

Sprint-3

Application Building

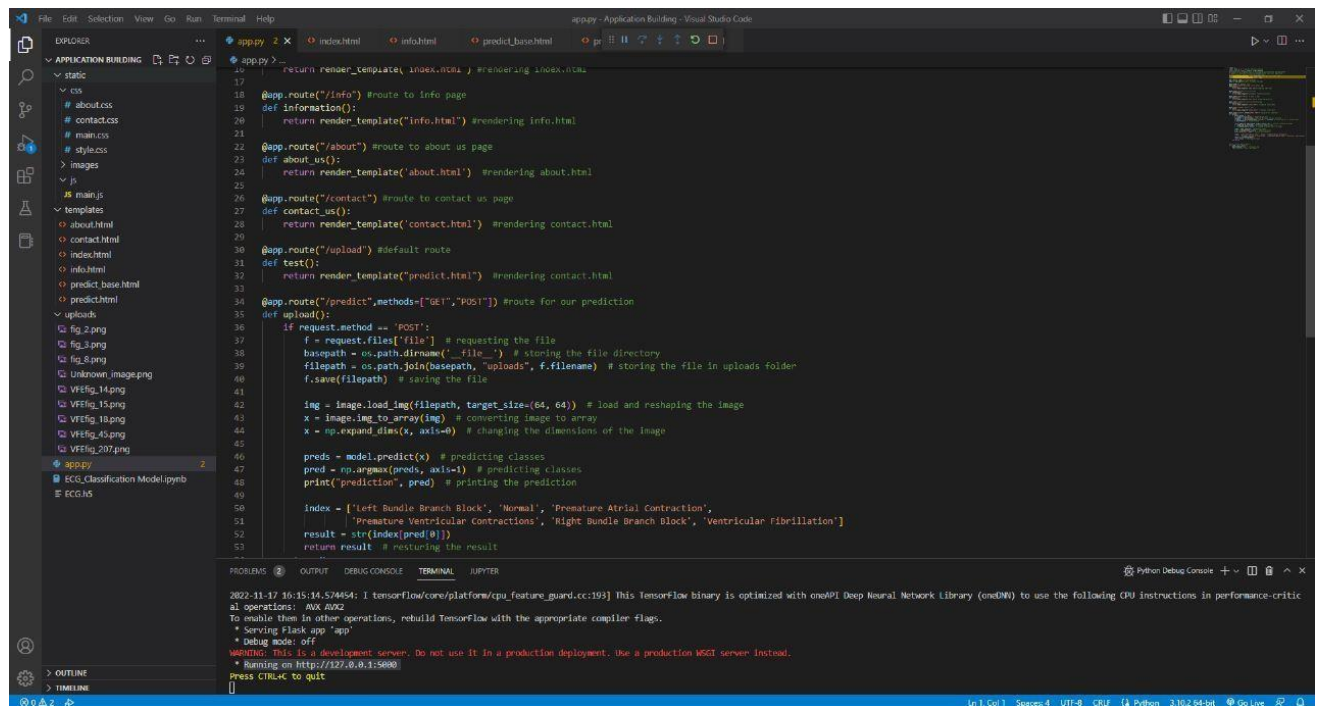
RUN THE APP

Date	13Nov 2022
TeamID	PNT2022TMID49645
ProjectName	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

TASK:

Run The App.

