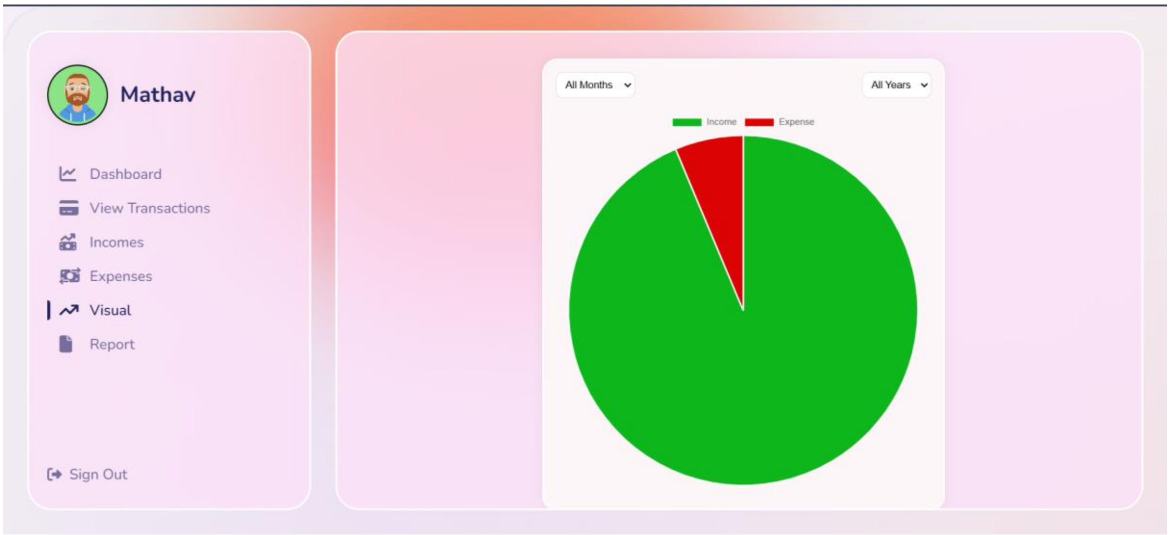
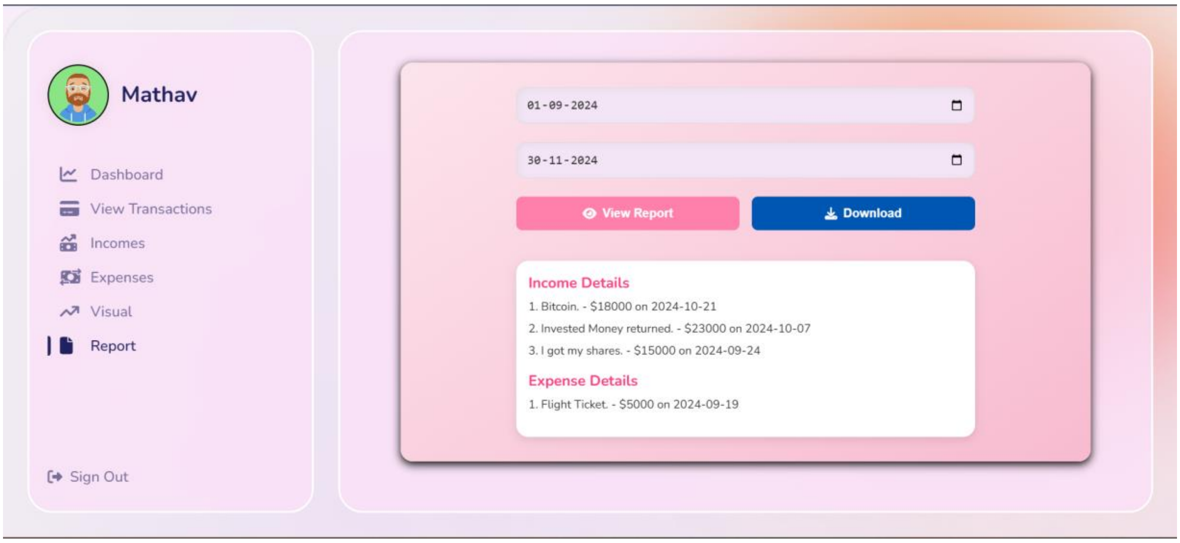


FIGURE A2.4 VISUAL PAGE



01-09-2024

30-11-2024

Income Details

1. Bitcoin. - $18000 on 2024-10-21
2. Invested Money returned. - $23000 on 2024-10-07
3. I got my shares. - $15000 on 2024-09-24

Expense Details

1. Flight Ticket. - $5000 on 2024-09-19

FIGURE A2.5 REPORT PAGE

REFERENCES

1. Fisher, R. (2018). *Building Scalable Web Applications with Node.js and MongoDB.* Publishing.

2. Gormley, C., & Tong, W. (2018). *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage.* O'Reilly Media.

3. Kim, H., & Yoo, J. (2021). "A Study on the Use of Firebase for Scalable Web Applications." *Proceedings of the International Conference on Web Technologies*, 34(5), 150-159.

4. Mehra, M., Kumar, M., Maurya, A., & Sharma, C. (2021). "MERN Stack Web Development." *Annals of the Romanian Society for Cell Biology*, 25(6), 11756-11761.

5. Smith, J., & Thomas, M. (2019). "Building Personal Finance Tools: A Guide to Expense Tracking Applications." *Journal of Financial Technology*, 15(3), 72-80.

6. Johnson, L., & Williams, P. (2020). *Advanced Data Visualization Techniques for Financial Analysis.* Wiley.

7. Patel, S., & Desai, R. (2019). "Utilizing MongoDB Atlas for Scalable Financial Applications." *International Journal of Database Management Systems*, 11(4), 88-96.

8. Kaur, G., & Singh, A. (2021). "ReactJS for Building Responsive User Interfaces: A Comparative Study." *Journal of Software Engineering and Applications*, 14(2), 75-82.

9. Roberts, D. (2020). *Practical React and Redux for Web Applications.* Apress.

10. Lee, S., & Park, H. (2022). "Enhancing User Experience in Financial Apps with Real-time Data Visualization." *International Journal of Financial Technologies*, 18(1), 102-110.