



Budgeting Application

BY:

GOKULA VARSHINI K - 23CDR048

HARINI M - 23CDR052

OVERVIEW

The Budgeting Application is helps users to create and manage monthly budgets. The application allows users to create monthly budgets by setting spending limits across various categories such as Food, Rent, Transportation, and Entertainment. Users can then track their actual expenses and compare them with their budgeted amounts and calculates the variance between planned and actual spending, helping users control their finances effectively.

FEATURES :

- Create monthly budget plans per category
- Add and track actual daily expenses
- View variance reports (budget vs actual)
- Store data persistently using RDBMS
- Multi-user support (user-based data separation)

FEATURES :

- **User Registration & Login:** Secure authentication to access personal budget data.
- **Budget Creation:** Users can define budget categories (e.g., Food, Rent, Travel) and Set expected spending for each category and View history of expenses.
- **Variance Calculation:** System calculates the difference between actual spending and budgeted amount. Visual indicators (e.g., red for overspending).
- **Reports & Analytics:** Monthly budget vs. actual comparison.

TECH STACK :

- Frontend: HTML, CSS, JavaScript (React.js)
- Backend: Node.js / Java / Python (Flask)
- Database (RDBMS): MySQL
- API Testing: Postman

Table Structure :

1. Users Table - To support multiple users with authentication. & Ownership

- id - Primary Key
- username
- email
- password

Relationship - One user can have many budgets and expenses

2 . Categories - Stores unique spending categories (e.g., Food, Rent, Travel)

- id - Primary Key
- name

Relationship - Stores unique spending categories (e.g., Food, Rent, Travel)

3. BudgetItems Table - Store Monthly Budget Plans

- user_id - Foreign Key (users(id))
- categories_id - Foreign Key (categories(id))
- budgeted_amount
- month

Relationship - One user Many budget items

One category Many budget items

4. Expenses Table - Log Real Expenses (Stores each transaction/expense made by a user.)

- user_id - Foreign Key (user(id))
- category_id - Foreign Key (categories(id))
- amount
- date

Relationship - One user Many expenses

One category Many expenses

Personal Budgeting Data Model

