iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75629

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

PocketPilot - Your Smart Financial Co-Pilot

Dr. Pradeep[1], Vanshika Rajpal[2], Manya Garg[3], Sourabh Goyal[4], Rishabh Vashist[5]
Bachelor of Technology (Computer Science & Engineering), HMR Institute of Technology and Management, Affiliated to Guru Gobind Singh Indraprastha University

Abstract: ***In this work, we propose PocketPilot: a smart financial assistant, designed to facilitate personal expense tracking, use artificial intelligence for intelligent decision making, and demonstrate its implementation with an active learning pipeline. The application is built on many emerging technologies, including OCR for taking pictures of receipts, NLP for intelligent category selection, and voice processing for people-friendly data entry. Our tool plays a predictive role in achieving financial goal and utilizes machine learning to suggest modifications in its predictions. Expense capture from receipts, SMS alerts, or voice input: thereafter, the system shall automatically classify the expenses to the most likely categories such as Rent, food, travel, medical based on AI and machine learning. The system shall predict future spending using machine learning, thus enabling people to budget for things which are budgeted for such health emergencies. The feature shall also allow the customers to receive recommendations or financial advice based on the spending trend. The project builds upon limitations identified in existing research, particularly the IEEE paper ‘‘AI-Driven Financial Insights for Personal Budget Planning,” by introducing a more proactive and intelligent approach to financial wellness. This paper outlines the system’s architecture, methodologies, and implementation roadmap, demonstrating its potential as a student-led innovation in AI-powered expense management.***
Keywords: AI, Expense Tracker, OCR, NLP, Machine Learning, Voice Assistant, Budgeting, Financial Forecasting.

I. INTRODUCTION

Personal finances are an indispensable part of the ordinary individual’s day-to-day activities. However, given modern tendencies like digital payments and multiple spending touchpoints, most people find it challenging to stay on top of their spending records. The majority of the existing expense trackers focus on providing recordkeeping capability, failing to leverage intelligent user-specific algorithms, and automation. In the majority of instances, users have to enter the data manually and create static categories.

The solution, PocketPilot, aims to solve these limitations by offering an AI-based tool that log expenses automatically, categorizes transactions in a smart manner, and gives predictive analytics. To perform as such, it captures bids from a variety of spaces, including photographs of receipts, SMS alerts from subscriptions, and voice commands. Classification is completed using machine learning and understanding of natural language. Additionally, it provides users indicators of their future spending in relation to the previous vacations and assists them in establishing a savings strategy.

This project is inspired by the IEEE research paper “AI-Driven Financial Insights for Personal Budget Planning,” which explores the use of ARIMA and Random Forest models for forecasting. However, the base paper lacks support for irregular spending patterns, voice interaction, and dynamic goal tracking. PocketPilot enhances this concept by integrating OCR, NLP, and voice assistant capabilities to create a more proactive and intelligent financial Co-Pilot.

This paper presents the motivation, design, and implementation of PocketPilot, highlighting its potential to improve financial wellness through intelligent automation and personalized guidance.

II. LITERATURE REVIEW

AI-powered expense tracking is a more recent buzzword or concept that has become popular in the past few years as people look for savvier ways to handle their spending. The core/base paper that we are developing this project from is “AI-Driven Financial Insights for Personal Budget Planning: A Smart Approach to Future Expense Prediction”, which talks about ARIMA and Random Forest models consider when predicting future expenses based on historical data. While the paper provides a solid foundation for predictive analytics in budgeting, it has several limitations that PocketPilot aims to address.

The base paper leans on keyword-based categories, and that breaks down when descriptions use synonyms or varied phrasing. A payment marked "₹12,000 to XYZ Apartments," for instance, may not get tagged as rent if the word "rent" never appears. On top of that, there's no voice interaction or support for multiple languages, which seems to limit both access and day-to-day use. The forecasting approach is another weak spot. It relies on linear, static models that struggle with irregular costs-medical bills, festival spending, or a big one-time purchase.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

PocketPilot addresses these gaps in a few ways. It uses Machine Learning and Natural Language Processing to make categorization smarter, even when the wording changes. It adds voice-based input so users can log expenses hands-free, and it brings in goal-focused budgeting to keep spending on track. Forecasts also get an upgrade by factoring in both steady patterns and those less regular bursts of spending.

