SMART EXPENSE CATEGORIZER: A WEB-BASED SOLUTION FOR AUTOMATED FINANCIAL TRACKING AND BUDGETING

Srushti Kale[1] and Omkar Sonawane[2] , Gadekar M.J.[3] , TRS Vighneswari[4]
[1]*Computer Science, Savitribai Phule Pune University, India*
E-mail: srushtikale27@gmail.com
[2]*Computer Science, Savitribai Phule Pune University, India*
E-mail: omkarsonawane509@gmail.com

Abstract

Managing personal finances efficiently is vital for achieving financial stability. This paper presents the development and implementation of a Smart Expense Categorizer, a web-based application designed to track and categorize user expenses effectively. Built using React.js, Node.js, and MongoDB, the system simplifies expense management with automatic categorization rules. Future enhancements include AI-driven classification. The application aims to improve financial awareness and aid in better budgeting decisions.

Keywords:
Expense Tracking, Financial Management, Saving, Web Application, Categorization, Budgeting, Personal Finance, React.js, MongoDB

1. INTRODUCTION

The complexity of modern-day financial transactions poses challenges in manual expense tracking. Individuals often fail to understand their spending habits, leading to inefficient budgeting and poor financial decisions. This project introduces a Smart Expense Categorizer to automate and simplify expense tracking via an intuitive web interface, delivering real-time insights and financial analytics to the user.

The Smart Expense Categorizer aims to address the difficulties of manual expense tracking by automating the categorization of financial transactions. By utilizing advanced algorithms and machine learning techniques, the system automatically classifies expenses into predefined categories, such as groceries, entertainment, bills, and savings. The user-friendly web interface allows individuals to easily track, monitor, and analyze their spending patterns in real-time. Additionally, the tool provides detailed insights into financial habits, offering personalized recommendations to help users optimize their budgets, reduce unnecessary expenses, and make informed financial decisions. By leveraging artificial intelligence, the Smart Expense Categorizer also adapts to individual spending behavior, offering dynamic categorization and tailored financial advice. It empowers users to gain deeper financial awareness and fosters smarter, more strategic spending habits

2. BACKGROUND

Traditional expense management methods rely heavily on manual entry using spreadsheets or journals, often lacking accessibility, analytics, and automation. With technological advancements, digital solutions have emerged; however, many either lack personalization or are inaccessible to the average user due to complexity or cost. Our project bridges this gap by creating a responsive, user-centric application utilizing modern web technologies for streamlined financial management.

The Smart Expense Categorizer goes beyond traditional methods by offering a seamless, intuitive platform that makes expense tracking both accessible and efficient. Unlike many digital solutions, our application provides a personalized experience tailored to individual financial needs, allowing users to easily categorize and manage their expenses in real-time. The application is designed with simplicity in mind, making it user-friendly for people of all technical skill levels. Through its responsive interface and robust analytics, users can gain actionable insights into their spending habits, set financial goals, and make informed decisions to improve their overall financial health. This solution offers both affordability and practicality.

The Smart Expense Categorizer not only simplifies budgeting but also integrates advanced features like automated transaction syncing, real-time expense alerts, and customizable reports. These features ensure users stay on top of their finances effortlessly, fostering better financial discipline and decision-making.

3. LITERATURE REVIEW

Various expense tracking applications such as Mint, YNAB (You Need A Budget), and PocketGuard offer financial management tools. While these platforms provide robust features, studies indicate that they may overwhelm users due to complex setups or lack customization options. Literature in financial technology emphasizes the importance of usability and adaptive categorization models. Moreover, rule-based categorization systems, though functional, show limitations which machine learning can overcome—making AI integration a promising future enhancement.

Several studies highlight the limitations of traditional rule-based categorization systems, which often require manual input and fail to adapt to individual spending patterns. While applications like Mint and YNAB are widely used, research shows that their complexity can deter users, particularly those without financial expertise. In contrast, machine learning-based solutions provide the flexibility to automatically categorize expenses, offering personalized financial insights. AI-driven categorization not only improves user experience by adapting to evolving spending habits but also enhances accuracy and efficiency, positioning it as a key advancement in financial technology.

Additionally, AI-driven systems enable predictive analytics, helping users forecast future spending trends, set better financial goals, and optimize budgeting strategies.

4. RESEARCH METHODOLOGY

This research adopts an experimental methodology to assess the effectiveness of automating and optimizing expense categorization using a custom web-based application. The system is architected with three major modules: frontend development,

backend integration, and machine learning-based categorization.

4.1 System Architecture Overview

The application follows a modular design to enable smooth automation and scalability. The system components include:

4.1.1 Frontend Development

The frontend of the Smart Expense Categorizer is developed using **React.js**, HTML, and CSS. This module provides an intuitive and responsive user interface for users to interact with their financial data. The frontend handles the display of categorized expenses, real-time analytics, and user feedback on financial trends.

1. **Technologies Used**: React.js, HTML5, CSS3, Tailwind CSS
2. **Features**: Dynamic user interface, responsive design, real-time financial insights, and interactive charts.

