DOCUMENTATION

Project Title: Budget Wise AI

Milestone 1: Weeks 1–2

Module 1: User Authentication & Basic Transaction Input

Objective

To develop a secure user authentication system and a Streamlit interface that allows users to manually input and view financial transactions.

High-Level Requirements

1. User Registration

- **Implement a registration form within Streamlit for new users.**
- **Collect email, password, and name.**
- **Store credentials securely in a SQLite/MySQL database.**
- **Passwords are hashed using the bcrypt library before storage.**
- **Generate and store a JWT (JSON Web Token) for session-based authentication.**

2. Login System

- **Create a Streamlit login page.**
- **Verify user credentials with the stored hash in the database.**
- **Upon successful login, store a session state variable (st.session_state["authenticated"] = True) to manage user sessions.**
- **Redirect users to the main dashboard page after login.**

3. Profile Management

- **Display basic user information: Name, Email, and Date Joined.**
- **Allow users to view a quick summary of:**

- **Total Income**
- **Total Expense**
- **Net Balance**

- **Include an option to log out by clearing the Streamlit session state.**

4. Manual Transaction Input

- **Create a Streamlit form for adding transactions with the following fields:**
  - **Date (default: today's date)**
  - **Description (text input)**
  - **Amount (numeric input)**
  - **Type (select box: Income / Expense)**
- **Store each transaction in the database, linked to the logged-in user.**
- **Display all transactions in a data table using st.dataframe() or st.table().**

Component	Technology
Web Framework	Streamlit
Backend	Python
Database	SQLite / MySQL
Authentication	JWT / Streamlit Session State
Security	bcrypt (Password Hashing)
Data Handling	Pandas
Visualization	Plotly / Matplotlib

Deliverables

- **Fully functional registration and login pages.**
- **Secure session-based authentication using Streamlit's session state.**

- **Simple form-based transaction entry with database storage.**
- **Transaction table display on user dashboard.**

Expected Output

- **Registered users can log in securely.**
- **Users can input and view their transactions.**
- **Data persists across sessions and is isolated per user.**

Milestone 2: Weeks 3–4

Module 2: Transaction Categorization & Basic Reporting

Objective

Enhance the system by adding automatic transaction categorization and a dynamic reporting dashboard.

High-Level Requirements

1. Automated Categorization

- **Use a rule-based or keyword-matching approach to classify transactions.**
- **Example logic:**
    - **If description contains "grocery", "supermarket" → Category: *Groceries***
    - **If description contains "bus", "cab", "fuel" → Category: *Transport***
    - **If description contains "rent", "lease" → Category: *Housing***
- **Allow manual editing of the auto-assigned category via dropdown in the Streamlit table.**

2. Spending Summary Reports

- **Use Pandas to process transactions and generate:**
  - **Total spending per category**
  - **Monthly income and expense summaries**
  - **Income vs. Expense comparison charts**
- **Enable filtering by date range or category using Streamlit widgets (st.date_input, st.selectbox).**

3. Dashboard View

- **Display an interactive Streamlit dashboard containing:**
  - **Recent Transactions Table**
  - **Pie Chart (category-wise spending)**
  - **Bar Chart (monthly summary)**
  - **KPIs showing total income, total expense, and balance**
- **Use Plotly or Matplotlib for data visualization.**

Component	Technology
Web Framework	Streamlit
Data Processing	Pandas
NLP / Categorization	Simple Keyword Matching / NLTK
Visualization	Plotly / Matplotlib / Seaborn
Database	SQLite / MySQL

Deliverables

- **Categorization logic for transactions.**
- **Streamlit dashboard showing spending insights.**
- **Interactive charts and summaries.**
- **Filterable reports by date and category.**

Expected Output

- **Transactions are auto-categorized and editable.**
- **Dashboard displays total income, expenses, and category spending visually.**
- **Reports update dynamically as new transactions are added.**

Table	Columns
users	id (INT, PK), name (TEXT), email (TEXT), password_hash (TEXT), created_at (DATETIME)
transactions	id (INT, PK), user_id (INT, FK), date (DATE), description (TEXT), amount (FLOAT), type (TEXT), category (TEXT)

Milestone	Duration	Focus
1	Weeks 1–2	Streamlit Authentication + Manual Input
2	Weeks 3–4	Categorization + Reporting + Dashboard