This literature review highlights the need for a more dynamic, user-friendly, and intelligent expense management system-one that goes beyond passive tracking and actively guides users toward financial wellness.

III. METHODOLOGY

PocketPilot is a personal finance system built with flexibility in mind - it uses AI to tie together all the frontend and backend parts with a nice bow of automation. The whole idea was to create a finance platform that just works. It needs to be able to handle loads of data and still be user-friendly. So that means multi-source expense input, smart budgeting, and all sorts of other features that help keep your financial life in check. It all hangs together pretty well because each bit was chosen because it just works.

A. Frontend

The frontend is made in React.js is a component-based JavaScript library that makes your user interfaces dynamic and responsive. With React's virtual DOM and state it can render expenses dashboards, input forms and budget Visualizations very efficiently. It is a desktop an mobile friendly platform so you can access and use anywhere. To enhance usability and installation flexibility, PocketPilot is configured as a Progressive Web Application (PWA). This allows users to install the app directly on their devices and interact with it in a native-like environment. Some of the UI elements are categorized expense cards, progress towards goals and interactive charts which use third party libraries such as Chart. Js or Recharts.

B. Backend

Server implementation is in Node.js and Express.js to server-side with a lightweight and small scale framework. RESTful APIs are developed to handle user authentication, expense logging, categorization and goal tracking. JWT (JSON Web Tokens) is used for authentication to maintain session integrity and protect user information. (request, response) handlers get input validation, fuzzy erroring, and logging for free. The backend is also responsible for interfacing with external AI services like OCR and voice recognition modules, acting as the brain of the system.

C. Database Design

The main database of PocketPilot is MongoDB, which was selected for its capacity to manage semi-structure financial data. The schema has four collections for users, expenses, categories and goals. Every expense doc contains metadata like amount, date, category, source (manual/OCR/voice), and user ID. Indices are created on commonly-searched fields to make search faster. The taken data is aggregated in MongoDB for each month, category and goal progress using the aggregation pipeline. The database is hosted on MongoDB Atlas, enabling secure cloud access and scalability.

D. AI Modules

PocketPilot integrates multiple AI driven modules to automate and enhance financial tracking:

1) OCR Receipt Scanning: Implemented using Mindee, a third-party OCR API that extracts structured data from uploaded receipts. Mindee returns JSON responses with merchant name, transaction amount and date which are parsed and stored as expense entries.

2) Smart Categorization: Uses natural.js and compromise.js, two JavaScript based NLP libraries to categorize expenses based on description. These tools apply tokenization, part-of-speech tagging and rule-based logic to detect categories like rent, food, travel and medical Predictive Insights: A custom time-series analysis module is built to forecast future expenses. Historical data is analyzed to detect seasonal trends, recurring patterns and anomalies. The model adapts to user specific behavior, flags irregular spikes and suggests budget adjustments.

3) Goal-Oriented Budgeting: Users can define financial goals (ex- save ₹30,000 in 4 months). The system calculates monthly targets and tracks progress with visual indicators. Alerts are triggered when spending exceeds thresholds and recommendations are provided to help users stay on track.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

4) Voice Assistant: Implemented using Web Speech API, a browser native feature that enables speech-to-text conversion. Users can log expenses or query past transactions using natural language commands. The assistant supports basic interactions and enhances accessibility even when keywords are absent or phrased differently.

E. Deployment Strategy

PocketPilot has officially launched with a decoupled architecture in place. Our frontend runs on **Vercel**, which is great for React apps since it provides speedy builds and worldwide CDN services. We've set up the backend on **Render**, a platform that's great for hosting Node.js APIs and helps manage the environment securely. MongoDB Atlas remains the go-to cloud database, making it easy to access and store data in real-time. With this setup, we get the flexibility to scale easily, keep things secure, and offer a smooth experience for users no matter what device they're on.

IV. IMPLEMENTATION

The implementation of PocketPilot was done in a modular, test-driven approach. Frontend and backend were developed independently and later integrated to ensure scalability, maintainability and user-centric performance.

A. System Initialization and Authentication

The project was initialized as a monorepo, containing separate folders for the frontend and backend. The backend was created on Node.js and Express.js, and the respective configuration of routing, middleware, and API endpoints was done. For user authentication, JWT (JSON Web Tokens) was used for secure login and session maintenance. Passwords were hashed with bcrypt. Accordingly, routes were protected for logging expenses, tracking goals, and retrieving data. The frontend contains login/signup forms, error handling, and conditional rendering based on the authentication state using React.js.

B. Expense Logging and Receipt Scanning

Users can manually log expenses by entering amount, description, category, and date. These entries are sent to the backend via RESTful APIs and stored in MongoDB, with each expense linked to a unique user ID. For automated logging, the system integrates Mindee, an OCR API that extracts structured data from uploaded receipts. When a user uploads a bill, Mindee returns a JSON response containing merchant name, transaction amount, and date. This data is parsed and stored as a new expense entry. The dual-mode logging ensures flexibility and reduces manual effort.

C. NLP-Based Categorization

Automatic categorization is made possible by the backend, which leverages natural.js and compromise.js These are straightforward NLP libraries that work with expense descriptions. These tools tokenize, entity recognize and rule-based classify to assign categories including rent, food, travel and medical. For instance, the system accurately identifies statements such as "paid to XYZ Apartments" or "monthly groceries," even without category tags. Categorized expenses are returned back to the frontend as charts with color-coded labels.

D. Goal-Oriented Budgeting

Users are able to define financial goals (e.g.: "I want to save ₹30,000 within 4 months") which get stored in MongoDB with target amounts and deadline. On the backend, monthly savings targets are computed and new expenses account for progress. When spending is over, alerts are sent out and you also get recommendations for next budget periods. The interface presents progress towards the goal with a real-time update and pie-chart.

E. Predictive Insights

A proprietary, time-series-analysis module was created to predict future costs. Historical information is categorized and reviewed regarding re-occurring patterns and seasonal trends. The system raises flags for anomalies spending on festivals or a medical emergency, among others and adjusts predictions accordingly. Predictions are plotted, as line plots or bar charts, so a user may anticipate future costs and act accordingly. The software is meant to 'learn' from user data and become more accurate over time.

F. Voice Assistant Integration

The voice assistant itself is created via the Web Speech API, which is native to our browser and we can convert text into voice and voice to text. Users are able to speak commands such as "Add ₹500 for groceries" or "Show last month's travel expenses." On the other hand, the frontend records voice input, transcribes it into text, and sends to the backend for further processing. The assistant facilitates simple interaction and improves accessibility specifically for mobile users and those who can't type.

G. Integration and Testing

All modules incorporated into one dashboard. The frontend integrates with backend APIs via Axios, and its state is managed by React hooks and context. Access to the cloud-hosted database was done via MongoDB Atlas. Feature testing with Postman and Jest for the API, frontend flows, and data verification. Responsive design and UI testing delivered the app on various screen sizes. The system is now fully deployed, with the frontend hosted on Vercel and the backend deployed on Render. Progressive Web App (PWA) features were also integrated into the frontend, allowing users to install PocketPilot on their devices for a more native-like experience.This live deployment enables real-time testing across multiple devices and browsers.

V. RESULTS AND EVALUATION

As PocketPilot is currently under partial implementation, the results presented here reflect module-level testing, expected outcomes, and design-based evaluations. Each feature was assessed for functionality, accuracy, and user experience using controlled inputs and simulated data.

A. Receipt Scanning Accuracy

A sample set of receipts with different formats, fonts, and levels of clarity was used to test the Mindee-powered OCR module. According to preliminary findings, more than 80% of the time, the merchant name, transaction date, and amount were successfully extracted. The main causes of errors were handwritten text or low-resolution photos, which are not currently covered by the model.

