Back-End Development Phase

Date	15 April 2025
Team ID	SWTID1744094910
Project Title	Personal Expense Tracker App
Maximum Marks	5 Marks

Database Schema Design

1. Core Tables

1.1 users

Column	Type	Description
user_id	SERIAL	Primary key
email	VARCHAR(255)	Unique, indexed
password_hash	VARCHAR(255)	BCrypt hashed
name	VARCHAR(100)	
created_at	TIMESTAMP	Default: CURRENT_TIMESTAMP
updated_at	TIMESTAMP	Default: CURRENT_TIMESTAMP

Indexes:

- UNIQUE (email)

1.2 categories

Column	Type	Description
category_id	SERIAL	Primary key
user_id	INTEGER	Foreign key: users.user_id
name	VARCHAR(50)	E.g., "Food", "Travel"
type	VARCHAR(10)	expense or income
created_at	TIMESTAMP	Default: CURRENT_TIMESTAMP

Indexes:

- INDEX (user_id)

1.3 expenses

Column	Type	Description
expense_id	SERIAL	Primary key
user_id	INTEGER	Foreign key: users.user_id
category_id	INTEGER	Foreign key: categories.category_id
amount	DECIMAL(15,2)	Positive value
date	DATE	
payment_method	VARCHAR(50)	E.g., "Cash", "Credit Card"
description	TEXT	Optional
group_id	INTEGER	Foreign key: groups.group_id (optional)
created_at	TIMESTAMP	Default: CURRENT_TIMESTAMP

Indexes:

- INDEX (user_id, date)
- INDEX (category_id)

1.4 budgets

Column	Type	Description
budget_id	SERIAL	Primary key
user_id	INTEGER	Foreign key: users.user_id
category_id	INTEGER	Foreign key: categories.category_id
amount	DECIMAL(15,2)	Budget limit
period	VARCHAR(10)	monthly, weekly, custom
start_date	DATE	
end_date	DATE	
created_at	TIMESTAMP	Default: CURRENT_TIMESTAMP

Indexes:

- INDEX (user_id, category_id)

2. Shared Expenses (Groups)

2.1 groups

Column	Type	Description
group_id	SERIAL	Primary key
name	VARCHAR(100)	E.g., "Family Budget"
created_by	INTEGER	Foreign key: users.user_id
created_at	TIMESTAMP	Default: CURRENT_TIMESTAMP

2.2 user_groups (Junction Table)

Column	Type	Description
user_id	INTEGER	Foreign key: users.user_id
group_id	INTEGER	Foreign key: groups.group_id
joined_at	TIMESTAMP	Default: CURRENT_TIMESTAMP

Composite Primary Key: (user_id, group_id)

2.3 expense_shares

Column	Type	Description
expense_id	INTEGER	Foreign key: expenses.expense_id
user_id	INTEGER	Foreign key: users.user_id
share_amount	DECIMAL(15,2)	Amount owed by the user
settled	BOOLEAN	Default: false
settled_at	TIMESTAMP	

Composite Primary Key: (expense_id, user_id)

3. Security & Performance

- Password Hashing: Use BCrypt with a cost factor of 12.
- Encryption: Encrypt sensitive fields (e.g., payment_method) using AES-256.
- Indexes:
  - users.email for fast login.
  - expenses.date for filtering by date.
  - budgets.period for budget tracking.
- Constraints:
  - CHECK (amount > 0) in expenses and budgets.
  - CHECK (type IN ('expense', 'income')) in categories.

Example Queries

1. Monthly Spending by Category

```sql
SELECT
  c.name AS category,
  SUM(e.amount) AS total_spent
FROM expenses e
JOIN categories c ON e.category_id = c.category_id
WHERE e.user_id = 1
  AND e.date BETWEEN '2024-04-01' AND '2024-04-30'
GROUP BY c.name;
```

2. Budge vs Actual Spending

```sql
SELECT
  b.amount AS budget_limit,
  COALESCE(SUM(e.amount), 0) AS actual_spending
FROM budgets b
LEFT JOIN expenses e
  ON b.category_id = e.category_id
  AND e.date BETWEEN b.start_date AND b.end_date
WHERE b.user_id = 1
  AND b.category_id = 3
GROUP BY b.amount;
```

3. Shared Expense Summary

```sql
SELECT
  u.name AS user,
  SUM(es.share_amount) AS total_owed
FROM expense_shares es
JOIN users u ON es.user_id = u.user_id
WHERE es.expense_id IN (
  SELECT expense_id
  FROM expenses
  WHERE group_id = 5
)
GROUP BY u.name;
```