# **📌 Case Study: InsureConnect**

*(A Digital Insurance Marketplace – inspired by RenewBuy)*

## **🎯 Application Goal**

A digital insurance marketplace where users (customers & agents) can:

* Compare **Motor, Health, and Life insurance policies**
* View **multiple quotes** from different insurers
* **Purchase policies** seamlessly
* Manage **renewals and claims assistance**

## **🛠 Tech Stack**

* **Backend:** Python + Django REST Framework (APIs)
* **Database:** PostgreSQL (structured data)
* **Caching:** Redis (fast insurer responses / session store)
* **Queue/Async:** Celery + RabbitMQ/Kafka (for heavy tasks like PDF generation, email/SMS)
* **Authentication:** JWT + OAuth (agents, customers)
* **Deployment:** Docker + AWS ECS (scalable microservices style)
* **Monitoring:** CloudWatch + ELK stack

## **📦 Core Modules**

### **1. Authentication & Authorization**

* Customer registration/login
* Agent login (separate role)
* JWT token-based authentication

### **2. Insurance Modules**

* **Motor Insurance:** Vehicle details → Quotes → Compare → Purchase → Policy PDF
* **Health Insurance:** Family details → Quotes → Compare → Purchase → Policy PDF
* **Life Insurance:** Term plan details → Quotes → Compare → Purchase → Policy PDF

### **3. Insurer Integrations (Adapter Pattern)**

* Different insurer APIs (HDFC, ICICI, Bajaj, etc.) return different formats
* Design a **common adapter interface** to normalize responses
* Use **mock APIs** for testing

### **4. Quote Engine**

* Takes customer input
* Calls all insurer adapters in **parallel**
* Aggregates & normalizes results
* Returns comparable quotes

### **5. Purchase & Policy Issuance**

* Customer selects a plan
* Make insurer API call (mocked)
* Generate policy document (**PDF**)
* Notify via **email/SMS**

### **6. Renewals & Reminders**

* Store expiry date in DB
* Send reminders (**async job with Celery**)

### **7. Admin & Analytics Dashboard**

* List policies sold (by insurer, premium, type)
* Track failed API calls
* Export reports

## **📊 Real-Time Use Cases**

1. **Customer buying car insurance**
   * Enters car number + details
   * Gets quotes from HDFC Ergo, ICICI Lombard, Bajaj Allianz
   * Compares premiums
   * Buys policy → PDF generated → Email sent
2. **Agent selling health insurance**
   * Logs in as agent
   * Helps customer enter details (age, family members)
   * Shows multiple insurer quotes (Star Health, Max Bupa, Care)
   * Customer selects plan
   * Policy purchased → Commission assigned to agent
3. **Life insurance purchase**
   * Customer enters details (income, age, smoker/non-smoker)
   * Fetches quotes (HDFC Life, ICICI Prudential, PNB MetLife)
   * Selects best plan
   * Buys → policy stored in DB
4. **Renewal reminder**
   * Motor insurance expiring in 7 days
   * Background task checks expiry
   * Sends automated **email + SMS**

## **📂 Implementation Roadmap**

1. Requirements (Q&A interview style)
2. High-Level Architecture (UML + Flow Diagram)
3. Database Schema Design (ER Diagram)
4. Authentication & User Roles (JWT)
5. Core Insurance Modules (Motor, Health, Life)
6. Insurer Integration Layer (Adapter Pattern)
7. Quote Engine (parallel calls + aggregator)
8. Purchase & Policy Issuance (mock APIs, PDF generation)
9. Renewal & Notification System (Celery + Redis)
10. Admin & Analytics Dashboard
11. Testing (unit tests, integration tests, mock insurer APIs)
12. Docker + Deployment (AWS ECS simulation)