

Ghar Ki Bachat – Hackathon Documentation

Team Name: Nexora

GitHub Repository: <https://github.com/pushpeshpant02/Ghar-Ki-Bachat>

Team Members:

1. Pushpesh Pant : Backend
2. Palak Pandey : Frontend
3. Manas Dobal: Database
4. Rohit Kumar Rathore: AI Models

1. Overview

Ghar-Ki-Bachat is a personal finance tracking system that allows users to add expenses, categorize transactions, view insights, and receive alerts about their financial habits. The system is designed to be fast, simple to use, and highly reliable, with built-in scalability and failure-handling capabilities.

2. System Architecture

The architecture follows a modular, layered approach to keep the application flexible, easy to maintain, and efficient.

2.1 Frontend Layer

- Handles user input (adding expenses, editing entries, viewing dashboards).
- Performs basic validation before sending data to the backend.
- Stores temporary data using local storage when offline.

2.2 Backend Layer

- Processes requests from the frontend.
- Applies business logic such as categorization, filtering, and validation.
- Ensures transactions are stored safely in the database.
- Handles authentication and user management.

2.3 Database Layer

- Stores all expense records, categories, budgets, and user profiles.
- Ensures fast retrieval and analytics-friendly structure.
- Maintains data consistency using ACID transactions.

2.4 AI Insight Layer

- Analyzes user expenses to find spending patterns.
- Generates alerts, cost-saving suggestions, and category breakdowns.
- Uses rule-based logic and ML-ready structure for future enhancements.

2.5 Analytics and Dashboard Layer

- Aggregates data from the database.
- Displays monthly reports, category charts, and spending trends.
- Provides alerts when spending crosses budget thresholds.

3. Data Flow Diagram (Explanation)

The system follows this clean data flow:

1. User Input
The user adds an expense (amount, category, notes).
2. Categorization
The system categorizes the expense automatically or uses a predefined rule.
3. Database Storage
The validated data is stored safely in the database using transactions.
4. AI Insight Layer
Data is analyzed to detect trends or generate alerts.
5. Dashboard + Alerts
Insights are shown immediately as charts, summaries, and warnings.

4. Scalability Strategy

Ghar-Ki-Bachat is designed to support growth from a small user base to thousands of active users.

4.1 Horizontal Scaling

- Application servers can be replicated.
- Load balancer distributes requests evenly.

4.2 Database Scaling

- Read-write separation using replicas.
- Indexing for faster search and filtering.
- Partitioning by user ID if needed in future.

4.3 Caching Layer

- Frequently accessed data (monthly totals, categories) stored in memory cache.
- Reduces database load and speeds up dashboard rendering.

4.4 Cloud Storage

- Files such as receipts can be stored in S3-like object storage.

5. Failure Handling

To ensure smooth functioning, the system incorporates multiple layers of protection.

5.1 Graceful Error Messages

The app never crashes. If an action fails, the user sees a simple message like:
"Something went wrong. Please try again."

5.2 Auto-Retry Mechanism

If a request fails due to temporary issues such as network delay, the system retries automatically before showing an error.

5.3 Offline Support

If the internet is unavailable:

- Expenses are stored locally.
- Synced automatically when connection returns.

5.4 Database Transaction Safety

All write operations use ACID transactions:

- Either fully saved
- Or not saved at all

No corrupted or half-saved entries.

5.5 System Monitoring

The backend monitors issues such as:

- Slow queries

- High load
- Failed API calls

Engineers can react before users experience issues.

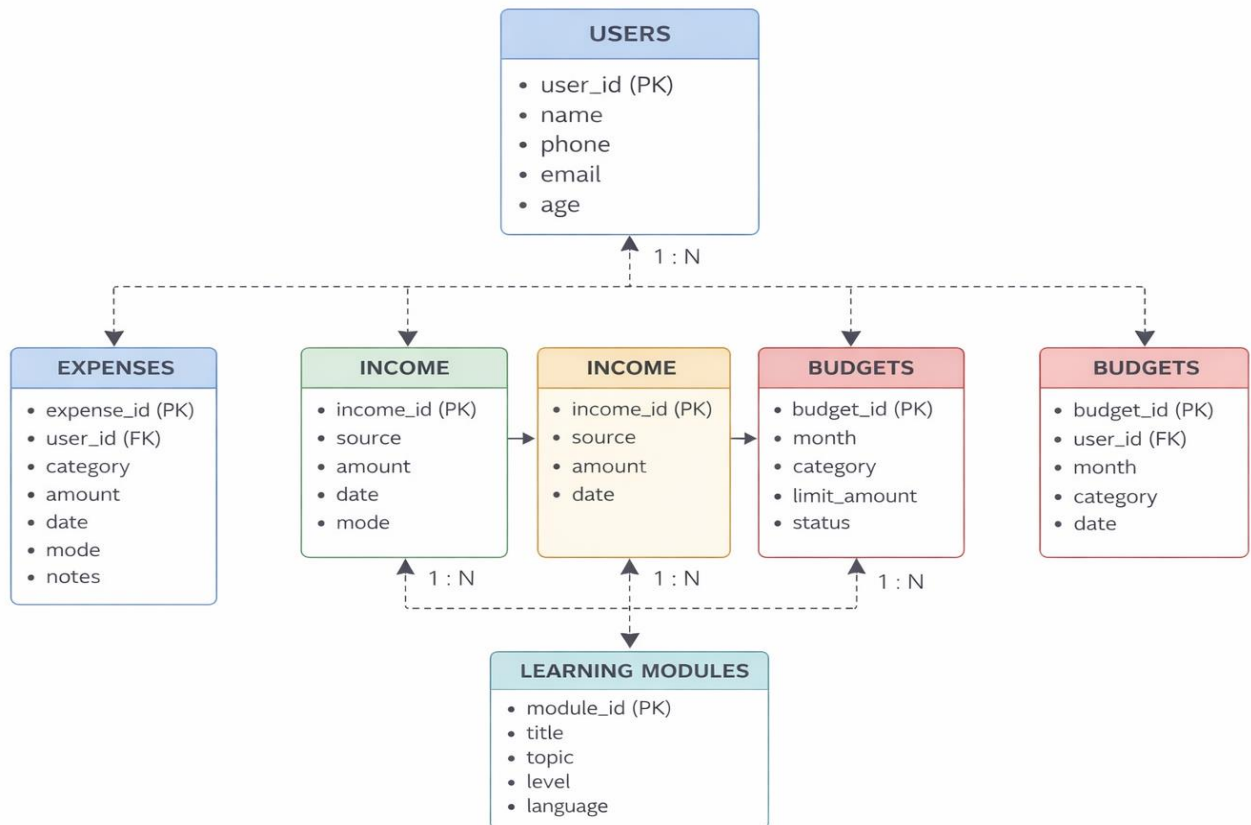
5.6 Failover Server

If the main server goes down, a backup server automatically takes over to avoid downtime.

5.7 Data Backup and Recovery

Automated backups ensure no permanent data loss. Restoration can be done quickly in case of an outage.

6. Database Schema (Simplified)



This diagram shows how all the data in the system is organized and connected.

1. Users Table (Main Table)

All other data belongs to a user.

Each user has:

- user_id
- name
- phone
- email
- age

This is the parent table.

2. Expenses

Every user can have many expense entries.

Each expense saves:

- category
- amount
- date
- payment mode
- notes

3. Income

Each user can have multiple income records.

It stores:

- income source
- amount
- date
- mode

4. Budgets

Users can set different budgets for different categories.

Each entry tracks:

- month
- category
- limit amount

5. Goals

Users can set saving/financial goals.

Each goal stores:

- goal name
- target amount
- timeline
- status

6. Advisory Logs

This table stores what advice the AI gives to each user.

It includes:

- suggestion
- risk level
- date

7. Learning Modules

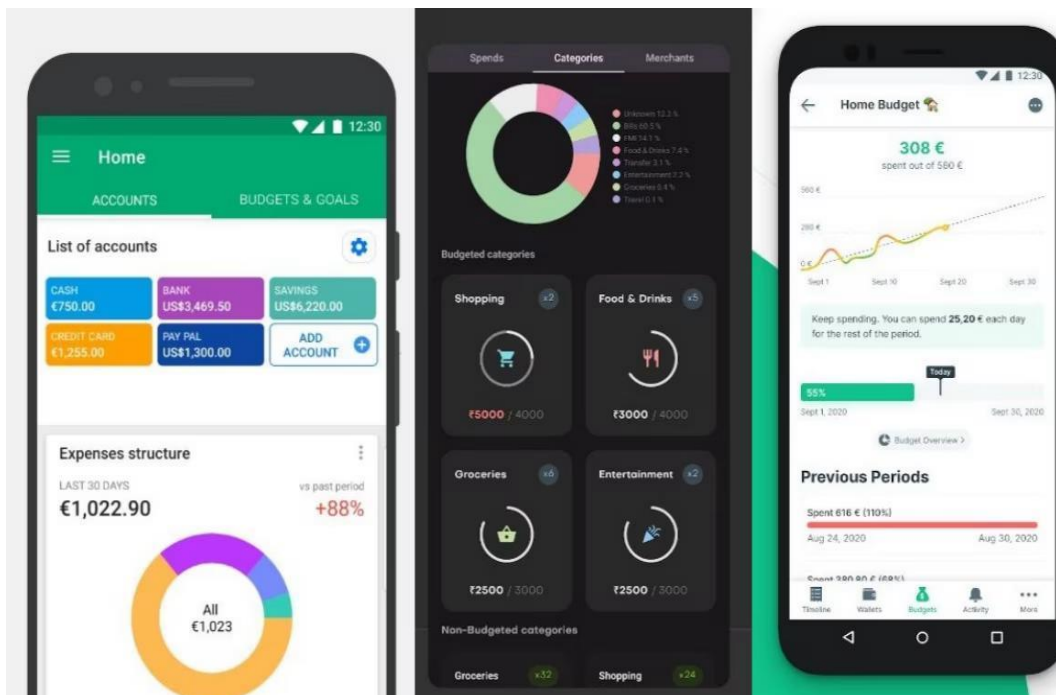
This is separate from users.

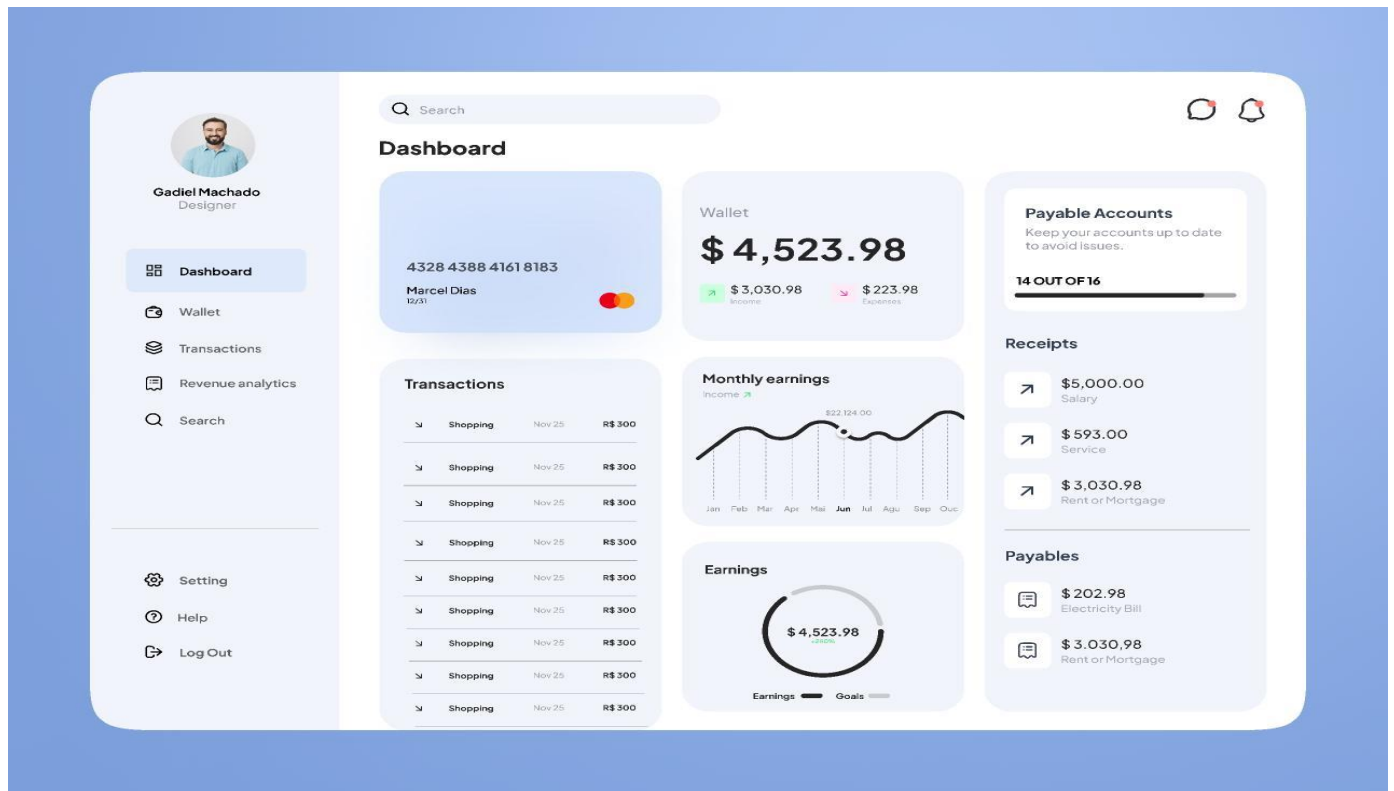
It contains educational content:

- title
- topic
- level
- language

7. UI DESIGN :

Below images show the ui design how our website or will look :





Conclusion

The “Ghar-Ki-Bachat” system provides a complete, user-friendly solution for managing personal finances. By integrating expense tracking, income management, budgeting, goal setting, AI-driven insights, and educational modules, the system empowers users to make informed financial decisions.