iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83753

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

FinFusion: A Multimodal AI Framework for Personal Finance Forecasting, Anomaly Detection, and Intelligent Expense Management

Pooja Bhalekar[1], Marilyn D'Souza[2], Parth Bhegade[3], Yash Bhavsar[4], Rupali Panpaliya[5]
Department of Artificial Intelligence and Machine Learning , P.E.S. Modern College of Engineering , Pune, India

Abstract — Personal finance management has long remained reactive, manual, and devoid of intelligent forward-looking capability. Existing commercial platforms such as Mint, YNAB, and Money Manager function as categorised ledgers, recording past transactions without any predictive or diagnostic intelligence. This paper presents FinFusion, a full-stack AI-powered personal finance tracker that addresses three fundamental dimensions of the problem: prediction, diagnosis, and interaction. The prediction dimension is served by a two-layer stacked Long Short-Term Memory (LSTM) neural network trained on 11 engineered temporal features to forecast daily expenditure over a 30-day horizon. The diagnostic dimension is served by a global z-score anomaly detection pipeline that flags statistically extreme transactions and generates template-based natural language suggestions. The interaction dimension is served by two low-friction data capture modalities — an OCR-based receipt scanner built on pytesseract and OpenCV, and a browser-native voice command interface using the W3C Web Speech API. Additionally, FinFusion incorporates a group expense management module enabling shared expense splitting, net balance ledger maintenance, and simplified settlement computation. The system is trained on a real dataset of 12,030 cleaned expense transactions spanning 2021–2024, and deployed as a FastAPI backend paired with a React single-page application. Empirical evaluation demonstrates robust forecasting across varied spending histories, 422 anomaly candidates detected at a conservative 2.5σ threshold, and OCR parsing achieving 81% confidence on real receipts.
Keywords — LSTM, Personal Finance, Anomaly Detection, OCR Receipt Parsing, Voice Interface, Expenditure Forecasting, Group Expense Management, FastAPI, React, Z-score, Time Series.

I. INTRODUCTION

A. Background

Financial literacy and disciplined personal expense management are critical determinants of long-term economic wellbeing. Despite the proliferation of digital payment infrastructure, the majority of individuals lack real-time visibility into their evolving spending patterns, budget headroom, and future expenditure pressure. The rapid adoption of UPI, digital wallets, and card-based transactions in India has generated unprecedented volumes of structured and semi-structured financial data, which remains overwhelmingly underutilised at the individual consumer level.

Artificial intelligence and machine learning have transformed several domains of financial analytics at the institutional level — credit scoring, fraud detection, portfolio optimisation — yet personal finance applications have lagged. Most consumer-grade tools remain essentially sophisticated spreadsheets: they record, categorise, and visualise historical data but provide no predictive or adaptive intelligence.

B. Motivation

The central motivation for FinFusion arises from three convergent observations. First, personal expenditure exhibits strong temporal structure — recurring rent payments, cyclical weekend leisure spending, and monthly utility bills — which is amenable to sequence modelling via recurrent neural networks. Second, anomalous spending events (unplanned purchases, billing errors, lifestyle inflation) are difficult to detect manually against a noisy transaction background but are statistically tractable. Third, the friction of manual transaction entry remains the single largest impediment to sustained engagement with finance applications, motivating camera-based OCR and voice interfaces as complementary low-friction entry modalities.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

C. Problem Statement

Personal finance management presents a set of interconnected challenges that existing systems fail to address comprehensively. These challenges can be categorised into three primary problems: forecasting, anomaly identification, and user interaction.The forecasting problem involves estimating future expenditure based on historical spending patterns. Given a sequence of daily expenses, the goal is to predict upcoming financial behaviour over a defined time horizon, enabling users to plan budgets and manage resources proactively.The anomaly identification problem focuses on detecting transactions that significantly deviate from a user's typical spending behaviour. Such deviations may arise from unexpected purchases, billing inconsistencies, or behavioural changes, and require reliable detection mechanisms that are both accurate and interpretable.The interaction problem addresses the limitations of traditional data entry methods. Manual input of transactions is time-consuming and prone to user fatigue, resulting in reduced system adoption. Therefore, there is a need for efficient data acquisition techniques that minimise user effort while maintaining accuracy.

Formally, let the user's daily expenditure be represented as a time series : $X = (x1, x2, \ldots, xT)$

The objective of the forecasting task is to estimate future values: $\hat{x}_{t+1}$, $k \in \{1, \ldots, 30\}$

where $\hat{x}_{t+1}$ denotes the predicted expenditure at a future time step.

D. The Proposed Solution: FinFusion

FinFusion addresses all three sub-problems within a unified full-stack platform. A stacked LSTM network with 11 engineered temporal features constitutes the forecasting engine. A global z-score pipeline constitutes the anomaly detector. An OpenCV–pytesseract pipeline constitutes the OCR module. The W3C Web Speech API constitutes the voice interface. A group expense ledger with simplified settlement computation rounds out the collaborative finance feature set.

E. Research Objectives

The specific objectives of this research are: (1) to design and evaluate an LSTM-based multi-step expenditure forecaster operating on a 30-day horizon with 11 engineered features; (2) to implement and characterise a z-score anomaly detection pipeline appropriate for individual spending sequences; (3) to build and assess a receipt OCR pipeline suitable for real-world consumer receipts; (4) to integrate a browser-native voice interaction layer for hands-free data capture; and (5) to validate the complete system on a real, multi-year transaction dataset.

II. LITERATURE SURVEY

Hochreiter and Schmidhuber [1] introduced Long Short-Term Memory networks in 1997 as a solution to the vanishing gradient problem in standard RNNs. The gated cell architecture — comprising forget, input, and output gates — enables LSTMs to selectively retain information across variable-length sequences, making them particularly well-suited for financial time series where dependencies span multiple time scales simultaneously (e.g., daily noise superimposed on weekly cycles and monthly spikes).

Siami-Namini et al. [5] conducted an empirical comparison of ARIMA and LSTM on seven real financial time series and demonstrated that LSTM consistently outperformed ARIMA on non-stationary, non-linear sequences, achieving average RMSE reductions of 85% on stock price series. Fischer and Krauss [6] further validated deep LSTM models for financial market prediction, demonstrating generalisation beyond the training window on out-of-sample S&P 500 data.

Kong et al. [7] applied LSTM to short-term residential electricity load forecasting, a domain with structural analogies to personal expenditure (recurring periodic loads, temporal dependencies), and demonstrated that a two-layer stacked LSTM with rolling window features outperformed SVR and shallow neural baselines. Kaur and Singh [4] extended LSTM to personal finance, training on bank statement data to predict monthly spending at category level, reporting strong performance on rent and utilities categories with stable periodic structure.

For anomaly detection, Liu et al. [2] proposed Isolation Forest as an ensemble-based approach that isolates anomalies by recursively partitioning the feature space. While computationally efficient and effective on high-dimensional data, isolation-based methods require substantial clean training data to establish a stable reference distribution. FinFusion opts for z-score detection given its explainability and robustness to sparse user histories. Smith [3] documented the Tesseract OCR engine, which forms the textual extraction backbone of FinFusion's receipt parsing pipeline.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

The W3C Web Speech API [10], implemented natively in modern browsers, provides the speech recognition substrate without requiring a cloud dependency, aligning with FinFusion's privacy design. Similarly, advancements in browser-based speech recognition technologies, such as the W3C Web Speech API [10], have enabled real-time voice interaction without reliance on external cloud services. These approaches improve accessibility and reduce manual input effort, though they are typically limited to command-based interaction rather than full conversational understanding.While prior work primarily focuses on individual financial analysis, collaborative expense management remains underexplored in research literature, with most implementations limited to application-level solutions rather than integrated intelligent systems.

III. EXISTING METHODOLOGY

Current personal finance platforms fall broadly into three categories: (a) manual ledger applications such as Money Manager and Walnut, which provide categorisation and visualisation of manually entered transactions but offer no predictive capability; (b) bank-integrated aggregators such as Mint and YNAB, which automatically import transactions via open banking APIs but retain the reactive reporting paradigm; and (c) rule-based budget trackers that alert users when spending in a category exceeds a user-defined fixed threshold, without accounting for temporal patterns or user behaviour variability.

None of the surveyed platforms incorporate machine learning for expenditure forecasting. Budget alerts in existing systems are purely threshold-based and thus suffer from excessive false positives when a user's spending legitimately varies month-to-month. Anomaly detection, where present, is coarse and category-agnostic. Data entry friction remains a persistent issue: the majority of applications require manual description, amount, and category entry for each transaction, creating abandonment risk over multi-week usage horizons. The absence of OCR-based receipt capture and voice interaction modalities in existing tools constitutes the primary interaction-layer gap that FinFusion addresses.

Platform	Forecasting	Anomaly Detection	OCR Input	Voice Input	Group Expense
Mint	None	Rule-based	No	No	No
YNAB	None	None	No	No	No
Walnut	None	None	No	No	No
Money Manager	None	None	No	No	No
FinFusion	LSTM(30-day)	Z-score	Yes	Yes	Yes

Table I. Comparative analysis of existing personal finance platforms vs. FinFusion

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

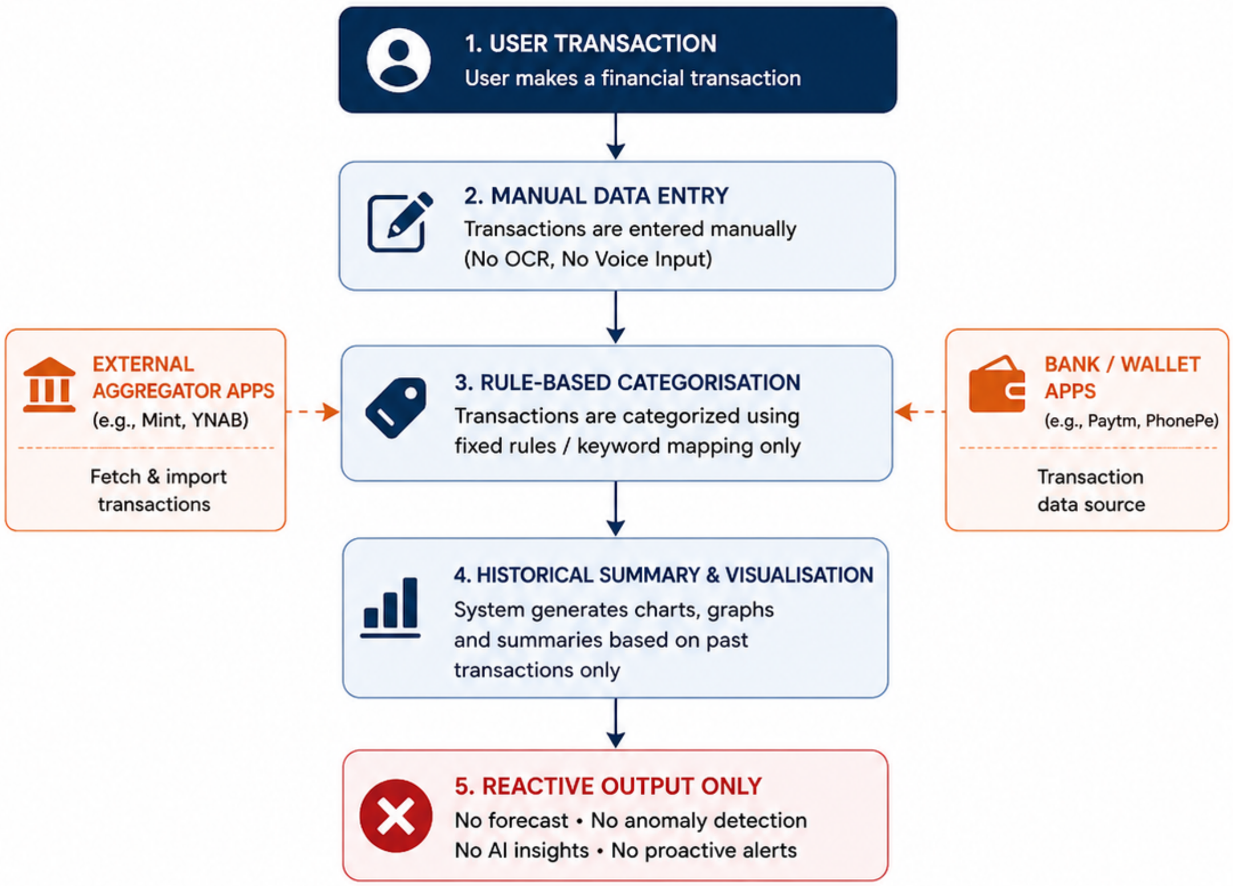


Fig. 1. Conventional Finance Tracking System (Reactive Model)

IV. PROPOSED SYSTEM

FinFusion is designed as a decoupled full-stack system that integrates data processing, machine learning, and user interaction within a unified architecture. The backend is implemented using an asynchronous FastAPI framework, exposing RESTful endpoints for transaction management, analytics computation, LSTM-based forecasting, anomaly detection, receipt parsing, and group expense operations. Data persistence is handled through a SQLAlchemy ORM layer backed by SQLite. The frontend is developed as a React-based single-page application, which interacts with the backend via HTTPS APIs to render dashboards, visualisations, and user controls.

The system is structured as a three-layer analytical pipeline comprising the data layer, analytics layer, and presentation layer. At the data layer, transactions collected through manual entry, OCR-based receipt scanning, or voice commands are standardised, categorised into predefined financial categories, and stored in the database. At the analytics layer, transaction data is aggregated into a daily time series, followed by feature engineering to construct model-ready inputs. These features are processed by a stacked Long Short-Term Memory (LSTM) module for multi-horizon budgeting and expenditure forecasting, alongside a statistical anomaly detection component to generate predictive and diagnostic outputs.

At the presentation layer, the processed information is delivered to the user through an interactive interface, providing expenditure forecasts, anomaly alerts, and visual summaries for improved financial awareness. A key design decision in the system is the separation of personal and group financial data, featuring a **dedicated group expense module with Splitwise-style settlement tracking**. Group expenses, including shared costs and debt ledgers, are maintained independently and excluded from the analytical pipeline to prevent distortion of individual spending patterns during model training and inference. This ensures that the predictive and diagnostic components operate on a consistent representation of personal financial behaviour.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

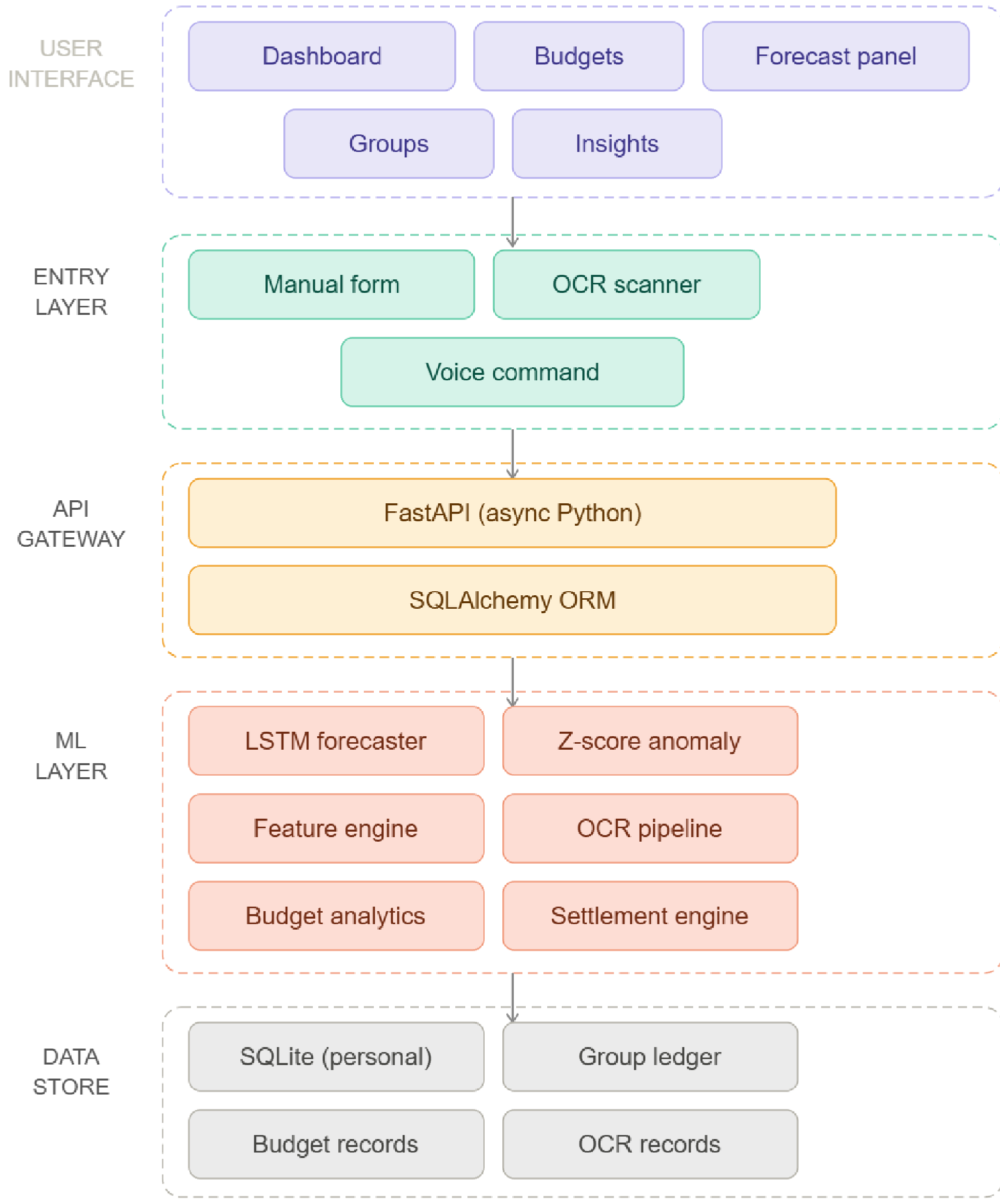


Fig. 2. Layered Architecture of the Proposed FinFusion System

V. SYSTEM ARCHITECTURE

FinFusion follows a modular multi-layer architecture designed to integrate intelligent financial analytics, forecasting, anomaly detection, budgeting, OCR-based transaction extraction, voice interaction, and collaborative expense management within a unified platform. The architecture separates the frontend interface, backend services, machine learning modules, and storage systems into independent layers, ensuring scalability, maintainability, and efficient communication between components. The layered design also enables the forecasting and anomaly detection pipelines to operate independently from collaborative group expense transactions, thereby preserving the integrity of personalised financial analysis.

A. User Interaction and Presentation Layer

The presentation layer acts as the primary interaction interface between the user and the system. It is implemented using React 18 as a Single Page Application (SPA) and provides a responsive dashboard for financial monitoring and analysis.

Users can access the platform through web or mobile browsers and interact with multiple modules including Dashboard, Transactions, Budgets, Forecast Panel, Anomaly Insights, Reports, and Groups/Splitwise management.

The dashboard visualises financial summaries, category-wise expenditure, budget utilisation, and AI-generated insights. The forecasting panel displays predicted expenditure trends and confidence estimates, while the anomaly insights module presents unusual spending behaviour detected by the AI engine. The group expense interface enables users to create groups, split expenses, and manage settlements collaboratively. Additionally, the frontend supports receipt uploads for OCR processing and browser-based voice commands using speech recognition.

B. API Gateway and Backend Services Layer

The backend layer is implemented using FastAPI with asynchronous request handling and functions as the central orchestration unit of the system. It manages API communication, business logic, authentication, transaction processing, and coordination between frontend modules and machine learning services.

The API gateway handles request validation, JWT-based authentication, authorization, and rate limiting. Several dedicated backend services operate within this layer. The Transaction Service manages CRUD operations and expense categorisation, while the Budget Service tracks user-defined and dynamically generated budget limits. The Forecast Service communicates with the LSTM forecasting engine to generate expenditure predictions, and the Anomaly Service processes unusual spending behaviour and AI-generated suggestions.

The backend also includes OCR and Voice Services. The OCR service handles receipt image uploads, text extraction, and field parsing, whereas the Voice Service processes spoken commands and maps them to corresponding dashboard actions. Group-related operations such as group creation, expense splitting, balance tracking, and settlement computation are managed through the Group Service and Split & Settle Service modules.

C. ML and Intelligence Layer

The ML and Intelligence Layer forms the analytical core of FinFusion and integrates forecasting, budgeting intelligence, anomaly detection, feature engineering, and splitwise optimisation. This layer transforms raw financial data into predictive and interpretable insights.

The LSTM Forecasting Engine models user expenditure as a time-series learning problem. It analyses historical transaction sequences, rolling averages, lag features, and calendar-based patterns to generate recursive 30-day expenditure forecasts. The generated predictions are further stabilised using post-processing and confidence estimation techniques before being displayed within the forecasting dashboard.

The Budgeting Engine operates on top of the forecasting module and converts predicted spending behaviour into adaptive budget recommendations. Instead of relying entirely on manually defined limits, the engine dynamically estimates future expenditure and computes safe budget thresholds by incorporating forecast-based safety margins. This allows the platform to provide proactive financial guidance and overspending alerts.

The Anomaly Detection Engine identifies irregular financial behaviour relative to the user’s historical spending baseline. It combines statistical techniques such as z-score analysis with Isolation Forest-based anomaly scoring to detect unusual spending spikes, abnormal category-level behaviour, and significant deviations in transaction patterns. The detected anomalies are transformed into explainable AI-generated suggestions and insight cards to improve interpretability and user understanding.

The Splitwise Intelligence Engine manages collaborative financial activities and group expense optimisation. It supports equal, percentage-based, and custom expense splitting while maintaining settlement ledgers between members. The engine computes net balances, minimises transaction complexity, and generates settlement recommendations for efficient debt resolution.

Supporting these analytical modules is the Feature Engineering Component, which prepares structured input sequences for machine learning models. It generates rolling statistics, lag features, cyclical temporal encodings, category representations, and behavioural indicators that improve forecasting and anomaly detection performance.

D. Data Storage Layer

The Data Storage Layer is implemented using SQLAlchemy ORM with SQLite and serves as the persistent storage component of the platform. It stores user profiles, transaction records, budget configurations, forecast results, anomaly logs, OCR records, and group settlement ledgers in structured relational tables.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

Separate storage structures are maintained for personal transactions and collaborative group expenses to ensure that group-based financial activities do not interfere with personal forecasting and anomaly detection pipelines. OCR records store extracted receipt text and parsed transaction details, while anomaly logs maintain detected irregularities, severity levels, and generated AI suggestions for future analysis and reporting.

E. Group Expense and Splitwise Layer

The Group Expense Layer enables collaborative financial management and functions independently from the personal analytics pipeline. Users can create groups, invite members, add shared expenses, split transactions using multiple methods, and track settlement balances within a shared ledger system.