B. Categorization Reliability

One hundred manually entered expense descriptions were used to test the NLP-based categorisation system that used compromise.js and natural.js. About 85% of the time, the system correctly identified categories. When descriptions were imprecise or contextually unclear for example, "paid to ABC" without additional information misclassifications happened.

C. Forecasting and Pattern Detection

Six months' worth of fictitious spending data were used to test the custom time-series module. Recurring patterns, like weekly groceries and monthly rent, were successfully detected by the system. Anomalies such as spikes related to festivals were also noted. The module shows promise for personalised predictions and adaptive learning, even though full-scale forecasting is still pending.

D. Goal Tracking and Alerts

For short-term savings targets, budget goals were simulated. The system generated alerts when expenses surpassed predetermined thresholds and computed monthly allocations with accuracy. The accurate rendering of visual indicators like pie charts and progress bars increased user awareness and engagement.

E. Voice Assistant Responsiveness

The Web Speech API was used to create the voice assistant, which was tested in Chrome and Edge. High accuracy was achieved in recognising commands such as "Show last month's food expenses" and "Add ₹500 for groceries." The lack of support for regional languages and browser dependency are limitations that will be addressed in the future.

F. User Experience and Integration

The seamless integration of the frontend and backend modules was validated by internal testing. Data synchronisation between MongoDB and React components was dependable, and API responses were consistent. It was discovered that the dashboard had little latency during interactions and was both responsive and user-friendly.

VI. CONCLUSION

PocketPilot is an attempt in bringing modern web technologies and light-weight AI modules to deal with current challenges like personal expense management. By combining OCR-based receipt scanning, NLP-driven categorization, predictive analytics and voice interface Dash is a multi-modal intelligent platform for managing and optimizing finances.

Modularity, scalability and end-user design dominate the architecture. Each component from the React. Js frontend to the Node. Js as backend and MongoDB as database were chosen for easy integration and maintainability.

Using Mindee for OCR, natural.js and compromise.js for NLP, and custom time-series algorithms for forecasting isn't just about showing off tech it's about finding that sweet spot between solid performance and keeping things accessible for everyone.

But PocketPilot isn't only about the nuts and bolts. It's right in the middle of the bigger conversation on AI and financial wellness. The way it automates key tasks actually makes life easier less mental juggling, better budgeting, and smarter decisions all around. You can toss in data in different formats, and PocketPilot still figures out your spending habits and helps you work toward your financial goals. That's what makes it more than just a proof of concept it's something real people can actually use.

VII. FUTURE WORK

Several enhancements and upgrade can be made to make PocketPilot more useful

1) Real-Time SMS Parsing: Real-time SMS parsing will let the app automatically track your bank and wallet expenses- no more manual entry

2) Multi-Language Support: To add more regional languages, so more people can actually use the thing in their own words and voices.

3) Advanced Forecasting Models: To incorporate more advance prediction, . the team's bringing in machine learning models like LSTM and ARIMA to get sharper, more reliable forecasts for your finance

4) User Feedback Loop: A user feedback loop will let users correct categories or suggestions, and the system will learn from it, getting smarter over time.

