WEB-BASED EXPENSE TRACKER WITH HANDWRITTEN TEXT RECOGNITION USING TESSERACT OCR

Hemalatha. S[1], Harish. T[2], Jagadeeshwaran. M[3], Maniraja. S[4]

[1]Professor, Department of Artificial Intelligence and Machine Learning, Sri Shakthi Institute of Engineering and Technology, An Autonomous Institution, Coimbatore-641062, India.

[2,3,4]Bachelor of Technology in Artificial Intelligence and Machine Learning, Second year, Sri Shakthi Institute of Engineering and Technology, An Autonomous Institution, Coimbatore-641062, India.

DOI: https://www.doi.org/10.58257/IJPREMS38320

ABSTRACT

In today’s fast-paced and digital-driven world, effectively managing personal finances has become both essential and challenging. The Expense Tracker project offers a comprehensive yet user-friendly web-based solution to help users seamlessly record, monitor, and analyze their expenses. Built using Flask for backend development, Chart.js for interactive data visualization, and Tesseract OCR for text recognition, the platform supports both manual input and handwritten text entry via a digital canvas. This flexibility ensures accessibility and convenience for users with diverse preferences.

The system goes beyond basic tracking by categorizing expenses, generating monthly summaries, and providing actionable financial insights through real-time visual analytics. By integrating affordability with innovative features, the Expense Tracker empowers individuals and small businesses to take control of their finances, fostering improved financial literacy and informed decision-making in today’s dynamic economic environment.

Keywords: Expense Tracking, Handwritten Text Recognition, Tesseract OCR, Flask, Data Visualization

1. INTRODUCTION

Managing personal finances is a critical aspect of modern life as individuals and small businesses navigate increasingly complex financial challenges. Traditional methods of tracking expenses, such as using pen and paper or spreadsheets, are often inefficient, time-consuming, and unsuitable for today’s fast-paced environment. Furthermore, many existing digital solutions either lack essential features or come with high costs, making them inaccessible for a broader audience. These limitations highlight the need for an affordable, user-friendly platform that simplifies expense tracking and empowers users to manage their finances effectively.

The Expense Tracker system is designed to bridge this gap by providing a web-based application that combines functionality with an intuitive user experience. Built using modern web technologies such as Flask for backend development, Chart.js for interactive data visualization, and Tesseract OCR for advanced handwritten text recognition, the platform offers robust features tailored to user needs. It supports both manual expense entry and OCR-based handwritten input, ensuring flexibility and convenience for users with diverse preferences. With capabilities such as expense categorization, monthly summaries, and real-time analytics, this system not only simplifies expense tracking but also provides actionable insights to help users make informed financial decisions. This paper delves into the system's development process, underlying technologies, and key features that make it an innovative solution for personal expense management.

2. METHODOLOGY

2.1 Backend Development

The backend of the Expense Tracker was built using Flask, a lightweight and flexible Python-based web framework that supports the development of secure, scalable, and efficient web applications. Flask was chosen for its simplicity and modularity, enabling the seamless implementation of core functionalities such as user authentication, data management, and API routing. Security was a primary focus in designing the user registration and login processes, with password hashing mechanisms employed to protect sensitive information. Flask's session management capabilities were used to maintain secure user sessions, ensuring a smooth and reliable user experience.

API endpoints were developed to facilitate communication between the frontend and backend, handling essential operations like processing user requests, managing database interactions, and executing operations for expense records. These endpoints also support advanced functionalities such as expense categorization and real-time data retrieval. Flask's integration capabilities further enabled the incorporation of third-party technologies like Tesseract OCR for text recognition and Chart.js for data visualization, resulting in a cohesive and highly functional backend architecture.

2.2 Frontend Design

The frontend of the Expense Tracker was crafted using HTML, CSS, and JavaScript to deliver an intuitive, visually appealing, and user-friendly interface. A responsive design was prioritized to ensure seamless accessibility across different devices, including desktops, tablets, and smartphones, catering to a wide range of users. The interface featured clear input fields, interactive buttons, and dynamic visual components that simplified navigation and allowed users to input expenses effortlessly. The inclusion of a digital canvas enabled users to provide handwritten inputs, which were processed by the OCR system for added flexibility.