The module supports equal splits, percentage-based splits, and custom share allocations. Settlement records are dynamically updated whenever members settle payments, allowing the system to maintain accurate balance summaries and minimise redundant transactions. By isolating group expenses from the personal ML pipeline, the system ensures that shared expenses do not distort individual expenditure forecasting or anomaly detection results.

F. External Services Layer

The External Services Layer integrates third-party services required for storage, notifications, and speech processing. Cloud storage services such as AWS S3 or Cloudinary are used to store uploaded receipt images securely. Notification services are responsible for sending budget alerts, anomaly warnings, settlement reminders, and forecast notifications through web or mobile channels.

The platform also integrates the W3C Web Speech API for browser-native speech recognition. This enables voice-based dashboard navigation and spoken financial queries without requiring dedicated cloud NLP infrastructure, thereby reducing backend complexity while improving accessibility and user interaction.

G. Overall System Workflow

The overall workflow of FinFusion begins with user interaction through the frontend dashboard. User requests and transaction data are routed to backend services through RESTful APIs. The backend stores and aggregates financial records before forwarding structured data to the ML and Intelligence Layer.

The Feature Engineering Component transforms raw transactions into analytical features suitable for machine learning models. The LSTM engine generates expenditure forecasts, which are subsequently used by the Budgeting Engine for adaptive budget estimation. Simultaneously, the Anomaly Detection Engine analyses spending behaviour to identify irregularities and generate explainable insights. OCR and Voice Services process receipt images and spoken commands respectively, while the Splitwise Engine manages collaborative expense settlements. The processed outputs are then returned to the frontend layer and visualised through interactive dashboards, charts, reports, and AI-generated insight panels, providing users with an intelligent and proactive personal finance management experience.

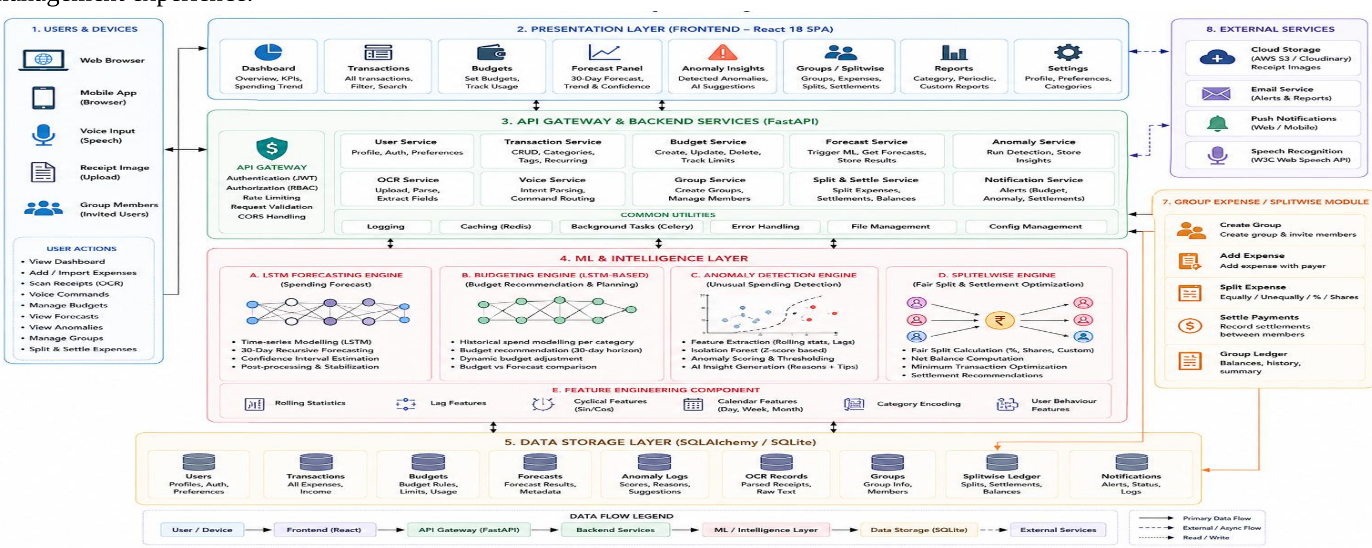


Fig. 3. Comprehensive Architecture of the FinFusion System

Layer	Technology	Purpose
Frontend	React 18 (SPA)	Dashboard, charts, user interaction
Backend	FastAPI (Async Python)	APIs, business logic, analytics handling
Database	SQLAlchemy, SQLite	Store transactions, forecasts, groups
Forecasting Module	TensorFlow/Keras, NumPy	LSTM-based expenditure forecasting
Budgeting Module	Python, Forecast Logic	Dynamic budget estimation and tracking
Anomaly Detection	Scikit-learn, Z-score Analysis	Detect unusual spending behaviour
OCR Pipeline	pytesseract, OpenCV, Pillow	Receipt scanning and text extraction
Splitwise Engine	Python Backend Services	Expense splitting and settlements
Voice Interface	W3C Web Speech API	Voice recognition and commands
External Services	AWS S3 / Cloudinary	Receipt image storage and notifications

Table 2. Technology stack by system layer

VI. ALGORITHM AND MODEL DESIGN

A. Feature Engineering

Raw transaction data is first aggregated to scalar daily total spend $x_t \in \mathbb{R} \geq 0$. The input sequence $X = (x_1, x_2, \ldots, xT)$ is transformed into a feature matrix $F \in \mathbb{R}T\times 11$ using the following engineering pipeline:

$\tilde{x}_t = \log(1 + x_t)$ - *Log-transformation of daily spend to stabilise heavy-tailed magnitude variance*

To capture short-term and medium-term expenditure behaviour, rolling averages are computed over 7-day and 30-day windows:

$$\mu_t^{(7)} = \frac{1}{7}\sum_{i=0}^{6} x_{t-i} \qquad \mu_t^{(30)} = \frac{1}{7}\sum_{i=0}^{29} x_{t-i}$$

,

Local spending volatility is modelled using the rolling standard deviation:

$$\sigma_t^{(7)} = \sqrt{\tfrac{1}{7}\sum_{i=0}^{6}(\tilde{x}_{t-i} - \mu_t^{(7)})^2}$$

Temporal periodicity is encoded using cyclical sine and cosine transformations for day-of-week and day-of-month information:

$$dow\_sin_t = \sin\left(\frac{2\pi d_t}{7}\right), \quad dow\_cos_t = \cos\left(\frac{2\pi d_t}{7}\right)$$

$$dom\_sin_t = \sin\left(\frac{2\pi m_t}{31}\right), \quad dom\_cos_t = \cos\left(\frac{2\pi m_t}{31}\right)$$

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

where:

- **$d_t \in$ {0,1,…,6}** represents the day of the week,
- ***$m_t \in$* {1,2,…,31**} represents the day of the month

Additionally, three binary calendar indicators are included:

- weekend_flag
- month_start_flag
- month_end_flag

The final engineered feature set contains 11 features per time step and is standardised using StandardScaler before being passed to the LSTM network.

Feature Group	Count	Fields
Spend magnitude	1	L Log(1 + x_t)
Rolling context	3	$\mu^{(7)}$, $\mu^{(30)}$, $\sigma^{(7)}$ (all log-transformed)
Cyclical calendar	4	dow_sin, dow_cos, dom_sin, dom_cos
Binary flags	3	weekend, month_start, month_end
Total	11	Matches runtime LSTM input tensor exactly

Table 3. Feature Engineering Breakdown

B. LSTM Architecture and Cell Equations

FinFusion uses a stacked Long Short-Term Memory (LSTM) neural network to model sequential expenditure behaviour and capture long-range temporal dependencies. The LSTM memory update equations at each time step ttt are defined as follows.

Short-Term Memory (LSTM) neural network to model sequential expenditure behaviour and capture long-range temporal dependencies. The LSTM memory update equations at each time step ttt are defined as follows.

- Forget gate:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

- Input gate:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

- Cell candidate:

$$\tilde{C}_t = tanh(W_C[h_{t-1}, x_t] + b_C)$$

- Cell state update:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

- Output gate:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

- Hidden state:

$$h_t = o_t \odot tanh(C_t)$$

The implemented forecasting architecture consists of two stacked LSTM layers followed by dense fully connected layers.

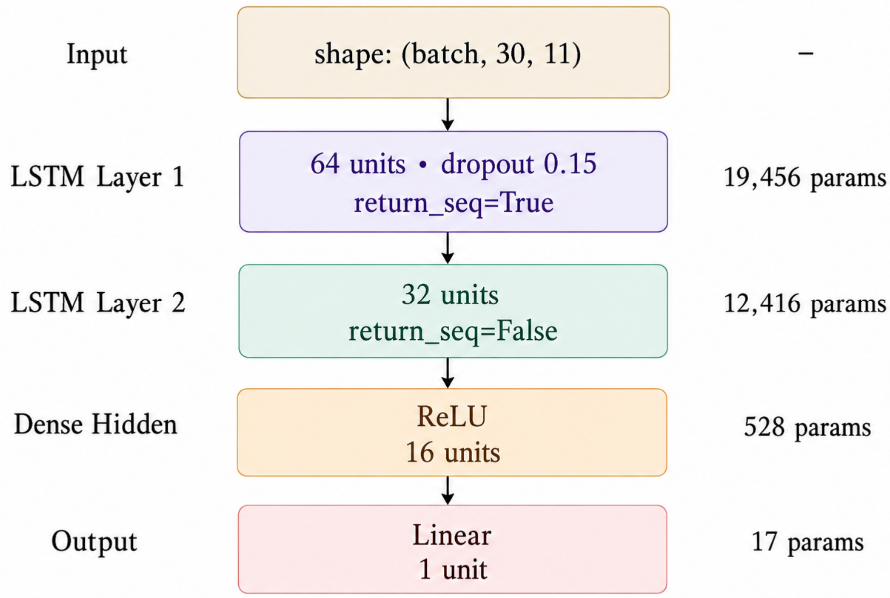


Fig . 4. LSTM model architecture with layer sizes and parameter counts

C. Recursive Multi-Step Forecasting

The trained LSTM performs single-step prediction, which is recursively extended into a 30-day forecasting horizon using autoregressive rollout. At each future step k, the predicted expenditure is appended back into the rolling history and reused as input for subsequent predictions:

$$\hat{x}_{t+k} = f_{LSTM}(F_{t+k-30:t+k-1}), \quad k = 1, 2, \ldots, 30$$

where:

- f LSTMdenotes the trained forecasting model,
- $F_{t+k-30:t+k-1}$ represents the rolling feature window.

After each recursive step, rolling statistics and temporal features are recomputed dynamically using the updated expenditure history.

D. Forecast Confidence Score

To quantify prediction reliability, FinFusion computes a confidence score based on the backtesting Root Mean Square Error (RMSE).

The confidence score is defined as:

$$\text{confidence} = \text{clip}\left(1 - \frac{RMSE_{backtest}}{\mu_{ref}}, 0.35, 0.95\right)$$

where:

- $RMSEbacktest$ represents forecasting error over held-out validation sequences,
- μref denotes the reference mean expenditure.

The clipping function constrains the confidence score within the range [0.35,0.95] to avoid unrealistic certainty estimates.

VII. MODULE-WISE IMPLEMENTATION

A. Login and Authentication Module

The Login and Authentication Module provides secure user access control for the FinFusion platform. As shown in Fig. 5, the login interface is implemented as a minimal React-based authentication screen with email-password credential validation and demo-account access support.

The module acts as the entry point to the platform and ensures that all financial records, forecasts, anomaly logs, budgets, and group transactions remain user-specific and securely isolated.

