The Expense Management System is a web application built with Angular that helps users manage shared expenses within groups. It allows users to create groups, add members, add expenses, and view all the groups they are part of. The system automatically calculates and shows how much the logged-in user owes ensuring transparency in expense sharing. The system integrates authentication with **JWT tokens** to ensure secure access, supporting both **admin** and **user** roles, and exceptions are managed using an **HTTP interceptor** for reliable error handling.

**Objective:**  
To build a secure and user-friendly expense management system that enables users to manage group expenses, track balances, and ensure transparency through role-based access and efficient expense calculation.

**2. Architecture Overview**

The app is structured in a feature-first style with clear separation of concerns:

* **Auth** → JWT-based authentication, middleware for protecting routes.
* **Backend** → server.js with Express API, authMiddleware.js for security and serverData.json for mock persistence.
* **Group** → Create groups and list all groups the user is part of.
* **Expense** → Add expenses within groups, calculate balances (owed/owes).
* **Core** → HTTP interceptor for exception handling, guards for authentication.
* **Shared** → Common Angular components, services, and utilities reused across modules.
* **Build/Deploy** → Production build artifacts inside dist/expense-tracking-project.
* **PWA** → Configured with ngsw-config.json for service worker support.

**Routing:** Routes cover authentication, group listing, and expense logging. Guards ensure authentication before accessing feature modules. Backend middleware enforces token-based security on API endpoints.

## 3. Key Features

* **Authentication & Authorization**
* Secure login & registration system.
* Guards enforce route protection for authenticated users only.
* **Group Management**
* Create groups and invite members.
* View all groups a user belongs to.
* **Dynamically add members** during or after group creation.
* **Expense Tracking**
* Add expenses with description, amount, and payer.
* Automatically split expenses among group members.
* **Balance & Settlements**
* Real-time calculation of who owes whom.
* **Dashboard**
* Displays an overview of all groups a user is part of, along with quick access to group details.
* **Reusable Components (Shared Module)**
* Custom UI components: forms, buttons, headers.
* Pipes & utilities for consistent data formatting
* **Caching Mechanism**
* Implements local caching to reduce redundant API calls and improve application performance.

## 4. Technical Highlights

* **Angular 18 with Lazy-Loaded Routes** – Offers modularity and performance optimization.
* **Centralized Type Safety** – Comprehensive data models (such as User, Group, Expense, Settlement) organized in a /models folder to ensure consistency.
* **Authentication & Guards** – Secure JWT authentication with route guards ensuring only authorized users access specific modules.
* **Dynamic Member Management** – Allow users to seamlessly add members to groups at runtime.