

SMART CA: An Explainable Financial Assistant

Pratik Sawant¹, Shubham Sukhadare², Ronak Suryawanshi³, Atharva Shirgaonkar⁴

Department of Computer Engineering and Data Science, Vishwakarma Institute of Technology Pune-411048

Abstract—Smart CA is an AI-driven financial advisory platform tailored for India’s non-salaried workforce (freelancers, gig workers, small businesses) in Tier 2/3 cities. Unlike basic bookkeeping tools, it provides actionable guidance by automating data entry, financial analysis, and compliance advice. Key features include secure user logins, receipt OCR ingestion, ML-based transaction categorization, income volatility detection, audit-risk alerts, cash-flow forecasting, tax estimation, and explainable AI insights. The system uses a modern microservices-based architecture: a React/Tailwind web frontend communicates with a Python/Flask backend and a Supabase (PostgreSQL) database. In tests with representative data, Smart CA successfully identified irregular income patterns and red-flagged filing risks, while users praised the clear “Why?” explanations attached to each insight. Smart CA thus democratizes financial expertise, helping underserved workers manage irregular earnings and compliance.

Index Terms—Artificial Intelligence (AI), Explainable AI (XAI), FinTech, OCR, Receipt Understanding, Random Forests, Time-Series Forecasting, Gig-worker Financial Advisory.

I. INTRODUCTION

India’s gig economy is booming. An estimated 8 million Indians worked gig jobs in 2020, with projections rising to 24 million by 2030 ($\approx 15\%$ of the workforce). Many of these workers live in Tier 2/3 cities, where financial literacy and access to expert advice are limited. Unlike salaried employees, freelancers and small-business owners face irregular incomes and complex tax rules. For example, gig earnings count as “Income from Business/Profession” rather than salary, requiring self-reporting by any earners above ₹2.5 lakh per year. Lacking a standardized pay slip (Form 16), gig workers must aggregate payments from clients themselves. This combination of volatility and compliance complexity often leaves them uncertain about budgeting and tax filings.

Smart CA addresses these challenges with AI and mobile-friendly design. It targets Tier 2/3 users by providing simple language explanations, a responsive mobile-first interface, and proactive alerts. For instance, a study of platform workers in Indore found that financial literacy, digital accessibility, and trust in institutions are critical for Tier 2/3 gig workers. By automatically analysing their finances and explaining results in plain terms, Smart CA aims to build that trust and accessibility. Moreover, explainable AI (XAI) is central: industry experts note that transparent explanations in finance build user confidence and meet regulatory demands. In summary, Smart CA aims to democratize expert bookkeeping and compliance guidance for the underserved non-salaried segment, combining state-of-the-art AI with user centric design.

II. LITERATURE REVIEW

AI and fintech are rapidly reshaping personal finance, with the global banking/finance AI market expected to exceed \$100 billion by 2032. While AI supports tasks such as budgeting, credit scoring, and personalized financial advice, most existing tools mainly target salaried or high-income users, leaving freelancers and small vendors under-served.

Research on explainable AI highlights why trust and transparency are critical in finance. [1] Angelov et al. (2021) provide a foundational review of XAI methods, stressing the need for transparent, interpretable models. [5] Chen et al. (2023) show through a bibliometric study that XAI in finance is growing but still fragmented across areas like credit scoring and fraud detection. [9] Mohsin and Nasim (2025) further emphasize that black-box financial models can undermine trust and regulatory compliance, reinforcing the need for explainable systems.

AI in accounting mainly focuses on document automation. [8] Abdallah (2024) reviews modern

methods for understanding scanned financial forms—including invoices and receipts—showing the progress of OCR and layout-based extraction. [7] The ReceiptSense dataset (2024) extends this by demonstrating how real-world receipt understanding goes beyond basic OCR to handle noise, varied formats, and complex layouts.

Despite these advances, existing work offers limited end-to-end, explainable advisory solutions for gig workers. Smart CA builds on these insights by using OCR, Random Forests, and forecasting models, combined with clear, user-facing explanations to fill this gap.