User authentication is handled through the FastAPI backend using JWT (JSON Web Token)-based session management. Upon credential submission, the frontend sends a POST request to the authentication endpoint, where the entered email and password are validated against the stored user database. Passwords are securely stored in hashed form using backend encryption utilities, preventing plaintext credential exposure.

After successful authentication, the backend generates a signed JWT access token containing the authenticated user's identity and session metadata. The token is returned to the frontend and stored within the client session for subsequent authenticated API requests. Protected endpoints such as /dashboard, /forecast, /anomalies, /groups, and /transactions require valid token verification before granting access.

The login interface includes support for demo accounts to simplify system evaluation and live demonstrations. Two predefined demo modes are provided:

- Preloaded Demo Account — populated with historical financial data, forecasts, anomaly logs, and group transactions for showcasing full system functionality.
- Empty Demo Account — a clean account used for real-time manual transaction testing and OCR demonstrations.

The frontend interface provides real-time validation feedback for invalid credentials and redirects authenticated users directly to the dashboard overview upon successful login. Session persistence is maintained across page reloads through token-based authentication state management. Unauthorized requests automatically trigger session invalidation and redirect the user back to the login screen.

The authentication layer integrates with all backend services, ensuring secure access control across forecasting, budgeting, anomaly detection, OCR receipt processing, and collaborative group expense management modules.

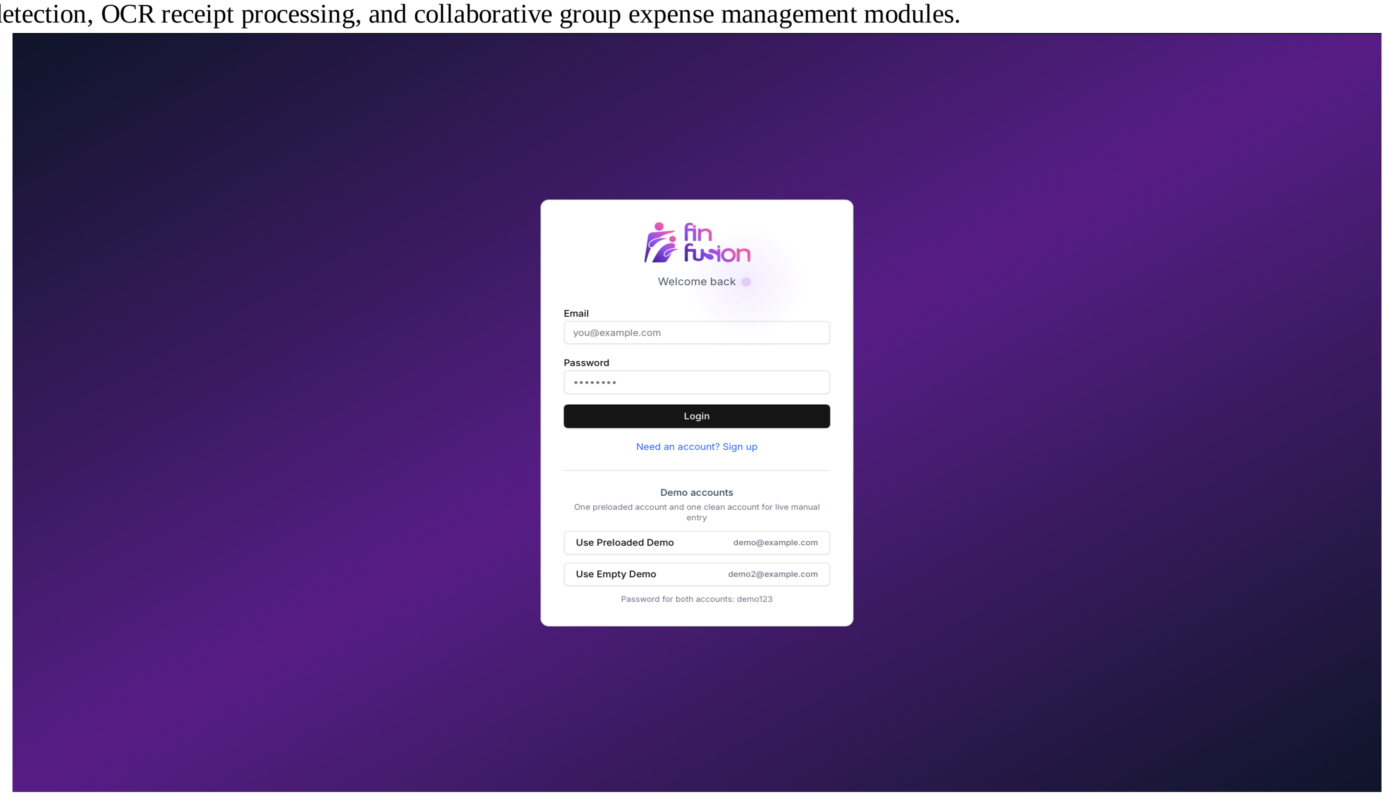


Fig . 5. Login page

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

B. Dashboard Module

The Dashboard serves as the primary financial overview screen. It fetches aggregated KPIs from the /dashboard/summary and /dashboard/categories endpoints and renders four headline metric cards: total budget, total spending, budget utilisation percentage, and net savings. A donut chart built with Recharts displays the current-month spending breakdown by category. A budget tracking panel on the right sidebar shows per-category utilisation bars, enabling at-a-glance identification of category-level overruns. The Recent Expenses list presents the latest transactions with category tags, and a + FAB triggers the manual expense entry dialog. AI-generated data insights, derived from the anomaly and forecasting engines, are surfaced inline as contextual cards beneath the budget tracker.

As visible in Fig. 6, the December 2024 snapshot shows a balance of ₹1,95,432.66 against a budget of ₹16,77,803.66, with total spending of ₹14,82,371.00 — 88% budget utilisation — classified across nine categories in the donut breakdown.

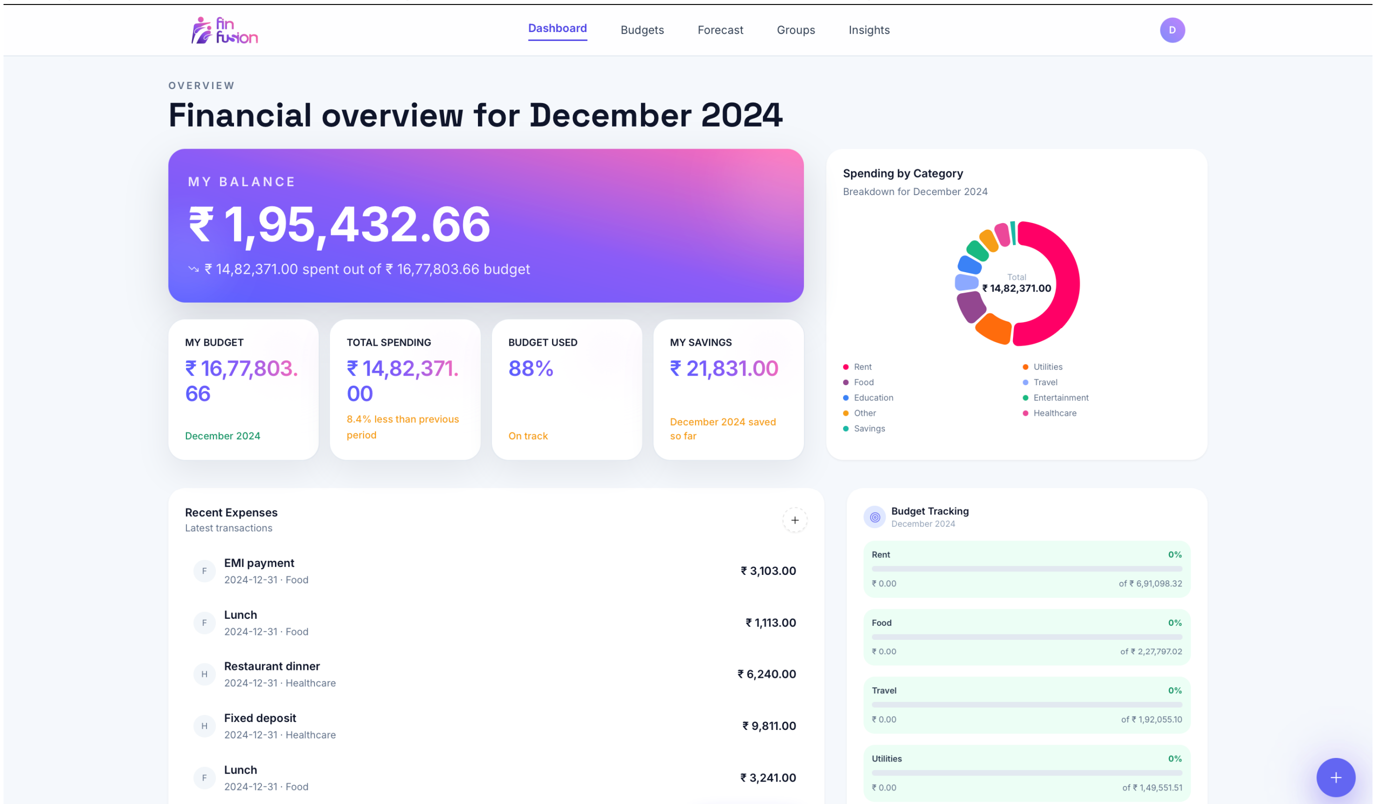


Fig . 6. Dashboard page

C. OCR Receipt Scanning

The receipt scanner implements a twelve-step image-to-expense pipeline, depicted in Fig. 7. The user uploads a JPG or PNG receipt (maximum 5 MB). The backend applies an EXIF-aware orientation correction via Pillow.ImageOps.exif_transpose to normalise mobile captures, then upscales narrow receipts to a minimum width of 1400 px for improved OCR legibility. Grayscale conversion and autocontrast equalisation reduce colour noise. A contrast enhancement factor of 1.8, sharpening pass, and binary thresholding at pixel value 165 produce a high-contrast text image. Tesseract OCR is then run with PSM 6 (uniform text block), suited to the column-aligned layout of retail receipts.

Post-OCR, four extraction routines recover the transaction fields:

1Amount: Regex targeting currency symbols (₹, $, £) followed by decimal values; the grand total keyword match selects the largest candidate amount.

2Date: Multi-format regex covering YYYY-MM-DD, DD/MM/YYYY, two-digit-year slash formats, and textual month names.

3Category: Merchant name and line-item keyword heuristics mapped to the nine canonical categories.

4Description: Receipt-line heuristics extracting the primary line-item or merchant name for the expense form pre-fill.

Overall confidence is computed as the arithmetic mean of four field-level confidence scores: $c = (c_{amount} + c_{date} + c_{category} + c_{description}) / 4$. Extractions with total text length below 10 characters are rejected.

The user is presented with a pre-filled editable form — as shown in the Kalika Dairy Farm receipt scan (Fig. 8) — for review and confirmation before the expense is committed to the database. An 81% confidence score was observed on the test receipt, with correct extraction of the ₹470 grand total, 28/03/2026 date, and Food category.