RUN ON LOCAL HOST (SCREEN SHOT):



```
16 return render_template("index.html") #rendering index.html
17
18 @app.route("/info") #route to info page
19 def info():
20     return render_template("info.html") #rendering info.html
21
22 @app.route("/about") #route to about us page
23 def about_us():
24     return render_template("about.html") #rendering about.html
25
26 @app.route("/contact") #route to contact us page
27 def contact_us():
28     return render_template("contact.html") #rendering contact.html
29
30 @app.route("/upload") #default route
31 def test():
32     return render_template("predict.html") #rendering contact.html
33
34 @app.route("/predict", methods=['GET', 'POST']) #route for our prediction
35 def upload():
36     if request.method == 'POST':
37         f = request.files['file'] # requesting the file
38         basepath = os.path.dirname(__file__) # storing the file directory
39         filepath = os.path.join(basepath, "uploads", f.filename) # storing the file in uploads folder
40         f.save(filepath) # saving the file
41
42         img = image.load_img(filepath, target_size=(64, 64)) # load and reshaping the image
43         x = image.img_to_array(img) # converting image to array
44         x = np.expand_dims(x, axis=0) # changing the dimensions of the image
45
46         preds = model.predict(x) # predicting classes
47         pred = np.argmax(preds, axis=1) # predicting classes
48         print("prediction", pred) # printing the prediction
49
50         index = ['Left Bundle Branch Block', 'Normal', 'Premature Atrial Contraction',
51                 'Premature Ventricular Contractions', 'Right Bundle Branch Block', 'Ventricular Fibrillation']
52         result = str(index[pred[0]])
53         return result # returning the result
```

2022-11-17 10:15:14.524454: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

* Serving Flask app "app"

* Debug mode: off

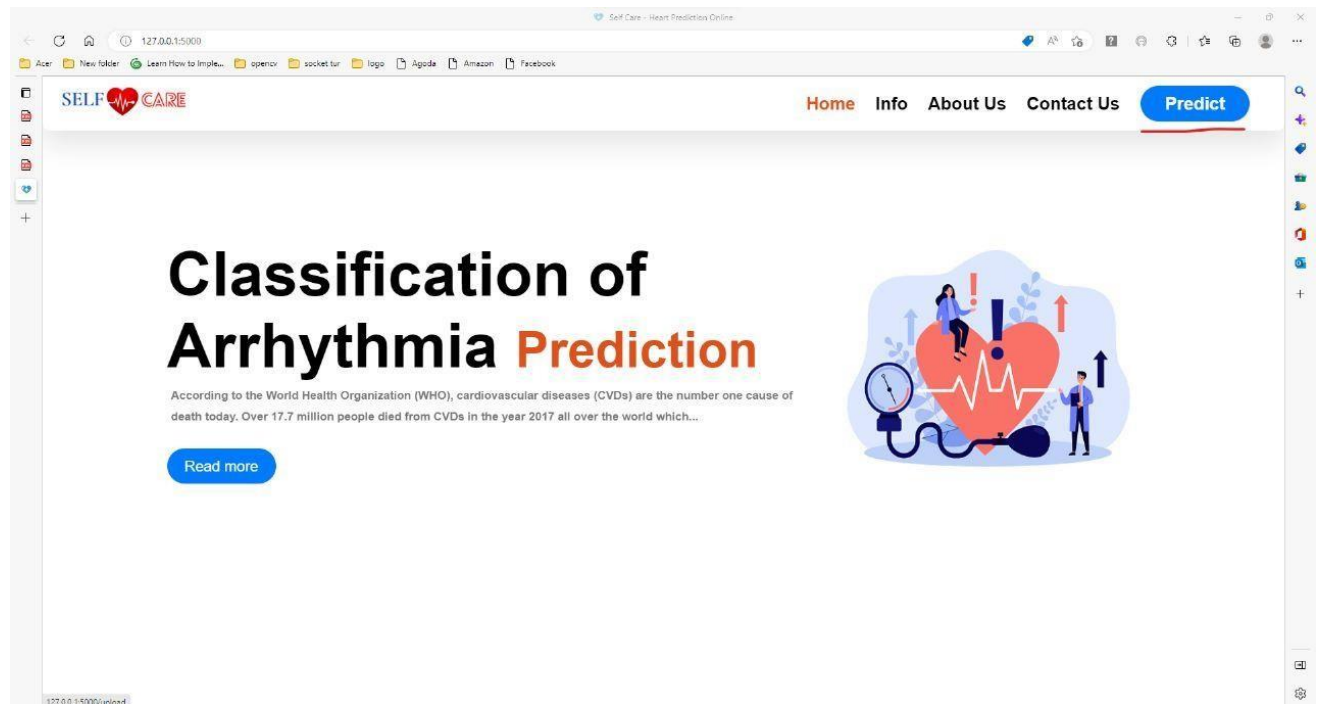
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

