# Build a full-stack web application that allows users to upload a **Dental X-ray DICOM image**

Company: Dobbe AI

Role: Data Science Intern

Duration: 3-6 months (Possibility of PPO depending upon performance)

# **Objective**

Build a full-stack web application that allows users to upload a **Dental X-ray DICOM image**, sends it to a **Roboflow object detection model**, displays **bounding boxes** for detected pathologies on the image, and passes the image + annotations to an **LLM** to generate a **textual diagnostic report**, which will be shown in a panel on the dashboard.

## **Tech Stack**

* **Backend**: FastAPI (Python)
* **Frontend**: ReactJS (preferred) or any SPA framework
* **Object Detection Model**: Roboflow API (cavities and periapical lesion detection)
* **LLM (Large Language Model)**: OpenAI GPT (or mock with dummy report generator if no key)
* **Image Format**: DICOM (.dcm)
* **Visualization**: Bounding boxes overlaid on original image
* **Packaging**: Docker (bonus)

## **Deliverables**

* GitHub repo with frontend and backend code.
* README with setup instructions.
* Backend must use FastAPI and expose required endpoints.

## **Sample Data**

Attached are a few sample **Dental DICOM (.dcm or .rvg)** X-ray images.

## **Functional Requirements**

### **1. Upload and Convert DICOM**

* Accept .dcm or .rvg file from frontend.
* Convert DICOM image to a viewable format (PNG or JPG).
* Display original image on the frontend.

### **2. Call Roboflow Model API**

Use the following Roboflow object detection model:

 [Roboflow Model](https://universe.roboflow.com/new-workspace-oorwh/adr/model/6)

* Endpoint: <https://detect.roboflow.com/adr/6>
* Required:
  + API key (Use your own free account )
  + confidence threshold: 30%
  + overlap threshold: 50%
* Pass the converted image to Roboflow API.
* Parse the annotations (bounding boxes, class labels, confidence scores).

### **3. Visualize Predictions**

* Draw bounding boxes on the image (with pathology name and confidence).
* Display annotated image on the main panel of the dashboard.

### **4. Generate Diagnostic Report with LLM**

* Pass image metadata + annotation JSON to an LLM (e.g., OpenAI GPT via API or simulate).
* Prompt Example:  
  You are a dental radiologist. Based on the image annotations provided below (which include detected pathologies), write a concise diagnostic report in clinical language.
* Output a brief paragraph highlighting:
  + Detected pathologies
  + Location if possible (e.g., upper left molar)
  + Clinical advice (optional)

### **5. Display Diagnostic Report**

* The LLM-generated report should be visible on the **right side of the dashboard**.

## 🖥️ **UI Suggestions**

Design a simple 2-panel UI:

+------------------------------------+------------------------+

| Left Panel (Image Viewer) | Right Panel (Report) |

| - Original or Annotated Image | - Diagnostic Report |

| - File Upload + Predict Button | |

+------------------------------------+------------------------+

## **Bonus**

* Dockerize the app (Dockerfile for backend and frontend)
* Add loading spinners for API calls
* Support multiple file uploads
* Unit tests for backend endpoints

## **Submission**

* Deadline: **2 days**
* Submit GitHub repo link
* Email subject: [Your Name] - Full Stack Dental X-ray Assignment