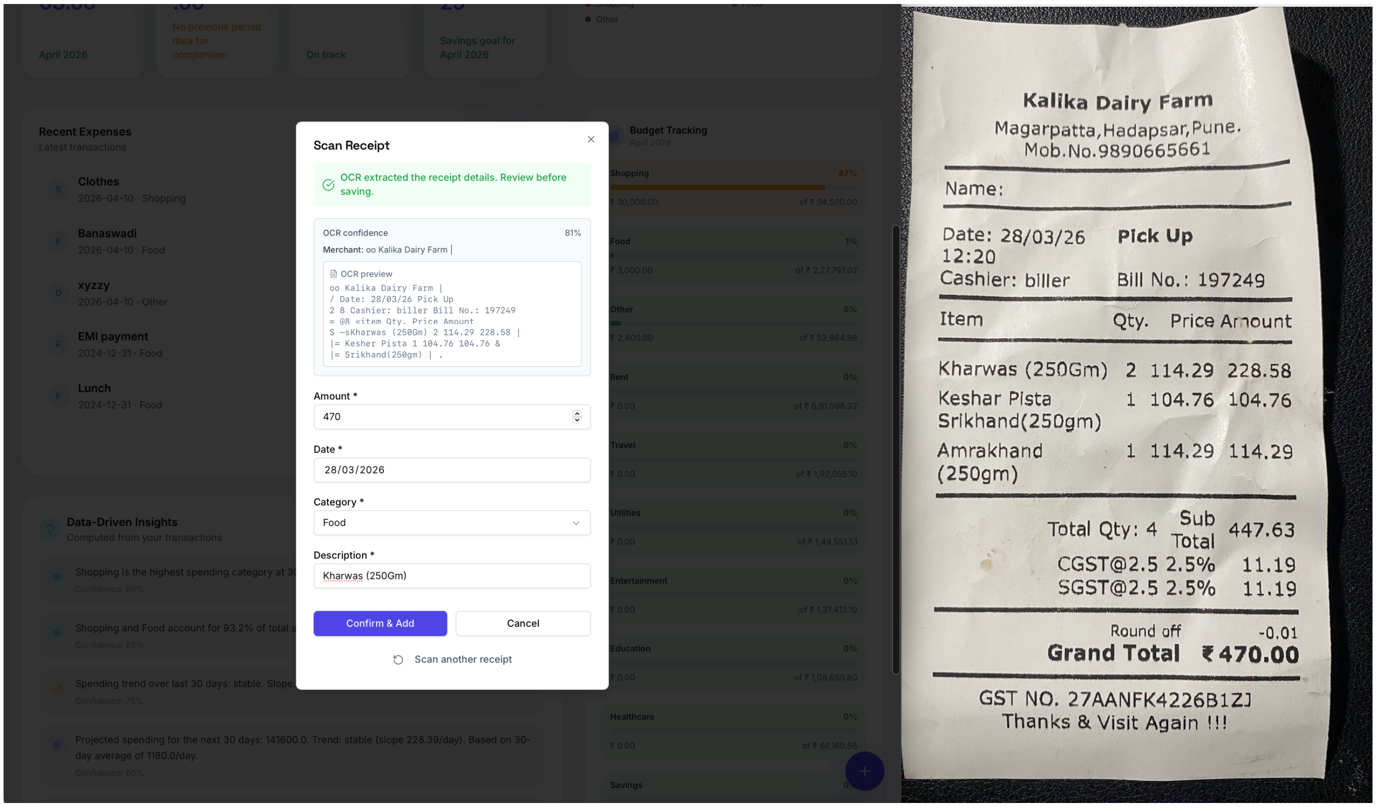


Fig . 7. OCR scanning receipt page

D. Manual Entry

The manual expense entry flow is accessible from a floating action button present on every screen. Users provide a description, amount, date, and category through a modal dialog. Category selection uses a dropdown populated with the nine canonical categories. The entry is immediately persisted through POST /transactions and reflected in all dependent views — the dashboard summary, budget gauges, and the forecasting engine's rolling history — without requiring a page reload, leveraging React's state management layer for real-time UI updates.

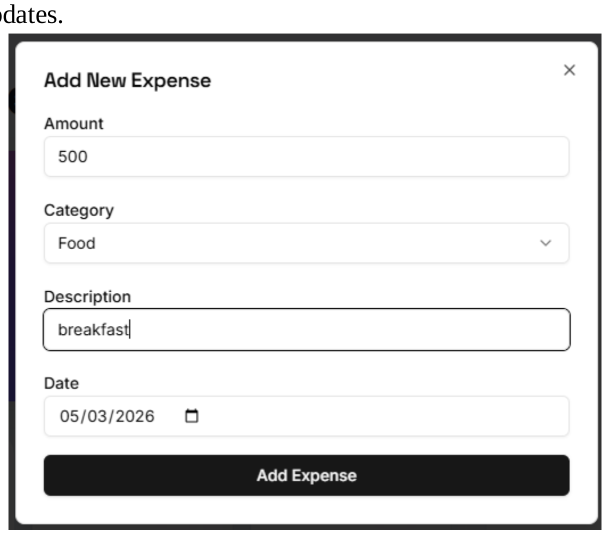


Fig . 8. Manual entry input

E. Voice-Based Input

The voice interaction layer is implemented as a browser-native frontend component using the W3C SpeechRecognition interface. No cloud speech service or backend NLP is involved: audio is processed locally by the browser engine, preserving user privacy and eliminating external API latency. The recognition pipeline proceeds through three stages — microphone activation, utterance capture in single-shot mode (continuous: false, interimResults: false), and transcript normalisation — before intent matching via keyword pattern extraction.

Six intent categories are supported: navigation (e.g., "go to forecast"), category filter ("show food expenses"), time filter ("show last month"), summary request ("what did I spend on healthcare"), forecast request ("predict next month"), and anomaly request ("show anomalies"). The UI indicator cycles through three states: Idle → Listening (green pulse) → Processing (amber spin). Unrecognised utterances return the indicator to Idle with a "please rephrase" prompt, while recognised commands are visually confirmed before execution.

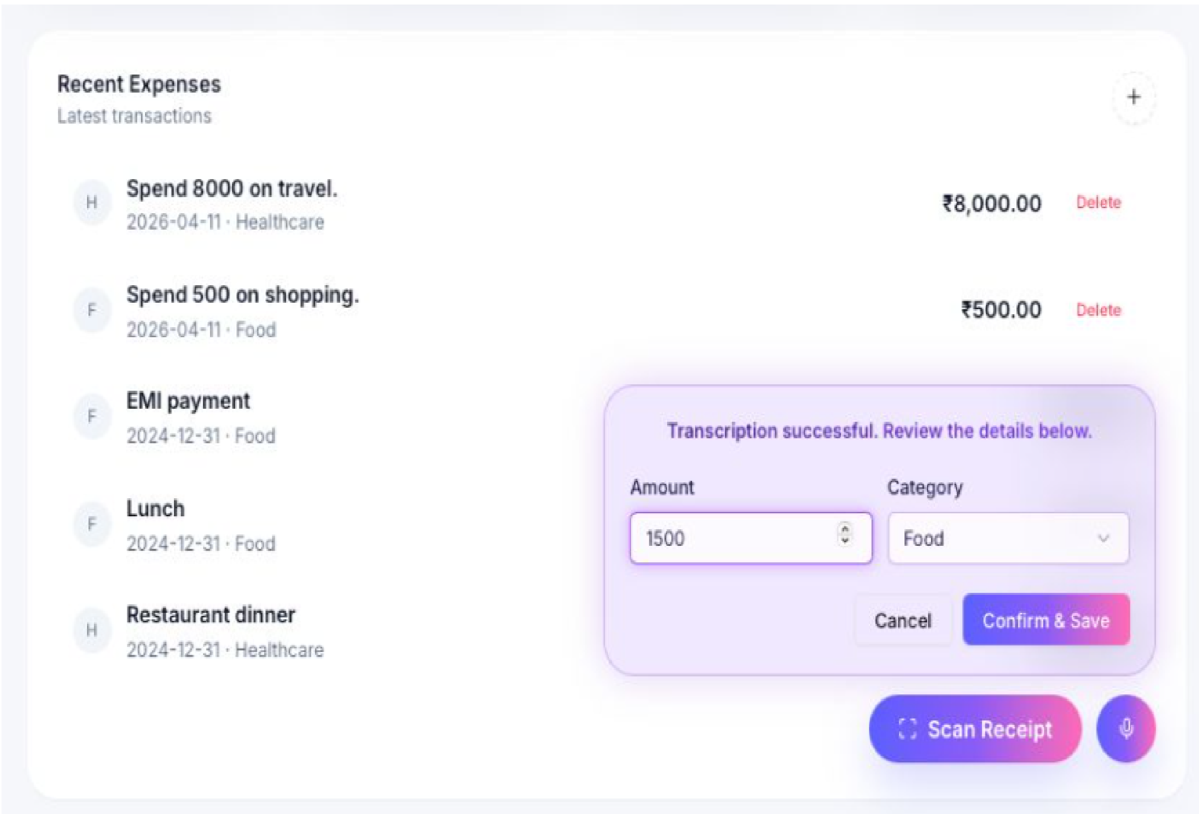


Fig . 9. Voice based input

F. Budgeting Module

The Budgets page (Fig. 10) displays a card-per-category layout under the current-month overview banner. Each card shows the category budget, an "Adjust" edit control, a visual utilisation progress bar, and the current spending amount with an On Track / Over Budget badge. A global overview banner at the top shows total utilised budget, percentage utilisation, and a data insight comparing current-month spend to the previous month — e.g., "Total spending decreased by 8.4% (2024-12 vs 2024-11): current ₹14,82,371 vs previous ₹16,18,238." This insight is generated by the analytics service at 90% confidence.

Budget categories are user-configurable: the "+ Add Category" control allows adding new spending buckets. The "Adjust" link per card opens an inline editor for the budget ceiling. Over-budget categories are highlighted in red with an explicit overage amount (e.g., Rent: Over by ₹1,19,255.39). Budget data is served through a dedicated analytics endpoint that computes current-period spend against stored budget limits and attaches the month-over-month insight from the anomaly/analytics service.

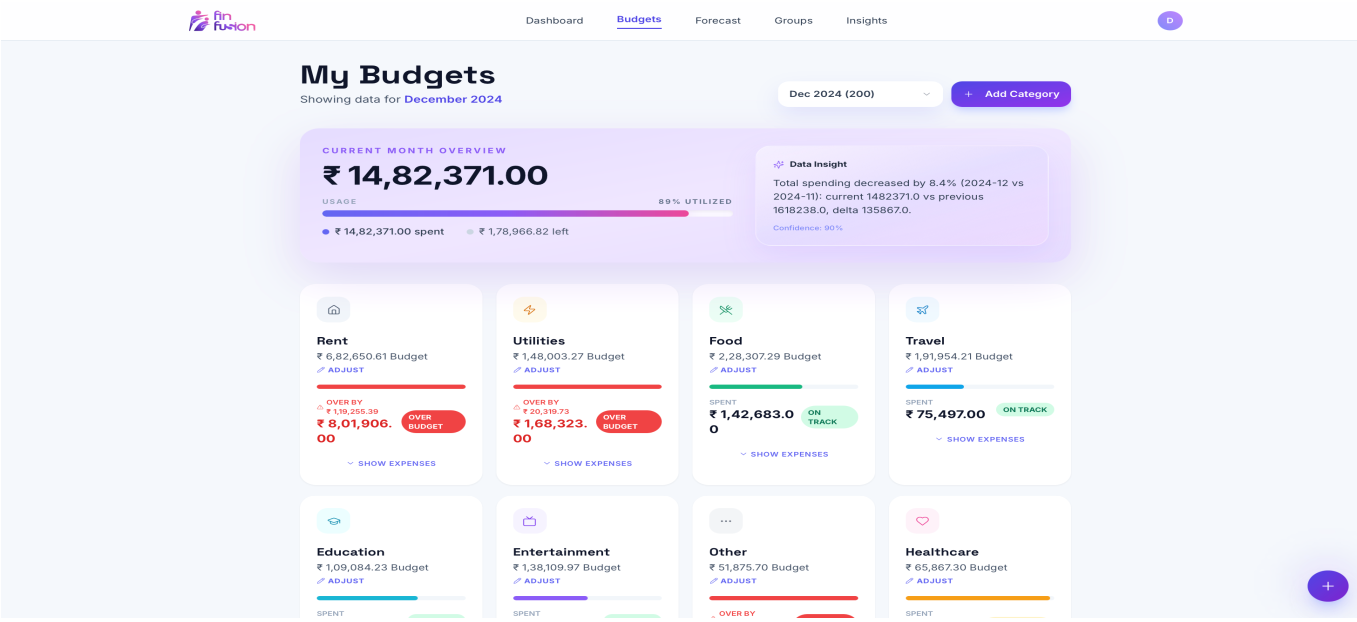


Fig . 10. Budget page

G. Forecasting Module

The Forecast page (Fig. 11) presents the 30-day projected spending headline, trend label, forecasting method, and confidence percentage. A daily/weekly toggle switches the chart granularity. The LSTM inference path executes at the GET /forecast endpoint: the backend assembles the last 30 days of aggregated daily spend, applies feature engineering, scales the feature matrix, and performs the 30-step recursive rollout. Post-processing applies expected-value blending (adaptive 0.65/0.35 to 0.20/0.80 neural-to-calendar blend), a dynamic floor of max(0.35 × expected, 0.20 × recent_mean), and a ceiling of max(2.4 × expected, recent_mean + 2.5 × recent_std).

If the LSTM path is unavailable (TensorFlow not loaded) or if aggregated history is below the 60-day minimum, the service automatically falls back to a statistical forecast combining moving average and linear trend extrapolation. Four derived insight cards accompany the forecast: (a) 30-day projected total, (b) trend direction (slope of linear fit over the forecast window), (c) peak spending day (arg max of the forecast series), and (d) top-category projection (distributed by trailing 120-day category share). The AI Insights panel on the right surfaces Liquidity Alert (peak outflow date) and Category Pressure alerts when any category is projected to exceed its historical cap.

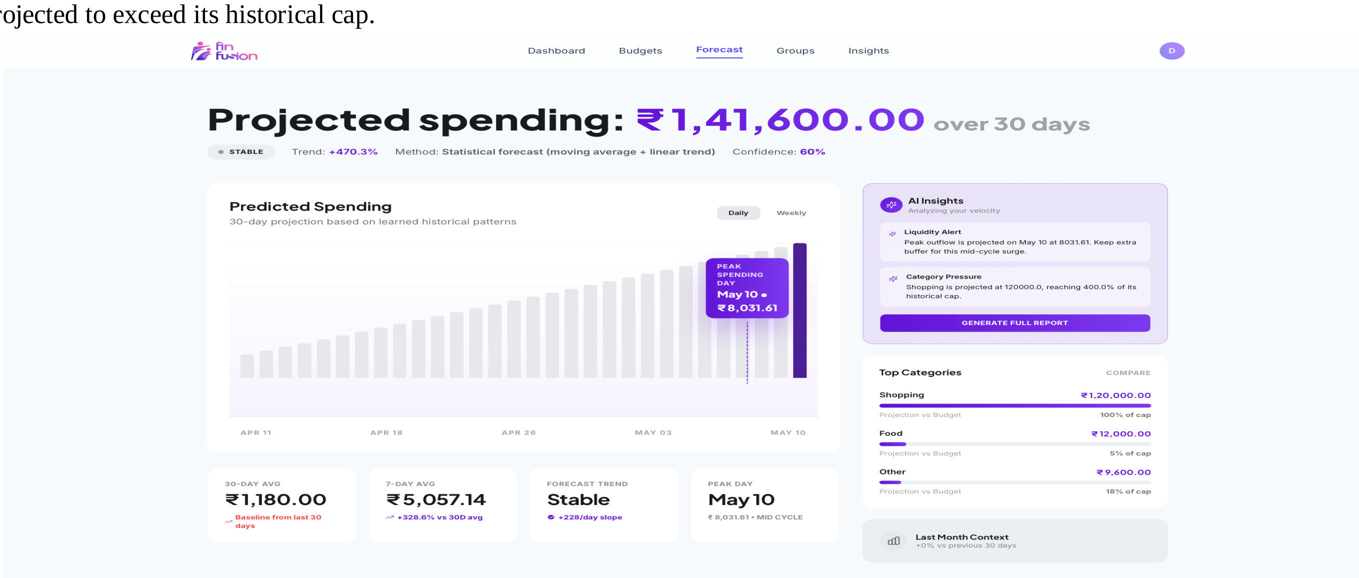


Fig . 11. Forecast page

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

H. Explainable AI and Insights

FinFusion's explainability layer operates at two levels. At the anomaly level, each flagged transaction is accompanied by a structured suggestion card containing the anomaly type, the triggering z-score, the absolute deviation from the user's mean (in ₹ and percentage), and a template-based natural language interpretation. For example, a transaction with z = 4.2 in the Healthcare category generates the card: "Unusually high Healthcare expense of ₹49,500 detected — 652% above your monthly norm of ₹6,584. This may indicate an unplanned medical event." The deterministic template approach ensures that the same anomaly always produces identical, auditable output — a deliberate design choice over LLM-generated text.

At the forecast level, the derived insights (trend slope, peak day, category pressure, liquidity alert) constitute a second layer of explainability, grounding the numeric forecast in human-interpretable observations. The dashboard's Data Insight cards also surface month-over-month spend comparison, confidence levels, and spending velocity observations that contextualise the quantitative outputs for non-technical users.

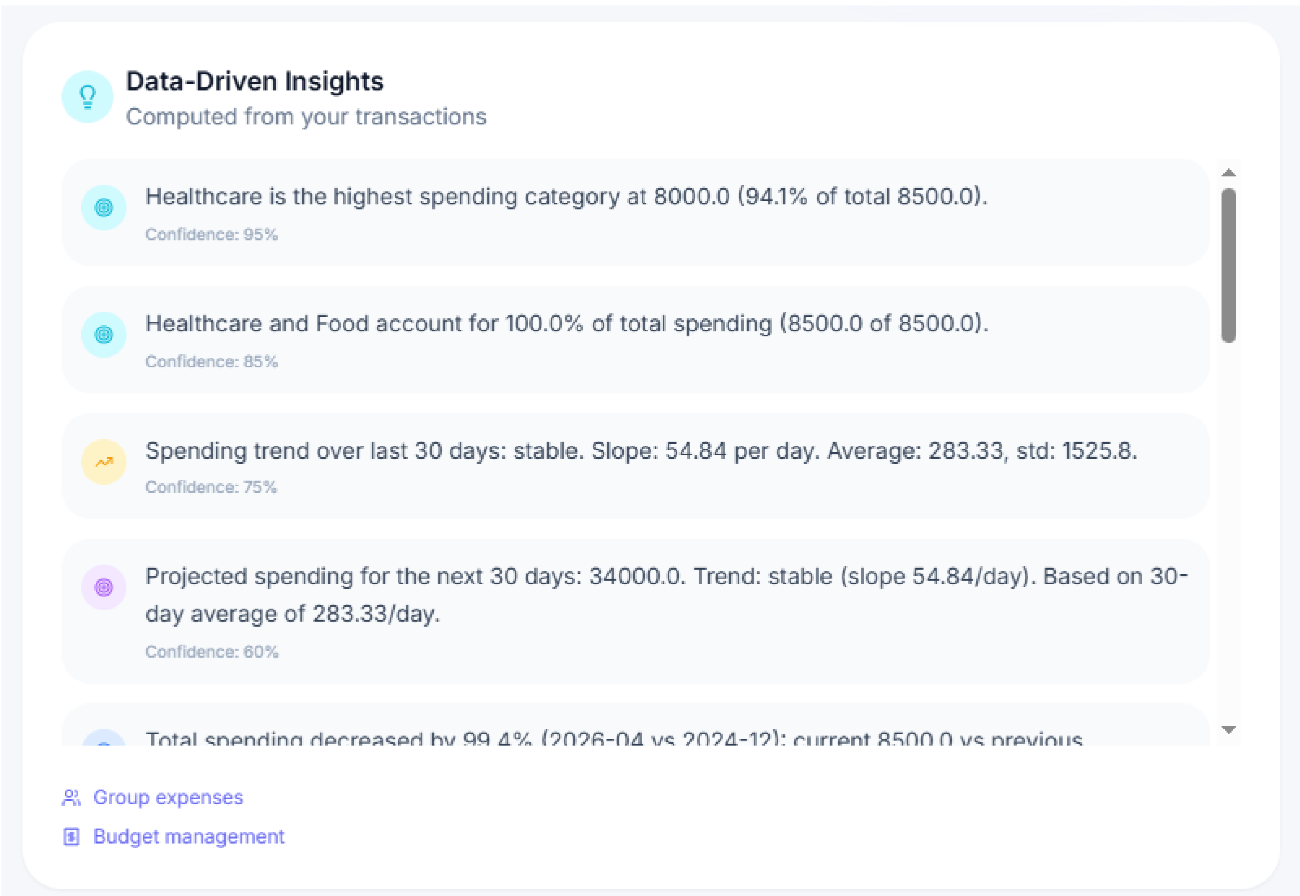


Fig . 12. AI Suggestion and Anomaly detection

I. Insights Page

The dedicated Insights page (Fig. 13) displays historical trends spanning the full dataset calendar range (2021-01-01 to 2026-04-10). A Monthly Spending Trend line chart shows total spend over time with Recent 12M and All Time toggle controls. A Category Trends Over Time stacked bar chart renders category-level contributions per month, enabling users to visually identify structural shifts — e.g., sustained increases in Rent or seasonal spikes in Travel. A Top Categories panel ranks all-time spending by category: Rent (₹2,81,15,610), Food (₹93,86,915), Travel (₹75,65,214), Utilities (₹58,81,271), Entertainment (₹50,01,180), and Education (₹39,85,575).

