Project Abstract: InsuraPro - Insurance Management Platform Using Spring Boot Microservices

The **InsuraPro** project aims to develop a comprehensive **insurance management platform** using **Spring Boot Microservices**, designed to provide users with a seamless experience in managing insurance policies, filing claims, and accessing real-time updates. The platform will offer services for various types of insurance, including health, life, automobile, and property insurance.

Key Features:

- **User Registration & Authentication**: Secure user registration, login, and profile management, with JWT role based authentication to distinguish between users
- **Policy Management**: Users can browse available insurance products, view policy details, purchase or renew insurance policies, and download policy documents.
- **Premium Payments**: Integration with payment gateways for users to pay premiums online through multiple payment options such as credit/debit cards, net banking, and digital wallets.
- **Claim Filing & Tracking**: Users can easily file claims for different types of insurance policies and track the claim status in real time.
- **Document Management**: A secure document repository for storing and managing insurance-related documents, such as policy papers, claims, and verification documents.
- **Policy Recommendations**: Personalized policy recommendations based on user needs and behavior, leveraging AI to suggest the best insurance plans.
- **Notifications & Reminders**: Automated notifications for policy renewal, premium payment deadlines, and claim status updates via email/SMS.
- **Agent Support**: A section for users to connect with insurance agents for support, queries, and additional services.

Architecture:

The application is built using **Spring Boot Microservices**, ensuring modularity, scalability, and flexibility:

- **User Service**: Handles user registration, authentication, and profile management.
- **Policy Service**: Manages insurance policies, including details, purchase, and renewals.
- **Payment Service**: Integrates with payment gateways to handle premium payments and renewals.
- **Claim Service**: Enables users to file, manage, and track insurance claims.
- **Notification Service**: Sends automated notifications and reminders.
- **Recommendation Service**: Uses machine learning to offer personalized policy suggestions.
- **API Gateway**: Manages external requests and routes them to the respective microservices.

Technology Stack:

- **Spring Boot 3.3.2** for microservice architecture and REST APIs.
- **Spring Security** for user authentication and role-based access.
- **MySQL/PostgreSQL** for relational data and **MongoDB** for unstructured document storage.
- **Spring Cloud** for service discovery, load balancing, and resilience.
- **React.js/Angular** for a modern, user-friendly front-end interface (optional).

Frontend Layer (OJET)

- SPA built with **Oracle JET components**.
- Role-based UI (User, Agent, Admin).
- Integration with backend via **Spring Cloud API Gateway**.
- Visualization of policies, payments, and claim statuses using **oj-table, oj-chart, oj-messages, oj-form-layout**.

Backend Layer (Spring Boot Microservices)

- **User Service** → authentication, profile.
- **Policy Service** → policy CRUD, renewals.
- **Payment Service** → premium payments.
- **Claim Service** → filing & tracking.
- **Notification Service** → emails/SMS.
- **Recommendation Service** → ML-based suggestions.
- **API Gateway + Eureka** → routing & discovery.

Database Layer

- PostgreSQL/MySQL → policies, claims, users, payments.
- MongoDB → document repository (policy docs, claim papers).

◆ OJET UI Requirements (per Feature)

1. User Registration & Authentication

- **OJET Screens**:
  - Login/Signup form (`oj-form-layout`, `oj-input-text`, `oj-button`).
  - JWT-based role handling → dashboard view changes per role.
- **APIs**: User Service → `/auth/register`, `/auth/login`.

---

2. Policy Management

- **OJET Screens**:
  - Policy catalog (`oj-table`, `oj-data-grid`) with filters (type: health, life, auto, property).
  - Policy details page with "purchase" or "renew" buttons.

  - Policy download in PDF.
- **APIs**: Policy Service → `/policies/list`, `/policies/purchase`, `/policies/renew`.

---

3. Premium Payments

- **OJET Screens**:
  - Payment form (oj-input-number, oj-select-single for payment method).
  - Confirmation modal with transaction summary.
  - Payment history in **oj-table**.
- **APIs**: Payment Service → `/payments/initiate`, `/payments/status`.

---

4. Claim Filing & Tracking

- **OJET Screens**:
  - Claim filing wizard (oj-form-layout, oj-file-picker for document uploads).
  - oj-table for claim status tracking (Pending, Approved, Rejected).
  - oj-progress-circle for claim progress visualization.
- **APIs**: Claim Service → `/claims/file`, `/claims/status`.

---

5. Document Management

- **OJET Screens**:
  - Document upload form (`oj-file-picker`).
  - Document repository (`oj-list-view` or card layout).
- **APIs**: Document Storage → `/documents/upload`, `/documents/list`.

---

6. Policy Recommendations (AI-driven)

- **OJET Screens**:
  - Recommendations card carousel with suggested policies.
  - oj-chart showing comparison between user’s current policy vs suggested.
- **APIs**: Recommendation Service → `/recommendations/user/{id}`.

---

7. Notifications & Reminders

- **OJET Screens**:
  - Notifications bell with dropdown (`oj-messages`).
  - Alert banners for upcoming premium deadlines.

- **APIs**: Notification Service → `/notifications/recent`.

---

8. Agent Support

- **OJET Screens**:
  - Chat/Contact agent form.
  - oj-table listing assigned agents.
- **APIs**: Agent/Support Service → `/agents/list`, `/agents/contact`.

◆ Extra Technical Stories (OJET Frontend)

- **US901**: As a user, I want a **dashboard in OJET** showing my policies, claims, and payments in one place.
- **US902**: As a user, I want to **track my claims visually** with progress indicators.
- **US903**: As an agent, I want an OJET view of **assigned clients and their policies**, so I can manage them easily.
- **US904**: As an admin, I want an **OJET analytics dashboard** for system usage, claims volume, and payments.