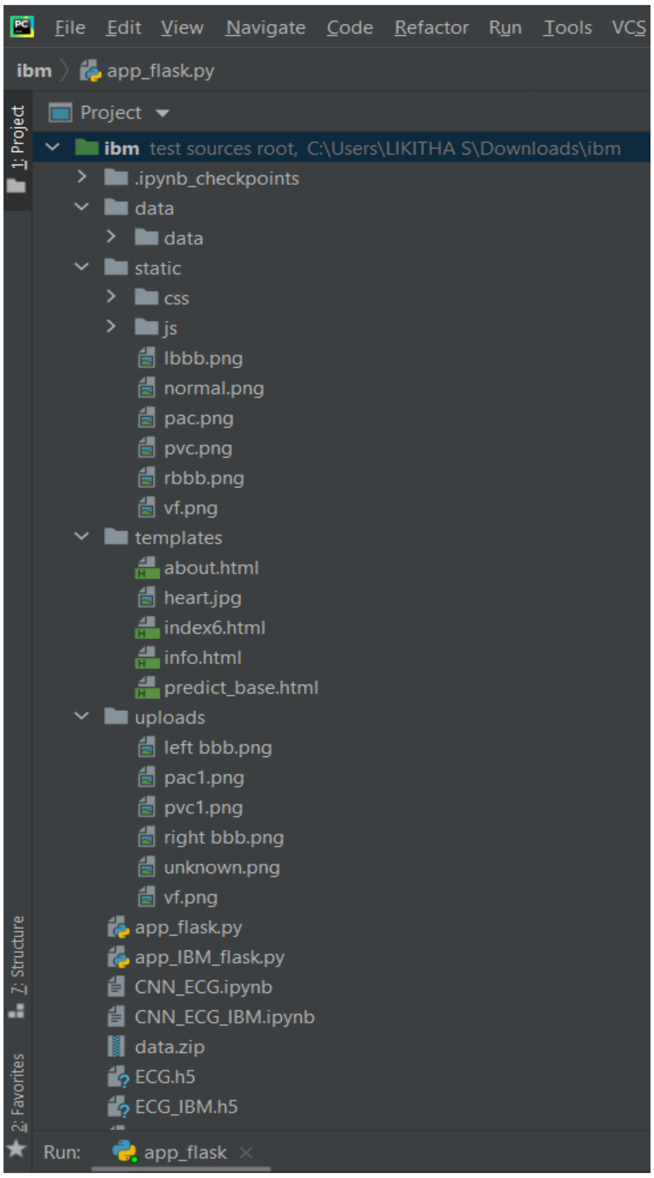
SPRINT-3

APPLICATION BUILDING

Build The Python Code

Date	19 Nov 2022
Team ID	PNT2022TMID01315
Project Name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

Project Structure:

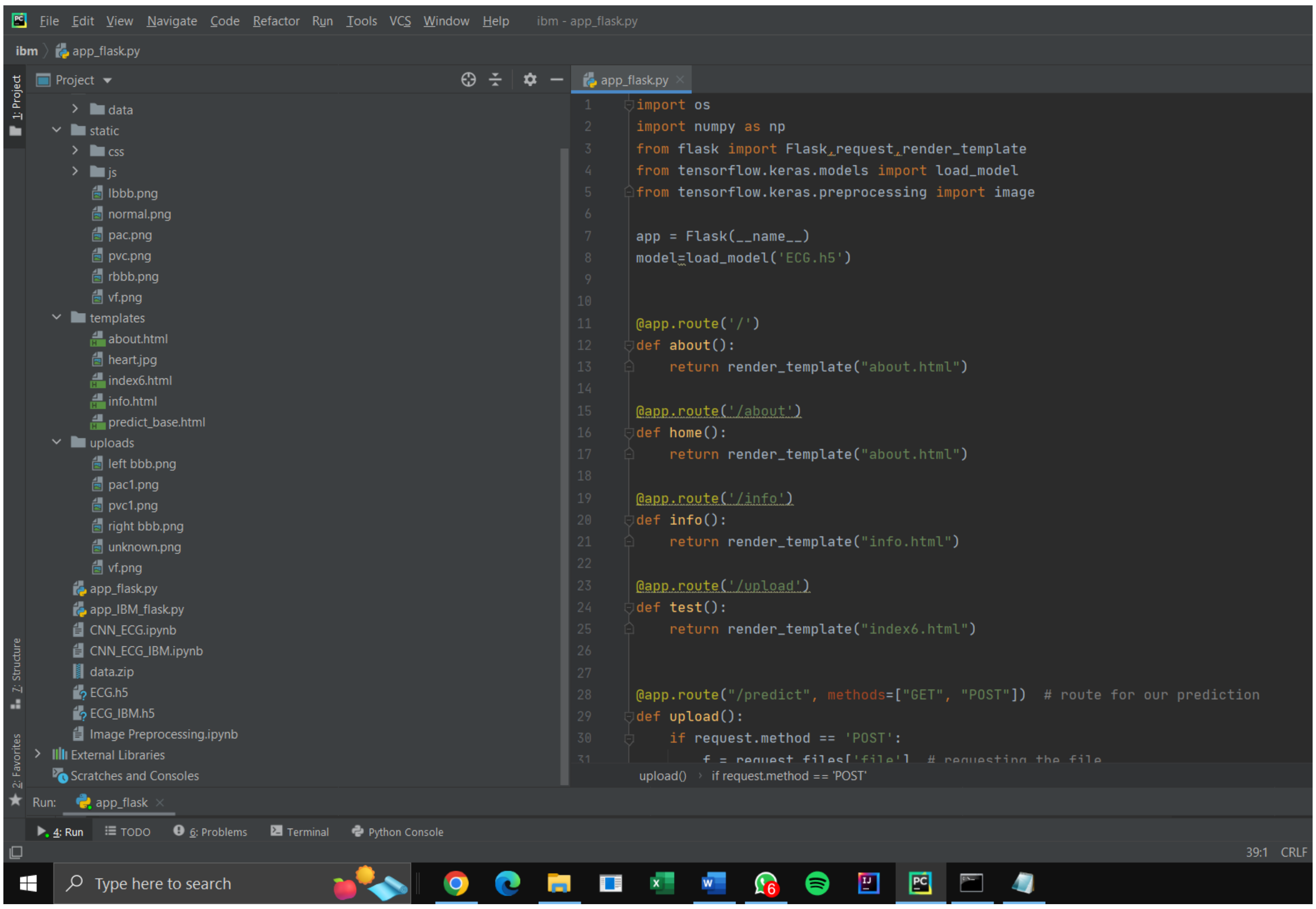


```
ibm  test sources root, C:\Users\LIKITHA S\Downloads\ibm
  .ipynb_checkpoints
  data
    data
  static
    css
    js
    lbbb.png
    normal.png
    pac.png
    pvc.png
    rbbb.png
    vf.png
  templates
    about.html
    heart.jpg
    index6.html
    info.html
    predict_base.html
  uploads
    left bbb.png
    pac1.png
    pvc1.png
    right bbb.png
    unknown.png
    vf.png
  app_flask.py
  app_IBM_flask.py
  CNN_ECG.ipynb
  CNN_ECG_IBM.ipynb
  data.zip
  ECG.h5
  ECG_IBM.h5
```

App_flask .py

```
import os
import numpy as np
from flask import Flask,request,render_template
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

app = Flask(__name__)
model=load_model('ECG.h5')


@app.route('/')
def about():
    return render_template("about.html")

@app.route('/about')
def home():
    return render_template("about.html")

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

@app.route('/upload')
def test():
    return render_template("index6.html")


@app.route("/predict", methods=["GET", "POST"])  # route for
our prediction
def upload():
    if request.method == 'POST':
        f = request.files['file']  # 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))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)

        pred = model.predict(x)
        y_pred = np.argmax(pred)
        print("prediction", y_pred)

        index = ['Left Bundle Branch Block', 'Normal',
'Premature Atrial Contraction',
                 'Premature Ventricular Contractions', 'Right
Bundle Branch Block', 'Ventricular Fibrillation']
        result = str(index[y_pred])

        return result  # resturing the result
    return None

if __name__=="__main__":
    app.run(debug=False)
```

App_flask.py screenshot:



```
import os
import numpy as np
from flask import Flask,request,render_template
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

app = Flask(__name__)
model=load_model('ECG.h5')


@app.route('/')
def about():
    return render_template("about.html")

@app.route('/about')
def home():
    return render_template("about.html")

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

@app.route('/upload')
def test():
    return render_template("index6.html")


@app.route("/predict", methods=["GET", "POST"])  # route for our prediction
def upload():
    if request.method == 'POST':
        f = request.files['file']  # requesting the file
```