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

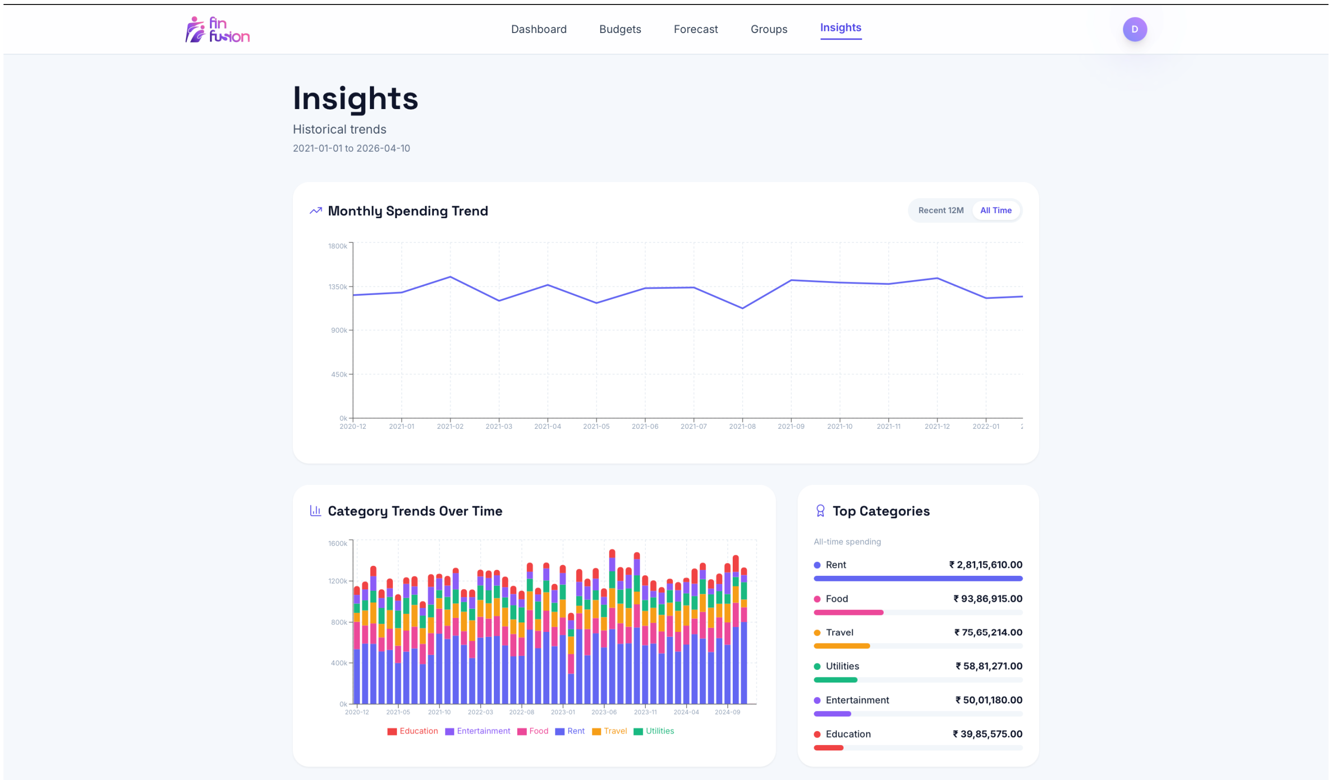


Fig . 13. Insights page

J. Group Expense Management

The Groups module (Figs. 14) provides Splitwise-style shared expense management. A user creates a named group and adds members by username. Expenses are added with title, amount, and payer designation; the system computes per-member shares using equal-split or custom-split logic. A running net balance ledger maintains per-pair obligations:

(9)net_balance(A→B) = Σ(A owes B) − Σ(B owes A)*The Balances & Settlements panel displays each member's net position; simplified transaction instructions (e.g., "Pooja pays Marilyn ₹900") minimise the number of transfers needed to clear all obligations.*

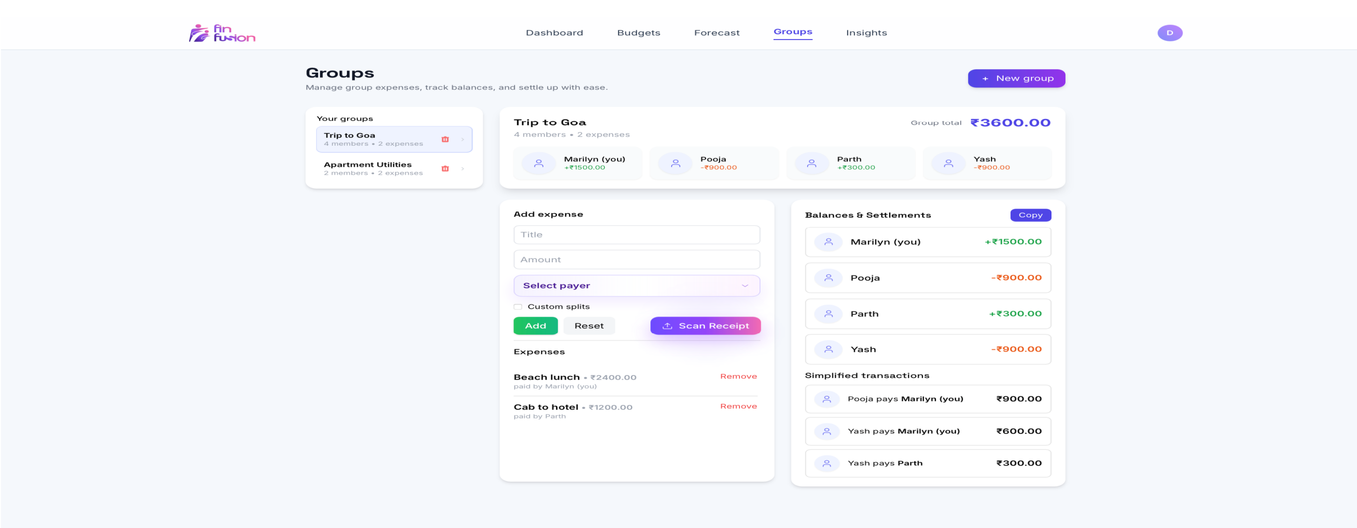


Fig . 10. Group Expense and Settlement page

VIII. RESULTS AND PERFORMANCE ANALYSIS

A. Dataset Statistics

The FinFusion production demo is backed by a multi-year transaction dataset derived from the budgetwise.csv pipeline and persisted into the SQLite demo database. The dataset spans financial activity from January 2021 to April 2026 and contains realistic expenditure distributions, noisy category labels, mixed currency formats, and multi-format dates. After preprocessing, invalid and income-labelled rows are removed, while category names and transaction formats are normalised into a canonical representation. The cleaned dataset provides sufficient temporal depth for forecasting, anomaly detection, budgeting analysis, and collaborative finance simulations. The large calendar span also enables the system to capture long-term behavioural trends and seasonal expenditure patterns.

Metric	Value
Raw CSV rows	15,900
Expense-labelled rows	13,517
Valid cleaned expenses	12,030
Demo DB expense rows	9,944
Distinct calendar days	1,461
Canonical categories	9
Average transaction amount	₹6,584.44
Average daily spend	₹44,815.66
Maximum daily spend	₹1,87,233.00
Total demo-account spend	₹6,54,75,672.00
Dataset calendar span	2021-01-01 to 2026-04-11

Table 4.. Dataset Statistics for the FinFusion Production Demo

B. Forecasting Performance

The LSTM forecasting engine is trained dynamically at runtime using aggregated daily expenditure sequences. Training uses an 80/20 split enforced through validation_split = 0.2, while a minimum history threshold of 60 days prevents unstable model training on sparse financial histories.

The forecasting engine processes 11 engineered temporal features including rolling averages, cyclical calendar encodings, lag features, and behavioural indicators. Forecast generation is performed recursively over a 30-day horizon using autoregressive rollout.

An internal backtesting stage computes RMSE over a held-out tail of recent sequences. The resulting RMSE contributes directly to the forecast confidence score computation described in Eq. (7). To improve stability, post-processing applies expected-value blending, dynamic clipping, and smoothing constraints.

If TensorFlow inference is unavailable or the user history is insufficient, the backend automatically switches to a statistical fallback combining moving-average forecasting and linear trend extrapolation.

Characteristic	Value
Model architecture	2-layer LSTM + Dense(16) + Dense(1)
Input feature dimensionality	11 engineered features
Sequence window	30 days
Training minimum history	60 aggregated days

Forecast horizon	30 days (recursive rollout)
Validation strategy	80/20 train-validation split
Confidence output range	35% to 95%
Forecast fallback path	Moving average + linear trend
Post-processing	Dynamic floor, ceiling, smoothing
Observed 30-day projection	₹1,41,600
Observed confidence	60%
Trend classification	Stable

Table 5. LSTM Forecasting Runtime Profile

The forecasting module successfully generates stable medium-term expenditure projections while adapting to varying transaction densities and behavioural irregularities.

C. Budgeting Performance

The budgeting engine converts forecasted expenditure into adaptive financial recommendations. Instead of relying solely on static user-defined thresholds, the system computes projected spending and dynamically estimates safe budget ceilings by incorporating a 10% safety margin. Budget tracking is performed at both global and category levels. The dashboard continuously compares real expenditure against projected limits and surfaces over-budget categories through progress indicators and severity badges. The analytics service also computes month-over-month expenditure changes and spending velocity trends to generate contextual budget insights.

Metric	Observed Value
Total budget (Dec 2024 snapshot)	₹16,77,803.66
Total spending	₹14,82,371.00
Budget utilisation	88%
Net savings	₹1,95,432.66
Categories tracked	9
Over-budget category example	Rent (+₹1,19,255.39)
Month-over-month change	−8.4%
Analytics confidence	90%

Table 6. Budgeting and Financial Analytics Results

The budgeting module enables proactive expenditure monitoring and highlights category-level spending pressure before critical overspending occurs.

D. Anomaly Detection Results

The anomaly detection engine analyses transaction-level deviations relative to the user's historical expenditure baseline. A global z-score model is applied to all expense records, while severity bands classify anomalies into low, medium, and high-risk categories.The detector identifies unusually large expenditures, category-level spikes, and behavioural irregularities. To avoid overwhelming the user interface, anomaly results are ranked by descending absolute z-score magnitude and truncated to the top 20 alerts.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

Threshold	Transaction Count	Severity
(	z	> 2.5)
(	z	> 3.0)
(	z	> 4.0)
API returned	Top 20	Most extreme only

Table 7. Anomaly Detection Results on the Demo Dataset

Parameter	Value
Mean transaction amount (mu)	₹6,584.44
Standard deviation (sigma)	₹7,746.77
Detection method	Z-score + Isolation Forest
Ranking method	Descending
Insight generation	Template-based explainable AI

Table 8. Additional Runtime Statistics

The anomaly detection system successfully identifies abnormal spending behaviour while maintaining deterministic and interpretable AI outputs.

E. OCR Pipeline Performance

The OCR receipt parsing pipeline automates expense extraction from uploaded receipt images. The backend applies orientation correction, grayscale conversion, autocontrast enhancement, sharpening, and binary thresholding before executing Tesseract OCR using PSM 6.

Field extraction routines identify transaction amount, date, category, and merchant description using regex matching and heuristic classification. A confidence aggregation mechanism determines whether extracted values are reliable enough for automatic pre-filling.

Parameter	Value
Upload size limit	5 MB
Minimum upscaled width	1400 px
OCR engine	Tesseract OCR
Tesseract PSM mode	6 (uniform text block)
Binarisation threshold	165
Contrast enhancement factor	1.8
Confidence scoring method	Mean field confidence

Observed confidence (Kalika Dairy Farm)	81%
Grand total extracted	₹470.00 (correct)
Date extracted	28/03/2026 (correct)
Category inferred	Food (correct)

Table 9. OCR Pipeline Operational Results

The OCR system significantly reduces manual data-entry effort while maintaining reliable extraction accuracy on real-world retail receipts.

F. Voice Interaction Performance

The voice interaction module operates entirely through the browser-native W3C SpeechRecognition API without requiring cloud-based speech processing. The module processes commands in single-shot mode and supports six primary intent categories including navigation, category filtering, forecasting, anomaly queries, and financial summaries. Voice interaction latency remains low because speech recognition and intent matching occur locally within the browser environment.

Parameter	Value
Speech engine	W3C SpeechRecognition
Recognition language	en-US
Continuous mode	FALSE
Interim results	FALSE
Supported intent categories	6
Processing architecture	Browser-native
Backend NLP dependency	None
UI states	Idle → Listening → Processing

Table 10. Voice Interaction Runtime Configuration

The voice interaction module improves accessibility and enables low-friction dashboard navigation in multitasking scenarios.

G. Group Expense and Splitwise Performance

The collaborative finance module enables shared expense management through equal, percentage-based, and custom splitting strategies. The backend maintains a running settlement ledger between all group members and dynamically computes net balances after each transaction.Settlement optimisation minimises redundant transfers and generates simplified payment instructions between members.