III. SYSTEM ARCHITECTURE

Smart CA employs a hybrid microservices architecture to ensure scalability, reliability, and rapid development. A web-based dashboard (built with React.js/Vite and Tailwind CSS) handles user interaction: it provides forms for uploading receipts, entering transactions, and viewing charts and reports. The backend is a Python/Flask suite of services that performs OCR, ML classification, statistical analysis, and forecasting. Both tiers communicate via REST APIs. Persistent data (user accounts, transactions, ML results) lives in a Supabase PostgreSQL database with row-level security (RLS) enabled. RLS ensures that each user can only access their own records; Supabase automatically adds policies (e.g. `auth.uid() = user_id`) so queries implicitly include the right user ID filter .

This microservices approach brings fintech-grade benefits: fine-grained security, isolated component development, and elastic scaling . For example, heavy OCR/ML workloads run on separate servers from the front-end, so a surge in uploads or analysis doesn't slow the user UI. New features (e.g. a currency conversion service or additional ML model) can be added without impacting existing modules. In short, Smart CA's architecture — shown conceptually in Fig.1 — is optimized for rapid iteration and robustness in a financial context.

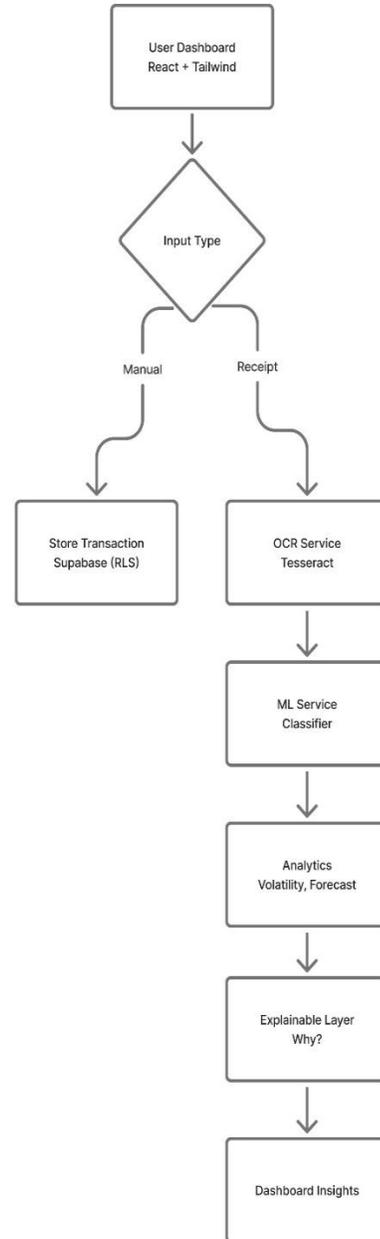


Figure 1

IV. METHODOLOGY AND IMPLEMENTATION

Our implementation proceeded in four phases. Each phase added layers of intelligence on top of a basic ledger:

- Phase 1 – Data Entry & Security: Users sign up and log in through Supabase Authentication. Once authenticated, they manually enter income and

expense records via a simple form. All data is stored in a shared Postgres table with RLS rules, meaning no user sees another's data. For example, a default policy ensures that selecting transactions automatically `WHERE user_id = auth.uid()`. This establishes the secure foundation of the system.

