Project Development Phase
Delivery of Sprint 4

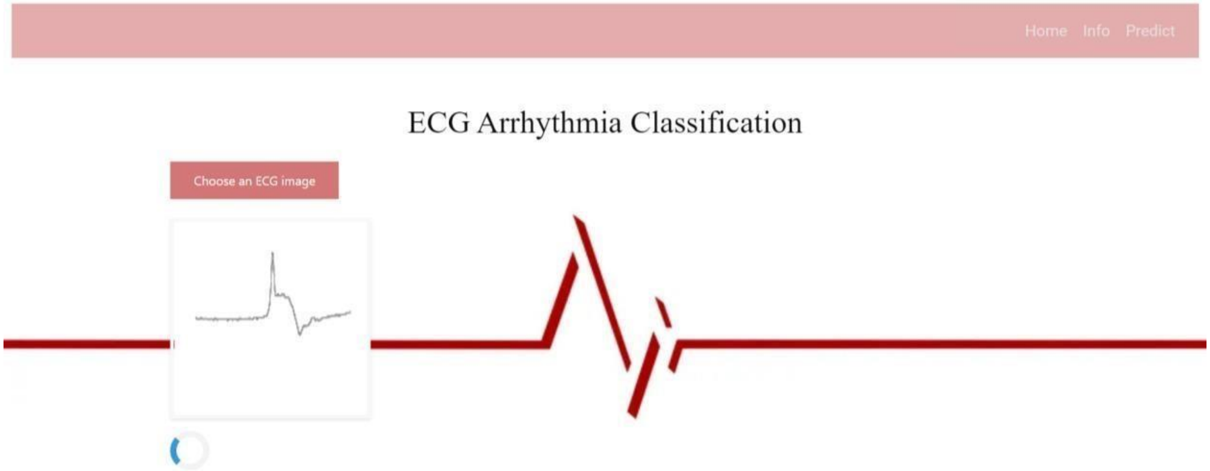
Team ID	PNT2022TMID16902
Project Name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

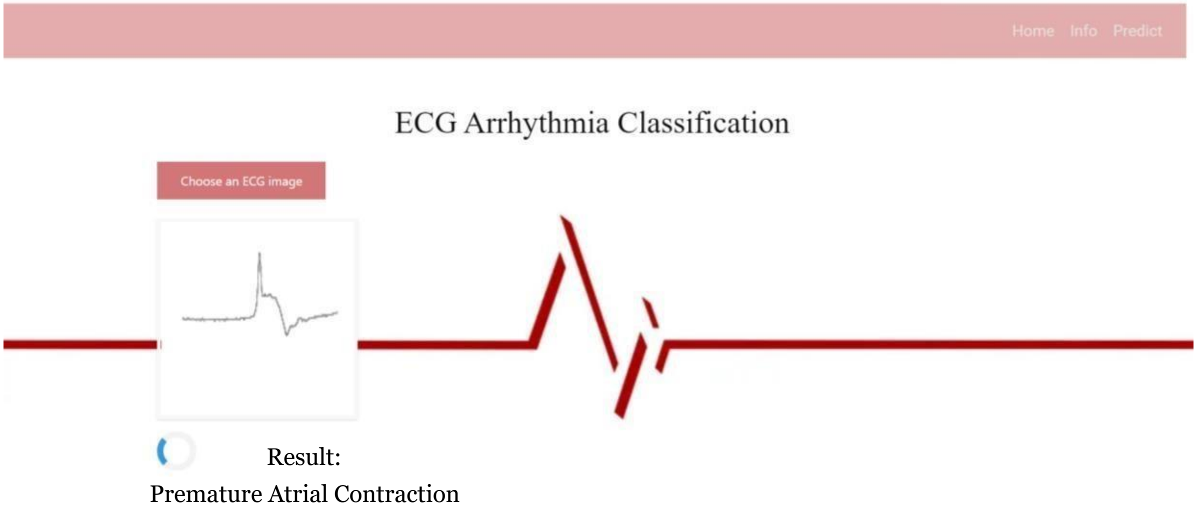
Code: Updated in GitHub in the Deliverables section in Sprint 4 folder.

