



## **AI UPI BUDGET ANALYZER**

Ms. J. Aiswarya, M.Sc., Assistant Professor,

Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

Anushri T, Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

### **ABSTRACT**

With the rapid expansion of digital transactions, especially through the Unified Payments Interface (UPI), the need for automated tracking and intelligent financial monitoring has become more relevant than ever. Modern users engage in frequent digital purchases—ranging from daily groceries to transportation—leading to an overload of unorganized transactional data in banking applications. Although banks provide statements and notifications, users often lack structured insights about their spending patterns.

This paper introduces the AI UPI Budget Analyzer, an intelligent web-based system designed to automate the categorization of UPI transactions, monitor personal budgets, and generate meaningful financial insights in real time. The system integrates QR code scanning for instant transaction capture, uses a rule-based keyword matching algorithm for accurate expense classification, and presents insights through interactive dashboards and visual analytics. Users receive real-time alerts during overspending, along with a calculated financial health score and personalized spending trends.

The system is implemented using React.js for the frontend, Node.js for backend logic, and SQLite for lightweight data storage. The overall architecture ensures scalability, modularity, and efficiency. Experimental evaluation shows that the integrated AI engine achieves high accuracy in categorizing transactions with minimal latency. The findings demonstrate that the



AI UPI Budget Analyzer can significantly improve personal expenditure management, reduce manual workloads, and promote healthier financial practices.

**KEYWORDS**— AI categorization, UPI transactions, budget monitoring, financial analytics, expense tracker, rule-based algorithm.

## I. INTRODUCTION

Over the last decade, digital payments in India have experienced massive transformation due to the introduction of the Unified Payments Interface (UPI). It has become one of the fastest-growing payment ecosystems globally, widely adopted across retail, transportation, food services, and peer-to-peer transfers. UPI's simplicity, reliability, instant processing, and integration with mobile devices have enabled users to perform transactions multiple times a day.

However, while UPI apps such as Google Pay, PhonePe, and Paytm allow users to view their transaction history, they lack automated categorization, budget tracking, and personalized financial insights. Everything from food bills to emergency expenditures appears as a plain transaction list, forcing users to manually review each entry while analyzing their spending habits.

This gap in digital finance management highlights the need for an intelligent system that can not only capture transactions but also process and interpret them. The AI UPI Budget Analyzer aims to solve this by

combining digital payment data with AI-driven categorization and budget analysis mechanisms. It allows users to understand their financial behavior, set budgets, and receive real-time alerts, transforming raw transactional data into meaningful financial knowledge.

## II. SYSTEM ARCHITECTURE AND DESIGN

The architecture of the AI UPI Budget Analyzer is built using a three-tier model consisting of:

1. Presentation Layer (Frontend)
2. Application Layer (Backend)
3. Data Layer (Database)

### A. Presentation Layer – React.js

The user interface is developed using React.js, known for its component-based architecture and real-time rendering capabilities. Key responsibilities of this layer include:

- Displaying dashboards and analytics



- Capturing UPI transactions using QR scanning
- Allowing users to manually enter expenses
- Providing budget setting functionalities
- Showing visual charts and financial summaries

The UI ensures clarity, user-friendliness, and responsiveness across devices.

#### B. Application Layer – Node.js & Express.js

The backend application handles:

- API creation and routing
- Categorization logic execution
- Budget monitoring operations
- Communication with the SQLite database
- Integration with payment verification systems like PayU

This layer forms the core processing engine and ensures secure data handling.

#### C. Data Layer – SQLite Database

SQLite is chosen due to:

- Low resource usage
- Easy integration with Node.js
- Efficient read/write operations

- Suitability for local storage requirements

The database stores:

- Raw transactions
- Categorized results
- Budget configurations
- Health score indicators
- Historical spending logs

The architecture ensures scalability, modularity, and enhanced system maintainability.

### **III. AI-BASED CATEGORIZATION METHODOLOGY**

At the heart of the AI UPI Budget Analyzer is the Rule-Based Keyword Matching Algorithm, designed for fast and accurate classification of UPI transactions.

#### A. Preprocessing of Transaction Data

Before classification, the system standardizes:

- Merchant names
- UPI transaction descriptions
- Payment reference strings

This involves:

- Removing special characters



- Converting text to lowercase
- Tokenizing words

For example:  
“Zomato\*UPI-Payment” → “zomato upi payment”

#### B. Keyword Matching Mechanism

The system checks processed words against predefined keyword lists such as:

- Food (zomato, swiggy, restaurant, cafe)
- Shopping (mynta, amazon, lifestyle)
- Travel (ola, uber, fuel station names)
- Bills (electricity, recharge, dth)
- Groceries (dmart, bigbasket, reliance fresh)

Each category includes multiple keywords to handle variations in merchant names.

#### C. Necessity Level Determination

Categories are labeled as:

- Essential (Groceries, Healthcare, Bills)
- Optional (Entertainment, Shopping)
- Luxury (Premium services, subscriptions)

This helps in calculating the financial health score.

#### D. Confidence Score Calculation

The system counts the number of keyword matches to generate a confidence percentage.

More matches = higher confidence.

#### E. Advantages of the Rule-Based Approach

- Real-time processing
- High interpretability
- No training data needed
- Easy to update categories

This lightweight approach ensures speed and accuracy for everyday financial tracking.

## **IV. DATA PROCESSING AND BUDGET ANALYSIS FRAMEWORK**

The system does more than categorize—it analyzes, aggregates, and interprets transactional data.

