Infosys | Springboard
Internship Program

EXPENSE MANAGEMENT SYSTEM

Group – 3

Mentor:
Rajesh Reddy

Students:
Swarnendu Roy

Aditya Kumar

Monalisa Sahoo

Alekhya Varanasi

Rohit Raj

USER STORIES

1)Add expense

- **Title:** Add Expense
- **As a:** user
- **I want to:** add a new expense entry with details like amount, category, date, and description
- **So that I can:** keep track of my expenses
- **Acceptance Criteria:**
    - Given a form to enter expense details, when I submit valid information, the expense should be saved successfully.
    - If any mandatory field is missing, an error message should appear.
    - The expense should appear in my expense list after saving.

2)View Monthly Expenses

- **Title:** View Monthly Expenses
- **As a:** user
- **I want to:** view a summary of my expenses for a specific month
- **So that I can:** understand my spending patterns
- **Acceptance Criteria:**
    - I should be able to select a month and see a list or summary of expenses.
    - The list should display each expense with category, amount, and date.
    - A total amount for the selected month should be displayed at the top.

3) Categorize Expenses

- **Title:** Categorize Expenses
- **As a:** user

- **I want to:** assign categories to my expenses (e.g., Food, Transport, Entertainment)
- **So that I can: analyze where most of my money goes**
- **Acceptance Criteria:**
    - A dropdown menu with predefined categories should be available when adding an expense.
    - I should be able to filter expenses by category on the summary page.
    - A visual representation (e.g., pie chart) should display expense distribution by category.

4) Set Budget Limits

- **Title:** Set Budget Limits
- **As a:** user
- **I want to:** set a monthly spending limit for each category
- **So that I can:** avoid overspending in certain areas
- **Acceptance Criteria:**
    - A budget setup screen where I can input a limit for each category.
    - Notifications when expenses in a category exceed the set limit.
    - A dashboard showing current spending vs. budget limits.

5) Export Expense Report

- **Title:** Export Expense Report
- **As a:** user
- **I want to:** export my expense history to a file format like CSV
- **So that I can:** keep records or analyze my expenses outside the system
- **Acceptance Criteria:**
    - An export button should generate a CSV with all expense details.
    - The file should include columns like Date, Category, Amount, and Description.
    - A prompt should confirm successful export and save location.

Functional Requirements

1) **User Registration and Authentication**

- The system shall allow users to register, log in, log out, and reset their password securely.

2) **Expense Management**

- The system shall allow users to add, edit, and delete expense entries, each including category, amount, date, and description, and display recent expenses in reverse chronological order.

3) **Category Management**

- The system shall allow users to create, edit, delete, and select from predefined categories.

4) **Budget Management**

- The system shall allow users to set a monthly budget for each category and notify them when expenses approach or exceed the budget.

5) **Reports and Analytics**

- The system shall generate a summary report of total expenses for a user-specified period, display expenses by category in a chart format, and allow users to export reports as a PDF.

6) **Payment Method Management**

- The system shall allow users to add, edit, and delete different payment methods (e.g., credit card, cash).

Non-Functional Requirements

1) **Performance**
   - The system should load the expense dashboard within 2 seconds and complete expense data queries and report generation within 5 seconds.
2) **Security**
   - The system shall use encryption to securely store user credentials and sensitive data, and automatically log out users after 15 minutes of inactivity.
3) **Usability**
   - The system should have a user-friendly interface that is easy to navigate, and input forms should provide real-time validation to ensure data accuracy.
4) **Reliability**
   - The system shall have an uptime of 99% to ensure availability and back up data regularly to prevent loss.
5) **Scalability**
   - The system shall handle a growing number of users and expenses without performance degradation.
6) **Compatibility**
   - The system should be accessible on web and mobile platforms, compatible with major browsers and devices.

DATABASE DESIGN

Database Tables *PK :- Primary Key, FK:- Foreign Key

1. Users

- Description: Stores user information.
- Fields:
  - user_id (PK): Unique identifier for each user.
  - username: Name of the user.
  - email: User's email for login and notifications.
  - password: Encrypted password for security.
  - created_at: Timestamp of account creation.

```
CREATE TABLE Users (
    user_id CHAR(36) PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

2. Categories

- Description: Stores categories of expenses, like Food, Travel, and Utilities.
- Fields:
  - category_id (PK): Unique identifier for each category.
  - category_name: Name of the category.
  - description: Additional information about the category.

```
CREATE TABLE Categories (
    category_id CHAR(36) PRIMARY KEY,
    category_name VARCHAR(50) NOT NULL,
    description VARCHAR(255)
);
```

.

3. Expenses

- Description: Contains records of each expense entry.
- Fields:
  - expense_id (PK): Unique identifier for each expense.
  - user_id (FK to Users): References the user who made the expense.
  - category_id (FK to Categories): References the category of the expense.
  - payment_method_id (FK to Payment_Methods): References the payment method used.
  - amount: The expense amount.
  - date: Date the expense was made.
  - description: Additional notes about the expense.

```
CREATE TABLE Expenses (
    expense_id CHAR(36) PRIMARY KEY,
    user_id CHAR(36) NOT NULL,
    category_id CHAR(36) NOT NULL,
    payment_method_id CHAR(36) NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    date DATE NOT NULL,
    description VARCHAR(255),
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (category_id) REFERENCES Categories(category_id),
    FOREIGN KEY (payment_method_id) REFERENCES Payment_Methods(payment_method_id)
);
```

4. Payment Methods

- Description: Stores available payment methods (e.g., Cash, Credit Card).
- Fields:
  - payment_method_id (PK): Unique identifier for each payment method.
  - method_name: Name of the payment method.
  - description: Details about the payment method.