JavaScript played a key role in enhancing the interactivity and responsiveness of the platform. It was used to dynamically update visual elements, such as charts, ensuring that users could instantly view changes when new expenses were added. By leveraging these technologies, the frontend not only provided a seamless and engaging user experience but also effectively supported the integration of backend functionalities, creating a cohesive and robust expense management system.

2.3 Optical Character Recognition (OCR) Integration

The Expense Tracker integrated Tesseract OCR, an open-source text recognition tool, to enable efficient input of handwritten expenses. Users could utilize a digital canvas to write or draw expense details, including the name, amount, and date. The Tesseract OCR engine processed the handwritten input, converting it into structured text data that was seamlessly incorporated into the system. This feature significantly reduced the need for manual data entry, offering a convenient and flexible way to record expenses.

To enhance the accuracy of text recognition, preprocessing techniques were employed before passing the input to Tesseract. These techniques included noise reduction, contrast adjustment, and binarization, ensuring better handling of diverse handwriting styles and improving the reliability of the OCR system. By combining advanced text recognition with user-friendly functionality, the OCR integration added a layer of innovation to the platform, making it more accessible and efficient for users.

2.4 Database Management

SQLite was chosen as the database for its lightweight architecture, ease of integration, and efficient data handling capabilities, making it ideal for the Expense Tracker system. It acted as the central repository for storing various types of data, including user credentials, expense records, categories, and budget-related details. The database schema was meticulously designed to optimize performance during data retrieval and update operations. Logical relationships were established between tables, ensuring data integrity and enabling features like user-specific expense categorization and advanced filtering. To protect sensitive financial data, encryption techniques were applied to the database, providing an additional layer of security. This ensured that user information, including personal and financial details, was safeguarded against unauthorized access. The use of SQLite not only streamlined the system's operations but also supported scalability, making it suitable for both individual users and small businesses. Its robust design allowed the database to handle the diverse functionalities of the Expense Tracker efficiently.

2.5 Data Visualization

Chart.js, a powerful JavaScript-based library, was integrated into the Expense Tracker to create dynamic and interactive data visualizations. Key financial insights, such as expense trends over time, were represented using line charts, while pie charts provided a clear breakdown of spending across different categories. These visual elements enabled users to identify patterns in their spending habits, offering actionable insights to support better financial decision-making.

The frontend leveraged JavaScript to dynamically update these visualizations in real time whenever new expenses were added, edited, or categorized. This seamless interactivity ensured that users could instantly view changes and monitor their financial progress. By combining intuitive visuals with real-time updates, Chart.js enhanced the user experience and made complex financial data easily comprehensible.

2.6 System Scalability and Performance

The Expense Tracker was designed with scalability as a key consideration, ensuring that it could efficiently handle an increasing number of users and growing volumes of data over time. Flask’s modular and extensible architecture allowed for the easy addition of new features, such as advanced analytics and third-party service integrations, without compromising system performance. This approach ensured that the platform could evolve and expand to meet future needs while maintaining stability. SQLite, though a lightweight database, was carefully optimized to handle concurrent user operations without significant performance bottlenecks. The system's architecture was designed for high uptime and minimal latency, ensuring that users could interact with the platform smoothly even during periods of increased activity. By focusing on both scalability and performance, the Expense Tracker offers a reliable and efficient solution for managing personal finances at scale.

2.7 Security Measures

Security was a top priority throughout the development of the Expense Tracker to safeguard user data and ensure privacy. User credentials were securely stored using hashed passwords, utilizing industry-standard hashing algorithms to protect against unauthorized access. Session management techniques were implemented to prevent session hijacking and unauthorized access during user interactions.

To further enhance data protection, encryption was applied to sensitive financial records, both during storage and transmission, ensuring that user data remained secure at all times. The system adhered to relevant data privacy standards and regulations, ensuring compliance and maintaining user trust. These security measures combined to create a reliable and safe environment for managing sensitive financial information.

