**Disease Detective Suite**

**Objective of the project** :-

Disease Detective Suite is a Web-Application that leverages multiple modalities for accurate and efficient diagnosis, it caters several use cases:

1. Early detection of common medical diseases based on patient records, including symptoms and demographic information
2. Detection of presence of skin-cancer from an image of skin provided by the user
3. Providing a descriptive summary from a patient’s blood report.

**Configuration of the project** :-

* Backend : Flask
* Frontend : React.js
* Technologies : Machine Learning, Deep Learning, Generative AI

**To start the project** :-

1. Go to the **Disease-Backend** directory, create a virtual environment , activate it and install all the dependencies mentioned in **requirements.txt** using command **‘pip install -r requirements.txt’**.After that, enter the command ‘**python app.py’** to start the flask backend.
2. Go to the **disease-app**, inside the **Disease-Frontend** directory ; install the required dependencies using ‘**npm install’** command. After that, enter the command ‘**npm start’** and it should direct you to the browser where the web application is now running.

**Workflow of the Web-Application** :-

1. The web application showcases the three use cases, each with a different tab: **Symptoms-Based**, **Image-Based**, and **GenAI-Based**.
2. In the **Symptoms-Based** tab, the user is presented with several drop-down choices of - disease; symptoms such as fever, cough, fatigue; demographic information such as age, gender; and also asking for blood-pressure, cholesterol levels of the user. After the user finishes making their selection and clicks on Submit, the result will be displayed, that is, **Positive/Negative**; specifying whether the user has that particular disease.
3. In the **Image-Based** tab, the user can simply upload an image of a potential skin cancer and submit it. The result will be displayed to him, that is, whether their skin is actually **Cancerous** or **Non-Cancerous**.
4. In the **GenAI-Based** tab, the user can simply upload a pdf of a blood report and submit it. As a result, a descriptive **summary** of the report will be displayed to the user.

**Algorithms Implemented** :-

1. For **Symptoms-Based**, **Decision tree** algorithm with **f\_classif** as feature selection method was trained on a dataset and implemented with the help of pipeline, column transformers and encoders like One-Hot, Ordinal encoders to handle categorical variables.
2. For **Image-Based**, a custom **Convolutional neural network**(CNN) was built and trainedon a dataset of images whose task was to perform image classification with the help of tensorflow and keras.
3. For **GenAI-Based**, OpenAI’s **GPT-4** was used to generate summary from the input pdf file with the help of embeddings, vector database and Retrieval-augmented generation(RAG).