Unit 8D, First Floor, Pangaea Office Park,

Plot No.2374, Off Great East Road,

Lusaka, Zambia

**Software Engineer Practical Assessment**

### Practical Task: Insurance Claim Processing System Prototype

#### **Problem Overview:**

In the insurance industry, managing claims is a critical process that requires speed, accuracy, and transparency. Your task is to create a **web-based insurance claim processing application** that streamlines claim submissions, provides real-time claim status updates, claims approvals and discharge process.

#### **Task Description:**

Design and develop a **web application prototype** that allows a user to:

1. **Submit an insurance claim** with required information (e.g., claim type, claimant details, description of the incident, supporting documents).
2. **View the claim status** (Pending, Under Review, Approved, Denied).
3. **Automation analysis**: Use AI to perform basic validation on the claim submission (e.g., checking if all necessary fields are filled, if the claim description sounds reasonable, etc.).
4. **Claim approval Maker-checker for the backend admins.**
5. **Discharge process to settle the insured or claimant.**
6. **Backend system** to manage claim data and allow updates in real-time (e.g., marking a claim as “approved” or “denied”).

### **Requirements:**

1. **Frontend**:
   * A simple user interface that allows a user to submit a claim.
   * Display the status of the claim and offer functionality for the user to track the claim's progress.
2. **Backend**:
   * Handle the storage of claim data (e.g., in a database or temporary data storage).
   * Implement an API that allows frontend and backend communication.
   * Manage different states of the claim (e.g., Pending, Under Review, Approved, Denied).
3. **AI Integration** (Optional but encouraged):
   * Implement an AI-based check (you can use basic natural language processing or machine learning models) to validate that the description of the claim appears reasonable or checks for missing information.
4. **Additional Functionality** (Optional but encouraged):
   * Use a simple algorithm to prioritize claims based on urgency (e.g., critical claims or high-value claims).
   * Enable claim status updates via a notification system (e.g., an email or SMS).

### **Tools You Can Use:**

* **Frontend**: React.js, Next.js, or any JavaScript framework of your choice.
* **Backend**: Node.js, Express.js, or any backend framework you prefer. You can also use cloud functions like AWS Lambda.
* **Database**: Any database solution (e.g., Postgres, MongoDB, MySQL) or even in-memory storage if it’s a small prototype.
* **AI/ML Tools**: You can integrate a basic **NLP model** (using tools like OpenAI's API, Hugging Face, or pre-trained models) to analyze claim descriptions and validate content.
* **Authentication**: Optional - Implement basic authentication (JWT, OAuth) to track users and their claims.

### **Submission Guidelines:**

* **Code**: Share the full source code of the application, along with instructions for how to run it locally or deploy it. (Git Repository to be provided)
* **Documentation**: Briefly explain the choices you made during the development process (tools, libraries, design patterns, etc.), and how you would improve the system further if given more time.
* **AI Integration**: If AI/ML tools were used, explain how the model works, its limitations, and how you might improve it in a real-world scenario.