

PERSONAL EXPENSE TRACKER

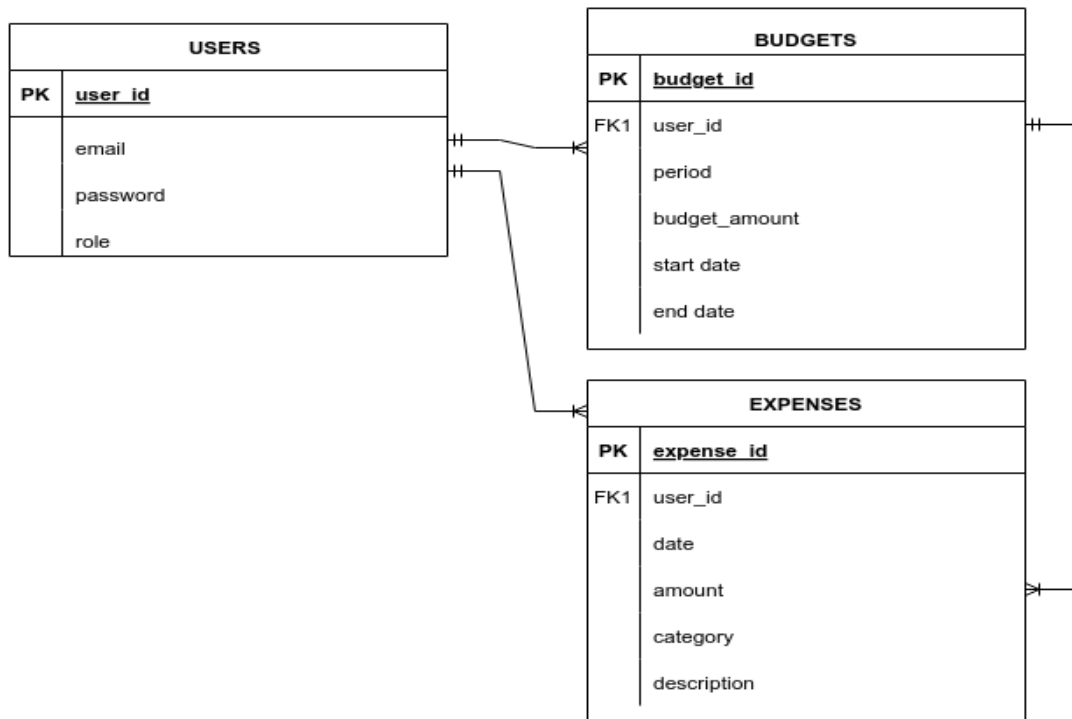
(Sidhvi Nuvvula, Madhuritha Alle, Saranya Roy, Usharani Lankalapalli, Chandana Rajashekhar, Dhyan Pandya)

1. Problem Definition & Requirements

- **Problem Statement:**
Managing personal finances manually can be cumbersome and error-prone. The Expense Tracker project aims to provide a streamlined digital solution for individuals to track their expenses, manage budgets, and analyze spending patterns efficiently.
- **Requirements Gathering**
Functional Requirements:
 - User authentication with role-based access (admin/user).
 - Create, read, update, delete (CRUD) operations for expenses and budgets.
 - Dashboard for tracking and visualizing expenses.**Non-Functional Requirements:**
 - Secure data management using MySQL.
 - Responsive user interface with Streamlit.
 - Scalable database design for future enhancements.
- **Scope and Feasibility**
The project focuses on tracking individual expenses and managing budgets. Features include user management, role-based access, and real-time updates. Using Python, Streamlit, and MySQL ensures feasibility and scalability.

2. Database Design

- **Entity-Relationship Diagram (ERD):**
The database includes the following entities:
 - Users: user_id, email, password, role
 - Expenses: expense_id, user_id, amount, category, date, description
 - Budgets: budget_id, user_id, budget_amount, period, start_date, end_date



- **Normalization:**

Normalization was applied to reduce redundancy and maintain data integrity. All tables are in at least 3rd Normal Form (3NF).

- **Table Design**Example of table structure:

```

CREATE TABLE users (
  user_id INT AUTO_INCREMENT PRIMARY KEY,
  email VARCHAR(255) UNIQUE NOT NULL,
  password VARCHAR(255) NOT NULL,
  role ENUM('admin', 'user') DEFAULT 'user' NOT NULL
);
  
```

```

CREATE TABLE expenses (
  expense_id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT,
  amount DECIMAL(10, 2),
  category VARCHAR(50),
  date DATE,
  description VARCHAR(255),
  FOREIGN KEY (user_id) REFERENCES users (user_id),
  FOREIGN KEY (budget_id) REFERENCES budgets (budget_id)
);
  
```

```
FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE  
CASCADE
```

```
);
```

```
CREATE TABLE budgets (
```

```
budget_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
user_id INT,
```

```
budget_amount DECIMAL(10, 2),
```

```
period VARCHAR(20), -- 'weekly' or 'monthly'
```

```
start_date DATE,
```

```
end_date DATE,
```

```
FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE  
CASCADE
```

```
);
```

3. Implementation:

- SQL Queries:

```
INSERT INTO users (user_id, email,password) VALUES
```

```
(1, 'Alice Smith', 'alice@example.com'),
```

```
(2, 'Bob Johnson', 'bob@example.com'),
```

```
(3, 'Carol Williams', 'carol@example.com');
```

```
INSERT INTO users (email, password, role)
```

```
VALUES ('sidhvinuvvula16@gmail.com', 'Sidhu@1416', 'admin');
```

```
SELECT * FROM USERS;
```

```
INSERT INTO expenses (expense_id, user_id, date, amount, category,  
description) VALUES
```

```
(1, 1, '2024-10-31', 32.97, 'Entertainment', 'Sample expense data'),
```

```
(2, 2, '2024-10-24', 121.75, 'Dining Out', 'Sample expense data'),
```

```
(3, 2, '2024-10-25', 50.67, 'Entertainment', 'Sample expense data'),
```

```
(4, 1, '2024-11-08', 61.04, 'Utilities', 'Sample expense data'),
```

```
(5, 3, '2024-11-09', 141.26, 'Dining Out', 'Sample expense data');
```

```

SELECT * FROM expenses;
INSERT INTO budgets (budget_id, user_id, budget_amount, period,
start_date, end_date)
VALUES
(1, 1, 794.45, 'weekly', '2024-10-13', '2024-11-12'),
(2, 3, 425.87, 'weekly', '2024-10-13', '2024-11-12'),
(3, 2, 890.11, 'monthly', '2024-10-13', '2024-11-12'),
(4, 3, 339.07, 'monthly', '2024-10-13', '2024-11-12'),
(5, 3, 224.22, 'weekly', '2024-10-13', '2024-11-12');
SELECT * FROM budgets;

```

- Data Integrity:

1. Primary Keys

Definition: A primary key is a column (or a combination of columns) in a table that uniquely identifies each row in that table.

Usage in the Project:

Primary keys ensure that each record in the database tables is unique and can be referenced by other tables. Examples:

- **Users Table:** The user_id column serves as the primary key, uniquely identifying each user.
- **Expenses Table:** The expense_id column acts as the primary key, uniquely identifying each expense entry.
- **Budgets Table:** The budget_id column is the primary key, uniquely identifying each budget entry.

2. Foreign Keys

Definition: A foreign key is a column in one table that establishes a relationship between the data in two tables. It references the primary key in another table.

Usage in the Project:

Foreign keys are used to create relationships between tables, ensuring referential integrity:

- **Expenses Table:**

- The user_id column is a foreign key referencing the user_id in the users table. This ensures that each expense is associated with a valid user.
- **Budgets Table:**
 - The user_id column is a foreign key referencing the user_id in the users table. This ensures that each budget is tied to a valid user.

3. Constraints

Definition: Constraints are rules enforced on data in the database to ensure validity, integrity, and performance.

Usage in the Project:

- **NOT NULL Constraint:**
 - Ensures that critical fields like email, password, and role in the users table cannot be empty.
 - Applied to fields like amount and category in the expenses table to ensure completeness.
- **UNIQUE Constraint:**
 - Enforced on the email column in the users table to prevent duplicate user registrations.
- **CHECK Constraint:**
 - Could be used to ensure valid role values (e.g., user or admin) in the users table.
 - Example: Ensures amount values in the expenses table are positive.
- **Default Values:**
 - Default roles are set as user for new registrations unless specified otherwise.
- **Referential Integrity:**
 - Foreign keys ensure that records in the expenses and budgets tables are tied to valid user_id values in the users table. Deleting or updating a user cascades or restricts associated records based on the foreign key configuration.
- **Test Data:**
SELECT * FROM users

	user_id	email	PASSWORD	role	
	1	alice@example.com	Alice@123	user	
	2	bob@example.com	Bob@123	user	
	3	carol@example.com	Carol@123	user	
	4	sidhvi@gmail.com	Sidhvi@123	user	
	5	saranyaroy@gmail.com	Saranya@123	user	
	6	madhuritha@gmail.com	Madhu@123	user	
	7	eharsha05@gmail.com	Harsha	user	
	8	sidhvinuvvula16@gmail.com	Sidhu@1416	admin	

SELECT * FROM expenses

	expense_id	user_id	amount	category	date	description	created_at	
	1	1	32.97	Entertainment	2024-10-31	Sample expense data	2024-11-18 12:09:07	
	2	2	121.75	Dining Out	2024-10-24	Sample expense data	2024-11-18 12:09:07	
	3	2	50.67	Entertainment	2024-10-25	Sample expense data	2024-11-18 12:09:07	
	4	1	61.04	Utilities	2024-11-08	Sample expense data	2024-11-18 12:09:07	
	5	3	141.26	Dining Out	2024-11-09	Sample expense data	2024-11-18 12:09:07	
	6	6	6.98	6.98	2024-11-22	Food at Nawabi's	2024-11-21 23:28:25	
	7	7	1000.00	car	2024-11-22	repair	2024-11-22 10:02:15	
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

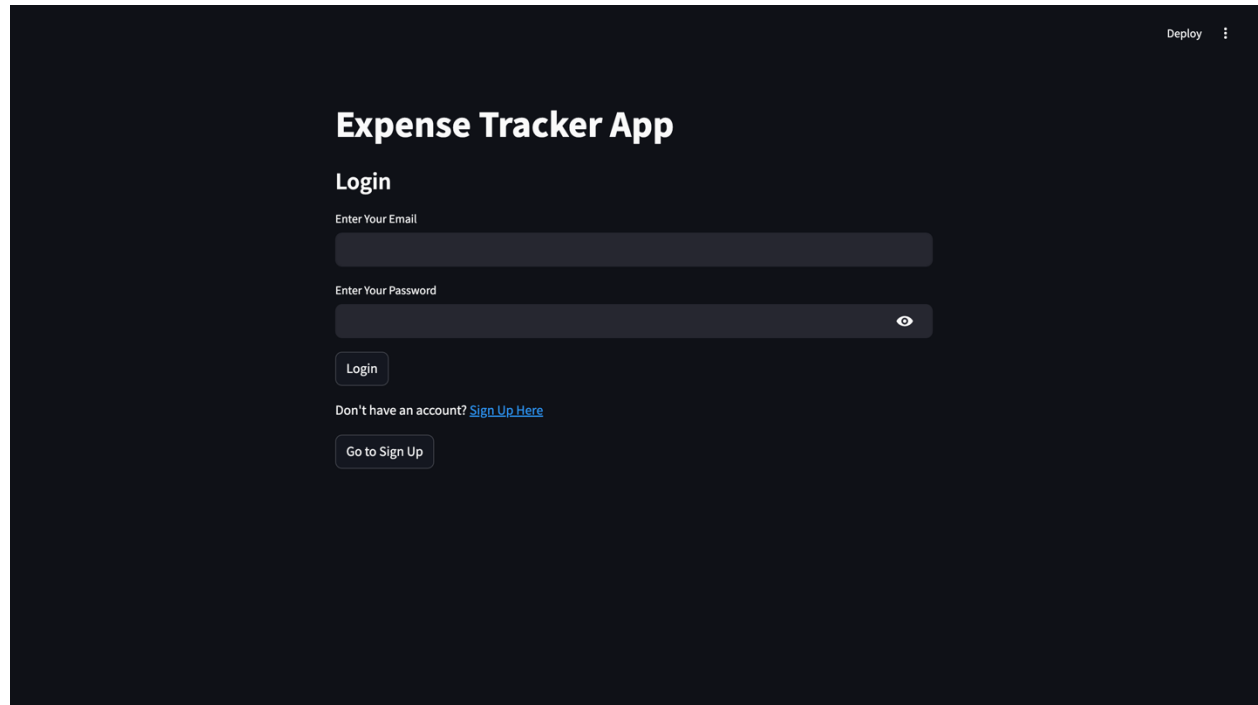
SELECT * FROM budgets

	budget_id	user_id	budget_amou...	period	start_date	end_date	
	1	1	794.45	weekly	2024-10-13	2024-11-12	
	2	3	425.87	weekly	2024-10-13	2024-11-12	
	3	2	890.11	monthly	2024-10-13	2024-11-12	
	4	3	339.07	monthly	2024-10-13	2024-11-12	
	5	3	224.22	weekly	2024-10-13	2024-11-12	
	6	1	0.00	weekly	2024-11-19	2024-11-19	
	7	1	0.00	weekly	2024-11-19	2024-11-19	
	8	1	0.00	weekly	2024-11-19	2024-11-19	
	10	7	5000.00	weekly	2024-11-22	2024-11-29	
	12	6	250.00	weekly	2024-11-22	2024-11-29	
	14	6	50.00	weekly	2024-11-29	2024-12-05	
	NULL	NULL	NULL	NULL	NULL	NULL	

4. Application Integration (15 points)

- **User Interface**

- Tools: Streamlit for the front-end.
- Features: Login page, expense management, and budget overview
- Screenshots of key UI components:



Expense Tracker App

Login

Enter Your Email

sidhvinuvvula16@gmail.com

Enter Your Password

.....

Press Enter to apply

Login

Don't have an account? [Sign Up Here](#)

Go to Sign Up

Navigation

Logout

Tracker Menu

Users

Welcome,
[sidhvinuvvula16@gmail.com!](#)

Manage Users

	User ID	Email	Role
0	1	alice@example.com	user
1	2	bob@example.com	user
2	3	carol@example.com	user
3	4	sidhi@gmail.com	user
4	5	saranyaroy@gmail.com	user
5	6	madhuritha@gmail.com	user
6	7	eharsha05@gmail.com	user
7	8	sidhvinuvvula16@gmail.com	admin

Add New User

New User Email

New User Password

Navigation

Logout

Tracker Menu

Expenses

Deploy

Welcome,
sidhvinuvvula16@gmail.com!

Manage Expenses

	Expense ID	User ID	Amount	Category	Date	Description
0	1	1	32.97	Entertainment	2024-10-31	Sample expense data
1	2	2	121.75	Dining Out	2024-10-24	Sample expense data
2	3	2	50.67	Entertainment	2024-10-25	Sample expense data
3	4	1	61.04	Utilities	2024-11-08	Sample expense data
4	5	3	141.26	Dining Out	2024-11-09	Sample expense data
5	6	6	6.98	6.98	2024-11-22	Food at Nawabi's
6	7	7	1,000	car	2024-11-22	repair

