Project Development Phase

Delivery of Sprint 4

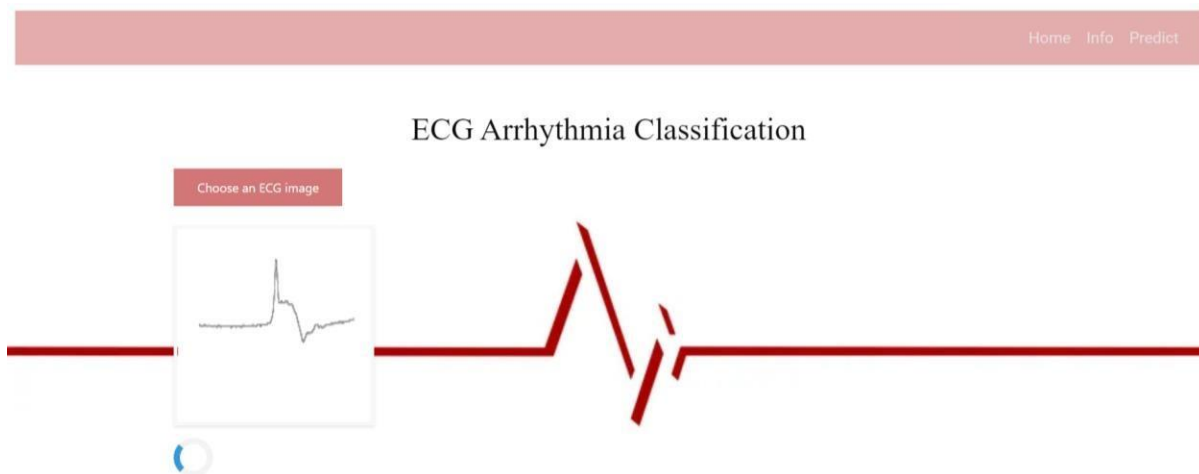| Date	11 November 2022
Team ID	PNT2022TMID16980
Project Name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

Code: Updated in GitHub in the Deliverables section in Sprint 4 folder.

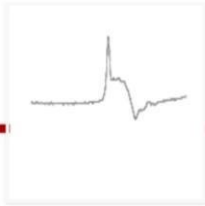**Description of USN and Screenshots:**

USN-6:

As a user, I can upload an ECG image and view the result. The type of Arrhythmia such as Left Bundle Branch Block, Normal, Premature Atrial Contraction, Premature Ventricular Contractions, Right Bundle Branch Block and Ventricular Fibrillation is displayed.

Screenshot:

ECG Arrhythmia Classification

Choose an ECG image

Result:
Premature Atrial Contraction

USN-7:

As a user, I can upload any ECG image and view the result. The algorithm is designed to denote the type of Arrhythmia such as Left Bundle Branch Block, Normal, Premature Atrial Contraction, Premature Ventricular Contractions, Right Bundle Branch Block and Ventricular Fibrillation.
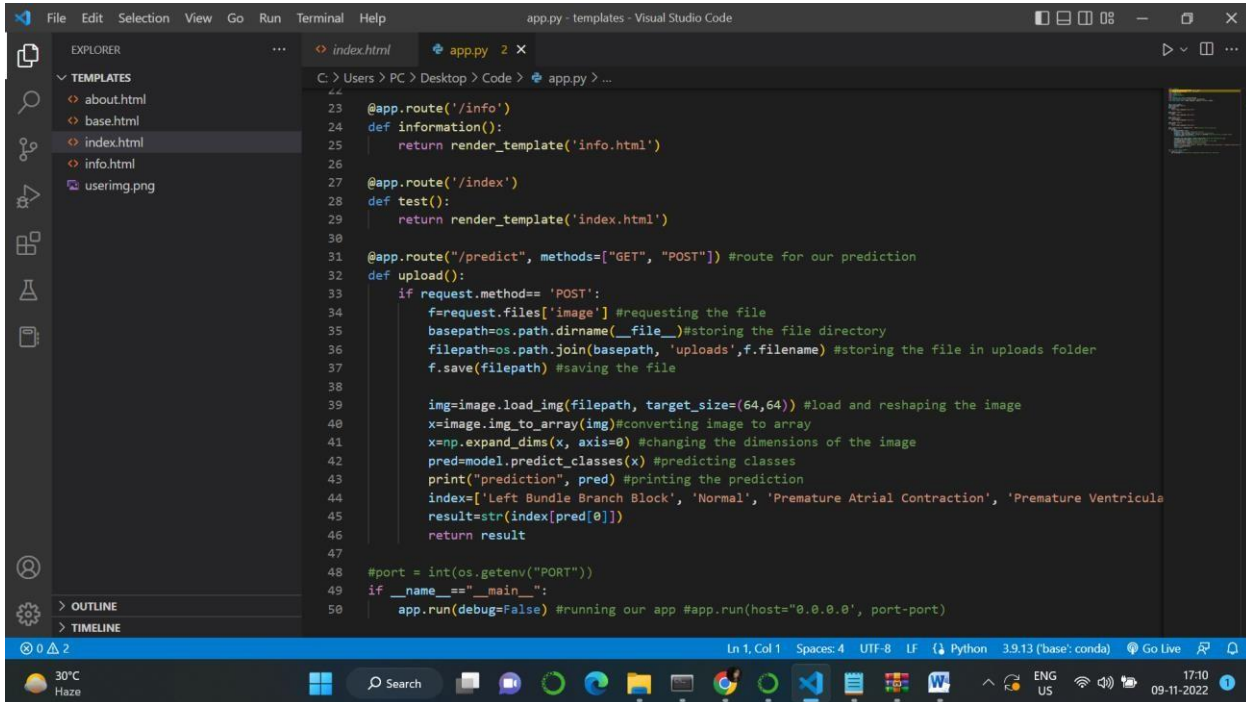
The HTML file used to build the Info page includes:


```html
            padding: 20px;
    }
    </style>
    <div class="heading">
        ECG Arrhythmia Classification
    </div>

<div>
    <form action="" id="upload-file" method="post" enctype="multipart/form-data">
    <label for="imageUpload" class="upload-label">
        Choose an ECG image
    </label>
        <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
    </form>

    <div class="image-section" style="display:none;">
        <div class="img-preview">
            <div id="imagePreview">
            </div>
        </div>
        <div>
            <button type="button" class="btn btn-primary btn-lg " id="btn-predict">Predict!</button>
        </div>
    </div>

    <div class="loader" style="display:none;"></div>

    <h3>
        <span id="result" style="font-size:17px;"> </span>
```

The Python code behind classification of Arrhythmia :

```python
23      @app.route('/info')
24      def information():
25          return render_template('info.html')
26
27      @app.route('/index')
28      def test():
29          return render_template('index.html')
30
31      @app.route("/predict", methods=["GET", "POST"]) #route for our prediction
32      def upload():
33          if request.method== 'POST':
34              f=request.files['image'] #requesting the file
35              basepath=os.path.dirname(__file__)#storing the file directory
36              filepath=os.path.join(basepath, 'uploads',f.filename) #storing the file in uploads folder
37              f.save(filepath) #saving the file
38
39              img=image.load_img(filepath, target_size=(64,64)) #load and reshaping the image
40              x=image.img_to_array(img)#converting image to array
41              x=np.expand_dims(x, axis=0) #changing the dimensions of the image
42              pred=model.predict_classes(x) #predicting classes
43              print("prediction", pred) #printing the prediction
44              index=['Left Bundle Branch Block', 'Normal', 'Premature Atrial Contraction', 'Premature Ventricula
45              result=str(index[pred[0]])
46              return result
47
48      #port = int(os.getenv("PORT"))
49      if __name__=="__main__":
50          app.run(debug=False) #running our app #app.run(host="0.0.0.0", port-port)
```