3. SOME IMPORTANT SOFTWARE USED AND ITS DESCRIPTION

3.1 Flask

Flask is a lightweight, flexible Python-based web framework that played a crucial role in the backend development of the Expense Tracker. Known for its simplicity and modularity, Flask allows developers to build secure and scalable web applications with ease. It provides essential tools and libraries to streamline web development, including those for handling HTTP requests, managing user sessions, and routing APIs.

Flask's modular architecture made it an ideal choice for the Expense Tracker project, as it allowed for easy integration with third-party libraries and technologies, such as Chart.js for data visualization and Tesseract OCR for text recognition. The framework enabled the implementation of secure user authentication, efficient database operations, and seamless communication between the frontend and backend. Its minimalistic nature also ensured that the system could remain lightweight while maintaining high performance and flexibility.

3.2 SQLite

SQLite is a lightweight, serverless relational database management system that was utilized for storing and managing key user data in the Expense Tracker. This included user credentials, expense records, categories, and budget-related information. SQLite's simplicity and efficiency made it an ideal choice for the project, as it provided fast data retrieval, reliable storage, and minimal overhead for managing local databases.

Its seamless integration with Flask ensured smooth backend operations, allowing for efficient interaction between the database and the web application. SQLite’s minimal setup and low maintenance requirements helped reduce development complexity, making it an optimal solution for the Expense Tracker's needs. Despite its lightweight nature, SQLite proved capable of handling the necessary data operations while maintaining a high level of performance.

3.3 Tesseract OCR

Tesseract OCR is an open-source optical character recognition engine developed and maintained by Google. It was integrated into the Expense Tracker to recognize and extract text from handwritten inputs on a digital canvas. By processing user-drawn data, Tesseract converted handwritten expense details into structured text fields such as the expense name, amount, and date.

This OCR functionality significantly improved the user experience by eliminating the need for manual data entry, making the platform more efficient and user-friendly. The ability to capture handwritten expenses through OCR not only added convenience but also made the system more accessible for users who preferred writing over typing, thereby enhancing overall usability.

3.4 Chart.js

Chart.js is a robust JavaScript library that enables the creation of interactive and visually engaging charts. In the Expense Tracker, it was used to visualize key financial data, including expense trends, category breakdowns, and budget comparisons. The library provides a wide range of chart types, such as line and pie charts, allowing users to easily understand their spending patterns and track financial goals.

One of the key features of Chart.js is its ability to support dynamic updates, ensuring that charts are automatically refreshed in real-time whenever users add or modify their expense data. This seamless integration of real-time visualization provided users with an interactive and informative experience, making it easier to make data-driven financial decisions.

3.5 HTML, CSS, and JavaScript

HTML, CSS, and JavaScript are foundational frontend technologies that were employed to build the user interface of the Expense Tracker. HTML provided the structure and layout for the content, ensuring a well-organized foundation for the platform. CSS was used to style and enhance the visual appearance of the application, creating an appealing and professional design. JavaScript added interactivity and dynamic features, making the platform more engaging and user-friendly. Together, these technologies enabled the creation of a responsive and intuitive interface that ensured seamless accessibility across various devices, including desktops, tablets, and smartphones. The combination of structured content, visually appealing design, and interactive elements provided a smooth user experience, enhancing overall functionality and ease of use.

3.6 Python

Python was the primary programming language used for the development of the Expense Tracker. Known for its versatility and ease of use, Python enabled the seamless integration of various technologies, including Flask for backend development and Tesseract for optical character recognition (OCR) processing. The language's robust ecosystem of libraries and frameworks simplified complex tasks such as data handling, text recognition, and data visualization, making development more efficient.

With its rich set of tools and frameworks, Python ensured that the Expense Tracker could be built with scalability, performance, and ease of maintenance in mind. This contributed to the overall success of the project, enabling the creation of a user-friendly and efficient platform that redefines how users manage and track their personal finances.