#### A. Transaction Aggregation

The system organizes data based on:

- Daily, weekly, and monthly spending
- Category-wise distribution



- Trend detection based on historical cycles

## B. Budget Monitoring

Users can set budgets for:

- Total monthly spending
- Category-wise spending (e.g., Food budget: Rs. 2000)

The system continuously checks:

- Remaining amount
- Utilized percentage
- Overspending occurrences

## C. Real-Time Alerts

The alert engine notifies users when:

- Spending exceeds 75% of a category budget
- Transaction pushes them into overspending
- Monthly limit is crossed

Alerts help users adjust spending behaviors in real-time.

## D. Financial Health Score Generation

The health score is calculated from:

- Spending vs. earnings pattern
- Essential vs. non-essential spending ratio
- Budget adherence level

- Frequency of overspending alerts

This score gives users a quick assessment of their financial discipline.

## V. IMPLEMENTATION DETAILS

Implementation involves seamless integration across frontend, backend, and database layers.

### A. Backend Initialization

Node.js server setup includes:

- API endpoints for CRUD operations
- Categorization engine integration
- Middleware for validation
- PayU verification API hooks

### B. Database Structure

SQLite tables include:

- transactions
- categories
- budgets
- health\_metrics

Each with constraints ensuring data integrity.

### C. Frontend Interface Development

React components handle:

- Transaction forms



- QR code capture
- Visual chart rendering
- Budget alerts

UI libraries like Recharts/Chart.js ensure dynamic visuals.

#### D. QR Code Extraction

The scanner reads:

- Amount
- Merchant name
- UPI ID
- Timestamp

This reduces manual errors and speeds up processing.

## VI. SYSTEM TESTING AND VALIDATION

Comprehensive testing ensures stability and performance.

#### A. Unit Testing

Tested modules include:

- QR scanner
- Keyword engine
- Database APIs
- Budget alerts

#### B. Integration Testing

Ensured smooth data flow between:

- React → Node
- Node → SQLite
- SQLite → Dashboard

#### C. System Testing

Simulated real scenarios with:

- High transaction frequency
- Mixed category transactions
- Ambiguous merchant names

#### D. Performance Testing

Measured:

- Response time (<200ms)
- Categorization speed
- Database read/write efficiency

The system performed efficiently even on low-power devices.

## VII. RESULTS AND DISCUSSION

Results show the system's robustness:

#### A. Categorization Accuracy

High accuracy achieved when keywords match merchant descriptions.

#### B. Budget Alerts



Users were instantly notified during overspending events.

### C. Dashboard Insights

Graphs such as:

- Pie charts displaying spending distribution
- Bar charts showing monthly comparison
- Line graphs for weekly trend

help enhance understanding.

### D. Financial Health Score

Users found the score easy to understand and useful for decision-making.

### E. Limitations

Challenges include:

- Unknown merchant names
- Abbreviated or inconsistent UPI descriptions

However, these can be improved in future versions using ML models.

## VIII. CONCLUSION

The AI UPI Budget Analyzer offers an intelligent and efficient solution for personal finance management. By automating UPI transaction categorization, integrating QR scanning, and providing

real-time financial analytics, it bridges the gap between digital payments and financial awareness.

The combination of a rule-based AI engine, secure payment verification, lightweight architecture, and visual dashboards makes the system highly practical for everyday users. Future enhancements may include:

- Machine learning-based classification
- Predictive financial behavior models
- Cloud-based multi-user support
- Integration with banking APIs
- Voice-based financial assistants

The proposed system demonstrates how AI-driven approaches can significantly improve digital financial management and

## REFERENCES

1. National Payments Corporation of India (NPCI). (2022). *Unified Payments Interface (UPI) – Product Overview and Statistics*. Retrieved from <https://www.npci.org.in>
2. Reserve Bank of India. (2023). *Digital Payments: Trends, Statistics, and Analysis*. RBI Bulletin. Retrieved from <https://www.rbi.org.in>



3. Gupta, S., & Sharma, R. (2021). *Adoption of Digital Payments in India: A Study on UPI Usage Trends*. Journal of Financial Innovation, 14(2), 45–57.
4. Kumar, A., & Raj, P. (2020). *AI-Based Personal Finance Management Systems: A Review of Techniques and Applications*. International Journal of Computer Applications, 176(32), 1–9.
5. Singh, M., & Thomas, A. (2022). *Rule-Based vs Machine Learning Approaches for Text Classification in Lightweight Systems*. Journal of Intelligent Computing, 10(4), 233–241.
6. Zhang, Y., & Li, X. (2019). *Keyword Matching and Rule-Based Classification for Automated Financial Categorization*. ACM Computing Surveys, 51(6), 1–30.
7. Wagh, A., & Patil, S. (2021). *Web-Based Budget Tracking Systems Using Interactive Dashboards*. International Journal of Web Technologies, 3(1), 27–35.
8. React.js Documentation. (2023). *React: A JavaScript library for building user interfaces*. Meta Platforms. Retrieved from <https://react.dev>
9. Node.js Foundation. (2023). *Node.js Documentation: Server-Side JavaScript Runtime*. Retrieved from <https://nodejs.org>
10. Hipp, R. D. (2022). *SQLite Documentation: Lightweight Embedded Database Engine*. SQLite Consortium. Retrieved from <https://sqlite.org>
11. PayU India. (2023). *UPI Payment Gateway Integration Guide*. Retrieved from <https://www.payu.in>
12. D'Silva, F., & Srinivasan, R. (2020). *Enhancing Personal Expense Management Through Financial Analytics*. Journal of Data Science and Applications, 12(3), 71–84.
13. Jain, P., & Mehta, V. (2022). *QR Code-Based Digital Transaction Systems and Security Considerations*. International Journal of Mobile Computing, 9(2), 89–101.
14. Patel, K., & Desai, R. (2021). *Visualization Techniques in Financial Dashboards: A Practical Review*. Journal of Visualization and Data Communication, 8(1), 56–67.