DURATION: 4 HOURS

DEVELOP SOFTWARE USING ARTIFICIAL INTELLIGENCE (AI) TOOLS/AI Agents FOR THE FOLLOWING USER STORY

# User Story

## Agent Performance & Commission Tracking System

As an insurance company administrator, I want a centralized system that records agent sales, calculates commissions based on predefined rules, and displays performance metrics, so that I can ensure transparency in payouts, motivate agents, and reduce manual errors.

## Steps

**Collect Agent Data**: Gather agent profiles (region, targets), sales records, and policy details.

**Commission Calculation Logic:**

- Define rules for commission based on policy type, premium amount, and tenure.
- Implement a rule engine for automated calculations.

**Performance Tracking:**

- Record every policy sold and link to the respective agent.
- Track KPIs and commission earned.

**Dashboard & Reporting:**

- Provide dashboards for agents and administrators.
- Enable data export for audits.

**User Access:**

- Login screens for administrators and agents.
- Role-based access to features.

## Data Requirements

- Agent profiles (name, region, targets)
- Policy sales data (policy type, premium, tenure)
- Commission rules and calculation criteria
- Performance metrics (KPIs, sales targets)
- Audit logs and exportable reports

## Expected Output

- Centralized agent management system
- Automated commission calculation and transparent payout reports
- Dashboards for agent KPIs and performance
- Exportable data for audits
- Reduced manual errors and improved agent motivation

## Key Features

1. Login screen for administrator & agents.
2. The system should allow adding and managing agent profiles with details like region and targets.
3. It should record every policy sold and link it to the respective agent.
4. Rule Engine for Commission calculations.
5. Commission calculation must follow predefined rules for policy type, premium amount, and tenure.
6. Agents should have access to a dashboard showing their KPIs and commission earned.
7. Data export functionality should be available for audits.

# Deliverables

## All the deliverables are expected to be of production-level quality, ensuring excellence in both content and look & feel.

1. Functional requirements document
2. Application Architecture and Architecture document
3. Database design / Data Models – No JSON input of data, use any RDBMS, deliverables include logical model, physical model and ER diagram
4. UI/Ux/Wireframe Design
5. Software code & Working Software
6. Functional Test scripts in a CSV or Excel
7. Traceability Matrix in CSV or Excel
8. End User manual - Screens and help guide
9. AI Agent – **"Code Quality Checker"** built with an AI framework and Streamlit or Gradio; takes code and checklist as input and provide observations against each checklist item and suggestions for improvements if any

# Instructions

**Use AI tools or Build Your Own AI Agents (BYOAA) to develop the software and deliverables**.

- **Selection of Tools:** Carefully evaluate and select the most suitable AI tools or frameworks for each phase of the project. If building custom AI agents, ensure they are scalable, maintainable. Additional weightage will be given if team builds their own AI agent to create the deliverables.
- **Quality Standards:** Adhere strictly to coding standards and best practices for software development. Apply consistent design patterns, code reviews, and automated testing to maintain high quality.
- **Integration:** Ensure seamless integration of components with the overall application architecture.
- **Testing:** Prepare comprehensive test cases, including both functional and non-functional scenarios.
- **Documentation:** Maintain clear and detailed documentation for all deliverables, including architecture diagrams, data models, and user manuals. Update documents regularly as changes are made.
- **User Experience:** Prioritise a user-centric approach in UI/UX design. Avoid using Streamlit or Gradio for the user interface.