* Running on <http://127.0.0.1:5000>

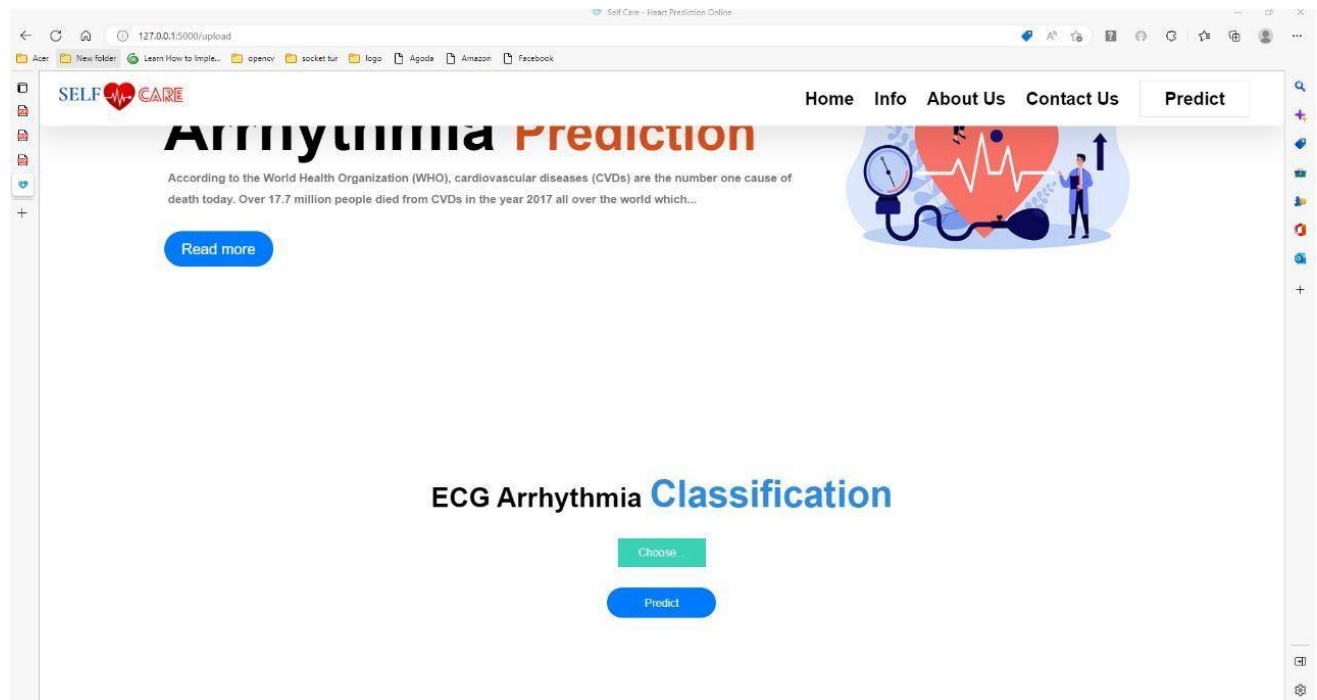
Press CTRL+C to quit

LOCAL HOST:

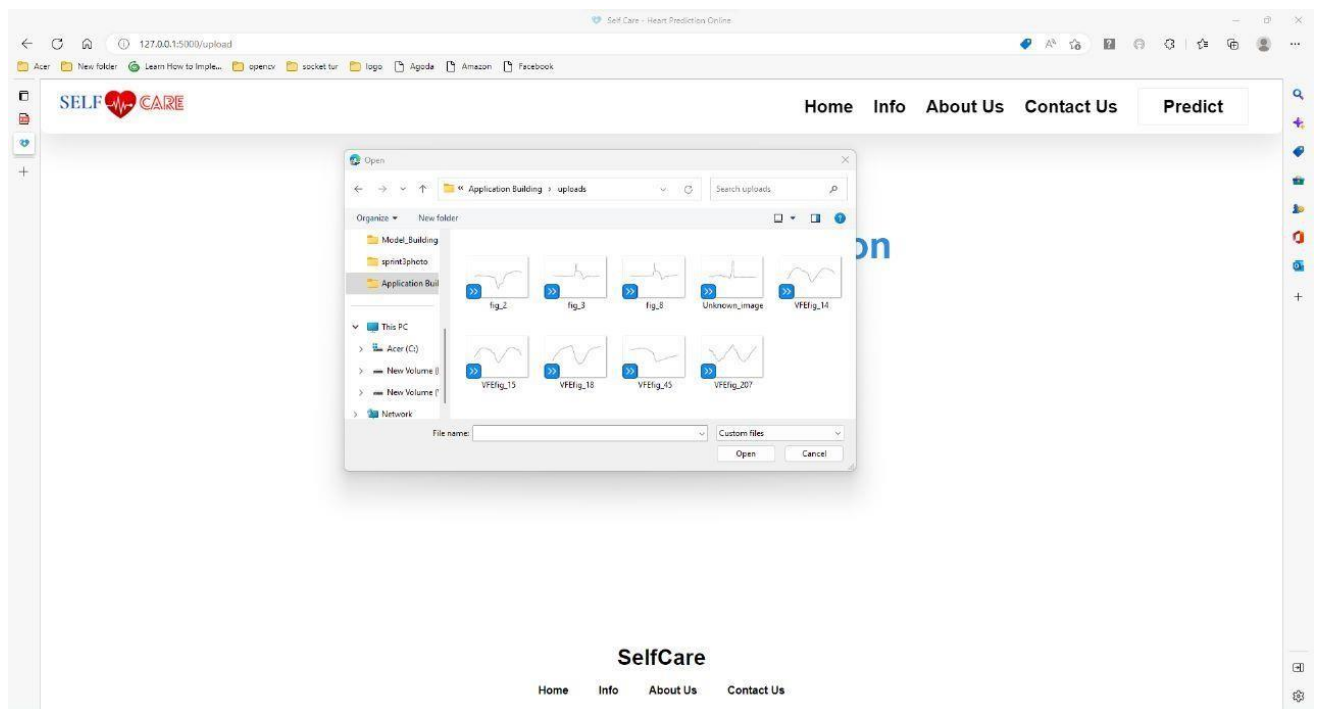
CLICK PREDICT (SCREEN SHOT)::