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:**





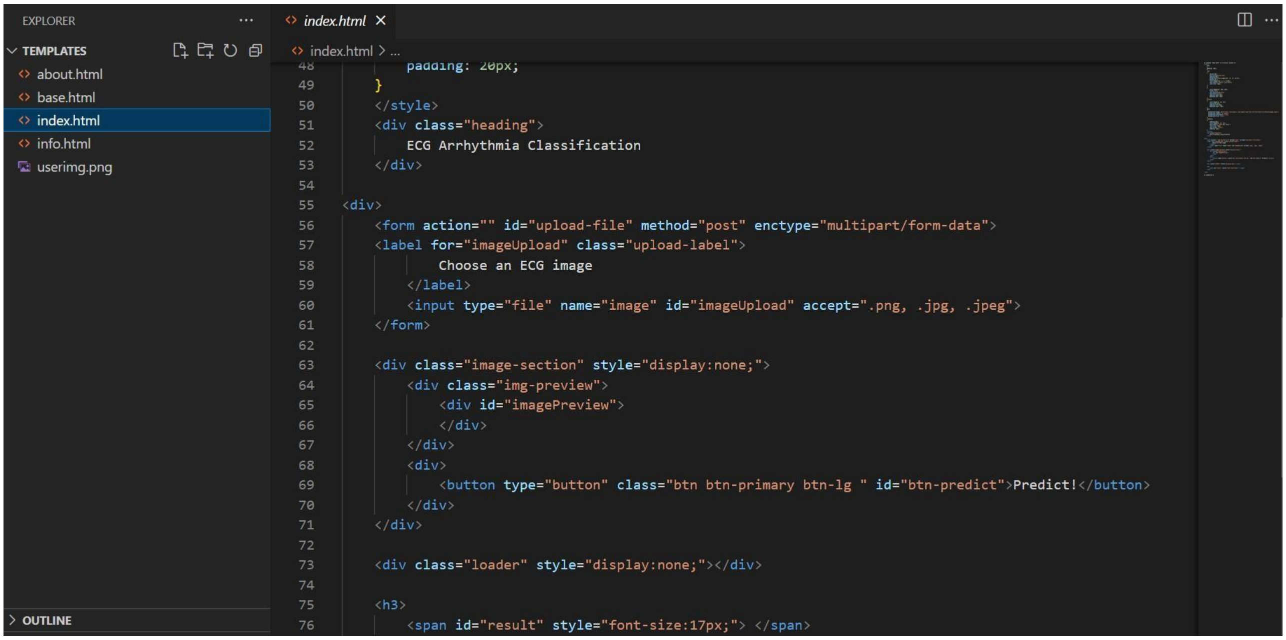
ECG Arrhythmia Classification

Result:
Premature Atrial Contraction

USN-7:

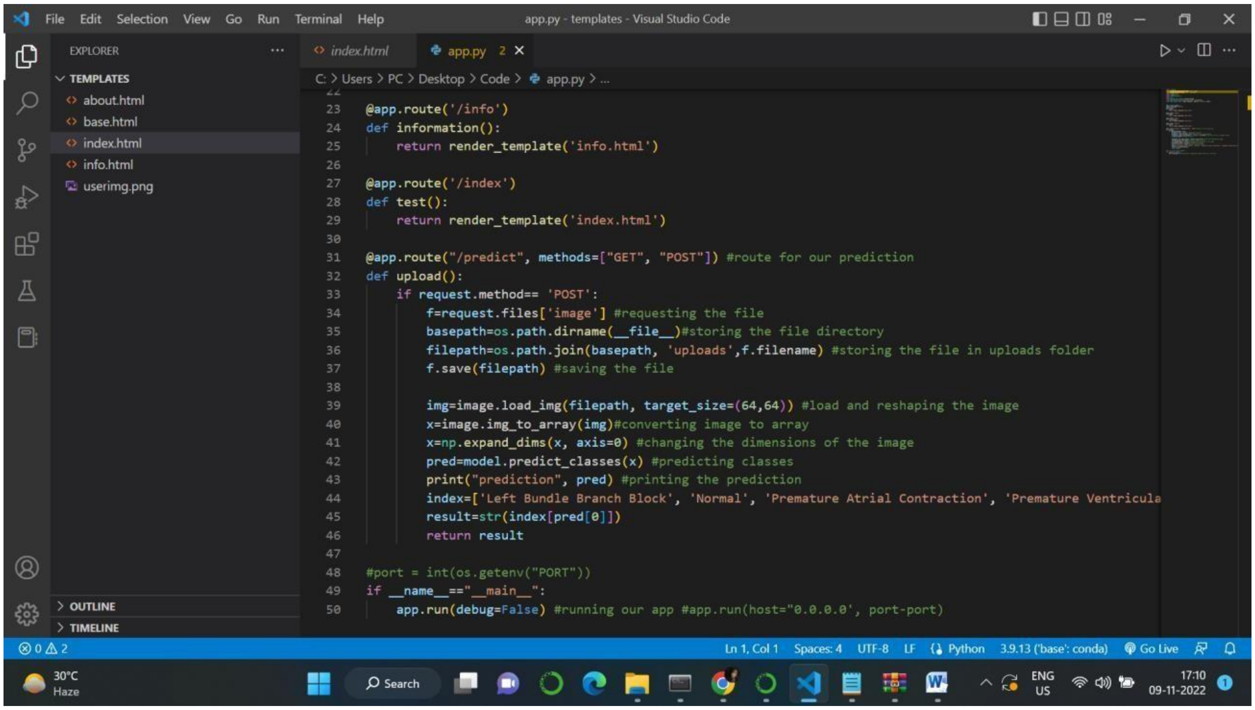
As a user, I can upload any ECG image and view the result. The algorithm is designed to denote the type of Arrhythmia such as Left Bundle Branch Block, Normal, Premature Atrial Contraction, Premature Ventricular Contractions, Right Bundle Branch Block and Ventricular Fibrillation.

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
@app.route('/info')
def information():
    return render_template('info.html')

@app.route('/index')
def test():
    return render_template('index.html')

@app.route("/predict", methods=["GET", "POST"]) #route for our prediction
def upload():
    if request.method== 'POST':
        f=request.files['image'] #requesting the file
        basepath=os.path.dirname(__file__)#storing the file directory
        filepath=os.path.join(basepath, 'uploads',f.filename) #storing the file in uploads folder
        f.save(filepath) #saving the file

        img=image.load_img(filepath, target_size=(64,64)) #load and reshaping the image
        x=image.img_to_array(img)#converting image to array
        x=np.expand_dims(x, axis=0) #changing the dimensions of the image
        pred=model.predict_classes(x) #predicting classes
        print("prediction", pred) #printing the prediction
        index=['Left Bundle Branch Block', 'Normal', 'Premature Atrial Contraction', 'Premature Ventricula
        result=str(index[pred[0]])
        return result

#port = int(os.getenv("PORT"))
if __name__=="__main__":
    app.run(debug=False) #running our app #app.run(host="0.0.0.0', port-port)
```