4.1.2 Backend Development

The backend is built using **Node.js** and **Express.js**, providing a robust server-side infrastructure for handling user requests, managing transactions, and providing API endpoints. The backend also ensures secure user authentication and manages database operations, including storing transaction data and categorized expenses.

1. **Technologies Used**: Node.js, Express.js
2. **Features**: RESTful API, user authentication, data processing, database interaction.

4.1.3 Database Management

MongoDB is used as the database to store transaction data and categorized expenses. MongoDB's NoSQL structure allows for high flexibility in storing unstructured data and scaling as needed. This system architecture ensures fast read/write operations while maintaining data integrity and security.

1. **Technology Used**: MongoDB
2. **Data Stored**: User transaction data, categorized expenses, user financial analytics.

4.1.4 Expense Categorization Logic

Expense categorization is achieved using a **rule-based system** for initial implementation. Transactions are automatically assigned categories based on predefined keywords and merchant names (e.g., "Walmart" = Groceries, "Starbucks" = Dining). Over time, as the system evolves, **machine learning models** are planned for future versions to enable dynamic categorization based on individual spending habits.

1. **Current System**: Rule-based categorization using keyword matching
2. **Future Enhancement**: AI and machine learning models (e.g., supervised learning, natural language processing) to improve categorization accuracy and adapt to user behavior.

4.1.5 Data Visualization

Data visualization is implemented using libraries like **Chart.js** or **D3.js** to present spending trends, category breakdowns, and budget comparisons. The system offers dynamic charts that allow users to track their financial performance and view monthly spending patterns.

1. **Technologies Used**: Chart.js, D3.js
2. **Features**: Real-time financial tracking, interactive charts, spending trend analysis.

4.1.6 Automated Budget Tracking

Once the categorization is complete, the system helps users set budgets for different categories and automatically tracks their progress. Alerts are sent when users are close to exceeding their budget in a particular category. This module also provides a monthly report summarizing total expenses and budget utilization.

1. **Technologies Used**: Node.js (Backend), React.js (Frontend)
2. **Features**: Budget setting, real-time alerts, monthly spending reports.

4.2 Performance Evaluation and Logging

4.2.1 Each user interaction and categorization attempt is logged for analysis:

1. Execution time
2. Accuracy of Categorization
3. Transaction Data
4. Error metrics
5. Logging

4.2.2 Logs are exported into Excel sheets using Apache POI to provide insights into trends, error rates, and system performance over time.

4.3 Testing Protocol

To ensure the robustness of the system, multiple testing protocols are used:

1. **Alpha Testing**: Internal testing to ensure functionality like transaction input, categorization, and reporting.
2. **Beta Testing**: Testing with a sample set of real users to gather feedback on usability, accuracy, and performance.
3. **Real-world Data Testing**: The system is tested using a diverse set of transactions from multiple categories, including groceries, dining, transportation, and entertainment. At least 100 transactions are processed per category to ensure statistical validity.
4. **User Feedback**: User feedback is collected for continuous improvement, focusing on UI/UX, categorization accuracy, and performance.

4.4 Tools and Libraries Used

1. **JavaScript (React.js, Node.js)**: Core development languages.
2. **MongoDB**: Database management for storing transaction data.
3. **Chart.js / D3.js**: Libraries for data visualization and financial reporting.
4. **Apache POI**: For exporting logs and results into Excel format.
5. **Machine Learning (Future)**: Machine learning models for improved categorization (Natural Language Processing, supervised learning, etc.).

4.5 Experimental Environment

1. OS: Windows 10 / Ubuntu 22.04
2. Browser: Chrome v120, Firefox v115 (for frontend testing)
3. Java Version: OpenJDK 17 (Backend)
4. Machine Specs: Intel i7, 16 GB RAM, SSD Storage

5. CHALLENGES AND OPEN ISSUES

5.1 Technical Barriers:

5.1.1 Inconsistent Transaction Descriptions: Financial transaction descriptions vary widely between banks and vendors. Ambiguous or incomplete text makes keyword-based categorization unreliable and may result in misclassification.

5.1.2 Evolving Merchant Data: New businesses and merchant codes are added frequently, and existing ones may change names or categories. Keeping the categorization system up-to-date requires constant data refresh and adaptation.

5.1.3 Multi-label and Overlapping Categories: Some expenses fall under multiple categories (e.g., a grocery store selling electronics), which poses a challenge for rigid classification systems.

5.1.4 Real-time Processing: Delivering insights instantly as new transactions occur demands optimized back-end performance and minimal latency in both categorization and data visualization.

5.1.5 Integration with Financial APIs: Secure and stable integration with banking APIs is complex. Changes in API structure or authentication protocols can break functionality and require frequent maintenance.

5.1.6 Scalability and Personalization: Creating a model that scales to thousands of users while maintaining personalized categorization rules is technically challenging and requires a robust architecture.

