**InsurAI Corporate Policy Automation and Intelligence System**

**Problem Statement**

In many insurance companies, corporate policy management is highly manual and prone to errors. Employees and managers face challenges such as:  
- Manual tracking of policy creation, renewal, and compliance  
- Delays in approvals and notifications  
- Difficulty in monitoring policy performance and employee adherence  
- Lack of analytics for decision-making  
  
These challenges lead to inefficiency, higher operational costs, and compliance risks.  
  
Solution: InsurAI automates and centralizes corporate insurance policy management while providing intelligent insights for decision-making, reducing errors and saving time.

**Features**

**User Management**

- Roles: Admin, Manager, Employee  
- Capabilities: Role-based dashboards, policy access, approval privileges  
- Authentication: JWT-based or Spring Security

**Policy**

- Create, edit, and delete corporate policies  
- Assign policies to departments or individuals  
- Track policy status (draft, pending approval, active, expired)

**Automation**

- Automated reminders for policy renewal  
- Notifications for approval or policy expiration  
- Scheduled reporting to management

**Analysis**

- Dashboard with policy stats: active, expired, pending  
- Department-wise compliance reports  
- Predictive analytics for policy renewal and risk assessment

**Security**

- Log all actions for audit trails  
- Data encryption at rest and in transit  
- Role-based access ensures sensitive information is protected

**System Architecture**

InsurAI follows a modern 3-tier architecture:

Frontend (React)

- Provides interactive user interfaces  
- Uses components for modular design  
- State management via Context API or Redux  
- API integration with backend using Axios Backend (Spring Boot)

- Implements business logic  
- Provides REST APIs for CRUD operations  
- Handles authentication, authorization, and validation  
- Integrates with MySQL for data storage

Database (MySQL)

- Stores users, policies, roles, and logs  
- Supports transactional operations  
- Enables data analytics queries

**Workflow**

1. Admin: Creates policies, manages users, monitors compliance  
2. Manager: Approves/rejects policies assigned to their department  
3. Employee: Views assigned policies, receives reminders, acknowledges compliance  
4. Automated System: Sends reminders and generates reports  
5. Analytics Engine: Aggregates policy data for management dashboards

**Approach to Development**

1. Requirement Analysis – Understand business needs and policy workflows  
   2. System Design – Define architecture, database schema, API endpoints  
   3. Backend Development – Implement business logic, API endpoints, database integration  
   4. Frontend Development – Build dashboards, forms, and views for each role  
   5. Integration & Testing – Connect frontend and backend, perform unit and integration tests  
   6. Deployment – Deploy on a server or cloud (optional Docker/Kubernetes)  
   7. Training & Documentation – Provide manuals and conduct sessions for end-users

**Benefits**

- Reduced manual effort and errors  
- Faster policy approvals and renewals  
- Real-time visibility into compliance  
- Improved decision-making through analytics  
- Scalable system for future expansion

**Future Enhancements**

- AI-powered policy recommendations  
- Integration with ERP and insurance provider APIs  
- Mobile app for employees to view policies on-the-go  
- Predictive analytics

insurai-corporate-policy-system/

├── backend/

│ ├── src/

│ │ ├── main/

│ │ │ ├── java/

│ │ │ │ └── com/

│ │ │ │ └── insurai/

│ │ │ │ ├── InsurAIApplication.java

│ │ │ │ ├── config/

│ │ │ │ │ ├── SecurityConfig.java

│ │ │ │ │ ├── WebConfig.java

│ │ │ │ │ ├── SwaggerConfig.java

│ │ │ │ │ └── ModelMapperConfig.java

│ │ │ │ ├── controller/

│ │ │ │ │ ├── PolicyController.java

│ │ │ │ │ ├── ClaimController.java

│ │ │ │ │ ├── CustomerController.java

│ │ │ │ │ ├── DashboardController.java

│ │ │ │ │ ├── AuthController.java

│ │ │ │ │ └── AIController.java

│ │ │ │ ├── service/

│ │ │ │ │ ├── PolicyService.java

│ │ │ │ │ ├── ClaimService.java

│ │ │ │ │ ├── CustomerService.java

│ │ │ │ │ ├── DashboardService.java

│ │ │ │ │ ├── AuthService.java

│ │ │ │ │ ├── AIService.java

│ │ │ │ │ ├── EmailService.java

│ │ │ │ │ └── ReportService.java

│ │ │ │ ├── repository/

│ │ │ │ │ ├── PolicyRepository.java

│ │ │ │ │ ├── ClaimRepository.java

│ │ │ │ │ ├── CustomerRepository.java

│ │ │ │ │ ├── UserRepository.java

│ │ │ │ │ └── AuditLogRepository.java

│ │ │ │ ├── model/

│ │ │ │ │ ├── Policy.java

│ │ │ │ │ ├── Claim.java

│ │ │ │ │ ├── Customer.java

│ │ │ │ │ ├── User.java

│ │ │ │ │ ├── AuditLog.java

│ │ │ │ │ ├── RiskAssessment.java

│ │ │ │ │ ├── PremiumCalculation.java

│ │ │ │ │ └── dto/

│ │ │ │ │ ├── PolicyDTO.java

│ │ │ │ │ ├── ClaimDTO.java

│ │ │ │ │ ├── CustomerDTO.java

│ │ │ │ │ ├── LoginRequest.java

│ │ │ │ │ ├── AuthResponse.java

│ │ │ │ │ └── DashboardStats.java

│ │ │ │ ├── exception/

│ │ │ │ │ ├── ResourceNotFoundException.java

│ │ │ │ │ ├── CustomExceptionHandler.java

│ │ │ │ │ └── ErrorResponse.java

│ │ │ │ ├── security/

│ │ │ │ │ ├── JwtUtil.java

│ │ │ │ │ ├── JwtRequestFilter.java

│ │ │ │ │ ├── UserDetailsServiceImpl.java

│ │ │ │ │ └── SecurityConstants.java

│ │ │ │ ├── util/

│ │ │ │ │ ├── PolicyNumberGenerator.java

│ │ │ │ │ ├── ClaimNumberGenerator.java

│ │ │ │ │ ├── DateUtils.java

│ │ │ │ │ └── ValidationUtils.java

│ │ │ │ └── ai/

│ │ │ │ ├── RiskAssessor.java

│ │ │ │ ├── FraudDetector.java

│ │ │ │ ├── PremiumCalculator.java

│ │ │ │ └── NLPProcessor.java

│ │ │ └── resources/

│ │ │ ├── application.properties

│ │ │ ├── application-dev.properties

│ │ │ ├── application-prod.properties

│ │ │ ├── data.sql

│ │ │ ├── schema.sql

│ │ │ └── logback-spring.xml

│ │ └── test/

│ │ └── java/

│ │ └── com/

│ │ └── insurai/

│ │ ├── service/

│ │ │ ├── PolicyServiceTest.java

│ │ │ ├── ClaimServiceTest.java

│ │ │ └── AIServiceTest.java

│ │ ├── controller/

│ │ │ ├── PolicyControllerTest.java

│ │ │ └── AuthControllerTest.java

│ │ └── InsurAIApplicationTests.java

│ ├── pom.xml

│ ├── Dockerfile

│ └── docker-compose.yml

├── frontend/

│ ├── public/

│ │ ├── index.html

│ │ ├── favicon.ico

│ │ ├── manifest.json

│ │ └── robots.txt

│ ├── src/

│ │ ├── components/

│ │ │ ├── common/

│ │ │ │ ├── Header/

│ │ │ │ │ ├── Header.js

│ │ │ │ │ └── Header.css

│ │ │ │ ├── Sidebar/

│ │ │ │ │ ├── Sidebar.js

│ │ │ │ │ └── Sidebar.css

│ │ │ │ ├── Footer/

│ │ │ │ │ ├── Footer.js

│ │ │ │ │ └── Footer.css

│ │ │ │ ├── LoadingSpinner/

│ │ │ │ │ ├── LoadingSpinner.js

│ │ │ │ │ └── LoadingSpinner.css

│ │ │ │ ├── Modal/

│ │ │ │ │ ├── Modal.js

│ │ │ │ │ └── Modal.css

│ │ │ │ └── Notification/

│ │ │ │ ├── Notification.js

│ │ │ │ └── Notification.css

│ │ │ ├── policies/

│ │ │ │ ├── PolicyList/

│ │ │ │ │ ├── PolicyList.js

│ │ │ │ │ └── PolicyList.css

│ │ │ │ ├── PolicyForm/

│ │ │ │ │ ├── PolicyForm.js

│ │ │ │ │ └── PolicyForm.css

│ │ │ │ ├── PolicyDetails/

│ │ │ │ │ ├── PolicyDetails.js

│ │ │ │ │ └── PolicyDetails.css

│ │ │ │ └── PolicyCard/

│ │ │ │ ├── PolicyCard.js

│ │ │ │ └── PolicyCard.css

│ │ │ ├── claims/

│ │ │ │ ├── ClaimList/

│ │ │ │ │ ├── ClaimList.js

│ │ │ │ │ └── ClaimList.css

│ │ │ │ ├── ClaimForm/

│ │ │ │ │ ├── ClaimForm.js

│ │ │ │ │ └── ClaimForm.css

│ │ │ │ ├── ClaimDetails/

│ │ │ │ │ ├── ClaimDetails.js

│ │ │ │ │ └── ClaimDetails.css

│ │ │ │ └── ClaimCard/

│ │ │ │ ├── ClaimCard.js

│ │ │ │ └── ClaimCard.css

│ │ │ ├── customers/

│ │ │ │ ├── CustomerList/

│ │ │ │ │ ├── CustomerList.js

│ │ │ │ │ └── CustomerList.css

│ │ │ │ ├── CustomerForm/

│ │ │ │ │ ├── CustomerForm.js

│ │ │ │ │ └── CustomerForm.css

│ │ │ │ └── CustomerDetails/

│ │ │ │ ├── CustomerDetails.js

│ │ │ │ └── CustomerDetails.css

│ │ │ ├── dashboard/

│ │ │ │ ├── StatsCards/

│ │ │ │ │ ├── StatsCards.js

│ │ │ │ │ └── StatsCards.css

│ │ │ │ ├── Charts/

│ │ │ │ │ ├── PremiumChart.js

│ │ │ │ │ ├── ClaimsChart.js

│ │ │ │ │ └── Charts.css

│ │ │ │ ├── RecentActivity/

│ │ │ │ │ ├── RecentActivity.js

│ │ │ │ │ └── RecentActivity.css

│ │ │ │ └── AIRecommendations/

│ │ │ │ ├── AIRecommendations.js

│ │ │ │ └── AIRecommendations.css

│ │ │ └── ai/

│ │ │ ├── RiskAssessment/

│ │ │ │ ├── RiskAssessment.js

│ │ │ │ └── RiskAssessment.css

│ │ │ ├── FraudDetection/

│ │ │ │ ├── FraudDetection.js

│ │ │ │ └── FraudDetection.css

│ │ │ └── ChatAssistant/

│ │ │ ├── ChatAssistant.js

│ │ │ └── ChatAssistant.css

│ │ ├── pages/

│ │ │ ├── Login/

│ │ │ │ ├── Login.js

│ │ │ │ └── Login.css

│ │ │ ├── Dashboard/

│ │ │ │ ├── Dashboard.js

│ │ │ │ └── Dashboard.css

│ │ │ ├── Policies/

│ │ │ │ ├── Policies.js

│ │ │ │ └── Policies.css

│ │ │ ├── Claims/

│ │ │ │ ├── Claims.js

│ │ │ │ └── Claims.css

│ │ │ ├── Customers/

│ │ │ │ ├── Customers.js

│ │ │ │ └── Customers.css

│ │ │ ├── Reports/

│ │ │ │ ├── Reports.js

│ │ │ │ └── Reports.css

│ │ │ ├── Settings/

│ │ │ │ ├── Settings.js

│ │ │ │ └── Settings.css

│ │ │ └── NotFound/

│ │ │ ├── NotFound.js

│ │ │ └── NotFound.css

│ │ ├── services/

│ │ │ ├── api.js

│ │ │ ├── auth.js

│ │ │ ├── policyService.js

│ │ │ ├── claimService.js

│ │ │ ├── customerService.js

│ │ │ ├── dashboardService.js

│ │ │ ├── aiService.js

│ │ │ └── reportService.js

│ │ ├── utils/

│ │ │ ├── auth.js

│ │ │ ├── helpers.js

│ │ │ ├── constants.js

│ │ │ ├── validators.js

│ │ │ └── formatters.js

│ │ ├── contexts/

│ │ │ ├── AuthContext.js

│ │ │ └── AppContext.js

│ │ ├── hooks/

│ │ │ ├── useAuth.js

│ │ │ ├── useApi.js

│ │ │ └── useForm.js

│ │ ├── styles/

│ │ │ ├── index.css

│ │ │ ├── theme.js

│ │ │ └── variables.css

│ │ ├── App.js

│ │ ├── App.css

│ │ ├── App.test.js

│ │ ├── index.js

│ │ └── store/

│ │ ├── index.js

│ │ ├── actions/

│ │ │ ├── authActions.js

│ │ │ ├── policyActions.js

│ │ │ ├── claimActions.js

│ │ │ └── customerActions.js

│ │ ├── reducers/

│ │ │ ├── authReducer.js

│ │ │ ├── policyReducer.js

│ │ │ ├── claimReducer.js

│ │ │ └── customerReducer.js

│ │ └── store.js

│ ├── package.json

│ ├── package-lock.json

│ ├── Dockerfile

│ ├── nginx.conf

│ └── docker-compose.yml

├── database/

│ ├── init.sql

│ ├── sample-data.sql

│ └── migrations/

│ ├── V1\_\_initial\_schema.sql

│ ├── V2\_\_add\_ai\_features.sql

│ └── V3\_\_add\_audit\_tables.sql

├── documentation/

│ ├── API\_Documentation.md

│ ├── User\_Guide.md

│ ├── Setup\_Guide.md

│ ├── Architecture.md

│ └── Database\_Schema.md

├── scripts/

│ ├── deploy.sh

│ ├── backup-db.sh

│ ├── restore-db.sh

│ └── test-api.sh

├── .gitignore

├── README.md

├── LICENSE

└── docker-compose.yml  
  
  
 **Foundation & Setup**

Initialize Project Structure

Set Up Backend (Spring Boot)

Set Up Frontend (React)

Configure Development Environment

Set Up Database

**Core Backend Development**

Design and Implement Database Schema

Create Data Models (JPA Entities)

Implement Repository Layer

Develop Service Layer with Business Logic

Build REST API Controllers

Implement Security & Authentication (JWT)

Test API Endpoints

**Core Frontend Development**

Set Up State Management (Redux)

Create API Service Layer

Implement Application Routing & Auth Guards

Develop Reusable UI Components

Build Main Application Pages

Connect Frontend to Backend API

**AI & Advanced Features Integration**

Develop Backend AI Service Layer

Create AI API Endpoints

Integrate AI into Business Logic

Build Frontend AI Components

Implement Additional Features (Reports, Dashboard Charts)

**Polish, Test, and Deploy**

Implement Comprehensive Error Handling

Write Unit and Integration Tests

Refine UI/UX and Styling

Dockerize Application

Deploy to Production Environment