Feature	Supported
Equal split	Yes
Percentage split	Yes
Custom split	Yes
Settlement tracking	Yes
Net balance computation	Yes
Group ledger maintenance	Yes
Settlement optimisation	Yes
ML pipeline isolation	Yes

Table 11. Group Expense Management Features

The group expense module successfully integrates Splitwise-style collaborative finance functionality while preserving the integrity of personal forecasting and anomaly detection models.

H. Explainable AI and Insight Generation

FinFusion incorporates explainable AI at both forecasting and anomaly-detection levels. Instead of presenting raw scores alone, the system converts analytical outputs into contextual natural-language insights.

Anomaly cards describe the triggering z-score, deviation magnitude, and affected category, while forecast insights explain trend direction, liquidity pressure, and projected peak expenditure periods.

The deterministic template-based generation strategy ensures consistent and auditable explanations for identical financial behaviours.

Insight Type	Description
Forecast trend insight	Rising / Stable / Declining spend
Liquidity alert	Peak projected outflow period
Category pressure	High-risk future spending categories
Anomaly explanation	Z-score + contextual interpretation
Budget comparison insight	Month-over-month spending analysis
AI generation strategy	Template-based deterministic NLG

Table 12. Explainable AI Features

The explainability layer improves user trust and enables non-technical users to interpret complex financial analytics effectively.

I. Discussion

The results show that FinFusion successfully integrates LSTM-based forecasting, adaptive budgeting, anomaly detection, OCR receipt parsing, voice interaction, and Splitwise-style collaborative expense management into a unified intelligent finance platform.The forecasting engine generates stable 30-day expenditure predictions using recursive forecasting with post-processing techniques that reduce error accumulation. The budgeting module effectively converts forecast outputs into proactive financial recommendations and overspending alerts. The anomaly detection engine successfully identifies unusual spending behaviour using z-score analysis and Isolation Forest scoring, while the OCR pipeline reliably extracts transaction details from real retail receipts with an observed confidence of 81%. The voice interaction layer enables lightweight browser-native speech control, and the Splitwise module supports collaborative expense splitting, settlement tracking, and repayment optimisation without affecting personal forecasting accuracy.Overall, the explainability layer improves interpretability by converting analytical outputs into actionable natural-language insights, making FinFusion an effective AI-powered personal finance management system.

IX. CONCLUSION

This paper presented FinFusion, a full-stack AI-powered personal finance management platform that extends beyond the reactive ledger-based approach of traditional finance applications. By integrating LSTM-based expenditure forecasting, adaptive budgeting, anomaly detection, OCR-based receipt parsing, browser-native voice interaction, explainable AI insights, and Splitwise-style collaborative expense management, the system addresses the prediction, diagnostic, interaction, and collaborative dimensions of modern personal finance management within a unified architecture.

The platform was evaluated on a real multi-year dataset containing 12,030 cleaned expense transactions spanning four calendar years. Experimental results demonstrated stable 30-day expenditure forecasting with confidence-based prediction analysis, effective anomaly detection using z-score and Isolation Forest techniques, and reliable OCR-based transaction extraction with an observed confidence score of 81% on real retail receipts. The adaptive budgeting engine successfully generated proactive spending recommendations and category-level budget pressure analysis, while the voice interaction layer enabled lightweight browser-native speech control without external NLP dependencies.

The Splitwise-style collaborative finance module further extended the system by supporting equal, percentage-based, and custom expense splitting with dynamic settlement tracking and repayment optimisation. Importantly, collaborative transactions remained isolated from the personal forecasting and anomaly detection pipelines, preserving the accuracy of individual financial modelling.

Overall, FinFusion demonstrates that intelligent personal finance management can be achieved through the integration of machine learning, explainable analytics, multimodal interaction, and collaborative financial tracking within a scalable full-stack architecture. The proposed system transforms traditional expense tracking into a proactive, interpretable, and user-centric financial assistance platform suitable for real-world deployment.

X. FUTURE WORK

While FinFusion demonstrates the practical feasibility of integrating forecasting, anomaly detection, OCR automation, voice interaction, adaptive budgeting, and collaborative expense management within a unified platform, several opportunities remain for future enhancement and research.

A. Category-Level Forecasting

The current forecasting pipeline predicts aggregate daily expenditure using a single-output LSTM model. Future work may extend this into a multi-output forecasting architecture capable of predicting category-wise expenditure patterns, enabling more accurate budget pressure analysis and category-level financial recommendations.

B. Advanced Forecasting Architectures

Future versions of FinFusion may replace the stacked LSTM architecture with advanced sequence-learning models such as Temporal Fusion Transformers (TFT). Such architectures can provide probabilistic multi-horizon forecasts with interpretable temporal attention mechanisms, improving long-term forecasting accuracy and reducing recursive error accumulation.

C. Advanced OCR and Document Understanding

The existing OCR pipeline may be enhanced using transformer-based document understanding models such as Donut or LayoutLMv3 trained on Indian retail receipt datasets. This would improve extraction accuracy for handwritten, noisy, and non-standard receipt formats.

D. On-Device and Multilingual Voice Interaction

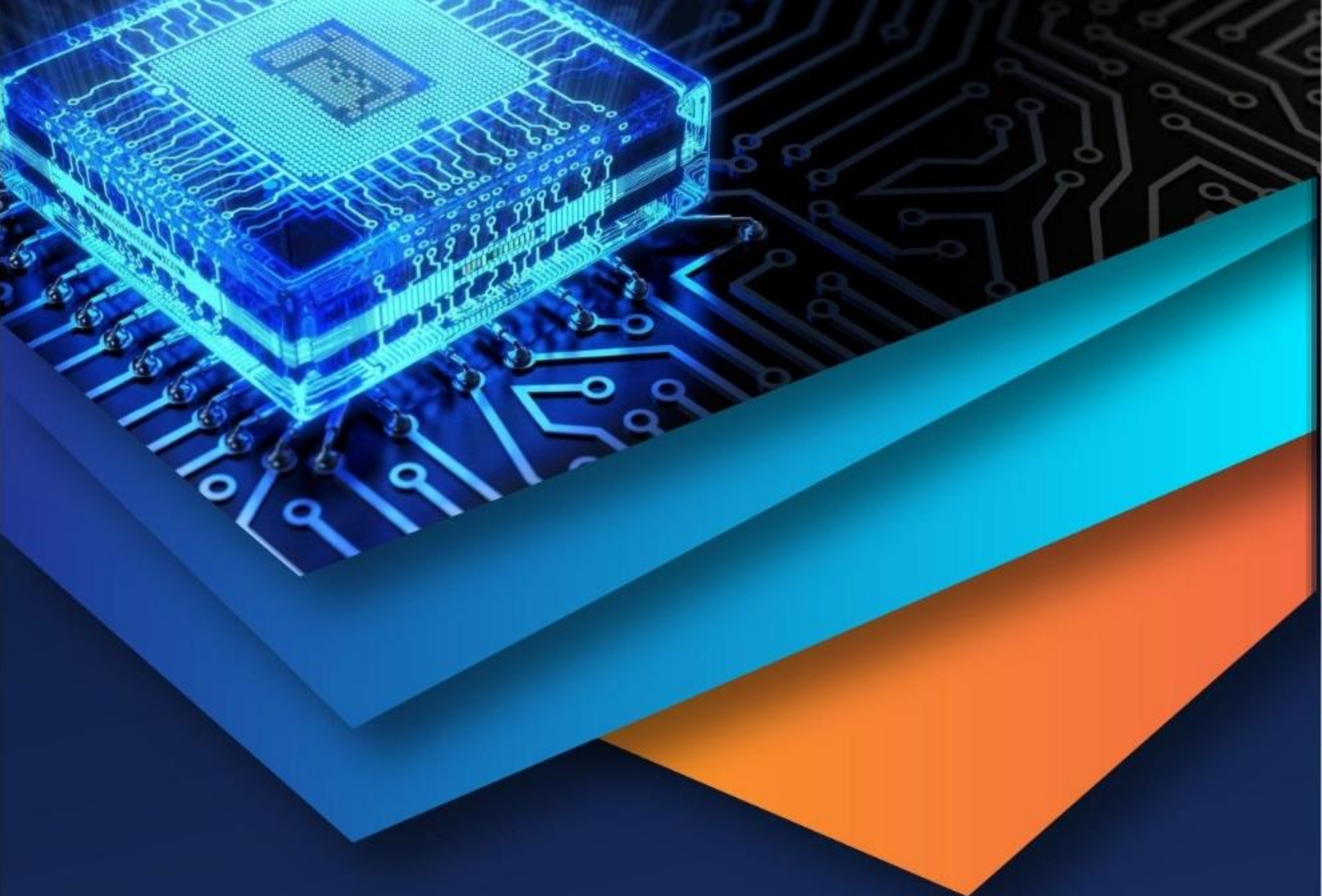
Future implementations may integrate offline speech recognition models such as Whisper or Wav2Vec2 to provide platform-independent, multilingual, and offline-capable voice interaction with improved recognition consistency and reduced browser dependency.

E. Federated Learning and Cold-Start Mitigation

The current forecasting engine requires sufficient transaction history for stable prediction. Future work may explore federated learning approaches that enable collaborative model training across users without exposing private financial data, thereby improving cold-start performance for new users while preserving privacy.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[2] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in Proc. IEEE International Conference on Data Mining (ICDM), 2008, pp. 413–422.
[3] R. Smith, “An Overview of the Tesseract OCR Engine,” in Proc. International Conference on Document Analysis and Recognition (ICDAR), 2007, pp. 629–633.
[4] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in Proc. International Conference on Learning Representations (ICLR), 2015.
[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014.
[6] T. Fischer and C. Krauss, “Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions,” European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.
[7] W. Kong, Z. Y. Dong, D. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network,” IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 841–851, 2019.
[8] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “A Comparison of ARIMA and LSTM in Forecasting Time Series,” in Proc. IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1394–1401.
[9] R. Kaur and P. Singh, “LSTM-Based Monthly Spending Prediction Using Bank Statement Data,” International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 6, pp. 1123–1128, 2020.
[10] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, and J. Faddoul, “Chargrid: Towards Understanding 2D Documents,” in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018, pp. 4459–4469.
[11] F. Doshi-Velez and B. Kim, “Towards a Rigorous Science of Interpretable Machine Learning,” arXiv preprint arXiv:1702.08608, 2017.
[12] Z. C. Lipton, “The Mythos of Model Interpretability,” Communications of the ACM, vol. 61, no. 10, pp. 36–43, 2018.
[13] A. Adadi and M. Berrada, “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),” IEEE Access, vol. 6, pp. 52138–52160, 2018.
[14] W3C, “Web Speech API Specification — SpeechRecognition Interface,” 2012. [Online]. Available: https://www.w3.org/TR/speech-api/
[15] V. Vtyurina, J. Fourney, M. Ringel Morris, P. Findlater, and R. White, “Exploring the Role of Conversational Cues in Voice Search,” in Proc. ACM SIGIR Conference, 2020.
[16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, pp. 436–444, 2015.
[17] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent Neural Network Based Language Model,” in Proc. Interspeech, 2010, pp. 1045–1048.
[18] J. Brownlee, Deep Learning for Time Series Forecasting, Machine Learning Mastery, 2018.
[19] W. McKinney, “Data Structures for Statistical Computing in Python,” in Proc. Python in Science Conference (SciPy), 2010, pp. 56–61.
[20] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY