INSURANCE MANAGEMENT SYSTEM

This project is a role-based Insurance Management System built using Angular as the frontend framework and JSON Server as a mock backend.
The system is designed with a layered architecture, separating UI, business logic, data access, and backend simulation.

The application supports three main user roles:

- Admin
- Agent
- Customer

Each role has controlled access to features using authentication, authorization, and route protection.

🔹 How the system works (end-to-end)

- Users authenticate using a mock JWT-based login system
- Role-based routing directs users to their respective dashboards
- Business logic is handled through Angular services
- Data is fetched and persisted using JSON Server via REST APIs
- Middleware-like behavior is achieved using Angular guards, interceptors, and services
- State is maintained using RxJS and localStorage
- Dashboards, analytics, and reports are dynamically updated based on data changes

Team wise work

TEAM MEMBER 1 (Sai Srihitha)

Module Name: Authentication & Admin Core Module

Role Description

Responsible for application structure, authentication flow, admin access, and overall integration of all modules.

What to Do

- Design overall application routing structure
- Set up role-based navigation for Admin, Agent, and Customer
- Implement login and registration for Agent and Customer
- Validate user credentials using JSON Server
- Implement mock JWT authentication on the frontend
- Store authentication state using `localStorage` (Remember Me)
- Handle role-based redirection after login
- Create route guards to protect secure pages
- Set up shared services for authentication and user state
- Create Admin Dashboard layout
- Display system overview data:
    - Total users
    - Total policies
    - Total claims
- Add navigation links from Admin Dashboard to:
    - Policy management
    - Claims management
- Handle final integration of all team modules
- Resolve merge conflicts and manage final Git integration

TEAM MEMBER 2 (Thanmai)

Module Name: Policy Management Module

Role Description

Responsible for managing the complete policy lifecycle, from viewing policies to administering them.

What to Do

- Display all available insurance policies for Customers and Agents

- Implement policy filtering based on:
    - Policy type
    - Premium range
    - Coverage amount
- Create a policy details view page
- Implement policy purchase flow:
    - Policy selection
    - Simple purchase form
    - Premium calculation logic
    - Assign purchased policy to customer
- Create Admin policy management screens
- Implement policy CRUD operations:
    - Add new policy
    - Edit existing policy
    - Delete policy
- Add UI-level renewal reminder indicators for expiring policies
- Coordinate with Auth module for admin-only access

TEAM MEMBER 3 (Koushik)

Module Name: Claims Processing Module

Role Description

Handles claim initiation by customers, claim tracking, and approval workflows for Admin and Agent roles.

What to Do

- Create claim filing form for Customers
- Allow customers to:
    - Select policy
    - Auto-fill customer and policy details
    - Enter claim amount and description
    - Submit claim request
- Store claim data using JSON Server
- Implement claim history view for customers
- Display claim status clearly:
    - Pending
    - Approved
    - Rejected
- Create claims review interface for Admin and Agent
- Allow Admin/Agent to:
    - View all submitted claims
    - Approve or reject claims

    - Add remarks for decisions
    - Update claim status
- Ensure claims update reflects correctly across dashboards

TEAM MEMBER 4(Umamaheswar)

Module Name: Customer & Agent Dashboard Module

Role Description

Manages dashboards and basic analytics for customers and agents.

What to Do

- Create Customer Dashboard
- Display customer-related data:
    - Active policies
    - Claim history
    - Payment status (static display)
- Create Agent Dashboard
- Display agent-related data:
    - Assigned customers
    - Policies sold by the agent
- Implement commission calculation logic based on number of policies sold
- Create basic reports and analytics views:
    - Policy sales summary
    - Claims count by status
    - Agent performance table
- Ensure dashboards update correctly when policies or claims change
- Work closely with Policy and Claims modules for data consistency