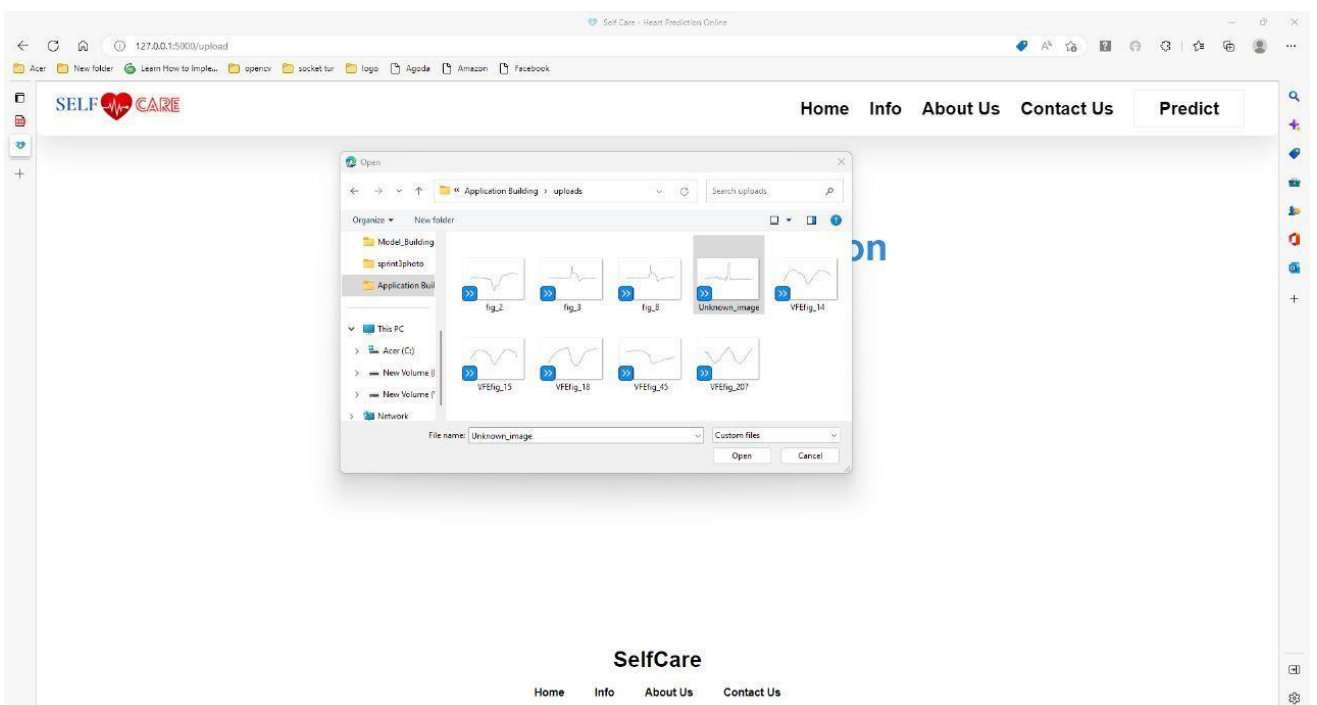
CLICK CHOOSE BUTTON (SCREEN SHOT):



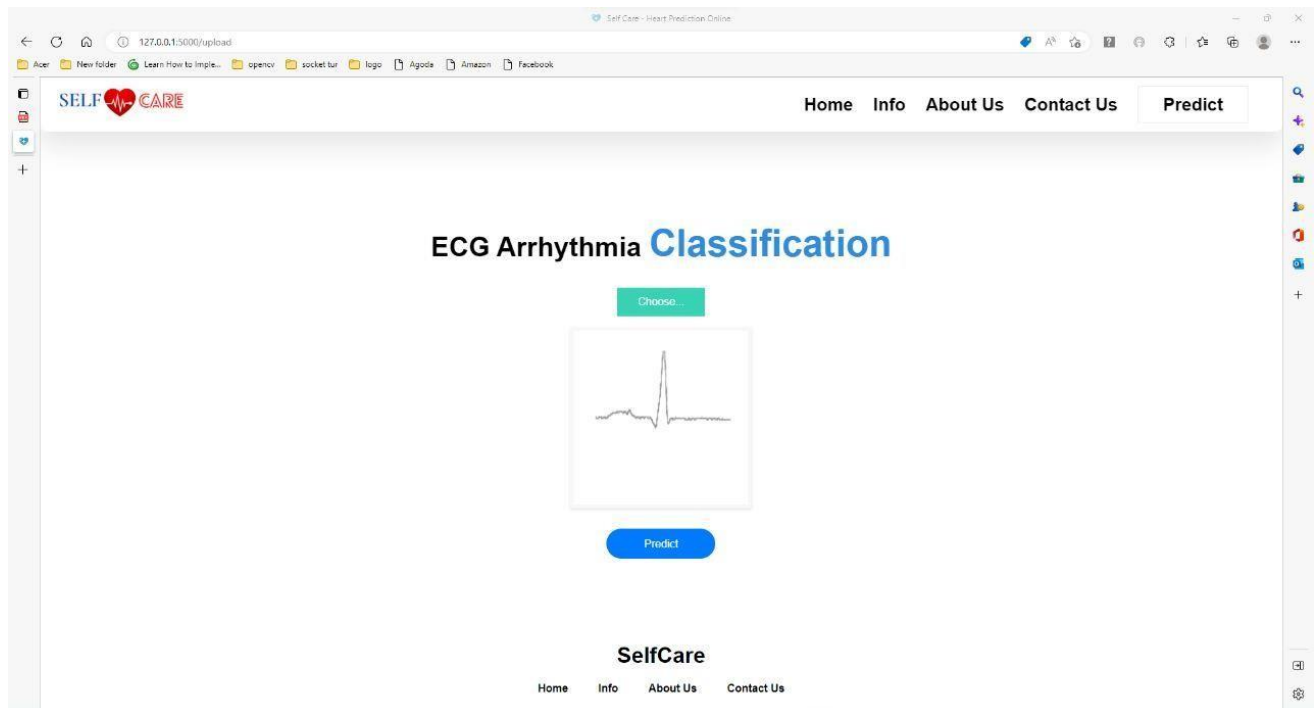
SELECT ANY FILE (SCREEN SHOT):



