Project Title: InsurAI – Corporate Policy Automation and Intelligence System
Name: Samir Shaikh

Sprint Backlog \ Execution Plan Document

1. Introduction

1.1 Purpose

The purpose of this document is to define the Sprint Backlog and Execution Plan for the InsurAI – Corporate Policy Automation and Intelligence System. This plan outlines how the project will be developed incrementally using Agile methodology, dividing work into manageable sprints with clearly defined goals, tasks, and deliverables.

1.2 Agile Approach

The InsurAI project follows an Agile-based incremental development approach. The complete system is divided into multiple sprints, where each sprint delivers a functional and testable component of the system. Continuous feedback and iterative improvement are emphasized throughout the development lifecycle.

2. Sprint Overview

Sprint	Duration	Focus Area
Sprint 1	1 Week	Project Setup & Core Backend Foundation
Sprint 2	1 Week	Authentication & Authorization
Sprint 3	1 Week	Policy Management Module
Sprint 4	1 Week	Claims Processing Module
Sprint 5	1 Week	Frontend Integration & Dashboards
Sprint 6	1 Week	AI Assistant, Testing & Deployment

3. Sprint-wise Backlog & Execution Plan

Sprint 1: Project Setup & Core Foundation

Sprint Goal

To establish the foundational structure of the InsurAI project, including development environment setup, project configuration, and database connectivity.

Sprint Backlog

- Create Spring Boot backend project using Maven
- Configure MySQL database connection
- Set up basic project folder structure
- Configure JPA and Hibernate
- Create initial database schema
- Initialize Git repository and version control

Deliverables

- Running Spring Boot application
- Successful database connection
- Initial project pushed to Git repository

Sprint 2: Authentication & Authorization

Sprint Goal

To implement secure user authentication and authorization using JWT-based security.

Sprint Backlog

- Design user and role entities
- Implement user registration API
- Implement login API with JWT token generation
- Configure Spring Security

- Implement role-based access control
- Test secured APIs using Postman

Deliverables

- Secure authentication APIs
- JWT-based access control
- Role-protected endpoints

Sprint 3: Policy Management Module

Sprint Goal

To enable administrators to manage insurance policies and allow customers to view policy information.

Sprint Backlog

- Create policy entity and repository
- Implement CRUD APIs for policies
- Implement policy search and listing APIs
- Validate inputs and handle exceptions
- Test policy APIs

Deliverables

- Fully functional policy management APIs
- Policy data stored and retrieved from database

Sprint 4: Claims Processing Module

Sprint Goal

To digitize the insurance claims process for customers and administrators.

Sprint Backlog

- Design claim entity and relationships
- Implement claim submission API
- Implement claim status tracking API
- Implement admin approval/rejection APIs
- Integrate email notifications for claim updates

Deliverables

- Claims submission and tracking feature
- Email notifications for claim status changes

Sprint 5: Frontend Integration & Dashboards

Sprint Goal

To integrate frontend React application with backend APIs and provide role-based dashboards.

Sprint Backlog

- Create React project structure
- Implement authentication UI
- Integrate login and registration APIs
- Build customer dashboard
- Build admin dashboard
- Integrate policy and claim APIs

Deliverables

- Functional frontend application
- Role-based dashboards
- API-integrated UI

Sprint 6: AI Assistant, Testing & Deployment

Sprint Goal

To integrate AI assistance, perform system testing, and prepare the application for deployment.

Sprint Backlog

- Integrate AI API for policy-related queries
- Implement AI chat interface in frontend
- Perform unit and integration testing
- Fix bugs and improve performance
- Prepare deployment configuration
- Create final project documentation

Deliverables

- AI-powered assistance feature
- Tested and stable application
- Deployment-ready build
- Final documentation

4. Definition of Done (DoD)

A sprint is considered complete when:

- All planned backlog items are implemented
- Code is tested and functional
- No critical bugs remain
- Features are integrated into the main branch
- Sprint deliverables are reviewed

5. Risks and Mitigation

Risk	Impact	Mitigation
Limited development time	High	Prioritize core features
AI API dependency	Medium	Use fallback responses
Integration issues	Medium	Early API testing