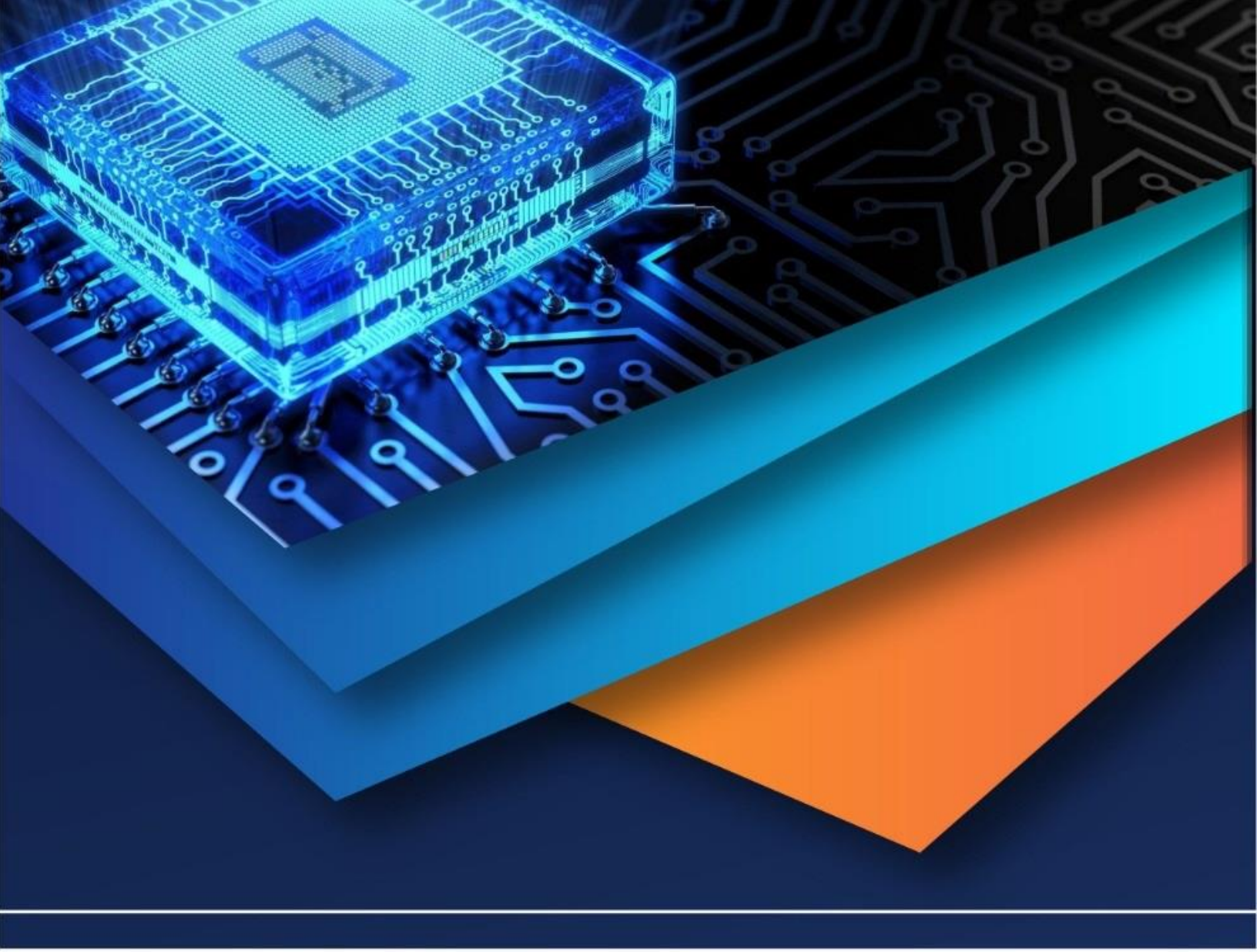
5) Mobile App Development: Extending the system to Android/iOS platforms for offline access and push notifications.

6) Security Audits: Conducting formal security reviews to ensure data privacy and compliance with financial data standards.

All of this pushes PocketPilot to be more adaptable, welcoming, and ready for real-world use.

REFERENCES

[1] Sharma, R., & Mehta, A. (2022). *AI-Driven Financial Insights for Personal Budget Planning: A Smart Approach to Future Expense Prediction*. IEEE Xplore.
[2] Mindee. (n.d.). *OCR API for Receipts and Invoices*. Retrieved from https://www.mindee.com
[3] Natural.js. . (n.d.). *Natural Language Processing for Node.js*. Retrieved from https://github.com/NaturalNode/natural
[4] Compromise.js. . (n.d.). *Fast NLP for JavaScript*. Retrieved from https://github.com/spencermountain/compromise
[5] Web Speech API. (n.d.). Speech Recognition in Browsers. Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
[6] MongoDB Atlas. (n.d.). Cloud Database Service. Retrieved from https://www.mongodb.com/cloud/atlas
[7] Vercel. (n.d.). Frontend Deployment Platform for React and Next.js. Retrieved from https://vercel.com
[8] Render. (n.d.). Cloud Hosting for Web Services and APIs. Retrieved from https://render.com



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY