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.

Problem Statement

Managing personal finances effectively is a common challenge for many individuals. People often lose track of their expenses and exceed their budgets due to the lack of a simple, user-friendly tool. There is a need for a digital solution that helps users monitor their income, allocate budgets by category, record expenses, and visualize spending patterns to ensure better financial control.

Solution Overview

The Simple Budgeting Application is a web-based tool designed to help users plan and track their monthly expenses. Key features include user registration, budget category management (e.g., food, travel, shopping), expense logging, variance calculation between budgeted and actual spending, and visual summaries. With an intuitive interface built using HTML, CSS, and Bootstrap, and backend logic powered by PHP or Python with a relational database (e.g., MySQL), the application empowers users to take control of their finances, avoid overspending, and improve savings.

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

TableStructure:

Column Name	Data Type	Constraints
id	INT(11)	PRIMARY KEY, NOT NULL, AUTO_INCREMENT
username	VARCHAR(50)	NOT NULL
email	VARCHAR(100)	NOT NULL, UNIQUE
password	VARCHAR(255)	NOT NULL
created_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP

Column Name	Data Type	Constraints
id	INT(11)	PRIMARY KEY, NOT NULL, AUTO_INCREMENT
budget_id	INT(11)	NOT NULL, INDEX, FOREIGN KEY (budget_id) references budgets(id) ON DELETE CASCADE
category	VARCHAR(100)	NOT NULL
amount	DECIMAL(10,2)	NOT NULL

Column Name	Data Type	Constrains
id	INT(11)	PRIMARY KEY, NOT NULL, AUTO_INCREMENT
budget_id	INT(11)	NOT NULL, INDEX, FOREIGN KEY (budget_id) references budgets(id) ON DELETE CASCADE
category	VARCHAR(100)	NOT NULL
amount	DECIMAL(10,2)	NOT NULL

Column Name	Data Type	Constraints
id	INT(11)	PRIMARY KEY, NOT NULL, AUTO_INCREMENT
budget_id	INT(11)	NOT NULL, INDEX, FOREIGN KEY (budget_id) references budgets(id) ON DELETE CASCADE
category_id	INT(11)	DEFAULT NULL, FOREIGN KEY (category_id) references category_budget(id) ON DELETE SET NULL
message	VARCHAR(255)	DEFAULT NULL
created_at	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP