**Final Project Specification & Database Design Document**

**1.Demo Scenario Overview**

**Specific Features to Present:**

This section outlines the core system functionalities that will be demonstrated during the final project presentation:

1. User registration and login: Secure account creation and access control.
2. Add income and expense records: Users can manually input transactions to track their daily finances.
3. View monthly financial summaries: The system summarizes a user’s financial activity by month.
4. Visualize expenses by category using charts: Expense data is converted into visual charts (e.g., pie charts) to help users quickly understand their spending habits.
5. Filter transactions by date and category: Users can locate specific transactions using flexible filtering options.

**User Actions in Demo:**

This section simulates the expected user interaction flow during the demo, showing how the system is used in real-life scenarios:

1. A user logs in to their MyFinance account using their credentials.
2. The user adds a new expense record, specifying the amount, category (e.g., food), and date.
3. They navigate to the financial reports page, where a pie chart displays their spending distribution for the month.
4. They use a filter to search for specific transactions, such as all dining expenses in the past two weeks.

**Functional Parts for Demo**

For the demo, we plan to implement the following core flows to ensure a clear and working user experience, even if some parts are simplified:

1. User Login (partial): A simplified login form will be available. It may use a hardcoded username and password for demonstration purposes instead of full authentication.
2. Add Transaction (fully functional): Users will be able to add a new income or expense record. This record will be stored in the database via a form using PHP and MariaDB.
3. View Transaction Table (functional): A page will display all transactions for the current user in a simple table format retrieved from the database.
4. View Report (pseudo implementation): A pie chart will be shown to represent the breakdown of expenses by category. For now, the chart will use static mock data instead of live database content.

**2. Planned URL Endpoints**

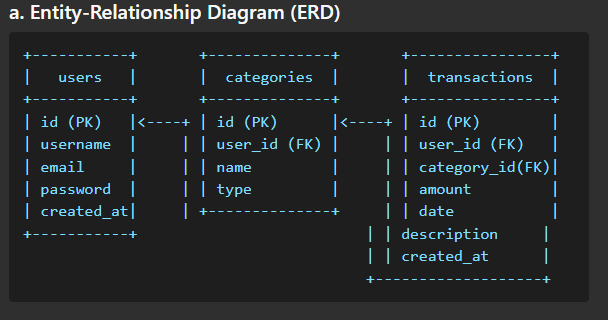
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **URL Path** | **HTTP Method** | **HTTP Variables** | **Session Variables** | **DB Tables** |
| /login.php | POST | Username,password | User\_id | SELECT user from users |
| /register.php | POST | Username,  password,  email |  | INSERT new user into users |
| /dashboard.php | GET |  | User\_id | SELECT transactions from transactions |
| /add\_transaction.php | POST | Amount,  category,date,  description | User\_id | INSERT  into transactions |
| /view\_report.php | GET | Month ,year | User\_id | SELECT+ GROUP BY from transactions |
| /logout.php | GET |  |  | Destory session |

Brief description

* login.php: Handles login form submission and validates credentials.
* register.php: Handles new user registration.
* dashboard.php: Displays all transactions for the logged-in user.
* add\_transaction.php: Processes the form to add a new transaction.
* view\_report.php: Shows a visual summary of expenses by category (uses static data if needed).
* logout.php: Logs the user out and ends the session.

**3. Database Design**

**a. Entity-Relationship Diagram (ERD)**

****

**b. Relational Model (Table Definitions)**

**users**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Constraints |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT |
| username | VARCHAR(50) | UNIQUE, NOT NULL |
| email | VARCHAR(100) | UNIQUE, NOT NULL |
| password | VARCHAR(255) | NOT NULL |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP |

**categories**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Constraints |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT |
| user\_id | INT | FOREIGN KEY REFERENCES users(id), NULLABLE |
| name | VARCHAR(50) | NOT NULL |
| type | ENUM('income','expense') | NOT NULL |

**transactions**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Constraints |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT |
| user\_id | INT | FOREIGN KEY REFERENCES users(id), NOT NULL |
| category\_id | INT | FOREIGN KEY REFERENCES categories(id), NOT NULL |
| amount | DECIMAL(10,2) | NOT NULL |
| transaction\_date | DATE | NOT NULL |
| description | TEXT |  |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP |

**c. Normalization**

* **First Normal Form (1NF):** All attributes are atomic and each column contains only one value.
* **Second Normal Form (2NF):** All non-key columns are fully functionally dependent on the entire primary key.
* **Third Normal Form (3NF):** No transitive dependencies; every non-key attribute depends only on the primary key.

**Design Decisions:**

* Separate categories table avoids duplication of category names and supports extensibility.
* Linking user\_id across tables supports multi-user environments.
* ENUM type in categories ensures consistency between income and expense classification.
* Use of foreign keys preserves referential integrity between tables.