Add New Expense

Amount

0.00

Category

Navigation

Logout

Tracker Menu

Expenses

Deploy

Add New Expense

Amount

0.00

Category

Date

2024/11/24

Description

Add Expense

Select Expense to Edit

1

Updated Date

2024/11/24

Updated Amount

32.97

Updated Category

Deploy

Navigation

Logout

Tracker Menu

Expenses

Updated Date

2024/11/24

Updated Amount

32.97

-

+

Updated Category

32.97

Updated Description

2024-10-31

Update Expense

Select Expense to Delete

1

Delete Expense

<

Deploy

Navigation

Logout

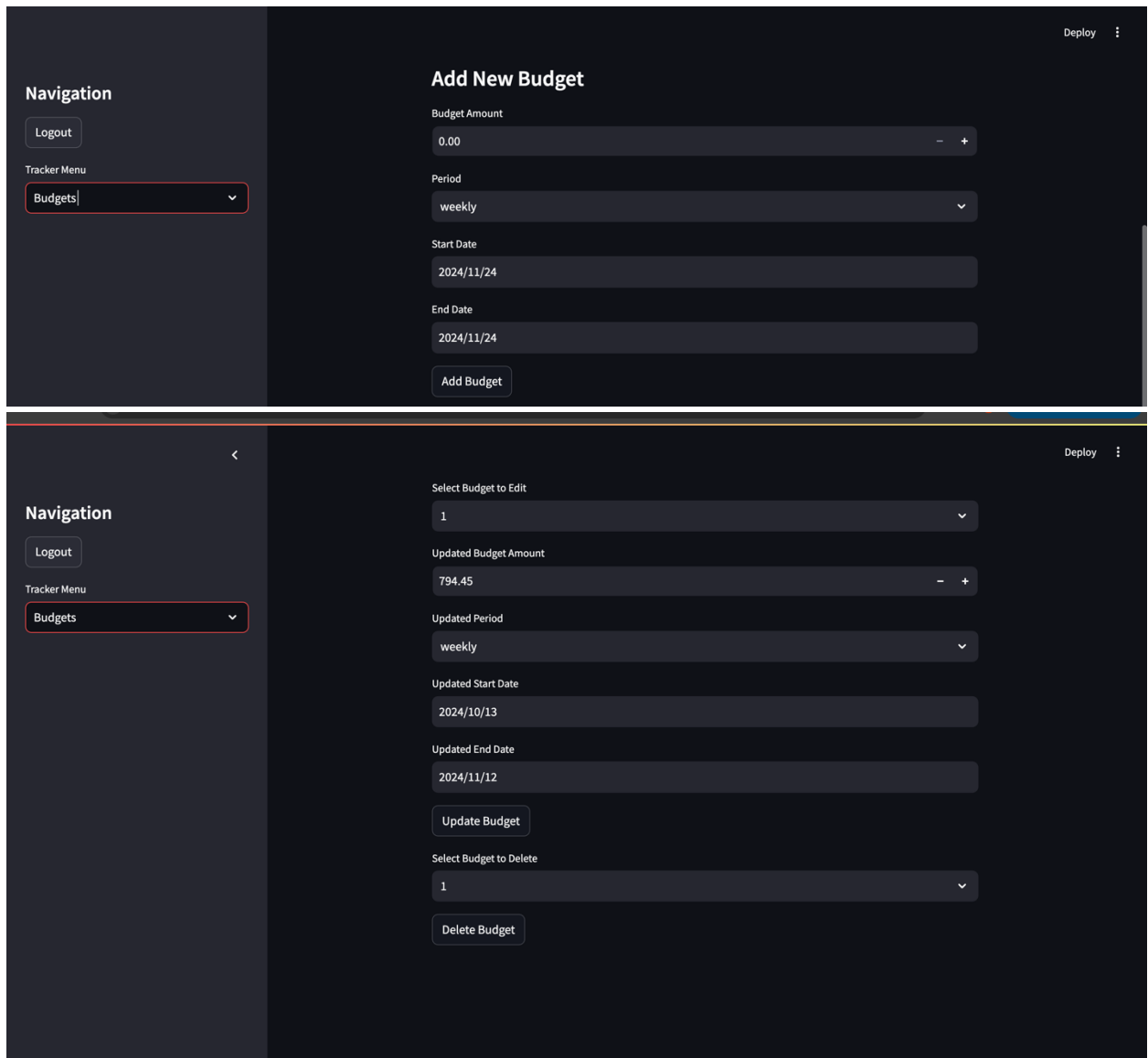
Tracker Menu

Budgets

Welcome,
sidhvinuvvula16@gmail.com!

Manage Budgets

	Budget ID	User ID	Amount	Period	Start Date	End Date
0	1	1	794.45	weekly	2024-10-13	2024-11-12
1	2	3	425.87	weekly	2024-10-13	2024-11-12
2	3	2	890.11	monthly	2024-10-13	2024-11-12
3	4	3	339.07	monthly	2024-10-13	2024-11-12
4	5	3	224.22	weekly	2024-10-13	2024-11-12
5	6	1	0	weekly	2024-11-19	2024-11-19
6	7	1	0	weekly	2024-11-19	2024-11-19
7	8	1	0	weekly	2024-11-19	2024-11-19
8	10	7	5,000	weekly	2024-11-22	2024-11-29
9	12	6	250	weekly	2024-11-22	2024-11-29



5. Teamwork & Collaboration :

1. Chandana and Usha:

Designed the database schema and ensured it met normalization standard.

Created SQL queries for CRUD operations

Developed the users, expenses, and budgets tables.

Ensured the proper use of primary and foreign keys.

Wrote table creation scripts and sample test queries.

Prepared the project report and documentation.

2. Dhyan Pandya and Saranya :

Designed and implemented the user interface using Streamlit.

Focused on creating a simple, functional UI for tracking expenses and managing budgets.

Created the login, signup pages

Created PowerPoint slides summarizing the key aspects of the project.

Practiced and delivered the presentation.

3. Sidhvi and Madhuritha:

Connected the Streamlit frontend to the MySQL backend.

Developed functions for executing CRUD operations.

Ensured data security by using prepared SQL statements.

Implemented Python functions to handle database queries.

Tested database connectivity and optimized CRUD operations.

Debugged and resolved issues related to data retrieval.