Technical Document – EvenUp

Project Name

EvenUp

General Objective

To design and implement a collaborative web application that enables users to manage shared expenses, balances, and pending payments in a fair, transparent, and simple way, facilitating financial organization within groups.

Specific Objectives

- Develop a RESTful API to handle users, groups, expenses, and payments through a backend built with Node.js and Express.
- Design and implement a relational PostgreSQL database to store structured information about users, groups, expenses, and balances.
- Integrate authentication and security using JSON Web Tokens (JWT) and password encryption.
- Build a responsive and user-friendly web interface with HTML, CSS (Bulma), and JavaScript.
- Deploy the project in a publicly accessible environment (e.g., Render, Netlify, GitHub Pages) to ensure availability for testing and real usage.

Problem Statement

In daily life, it is common for groups of people (friends, co-workers, or family members) to engage in activities that involve shared expenses such as trips, dinners, or collaborative projects. Managing these expenses manually often leads to confusion, mistakes, and conflicts, since it is not always clear who paid, how much each person owes, or what the final balance is for every member.

Although applications like Splitwise exist, they often present limitations in their free plans, are not tailored to specific user needs, or are just no intuitive.

This creates the need for EvenUp, a tailored solution that allows users to register expenses, assign participants, automatically calculate balances, and record payments within a simple and intuitive interface.

Project Scope

- Implementation of a backend in Node.js with Express to manage the logic for users, groups, expenses, payments, and balances.
- Design of a PostgreSQL database ensuring referential integrity, scalability, and traceability of information.
- Construction of a responsive web frontend using HTML, Bulma CSS, and JavaScript.
- Secure authentication and authorization with JWT, including session handling and basic user roles.
- Automatic balance calculations to determine debts and credits among group members.
- Deployment of the system in a publicly accessible environment, allowing evaluators and users to test the application without local setup.

User stories

- As a **user**, I want to log in with my username so that I can securely access my account.
- As a **user**, I want to recover my password so that I can regain access if I forget it.
- As a **user**, I want the app to be secure so no one can have access to my information without logging in.
- As a **developer**, I want the system to generate JWT tokens so that user sessions remain secure and valid.
- As a **user**, I want to create a group so that I can organize shared expenses with friends or colleagues.
- As a **user**, I want to be added to an existing group so that I can participate in its shared expenses.
- As a **group admin**, I want to edit group details (name, members) so that the group stays up to date.
- As a **group admin**, I want to delete a group so that inactive or unnecessary groups are removed.
- As a **user**, I want to view all my groups so that I can easily navigate between them.
- As a **group member**, I want to register a new expense so that the system updates the balances of all members.
- As a **group member**, I want to delete an expense so that invalid or mistaken records can be corrected.
- As a **group member**, I want to update an expense so that corrections or changes are reflected.
- As a **user**, I want to see a history of expenses so that I can track financial activity in the group.
- As a **developer**, I want the database to update balances automatically with triggers so that consistency is ensured without manual intervention.
- As a **user**, I want to register a payment to another group member so that the system reduces my debt.
- As a **user**, I want to edit payment details so that mistakes can be fixed.

- As a **user**, I want to delete a payment record so that invalid transactions are removed.
- As a **user**, I want to switch between light and dark themes so that I can choose the visual style that suits me best.
- As a **developer**, I want to define the workflow of the database so that it supports the application logic effectively.
- As a **developer**, I want to design the ER diagram so that the database structure is clear and well-documented.
- As a **developer**, I want to implement the database schema in SQL so that it is ready to be used by the backend.
- As a **developer**, I want to test database endpoints with Postman so that I can ensure the API works as expected.
- As a **developer**, I want to set up the frontend structure so that the web interface is organized and scalable.
- As a **developer**, I want to set up the backend structure so that the application logic is modular and maintainable.

Scrum methodology

Product owner: Luis Fernando Rodríguez López
Scrum Master: Luis Fernando Martínez Cervantes

- **Programmers:** Luis Fernando Rodríguez López - Luis Fernando Martínez Cervantes **-** Susana Silva Vallejo - José Tomás Loaiza Rodríguez - Samuel Rosero Álvarez