4. PROJECT ARCHITECTURE

The expense tracker web application is designed to facilitate the tracking of expenses through an innovative approach that leverages Optical Character Recognition (OCR) technology. Users can input their expense details, including the name and cost, by drawing directly on a canvas. This interactive process is supported by a frontend built with HTML, CSS, and JavaScript, which provides a user-friendly interface for expense input. The frontend features two canvases—one for the expense name and another for the cost—along with options to toggle between drawing and erasing. Users can also clear the canvas and initiate text recognition using OCR with a simple button click. Tesseract.js is employed to recognize the handwritten text from the canvases, and the recognized data is displayed in input fields for user confirmation before saving. The design is responsive, ensuring optimal usability on both desktop and mobile devices.

On the backend, the application is implemented using Flask or another web framework, which manages data processing and storage. It features REST API endpoints, such as /ocr_input (POST), which receive the recognized expense data from the frontend and save it to the database. The backend also includes data validation mechanisms to ensure that the received data is sanitized and properly formatted before storage. The database, which can be built using SQLite, MySQL, or PostgreSQL, is structured to store user expense records, including unique identifiers, expense names, costs, and timestamps.

The text recognition component utilizes Tesseract.js to process the canvas images and extract handwritten text, performing OCR processing on the client side. For data visualization, Chart.js is integrated into the application to present expense data in the form of charts, offering users valuable insights into their spending habits. The system flow begins with user input, where users draw their expense details on the canvases. Once the "Recognize" button is clicked, Tesseract.js processes the images to extract text, which is then validated on the client side to ensure correctness, particularly for numeric input related to costs. Upon clicking "Save Expense," the frontend sends a POST request to the backend with the recognized data, which is subsequently validated and stored in the database. Finally, the stored expense data is retrieved and visualized using Chart.js, providing users with a comprehensive overview of their expenses.

