**Part 5 - Deliverables**

* 1. A Gradioor script (app.py) that provides a simple UI for the model.
  2. Following are the instructions for deployment of our Gradio-based application on the Hugging Face Space, ‘IrumGilani/heart-disease-predictor’:-

• The content in the ‘requirements.txt’ file was automatically detected by the Hugging Face and the dependencies listed (gradio==5.1.0 joblib scikit-learn numpy) in it were installed. The progress of the process and error detection were monitored in the logs section of our Hugging Face Space, ‘‘IrumGilani/heart-disease-predictor.’

• We utilized Gradio’s tools and libraries to create an interactive user frontend/interface for our machine learning model. Our setup leverages Gradio’s API-style interface within Hugging Face Space (IrumGilani/heart-disease-predictor).

• By deploying the Gradio app on Hugging Face Space, we made both the frontend and backend accessible online, allowing users to interact with our ML seamlessly.

• Gradio simplifies the entire flow by allowing us to define inputs and outputs in the same code Gradio script file (app.py). Our Gradio-based API for heart disease prediction starts with the user entering health-related data through an easy-to-use web interface on Hugging Face Space, ‘IrumGilani/heart-disease-predictor.’ The backend, powered by Gradio, processes this input and feeds it into a pre-trained machine learning model that predicts the risk of heart disease. The result is then displayed back to the user, who can adjust the input to see how changes in data affect the prediction. This makes the model's insights accessible and interactive, facilitating real-time health risk assessment.

• The ‘app.py’ was the Gradio script on Hugging Face Space, ‘IrumGilani/heart-disease-predictor.’ When Gradio app was opened on the Hugging Face Space, input data fields appeared where input data was entered for all 19 variables. After hitting Submit button, the prediction result (“Risk of heart disease”) appeared.