```
CREATE TABLE Payment_Methods (
    payment_method_id CHAR(36) PRIMARY KEY,
    method_name VARCHAR(50) NOT NULL,
    description VARCHAR(255)
);
```

5. Budgets

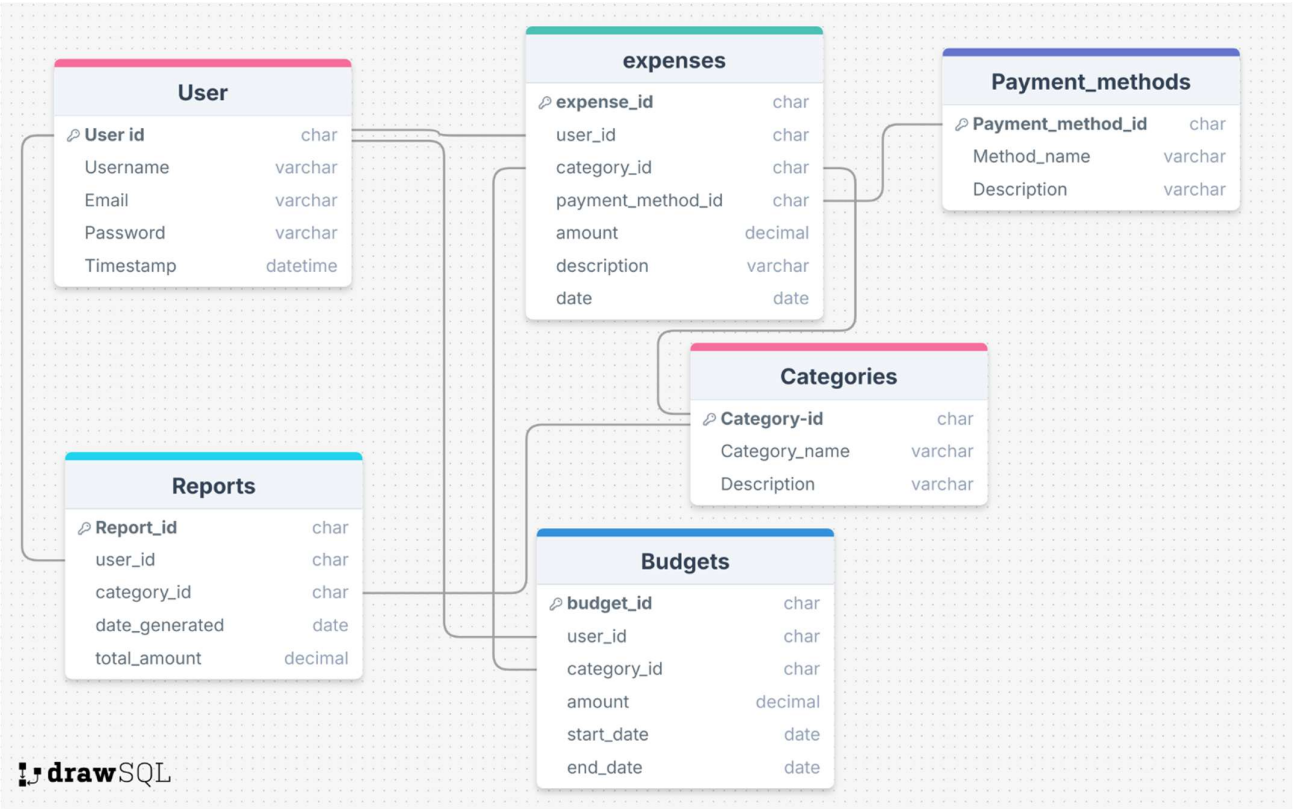
- Description: Defines budget limits for each user, often per category.
- Fields:
    - budget_id (PK): Unique identifier for each budget.
    - user_id (FK to Users): References the user who set the budget.
    - category_id (FK to Categories): The category the budget applies to.
    - amount: The budgeted amount.
    - start_date: When the budget starts.
    - end_date: When the budget ends.

```
CREATE TABLE Budgets (
    budget_id CHAR(36) PRIMARY KEY,
    user_id CHAR(36) NOT NULL,
    category_id CHAR(36) NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (category_id) REFERENCES Categories(category_id)
);
```

6. Reports

- Description: Stores generated reports for users.
- Fields:
    - report_id (PK): Unique identifier for each report.
    - user_id (FK to Users): References the user for whom the report is generated.
    - category_id (optional, FK to Categories): Category the report is focused on.
    - date_generated: Date the report was created.
    - total_amount: Summarized amount within the report.

```
CREATE TABLE Reports (
    report_id CHAR(36) PRIMARY KEY,
    user_id CHAR(36) NOT NULL,
    category_id CHAR(36),
    date_generated TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    total_amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (category_id) REFERENCES Categories(category_id)
);
```


User
User id char
Username varchar
Email varchar
Password varchar
Timestamp datetime
expenses
expense_id char
user_id char
category_id char
payment_method_id char
amount decimal
description varchar
date date
Payment_methods
Payment_method_id char
Method_name varchar
Description varchar
Categories
Category-id char
Category_name varchar
Description varchar
Reports
Report_id char
user_id char
category_id char
date_generated date
total_amount decimal
Budgets
budget_id char
user_id char
category_id char
amount decimal
start_date date
end_date date
drawSQL