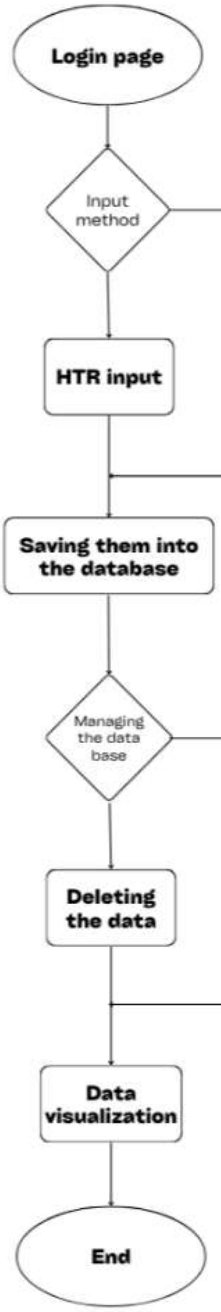


Figure 1: Project Flow

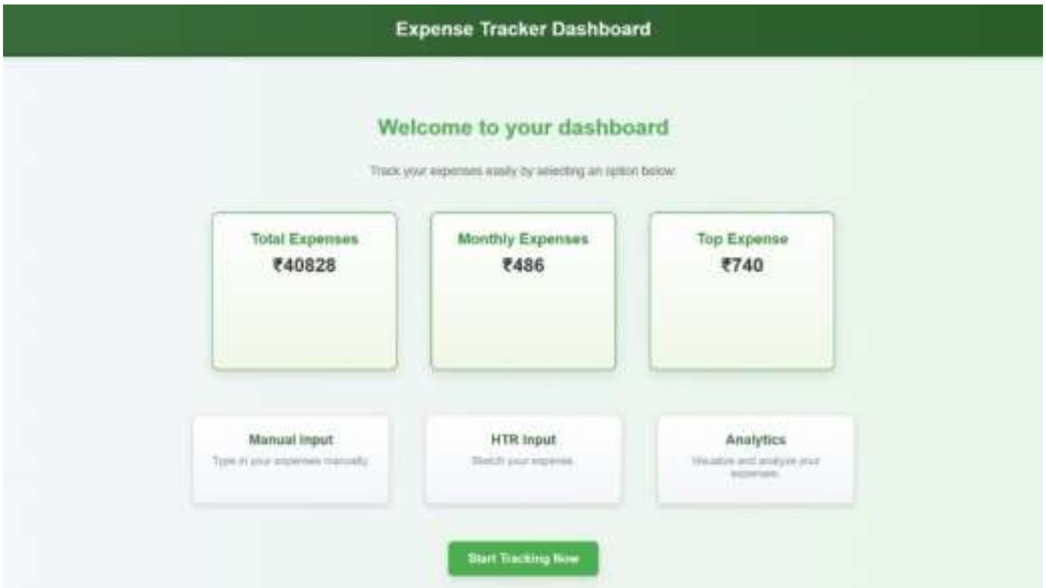
5. RESULTS AND DISCUSSION

The Expense Tracker project was developed to address the complexities of managing personal finances in a fast-paced world. By combining innovative technologies with a user-centered design, the system provided an effective solution for tracking, analyzing, and managing expenses. This section highlights the key results and discusses their implications.

5.1 User Interface and Experience

The frontend of the Expense Tracker was designed with a focus on simplicity, usability, and versatility, ensuring a smooth and intuitive user experience. Users could easily input their expenses manually or opt for handwritten data recognition, accommodating different user preferences and use cases. The integration of a digital canvas for handwritten input was a standout feature, enabling users to record their expenses in a natural and intuitive manner, making the platform more accessible.

Additionally, the responsive design ensured that the platform was fully accessible across various devices, including desktops, tablets, and smartphones. This responsiveness allowed users to manage their expenses conveniently on any device, ensuring that the user experience remained consistent and efficient, regardless of the platform. By prioritizing ease of use and accessibility, the system provided a seamless experience that catered to the needs of diverse users.



Expense Tracker Dashboard

Welcome to your dashboard

Track your expenses easily by selecting an option below:

Total Expenses ₹40828

Monthly Expenses ₹486

Top Expense ₹740

Manual Input — Type in your expenses manually.

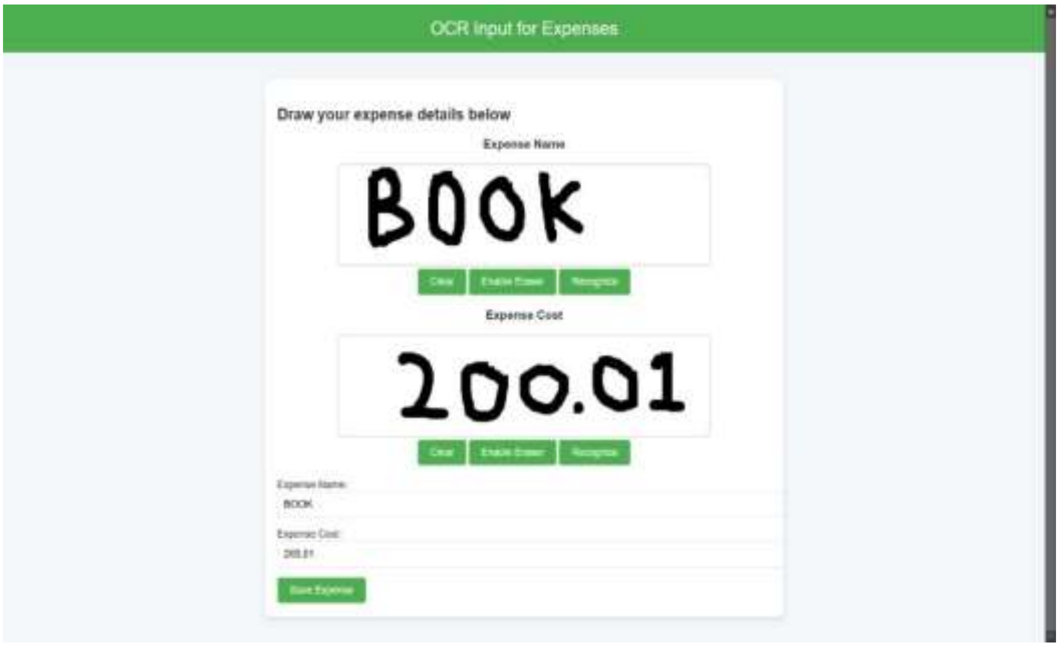
HTR Input — Sketch your expense.

Analytics — Visualize and analyze your expenses.

Figure 2: User Dashboard

5.2 Handwritten Text Recognition

One of the standout features of the system was the integration of Tesseract OCR for recognizing handwritten text. This feature significantly reduced the manual effort required to input expense details. Users could draw or write directly on a digital canvas, and Tesseract accurately extracted key details such as expense name, amount, and date. The system demonstrated high recognition accuracy for legible handwriting, with minor inaccuracies arising only in cases of extreme handwriting variations. This feature was particularly beneficial for users who preferred non-digital input methods.



OCR Input for Expenses

Draw your expense details below

Expense Name

BOOK

Expense Cost

200.01

Expense Name:
BOOK

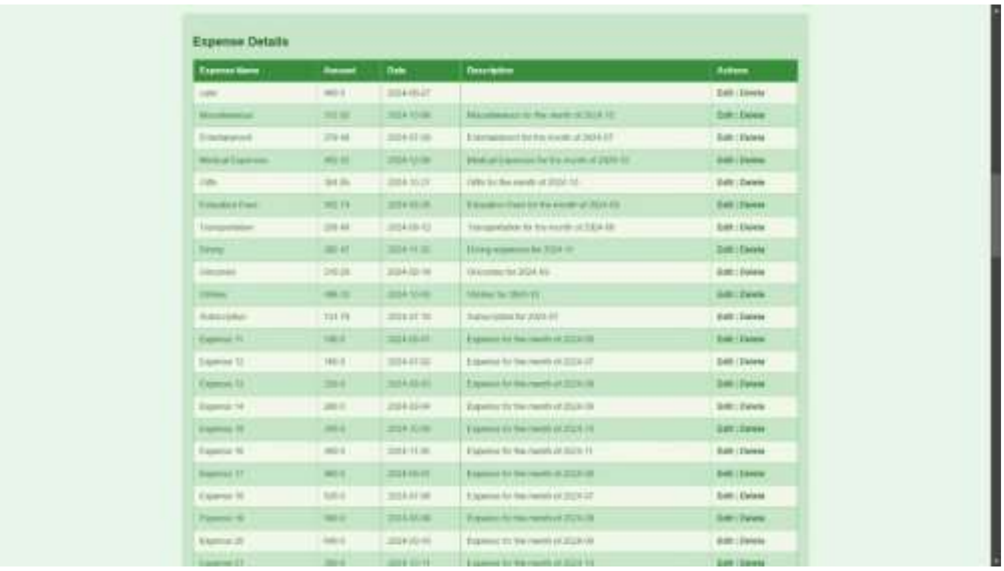
Expense Cost:
200.01

Figure 3: Handwritten Text Recognition

5.3 Expense Management

A standout feature of the Expense Tracker system was the integration of Tesseract OCR, which allowed for the recognition of handwritten text. This functionality greatly minimized the manual effort required to input expense details. Users could draw or write directly on a digital canvas, and Tesseract OCR would accurately extract key information such as the expense name, amount, and date.

The system demonstrated high recognition accuracy, especially for legible handwriting, with only minor inaccuracies occurring in cases of extreme variations in handwriting styles. This feature was especially beneficial for users who preferred handwriting over typing, providing a more intuitive and efficient way to record expenses. By leveraging OCR technology, the platform catered to a wider range of user preferences, making it more accessible and convenient for individuals who rely on traditional methods of input.



Expense Details

Expense Name	Amount	Date	Description	Actions
Lunch	[illegible]	2024-05-27		Edit | Delete
Miscellaneous	[illegible]	[illegible]	Miscellaneous for the month of [illegible]	Edit | Delete
Entertainment	[illegible]	[illegible]	Entertainment for the month of [illegible]	Edit | Delete
Medical Expenses	[illegible]	[illegible]	Medical Expenses for the month of [illegible]	Edit | Delete
Gifts	[illegible]	[illegible]	Gifts for the month of [illegible]	Edit | Delete
Education Fees	[illegible]	[illegible]	Education Fees for the month of [illegible]	Edit | Delete
Transportation	[illegible]	[illegible]	Transportation for the month of [illegible]	Edit | Delete
Dining	[illegible]	[illegible]	Dining expenses for [illegible]	Edit | Delete
Groceries	[illegible]	[illegible]	Groceries for [illegible]	Edit | Delete
Utilities	[illegible]	[illegible]	Utilities for [illegible]	Edit | Delete
Subscription	[illegible]	[illegible]	Subscription for [illegible]	Edit | Delete
Expense 11	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 12	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 13	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 14	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 15	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 16	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 17	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 18	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 19	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 20	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete
Expense 21	[illegible]	[illegible]	Expense for the month of [illegible]	Edit | Delete

Figure 4: Expense Management

5.4 Data Visualization

A key component of the Expense Tracker was its use of Chart.js for visualizing expense data. The platform provided dynamic, real-time charts that updated as users added or modified their expenses, allowing for an interactive and engaging experience. Line charts displayed trends in monthly spending, while pie charts offered category-wise breakdowns, providing users with clear insights into their financial habits.

These visualizations were instrumental in helping users identify spending patterns, compare actual expenses against their budgets, and make informed financial decisions. Feedback from initial testing indicated that the visual analytics were highly intuitive, enabling users to understand their financial data at a glance. This functionality greatly enhanced the overall user experience by offering actionable insights, contributing to better financial planning and management.

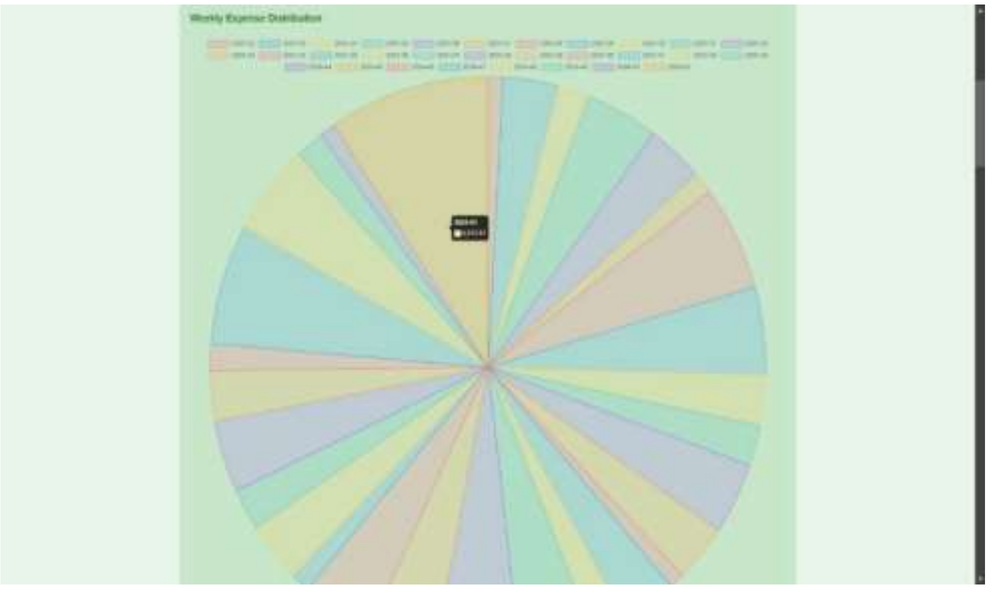


Figure 5: Data Visualization

5.5 System Performance and Scalability

The Expense Tracker system demonstrated strong performance and reliability, even under varying load conditions. SQLite, used for data storage, efficiently handled user data and expense records, ensuring fast data retrieval even as the database expanded. This efficiency contributed to the overall performance, allowing the platform to function smoothly with minimal delays.

Flask's modular architecture played a critical role in ensuring seamless communication between the frontend and backend, minimizing latency during user interactions. Additionally, the system's design was built with scalability in mind, enabling the addition of new features and support for a growing user base without compromising performance.

This scalable architecture ensures that the platform can evolve and accommodate future demands while maintaining high levels of performance.

5.5 Security Measures

Security was a fundamental consideration throughout the development of the Expense Tracker system. User credentials were stored securely using hashed passwords, ensuring that sensitive information was protected from unauthorized access. This approach minimized the risk of data breaches and enhanced the overall security of user accounts.

To further protect user data, sensitive financial information was encrypted during both storage and transmission, ensuring that it remained secure at all stages. The system adhered to industry-standard data privacy practices and regulations, which not only ensured compliance but also helped build user trust. These security measures were implemented to maintain a safe environment for users to manage their financial data with confidence.

5.6 Challenges and Solutions

Although the Expense Tracker project successfully met its goals, several challenges were encountered during development. One significant challenge was the accuracy of the OCR system, which relied heavily on the clarity of handwriting. To address this, preprocessing techniques were implemented to enhance the recognition of varying handwriting styles, improving the accuracy of the extracted data and minimizing errors in expense entries.

Another challenge was balancing performance with real-time updates for visualizations. As users interacted with the platform and added or modified data, it was essential that the charts updated instantly without causing performance degradation. This was resolved by optimizing database queries and frontend interactions, ensuring that updates were fast and efficient even as the dataset grew. These solutions helped maintain a smooth and responsive user experience while improving the overall performance and accuracy of the system.

5.7 Implications and Future Enhancements

The Expense Tracker offers a comprehensive and innovative solution for personal finance management, with standout features like OCR integration and dynamic data visualizations. These features differentiate the system from traditional tools, providing users with a more intuitive and efficient way to track and manage their expenses. The platform's scalability, security, and user-friendly interface have proven effective in meeting the needs of its target users.

In conclusion, future enhancements could expand the platform's functionality further. Mobile app support would allow users to manage their finances on the go, while advanced analytics could provide deeper insights into spending habits. Integration with external data sources, such as bank APIs, would enable automated expense tracking, reducing manual input. Additionally, further improving OCR accuracy through machine learning models and advanced preprocessing techniques could enhance recognition across a wider range of handwriting styles. These developments would further strengthen the system's capabilities, ensuring its continued relevance in an evolving digital landscape.

6. CONCLUSION

This project effectively combines Optical Character Recognition (OCR) with a user-friendly web application for tracking personal expenses. It allows users to manually input expenses or use a handwriting recognition feature for more convenient data entry. The application's homepage features a dashboard displaying quick statistics and a simple navigation interface between manual and OCR input options. A robust database management system securely stores user data, ensuring accurate record-keeping. This solution provides an easy and efficient way to manage personal finances. Future improvements can include enhanced analytics, customizable reports, and further optimization of the user interface for a more polished experience.

7. REFERENCES

[1] Y. Chen, "Design of a Personal Financial Planning Management Information System," 2010 International Conference on E-Business and E-Government, Guangzhou, China, 2010.

[2] A. Smith and J. Lee, "Expense Tracking with Tesseract Optical Character Recognition v5: A Mobile Application Development," 2023 IEEE International Conference on Mobile Computing and Technology (ICMCT), San Francisco, CA, USA, 2023.

[3] M. T. Johnson et al., "Mobile Bookkeeper: Personal Financial Management Application with Receipt Scanner Using Optical Character Recognition," 2021 IEEE Conference on Emerging Technologies for Financial Management (ETF), New York, NY, USA, 2021.

[4] S. Zhang and R. Kumar, "Advancing Optical Character Recognition for Handwritten Text: Enhancing Efficiency and Streamlining Document Management," 2024 IEEE Transactions on Intelligent Systems, vol. 56, no. 7, 2024.

[5] L. Williams, "WONGA: The Future of Personal Finance Management – A Machine Learning Approach," 2022 IEEE International Symposium on AI for Financial Applications (ISAF), London, UK, 2022.