BudgetWise AI based Expense Forecasting Tool

1. Introduction

In today's digital world, individuals generate large amounts of financial data through daily transactions. However, most people struggle to understand their spending patterns and forecast future expenses. BudgetWise is an AI-driven expense management and forecasting tool designed to help users make informed financial decisions.

The system allows users to track income and expenses, categorize transactions automatically, visualize spending patterns, and use machine learning models to forecast future expenses. With AI-powered insights, BudgetWise empowers users to improve financial discipline and achieve long-term savings goals.

2.Problem Statement

Most people struggle to manage their finances due to scattered transactions and limited insights from existing budgeting apps. Current systems only record expenses without automatic categorization or intelligent analysis. Users cannot predict future spending, identify patterns, or set meaningful financial goals. Poor visualizations make financial interpretation difficult, and weak security risks sensitive data. BudgetWise solves these issues by providing a secure, AI-driven platform with automated categorization, forecasting, and clear financial insights.

3. Objectives

The major objectives of the system are:

1. To provide a secure and user-friendly platform for managing expenses.
2. To automatically categorize transactions using rule-based/NLP techniques.
3. To generate intuitive financial reports for better decision-making.
4. To build a forecasting engine using Prophet for predicting future expenses.
5. To enable users to set financial goals and track their progress.
6. To offer interactive visual dashboards for detailed financial insights.

4. Existing System & Limitations

Existing System

- Manual budgeting (notebooks, spreadsheets)
- Basic expense-tracking apps
- Limited reporting and no forecasting capabilities

Limitations

- Time-consuming and prone to errors
- No AI-based predictions
- No automated categorization
- Limited data visualization
- No goal tracking
- Lack of advanced dashboards
- Weak user data security

5.Proposed System

The proposed system, BudgetWise, delivers an AI-driven financial analytics platform designed to simplify and enhance personal financial management. It provides secure user authentication through JWT-based login and registration, along with automated transaction categorization using keyword matching and NLP techniques. The system integrates Meta's Prophet model to forecast future expenses and offers interactive dashboards for visualizing spending trends, cash flow, and category-wise insights. Users can set and track financial goals, while an admin panel manages categories, rules, and system usage. The entire application is containerized using Docker, ensuring easy deployment and scalability. Together, these features create a complete end-to-end solution for intelligent financial planning.

6.Methodology

1. User Authentication & Data Entry: Users securely register/login and enter their income and expense data. This forms the primary dataset used throughout the system.

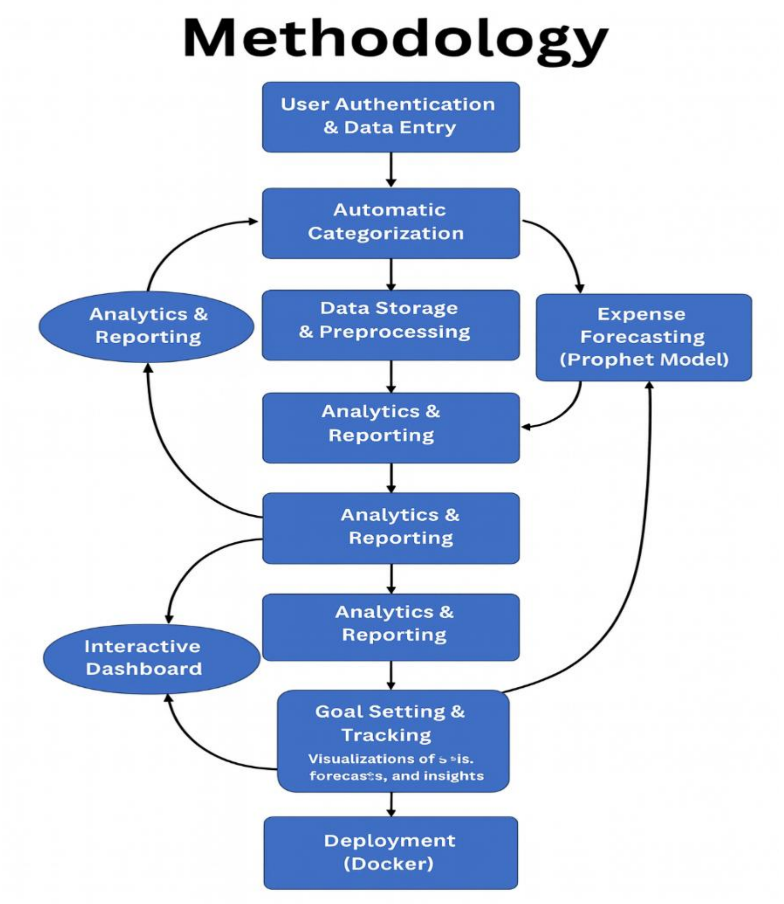
2. Automatic Categorization: Each transaction is analyzed using rule-based or NLP keyword matching to automatically assign categories (Food, Rent, Groceries, etc.), reducing manual work.

3. Data Storage & Preprocessing: All transactions are stored in a structured database. The system cleans, organizes, and converts the data into a time-series format for analytics and forecasting.

4. Analytics & Reporting: The system generates visual insights such as category-wise spending, monthly totals, and income vs. expense trends using Pandas, Plotly, and Matplotlib.

5. Expense Forecasting (Prophet Model): AI-based forecasting predicts future expenses by learning from the user's past spending patterns. This helps with budgeting and planning ahead.

6. Interactive Dashboard & Goal Tracking: All insights, forecasts, and goals are shown in an interactive dashboard created with Streamlit. Users can set savings goals, view progress, and understand their financial behavior clearly



6. System Architecture

6.1 Workflow Summary

1. **User Registration & Login**
   Secured using JWT authentication.

2. **Transaction Input**
   Users manually add transactions with date, description, amount, and type.

3. **Transaction Categorization**
   Rule-based keyword matching categorizes each transaction (Groceries, Rent, Travel, etc.).

4. **Data Storage**
   All data stored securely in a backend database.

5. **Reporting Engine**
   Generates monthly and category-wise summaries.

6. **Forecasting Engine (Prophet)**
   Creates time-series forecasts of future expenses.

7. **Visualization Layer (Streamlit/Plotly)**
   Displays charts, summaries, and goal progress.

8. **Admin Dashboard**
   Manage categories, rules, and system usage.

The system securely manages user expenses through JWT-based authentication, allowing users to input transactions that are automatically categorized using keyword rules and stored in a backend database. A reporting engine generates monthly and category-wise summaries, while the Prophet-based forecasting module predicts future spending trends. All insights are displayed through interactive Streamlit and Plotly dashboards, and an admin panel enables managing categories, rules, and overall system usage.