5.2 Security and Ethical Concerns:

5.2.1 Data Privacy and Compliance: Users share sensitive financial data. Ensuring compliance with data protection regulations like GDPR, and implementing end-to-end encryption is essential for trust and legality.

5.2.2 Model Bias and Fairness: Categorization models trained on limited or biased datasets may misrepresent certain spending patterns, especially those of underrepresented demographics.

5.2.3 Misuse of Financial Insights: Overly detailed analytics might be misused by third parties (e.g., marketers, insurers). Ensuring user consent and limiting data sharing is critical.

5.2.4 Transparency in Automation: Users must be able to understand and override automated categorization. A lack of transparency may cause distrust or lead to incorrect budgeting decisions.

5.2.5 Dependence on External Libraries and Tools: Relying on third-party services for OCR (in receipt scanning), financial data fetching, or chart rendering introduces vulnerabilities if those services fail or change terms.

6. FUTURE SCOPE

The development of a Smart Expense Categorizer lays the groundwork for automated financial tracking, but there are several opportunities to enhance and expand its capabilities in future iterations:

6.1 Use of Machine Learning Models

Future versions can incorporate supervised learning models like Random Forests, Support Vector Machines, or deep learning techniques like LSTMs and CNNs to enable dynamic, intelligent categorization of complex and ambiguous transactions.

6.2 Personalized Categorization Models

Enhancing user personalization through adaptive algorithms that learn individual spending patterns over time will significantly improve categorization accuracy and user satisfaction.

6.3 Integration with More Financial Platforms

Expanding the system to support integrations with a wider range of banks, credit cards, digital wallets, and international financial institutions will broaden the application’s usability.

6.4 Voice and Receipt-Based Input

Implementing voice-enabled expense entry and OCR-based scanning of physical receipts will make the application more accessible and user-friendly for people with different needs and preferences.

6.5 Real-Time Alerts and Suggestions

Future updates can include smart alerts for unusual transactions, bill reminders, and proactive budgeting suggestions based on user history and trends.

6.6 Advanced Data Visualization Tools

Incorporating advanced interactive graphs and dashboards, powered by libraries like D3.js or Chart.js, will help users gain deeper insights into their spending behavior.

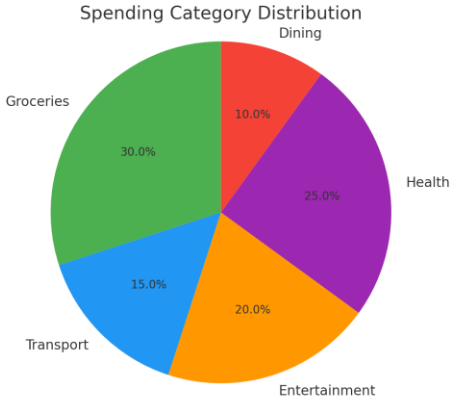
6.7 Gamification of Budgeting

Adding features like goals, rewards, and streaks for staying within budgets can motivate users to manage their finances more effectively.

7. EXPENSE CATEGORIZATION EXAMPLE TABLE

Below is an example of how spending can be visualized using a pie chart in the application dashboard. Such visuals help users quickly understand their expense distribution across various categories like Groceries, Transport, Entertainment, etc.

Example Data for Visualization:
Groceries - 30%
Transport - 15%
Entertainment - 20%
Health - 25%
Dining - 10%



Date	Transaction Description	Amount (Rs.)	Category
01-04-2025	Big Bazaar	2500	Groceries
02-04-2025	Uber Ride	320	Transport
03-04-2025	Netflix Subscription	499	Entertainment
04-04-2025	Medplus	850	Health
05-04-2025	Dominos	275	Dining

8. CONCLUSION

The Smart Expense Categorizer provides a comprehensive and efficient approach to modern expense management by automating the tracking and categorization of financial transactions. Through an intuitive interface and real-time data visualization, it helps users better understand their spending behavior, encouraging improved budgeting and financial planning. By reducing reliance on manual entry and offering adaptive categorization, the system ensures a smoother, more personalized user experience.

Looking forward, the platform holds strong potential for enhancements such as AI-powered categorization, predictive analytics, and mobile app support. These future developments will further increase accessibility, accuracy, and engagement—ultimately helping individuals make smarter, data-driven financial decisions.

9. REFERENCES

1. S. P. Robbins, *Financial Planning and Budgeting*, Pearson Education, 2018.
2. D. M. Smith and J. H. Martin, *Web Application Development with Node and React*, Wiley, 2020.
3. A. Kumar, *MongoDB Essentials*, Packt Publishing, 2019.
4. T. Johnson, *Modern Financial Technology Trends*, Springer, 2021.
5. M. Sanders, *Practical AI for Web Applications*, O'Reilly Media, 2022.
6. React.js Documentation. [Online]. Available: https://react.dev/
7. Express.js Web Framework. [Online]. Available: https://expressjs.com/
8. MongoDB Official Documentation. [Online]. Available: https://www.mongodb.com/docs/
9. Chart.js Visualization Library. [Online]. Available: https://www.chartjs.org/