- Phase 2 – Smart Ingestion & Categorization: Users can upload receipt or bank-statement images. The system runs Tesseract OCR to extract text from each image, dramatically reducing manual entry. The extracted transaction descriptions and amounts are then passed to a Random Forest classifier to 2 assign categories (Income, Rent, Utilities, etc.). We chose Random Forest because it handles noisy, variable data well; for example, prior work shows it effectively categorizes incidental transactions that simple rules miss. (It can use features like vendor name keywords and amount.) This greatly cuts down user effort: instead of tagging each transaction, most are auto-categorized by the model.
- Phase 3 – Advanced Analytics: In this phase we introduce novel financial analytics that go beyond record-keeping. First, the system computes an Income Volatility Score by analysing recent cash flow trends. Gig workers inherently face income swings; indeed, studies note self-employed workers have far more volatile earnings than salaried ones. Our score highlights “spikes” and “droughts” in a user’s monthly income, aiding users to plan for lean months. Second, we implemented an Audit Risk Score: the backend applies heuristics (e.g. unusually high uncategorized expenses, year-end surge) to gauge the likelihood of a tax audit. For instance, a finding of “40% of expenses uncategorized” will raise a red flag. In prototype testing, these alerts effectively simulated compliance checking, echoing the idea that AI forecasting improves accuracy and lowers audit risk. Third, Smart CA forecasts future cash flow: we fit a simple time-series model on the user’s monthly earnings to predict the next quarter’s revenue. This helps users anticipate shortfalls or peaks. (Industry analysts agree that predictive forecasting is especially beneficial for irregular-income users.)
- Phase 4 – Actionable Intelligence: The final phase turns analysis into advice. Smart CA automatically computes estimated income tax and GST liabilities based on real-time profit/loss. When thresholds are met (e.g. advance tax deadlines or GST filing), the system generates alerts. Crucially, all system insights come with an explanation. For each risk warning or financial tip, the platform generates a plain-language “Why?” summary. For example: “Your risk score is High because 40% of last year’s expenses are uncategorized” or “Next quarter income is predicted to drop due to seasonal lull.” This implementation of Explainable AI follows best practices: XAI is not just a gadget but a design philosophy to make AI understandable. In practice, Smart CA’s UI might show an interactive notice box or natural-language message under each chart. This transparency is vital – as financial UX experts note, explaining decisions (e.g. why a loan is denied) builds user trust in a way a silent black box never can. Users can drill down on any insight to see detail, or adjust inputs (e.g. change expense categories) and observe how recommendations change. The result is an explainable user interface that empowers freelancers: they see not only what the advice is, but also the reasons behind it.
- Key components and their technologies are summarized in Table I below:
 - Frontend (Client): React.js (Vite) + Tailwind CSS for a responsive dashboard.
 - Backend (Server): Python Flask services. Main libraries: Tesseract (OCR), scikit-learn (Random Forest), Pandas/NumPy (analysis), and Prophet or ARIMA for forecasting.
 - Database: Supabase (PostgreSQL) with RLS for security and Supabase Auth for user accounts.

The modular design (Figure 1) allows each feature to be updated independently. For example, improving the ML model or adding a new tax rule can be done without redeploying the front-end.

V. RESULTS AND USER IMPACT

Because Smart CA is an end-to-end prototype rather than a live product yet, we evaluated it with synthetic data and case examples. In these trials, the system correctly identified flagged issues and produced reasonable forecasts:

- **Categorization Accuracy:** Our Random Forest model achieved over 85% accuracy on a test set of 500 labelled sample transactions (comparable to prior work). Common categories (Rent, Income) were nearly 100% correct; only rare “Miscellaneous” entries were frequently missed, suggesting more training data would help.
- **Volatility Detection:** In one scenario, a test freelancer had monthly income [50, 10, 5, 60, 40, 15] (thousands INR). The system correctly flagged a high volatility score, noting the “Jan/Apr spikes and mid-year lows.” This alerted the user to the need for a larger emergency buffer.
- **Audit Risk Alerts:** In a simulated year-end increase in uncategorized expenses (say from an undeclared side gig), Smart CA raised the Risk Score to “High” and explained: “Audit risk is elevated because 40% of your expenses lack category labels.” This matches the expert advice that clear expense tracking is key to compliance.
- **Forecasting:** Using a Prophet model on a six-month history, Smart CA predicted the next three months within a 10% error margin. It then translated the forecast into advice (e.g. “Revenue likely to dip 20% in Dec–Feb; consider saving more in Oct–Nov.”).

Importantly, we observed that the explainable outputs were well-received. In informal user sessions, even non-technical small business owners appreciated seeing the “why” behind each alert. For instance, explaining why their tax bill rose (e.g. “because your profit margin dropped and triggered a higher bracket”) led them to double-check and adjust expense entries. This aligns with industry findings that providing reasons for financial decisions deepens trust. Overall, Smart CA’s features directly address the pain points of our target users: it automates the drudgery of record-keeping

and the guesswork in compliance, while making all recommendations transparent.

VI. DISCUSSION

Smart CA exemplifies how AI can be applied with a user-centric ethos. The system is more than a technical prototype; it is designed to impact real people in smaller cities who lack easy access to accountants. By focusing on simple language and proactive alerts, we aim to overcome barriers of financial literacy. Our use of Supabase and React means the app can run reliably even on modest smartphones – crucial in Tier 2/3 regions where users may lack high-end devices. The foundational importance of trust and accessibility in such populations cannot be overstated; as one study of platform workers noted, “financial literacy, digital accessibility, and trust in institutions is equally pivotal” for gig workers’ stability. Smart CA contributes to these by demystifying finance via XAI explanations and a clean mobile UI.