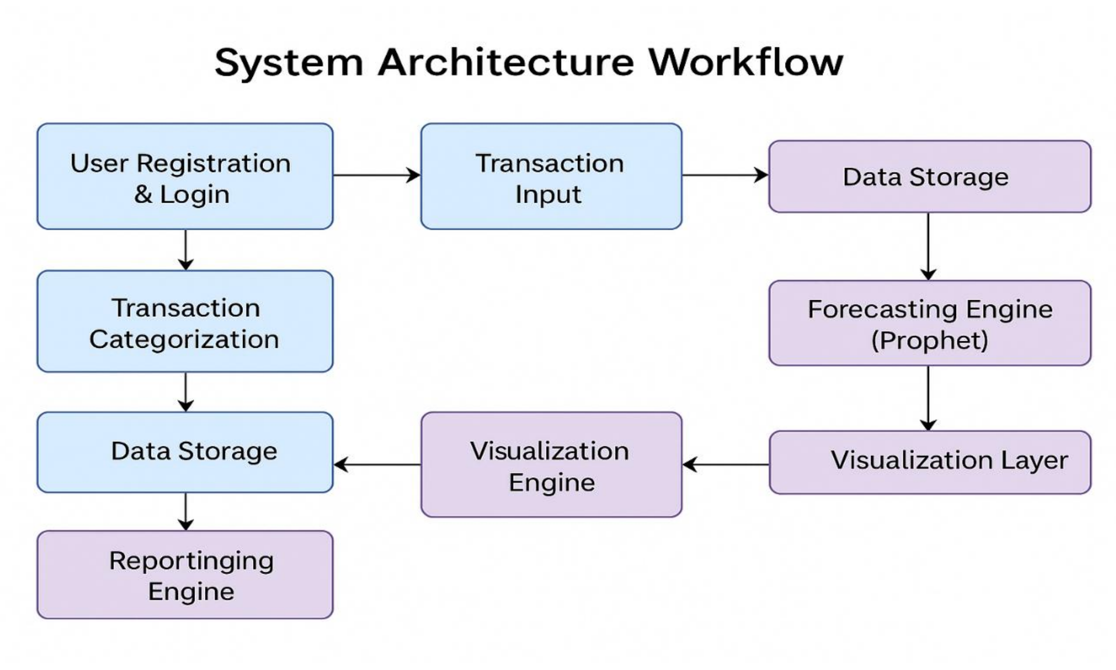
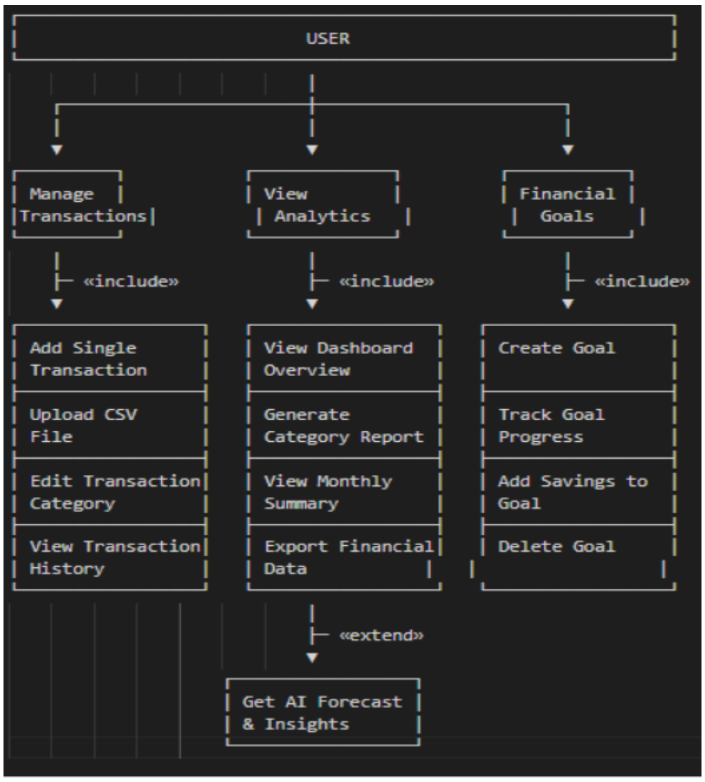

System Architecture Workflow
User Registration & Login
Transaction Input
Data Storage
Transaction Categorization
Forecasting Engine (Prophet)
Data Storage
Visualization Engine
Visualization Layer
Reportinging Engine

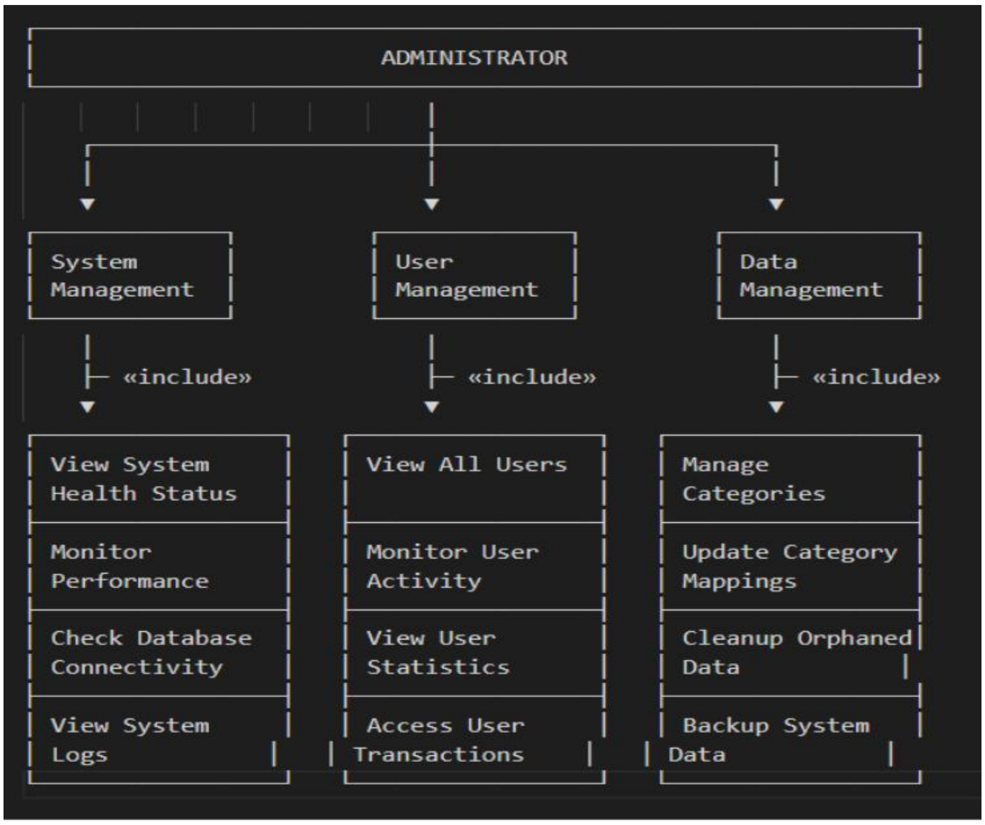
Figure 6.1: System Architecture

6.2 Use case Diagrams

6.2.1 User Use Case Diagram



6.2.2 Administrator Use Case Diagram



7. Modules Description

1. Authentication Module

- This module manages secure access to the system by handling user registration and login operations.
- It uses hashed passwords and JWT tokens to ensure only authenticated users can access protected features.
- Implements login validation, token creation, and token verification to maintain session security.
- Files like auth.py and models/user.py define user structure and authentication logic.
- Ensures privacy and protects sensitive financial records.

2. Transaction Management Module

- This is the core input layer where users add income and expense transactions.

- It handles CRUD operations—allowing users to add, update, view, or delete transactions.
- Stores transaction details such as date, amount, description, and type (income/expense).
- Routes in transactions.py and database models in models/transactions.py manage data flow.
- Serves as the foundational data source for categorization, reporting, and forecasting.

3. Categorization Engine

- Automatically assigns categories to user transactions using keyword-matching or simple NLP logic.
- Helps users avoid manual classification and reduces human error.
- Categories include Groceries, Bills, Food, Rent, Travel, etc.
- Implemented in categorizer.py, which scans transaction descriptions and applies rules.
- Supports better financial analysis by grouping similar expenses.

4. Reporting Module

- Converts raw transaction data into meaningful financial insights.
- Generates monthly summaries, category-wise spending charts, and income vs. expense comparisons.
- Uses libraries like Pandas, Matplotlib, or Plotly to create visual financial reports.
- Implemented in report_generator.py, responsible for analytical calculations.
- Enables users to easily understand their spending habits and patterns.

5. Forecasting Module (AI Model)

- Uses Meta's Prophet model to predict future expenses based on historical data trends.
- Analyzes daily, weekly, or monthly spending to estimate upcoming financial patterns.
- Helps users plan budgets and anticipate overspending before it occurs.

- Implemented in forecast.py, performing time-series forecasting.
- Introduces an intelligent, predictive element that traditional trackers lack.

6. Frontend Module (Streamlit Interface)

- Acts as the presentation layer where users interact with the system visually.
- Displays dashboards, reports, forecasting charts, and financial goal tracking.
- Provides a clean, intuitive UI built using Streamlit and Plotly.
- Fetches backend data and converts it into easy-to-understand visuals.
- All frontend logic is contained in the /frontend/*.py files.

7. Admin Panel Module

- Provides administrative control over system-wide resources and settings.
- Allows management of categories, transaction rules, and user activity monitoring.
- Ensures the application remains scalable, organized, and easy to maintain.
- Supports configuration updates without affecting end-users.
- Enhances overall system stability and consistency.

8. Tools & Technologies

1. Frontend Technologies

- Streamlit is used to build an interactive and lightweight web interface for dashboards, charts, and forecasting results.
- Plotly is used for dynamic visualizations such as line graphs, pie charts, and trend analysis.
- HTML and CSS support basic layout styling and enhance the visual presentation of the interface.

2. Backend Technology

- Python Flask serves as the backend framework, handling API routing, user authentication, data processing, and communication between the UI and database.

3. Database

- SQLite is used to securely store user data, transactions, categorized expenses, and forecasting records.
- Ensures efficient storage, retrieval, and management of financial data.

4. AI & Data Processing Technologies

- Prophet (Meta) is used to build the forecasting model that predicts future expenses using time-series analysis.
- Pandas helps in processing and cleaning financial data for analysis and model training.
- NumPy supports numerical computations and efficient data operations.

5. Visualization Libraries

- Matplotlib and Seaborn are used to generate static visual reports like bar charts and pie charts.
- Plotly provides interactive and real-time visual analytics for user dashboards.

6. Authentication Tools

- JWT (JSON Web Token) ensures secure login sessions and prevents unauthorized access.
- bcrypt is used for hashing passwords to protect user credentials.

7. NLP for Categorization

- NLTK / Keyword Matching techniques classify each transaction into categories such as groceries, rent, travel, etc., based on its description.

9.Database Design

9.1 Database Schema

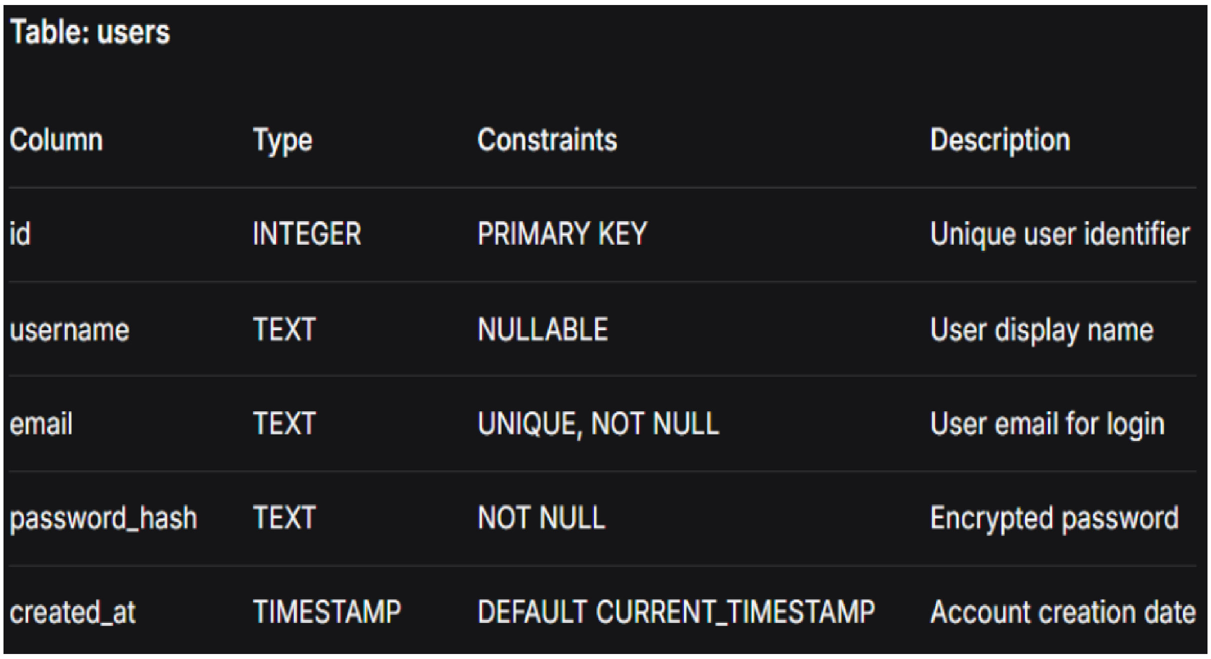


Table: users

Column	Type	Constraints	Description
id	INTEGER	PRIMARY KEY	Unique user identifier
username	TEXT	NULLABLE	User display name
email	TEXT	UNIQUE, NOT NULL	User email for login
password_hash	TEXT	NOT NULL	Encrypted password
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Account creation date

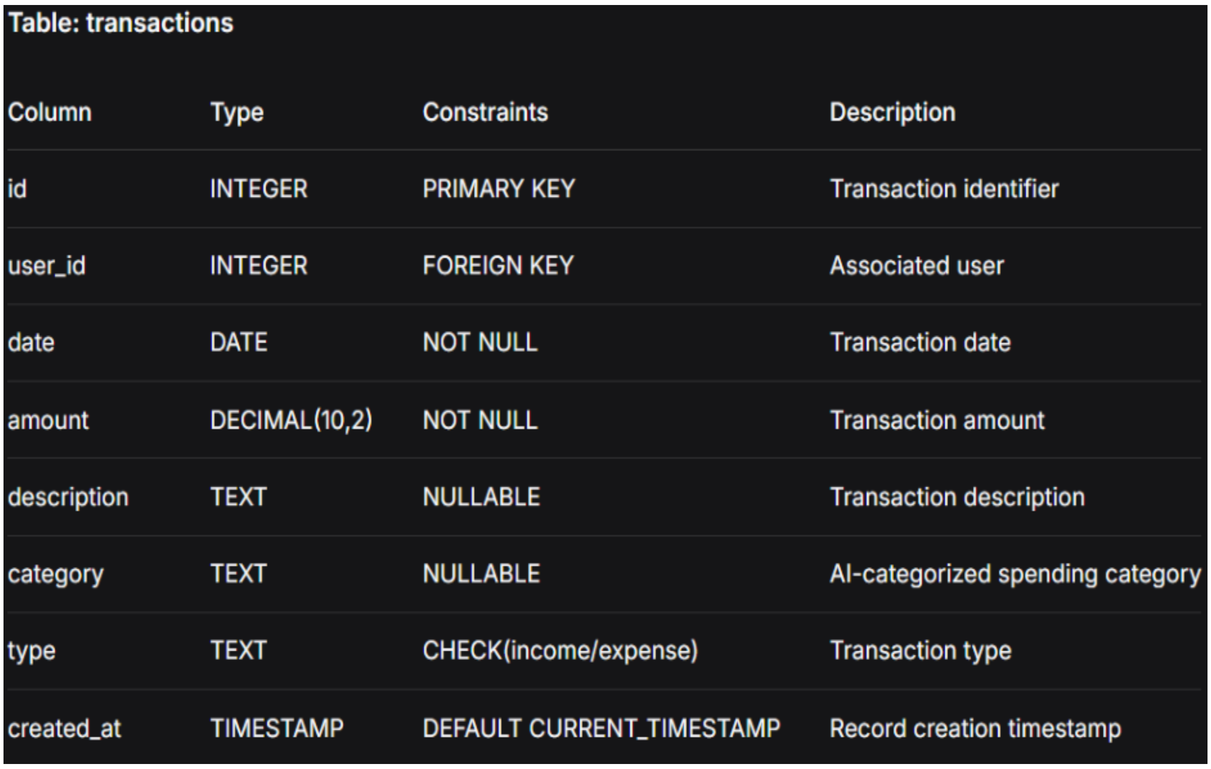


Table: transactions

Column	Type	Constraints	Description
id	INTEGER	PRIMARY KEY	Transaction identifier
user_id	INTEGER	FOREIGN KEY	Associated user
date	DATE	NOT NULL	Transaction date
amount	DECIMAL(10,2)	NOT NULL	Transaction amount
description	TEXT	NULLABLE	Transaction description
category	TEXT	NULLABLE	AI-categorized spending category
type	TEXT	CHECK(income/expense)	Transaction type
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Record creation timestamp

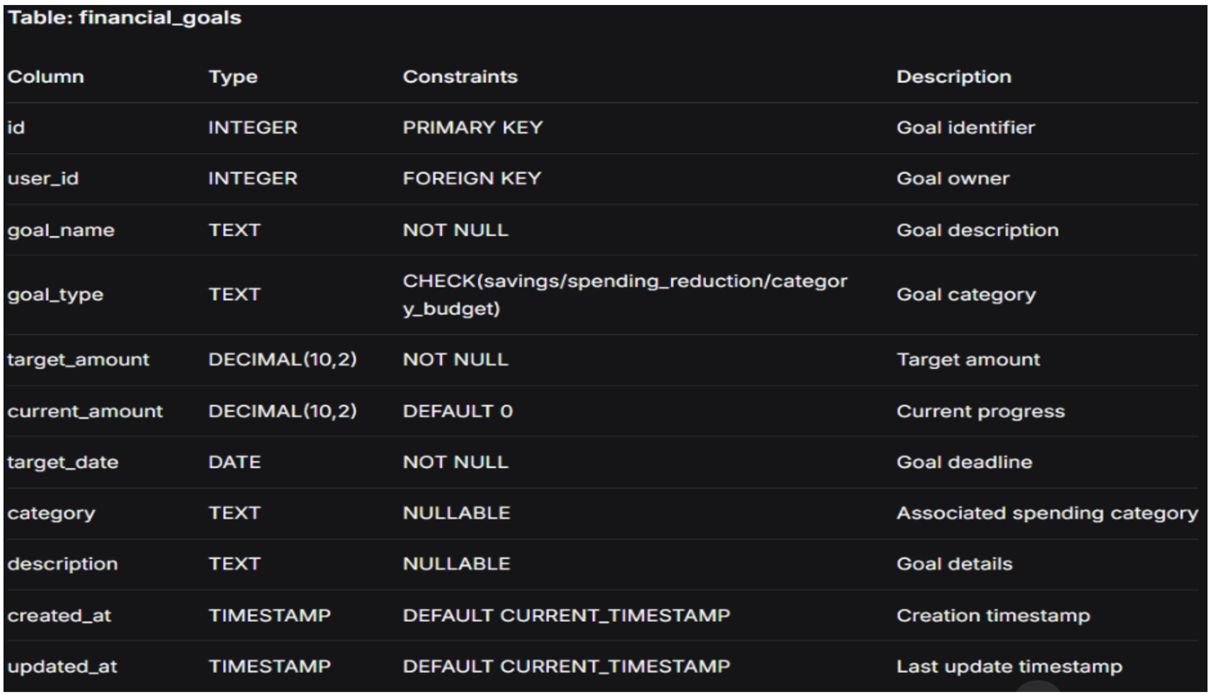


Table: financial_goals

Column	Type	Constraints	Description
id	INTEGER	PRIMARY KEY	Goal identifier
user_id	INTEGER	FOREIGN KEY	Goal owner
goal_name	TEXT	NOT NULL	Goal description
goal_type	TEXT	CHECK(savings/spending_reduction/category_budget)	Goal category
target_amount	DECIMAL(10,2)	NOT NULL	Target amount
current_amount	DECIMAL(10,2)	DEFAULT 0	Current progress
target_date	DATE	NOT NULL	Goal deadline
category	TEXT	NULLABLE	Associated spending category
description	TEXT	NULLABLE	Goal details
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Creation timestamp
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Last update timestamp

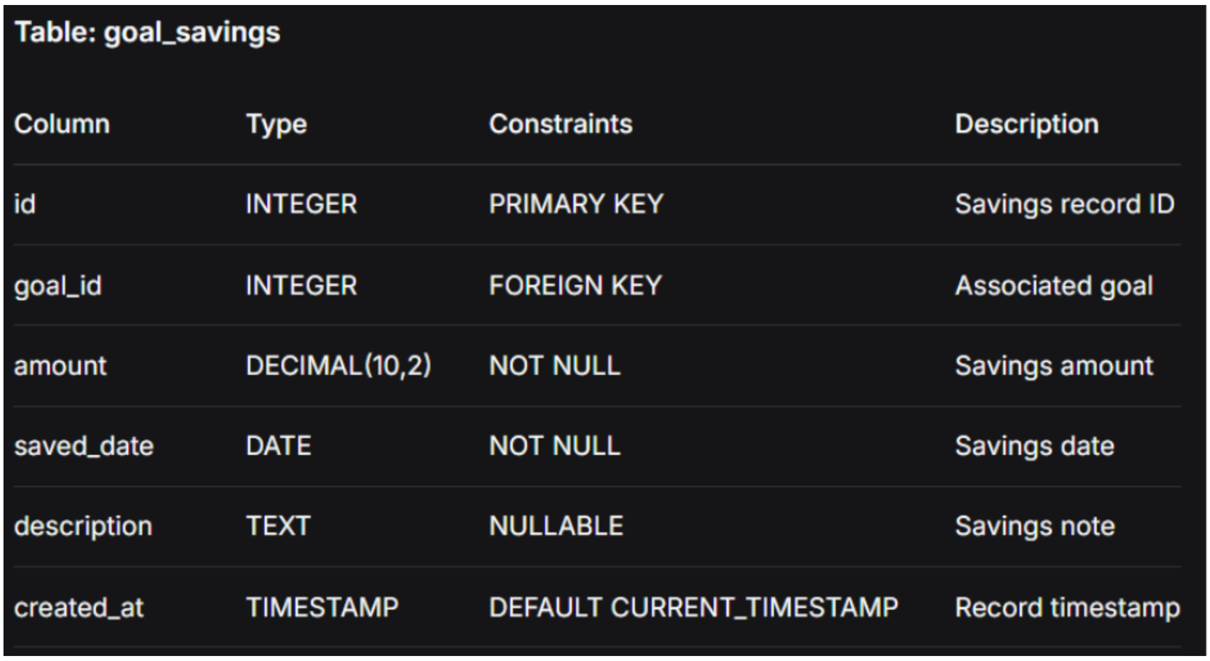


Table: goal_savings

Column	Type	Constraints	Description
id	INTEGER	PRIMARY KEY	Savings record ID
goal_id	INTEGER	FOREIGN KEY	Associated goal
amount	DECIMAL(10,2)	NOT NULL	Savings amount
saved_date	DATE	NOT NULL	Savings date
description	TEXT	NULLABLE	Savings note
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Record timestamp

9.2 Indexes and Optimization

Performance indexes

CREATE INDEX idx_transactions_user_id ON transactions(user_id); CREATE INDEX idx_transactions_date ON transactions(date); CREATE INDEX idx_financial_goals_user_id

ON financial_goals(user_id); CREATE INDEX idx_goal_savings_goal_id ON goal_savings(goal_id);

Auto-update timestamp trigger

CREATE TRIGGER update_financial_goals_timestamp AFTER UPDATE ON financial_goals FOR EACH ROW BEGIN UPDATE financial_goals SET updated_at = CURRENT_TIMESTAMP WHERE id = NEW.id; END;

10. Work Summary

Weeks 1: User Authentication & Basic Transaction Input

• User Registration: Implemented a secure user registration system with standard email/password and JWT (JSON Web Token) security.

• Login System: Developed a robust login mechanism to authenticate users.

• Profile Management: Created user profiles for managing their financial data.

• Manual Transaction Input: Designed a basic web interface allowing users to manually input individual dummy transactions (e.g., date, amount, description, type: income/expense).

Weeks 2: Transaction Categorization & Basic Reporting

• Automated Categorization: Implemented a rule-based or simple NLP (e.g., keyword matching using NLTK) system to automatically assign categories (e.g., 'Groceries', 'Rent', 'Transport') to transaction descriptions. Allow manual override.

• Spending Summary Reports: Developed functionality to generate basic reports showing total spending per category, monthly spending summaries, and income vs. expense over a period using Pandas.

• Initial Dashboard View: Enhanced the Ul to display a summary of recent transactions and a basic breakdown of spending by category (e.g., a pie chart or bar chart using Matplotlib/Seaborn).

Week 3: Forecasting Engine & Goal Setting

❖ Historical Data Preparation: Prepare historical transaction data in the format required by the forecasting model (e.g., time series of aggregated daily/weekly/monthly expenses per category).

❖ Prophet Integration: Integrate Prophet (Meta's forecasting library) to generate future expense forecasts for different categories or overall spending.

❖ Financial Goal Setting: Allow users to define financial goals (e.g., "Save $X by Y date," "Reduce spending in category Z by A%").

❖ Forecast Visualization: Update the dashboard to display the forecasted expenses alongside actual historical data (line chart using Matplotlib/Seaborn). Show projected savings based on forecasts.

Week 4: Advanced Visualization, Admin & Deployment

• Interactive Financial Dashboard: Develop a comprehensive, interactive financial dashboard (using Streamlit/Dash with Plotly/Matplotlib/Seaborn) that includes:

  ○ Detailed spending breakdown by custom periods

  ○ Income vs. Expense trends.

  ○ Forecasted cash flow.

  ○ Goal progress tracking.

• Admin Dashboard: Develop a basic admin interface:

  ○ Managing predefined transaction categories and rules.

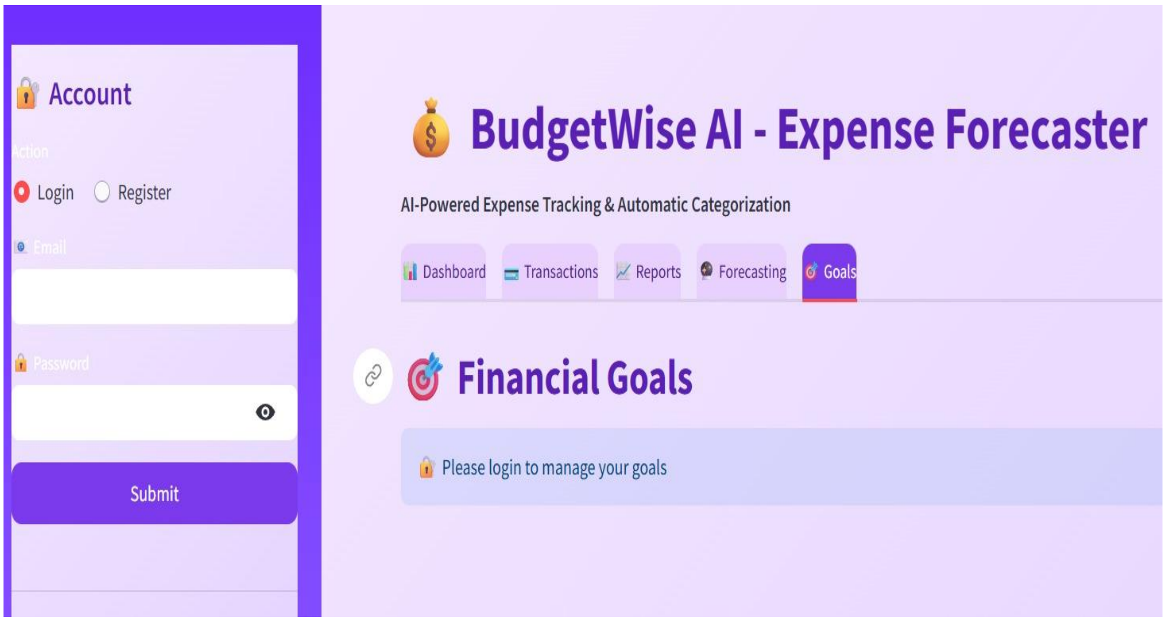
  ○ Monitoring system usage and data consistency.

• Deployment Preparation: Containerize the entire application using Docker. Prepare for deployment on free cloud platforms (e.g., Streamlit Community Cloud, Render).

• Documentation & Presentation: Finalize project documentation, code comments, and prepare for the final presentation.

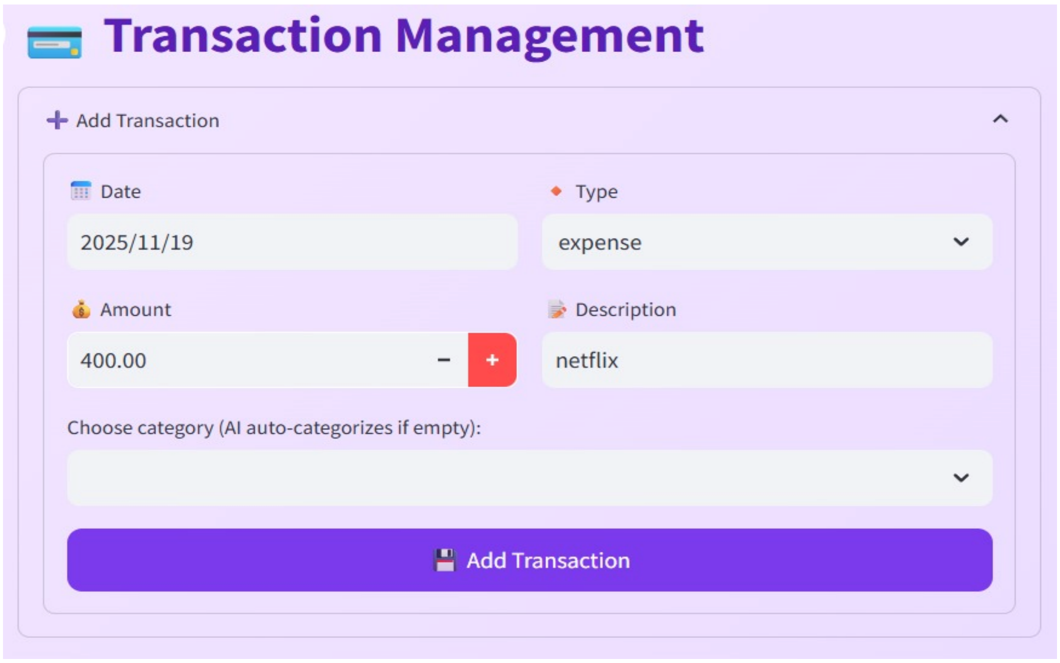
11.Snapshort

11.1 Login and Registration


Account
Action
Login
Register
Email
Password
Submit
BudgetWise AI - Expense Forecaster
AI-Powered Expense Tracking & Automatic Categorization
Dashboard
Transactions
Reports
Forecasting
Goals
Financial Goals
Please login to manage your goals

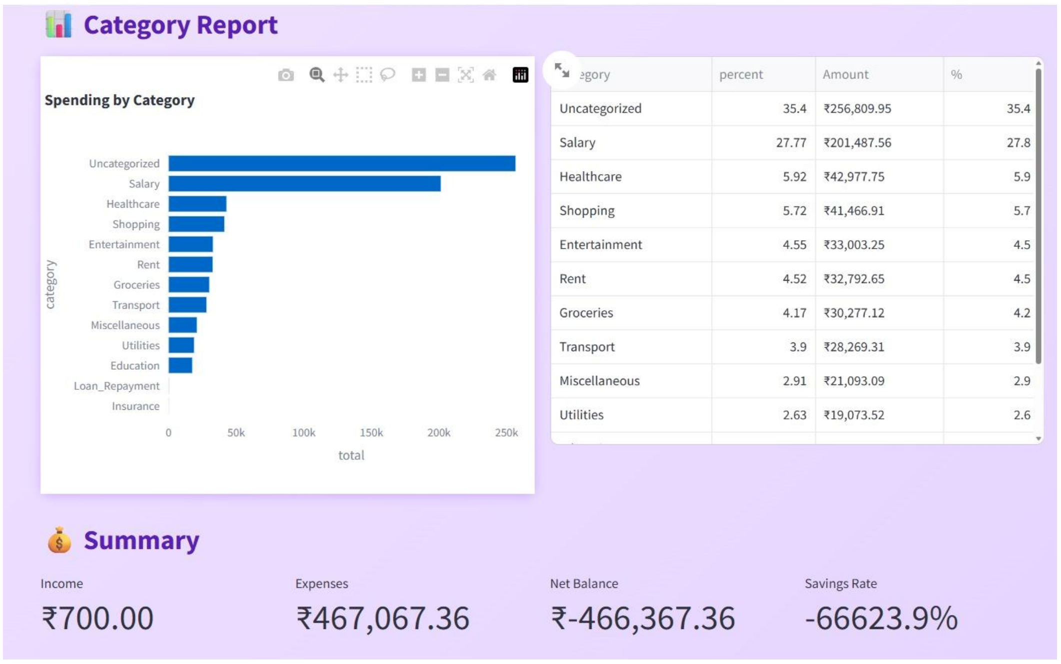
The interface displays the BudgetWise AI login and main dashboard, allowing users to securely access features like transactions, reports, forecasting, and financial goals through a clean and modern UI.

11.2 Transaction Management



Date: 2025/11/19
Type: expense
Amount: 400.00
Description: netflix
Choose category (AI auto-categorizes if empty):

11.3 Report on Category



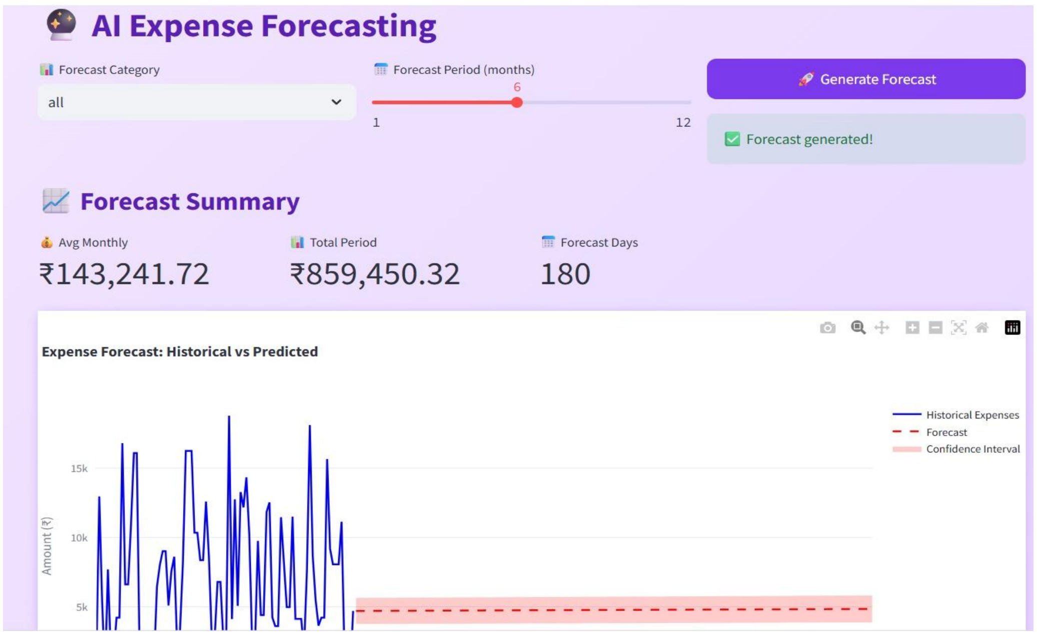
📊 **Category Report**

egory	percent	Amount	%
Uncategorized	35.4	₹256,809.95	35.4
Salary	27.77	₹201,487.56	27.8
Healthcare	5.92	₹42,977.75	5.9
Shopping	5.72	₹41,466.91	5.7
Entertainment	4.55	₹33,003.25	4.5
Rent	4.52	₹32,792.65	4.5
Groceries	4.17	₹30,277.12	4.2
Transport	3.9	₹28,269.31	3.9
Miscellaneous	2.91	₹21,093.09	2.9
Utilities	2.63	₹19,073.52	2.6

💰 **Summary**

Income	Expenses	Net Balance	Savings Rate
₹700.00	₹467,067.36	₹-466,367.36	-66623.9%

11.4 Forecast Summary



🔮 **AI Expense Forecasting**

📊 Forecast Category: all

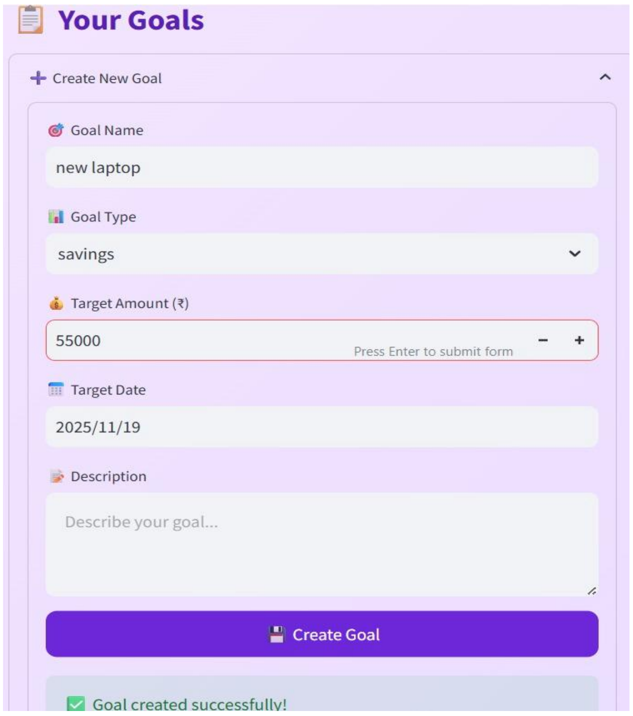
🗓️ Forecast Period (months): 6 (1 – 12)

✅ Forecast generated!

📈 **Forecast Summary**

💰 Avg Monthly	📊 Total Period	🗓️ Forecast Days
₹143,241.72	₹859,450.32	180

11.5 Goals Creation



Your Goals

Create New Goal

Goal Name: new laptop

Goal Type: savings

Target Amount (₹): 55000

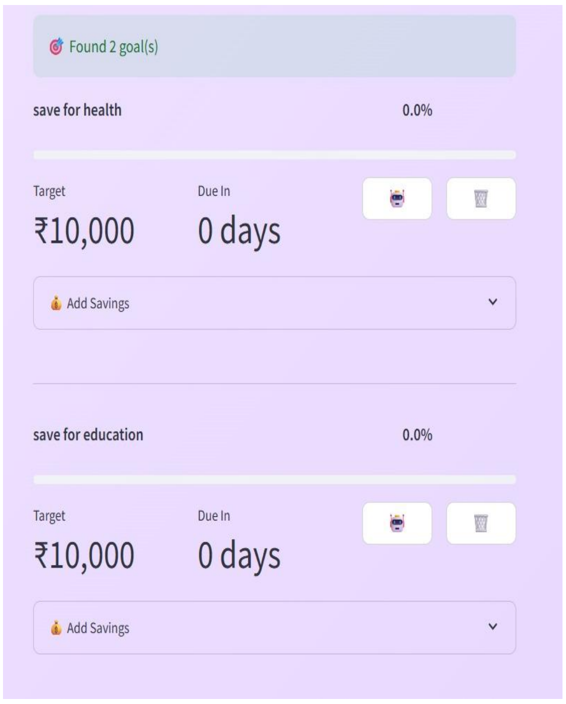
Press Enter to submit form

Target Date: 2025/11/19

Description: Describe your goal...

Create Goal

Goal created successfully!



Found 2 goal(s)

save for health 0.0%

Target ₹10,000 Due In 0 days

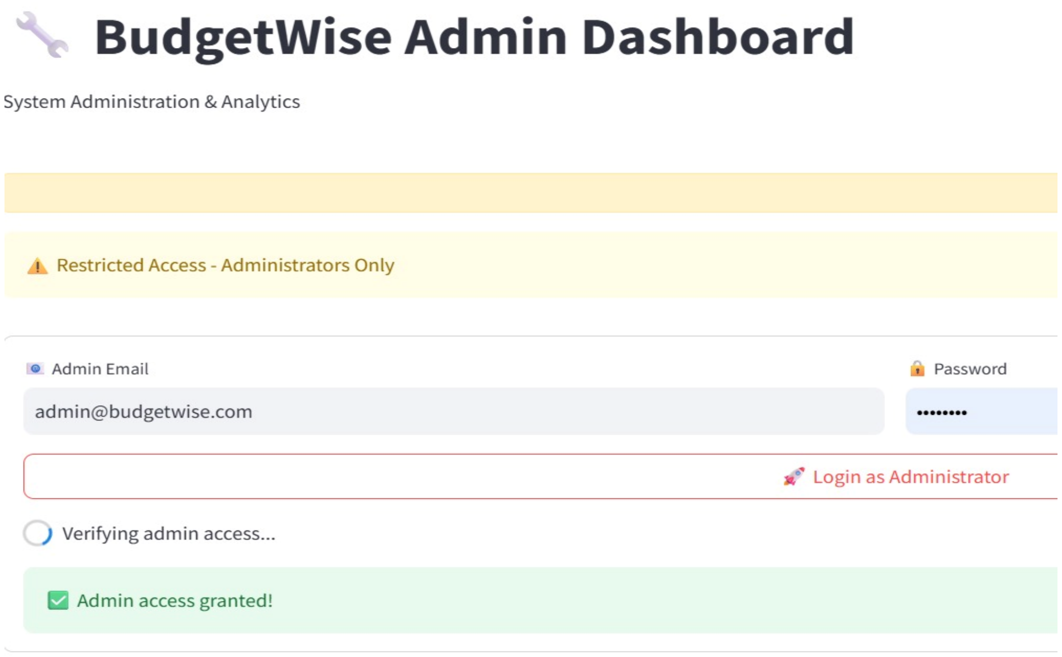
Add Savings

save for education 0.0%

Target ₹10,000 Due In 0 days

Add Savings

11.6 Admin Dashboard



BudgetWise Admin Dashboard

System Administration & Analytics

Restricted Access - Administrators Only

Admin Email: admin@budgetwise.com

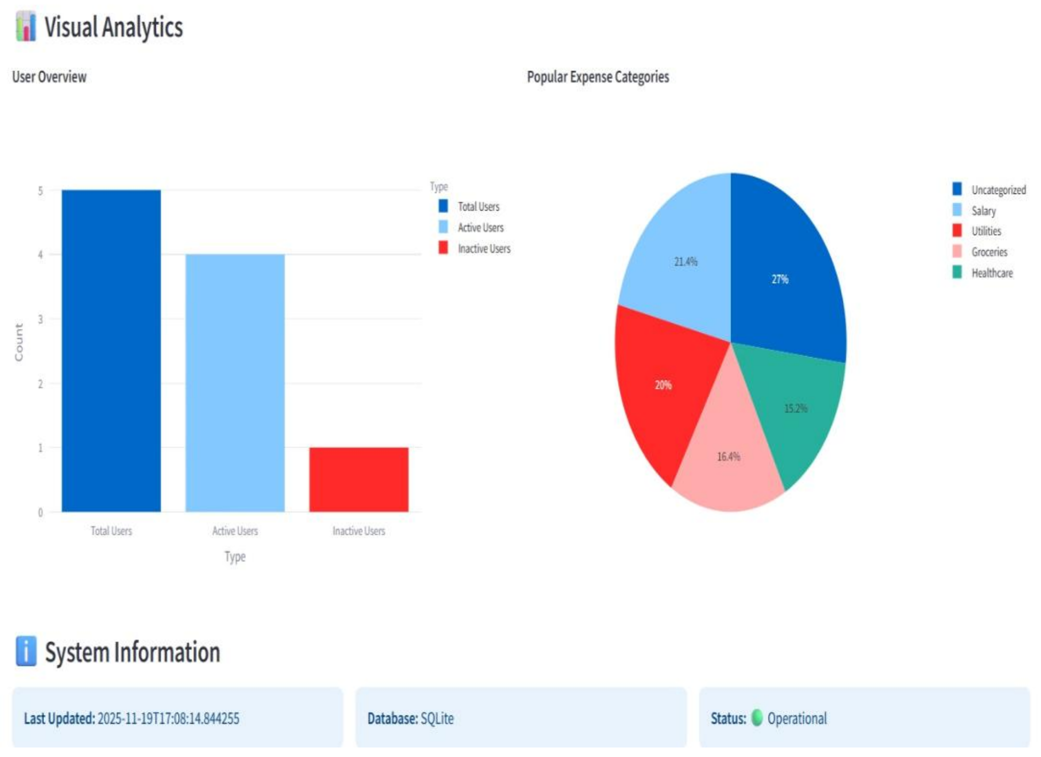
Password: ••••••••

Login as Administrator

Verifying admin access...

Admin access granted!

11.7 Visual Analytics



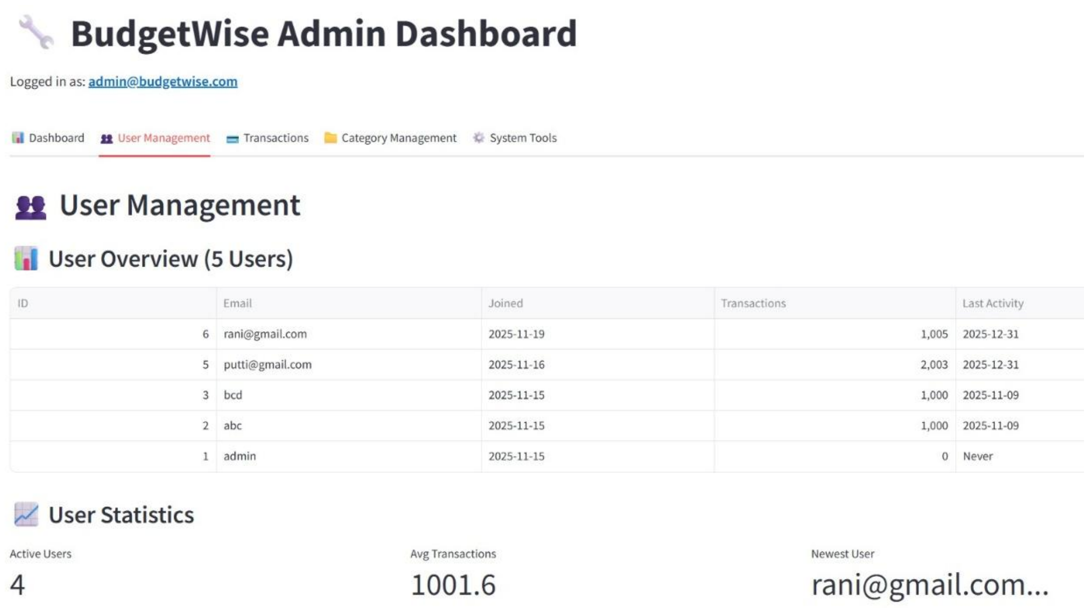
System Information

Last Updated: 2025-11-19T17:08:14.844255

Database: SQLite

Status: 🟢 Operational

11.8 User Management



BudgetWise Admin Dashboard

Logged in as: admin@budgetwise.com

User Management

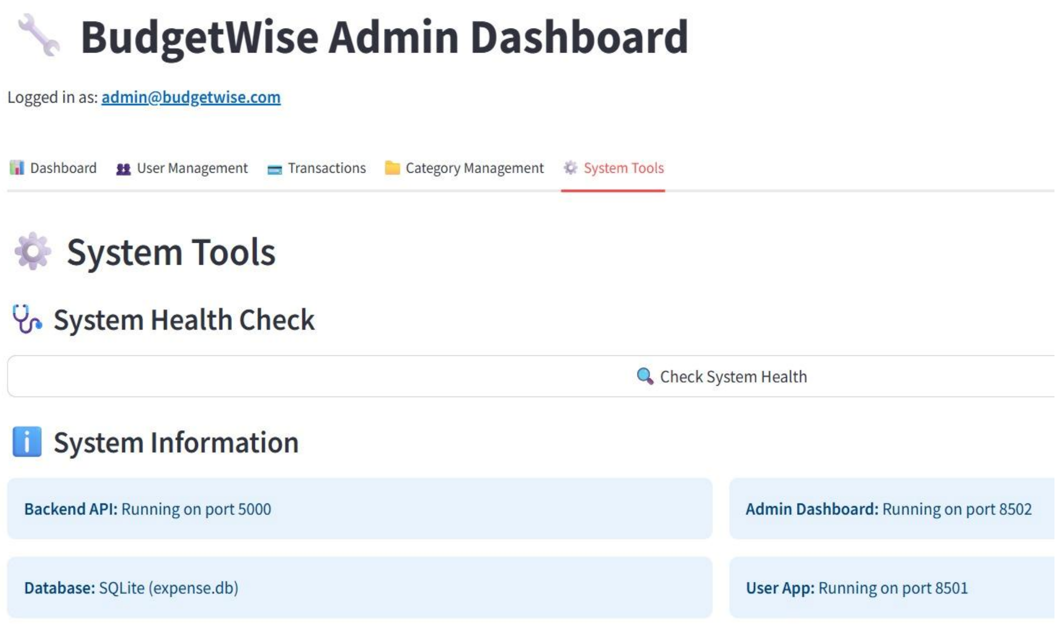
User Overview (5 Users)

ID	Email	Joined	Transactions	Last Activity
6	rani@gmail.com	2025-11-19	1,005	2025-12-31
5	putti@gmail.com	2025-11-16	2,003	2025-12-31
3	bcd	2025-11-15	1,000	2025-11-09
2	abc	2025-11-15	1,000	2025-11-09
1	admin	2025-11-15	0	Never

User Statistics

Active Users	Avg Transactions	Newest User
4	1001.6	rani@gmail.com...

11.9 System tools



Backend API: Running on port 5000

Admin Dashboard: Running on port 8502

Database: SQLite (expense.db)

User App: Running on port 8501

12.Future Enhancements

- Integration with banking APIs such as Razorpay or Plaid to automatically import transactions and minimize manual entry.
- Use of deep learning models to replace keyword-based categorization, improving accuracy and adapting to user behavior.
- Implementation of spending anomaly detection to notify users of unusual or unexpected financial activity.
- Support for multi-user family budgeting, enabling shared financial management within households.
- Development of a mobile application using Flutter or React Native for better accessibility and on-the-go usage.
- Introduction of voice-based transaction input to make data entry faster and more convenient.
- Ability to export financial reports in PDF or Excel formats for offline use and documentation.
- Smart bill reminder system to alert users about upcoming payments and due dates.

13. Conclusion

BudgetWise proves to be an effective AI-powered financial management solution by intelligently automating key tasks such as transaction categorization, expense analysis, and future spending prediction. Through the integration of machine learning forecasting, interactive visual dashboards, and personalized goal-tracking features, the system enables users to gain a clearer understanding of their financial behavior and make informed decisions with confidence. Its secure design, user-friendly interface, and analytical depth position BudgetWise as a reliable and comprehensive tool for modern personal finance planning, demonstrating how AI can significantly simplify and enhance everyday money management.