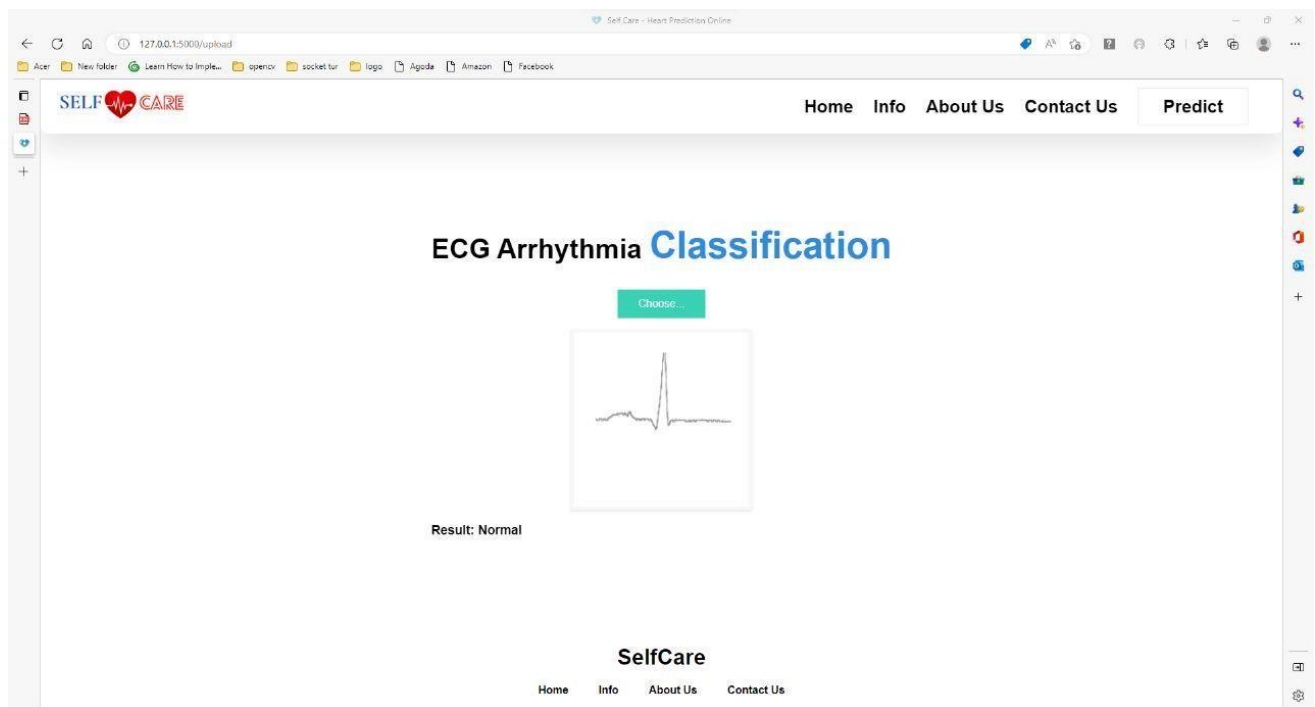
CLICK ANY ECG PHOTO (SCREEN SHOT):



CLICK PREDICT BUTTON (SCREEN SHOT):



SHOW ON RESULT (SCREEN SHOT):



PREDICT THE CORRECT RESULT