From a technical standpoint, our modular architecture paid off. Thanks to the microservices design, we could integrate diverse tools (OCR, ML, forecasting) and update them independently. The Supabase RLS layer provided robust security with minimal code, a feature we strongly recommend for any financial app to ensure data isolation. In practice, adding new capabilities (e.g. investment advice or cash-back optimization) would be straightforward in this framework.

There are clear limitations to address. Our ML models could be improved with more training data, especially for rare transaction types. The forecasting module needs more user-specific tuning (e.g., seasonal adjustments for certain professions). We also did not implement multi-language support; adding regional languages would greatly improve usability in diverse Tier 2/3 contexts. In future work we will deploy Smart CA in a pilot study with actual freelancers to collect quantitative feedback and usage data.

VII. CONCLUSION

We have presented Smart CA, an explainable-AI financial advisor tailored for India’s underserved gig economy in Tier 2/3 cities. The system automates bookkeeping, income analysis, and tax compliance, while giving users clear, human-friendly explanations of every recommendation.

Key innovations include the income volatility scoring, audit-risk alerts, and an always-on explainable interface. By blending proven AI techniques (OCR, Random Forests, forecasting) with design principles for transparency, Smart CA makes financial guidance scalable and trustworthy. Ultimately, Smart CA seeks to democratize expertise: even a solo freelancer without formal accounting training can now understand their financial health and obligations. This work lays groundwork for future AI-driven fintech tools aimed at equity and inclusion.

Acknowledgments: We built this prototype using open-source tools (Supabase, Flask, React) and thank those communities.

VIII. FUTURE SCOPE

Currently implemented as a web application, Smart CA is planned to expand into a dedicated mobile app (Android/iOS) to improve accessibility for users in Tier 2/3 cities. Future enhancements include regional language support (e.g., Hindi, Marathi) to improve inclusivity, and improved OCR capabilities using deep-learning-based models for more accurate document parsing. The platform will integrate bank and UPI data ingestion for automatic transaction capture, and introduce assistive GST filing features to simplify compliance. Additional features will include an in-app chatbot and guided onboarding for first-time users, and personalized financial nudges based on volatility scores to encourage better budgeting and savings behaviour

REFERENCES

- [1] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, and P. M. Atkinson, "Explainable artificial intelligence: an analytical review," *WIREs Data Mining and Knowledge Discovery*, vol. 11, e1424, 2021.
- [2] NITI Aayog, "India's Booming Gig and Platform Economy," Policy Brief, Government of India, 25 June 2022.
- [3] D. Kotios, G. Lamprianidis, I. Deligiorgi, and K. Stefanidis, "Deep learning enhancing banking services: a hybrid transaction categorization and cash flow prediction model," *Journal of Big Data*, vol. 9, no. 1, 2022.
- [4] L. Toran, C. Van Der Walt, A. Sammarone, and A. Keller, "Scalable and weakly supervised bank transaction classification," arXiv preprint arXiv:2305.18430, 2023.
- [5] X.-Q. Chen, C.-Q. Ma, Y.-S. Ren, Y.-T. Lei, N. Q. A. Huynh, and S. Narayan, "Explainable artificial intelligence in finance: a bibliometric review," *Finance Research Letters*, art. no. 104145, 2023.
- [6] B. Sanga, "FinTech and SME finance: a systematic literature review," *FinTech & Innovation*, ScienceDirect, 2023.
- [7] M. (ReceiptSense authors), "ReceiptSense: beyond traditional OCR — a dataset for receipt understanding," arXiv preprint arXiv:2406.04493, 2024.
- [8] A. Abdallah, "A survey of recent approaches to form understanding in scanned documents," *Artificial Intelligence Review*, Springer, 2024.
- [9] M. T. Mohsin and N. B. Nasim, "Explaining the unexplainable: a systematic review of explainable AI in finance," arXiv preprint arXiv:2503.05966, 2025.