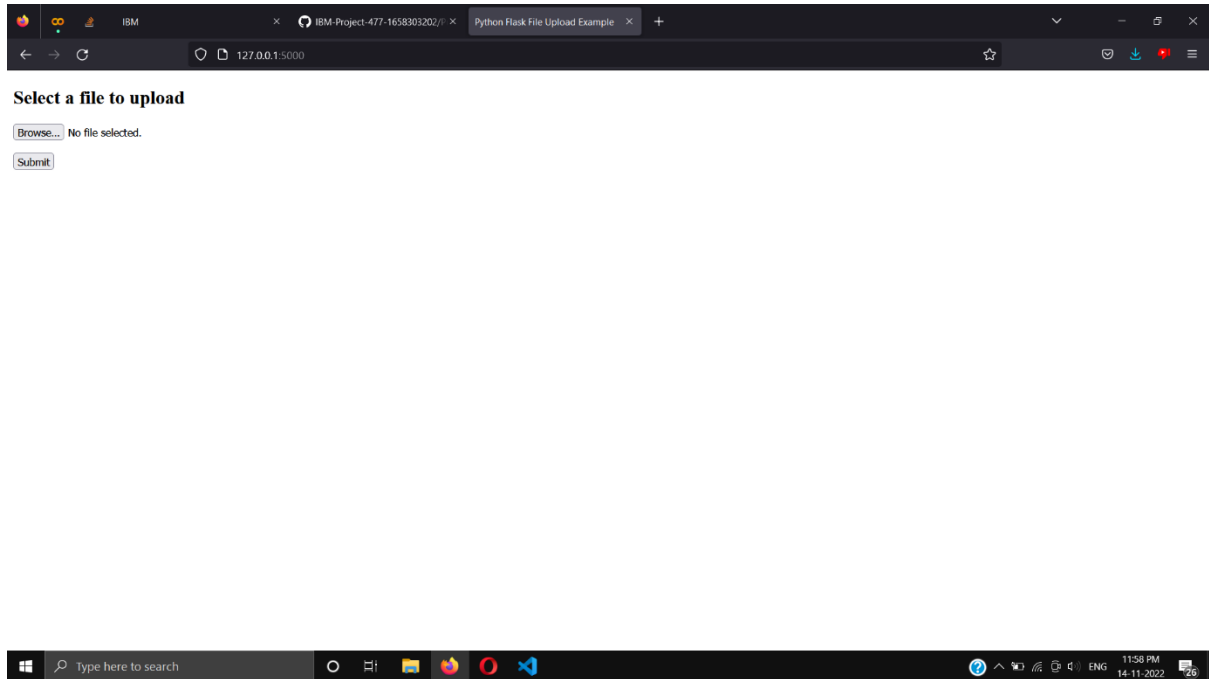


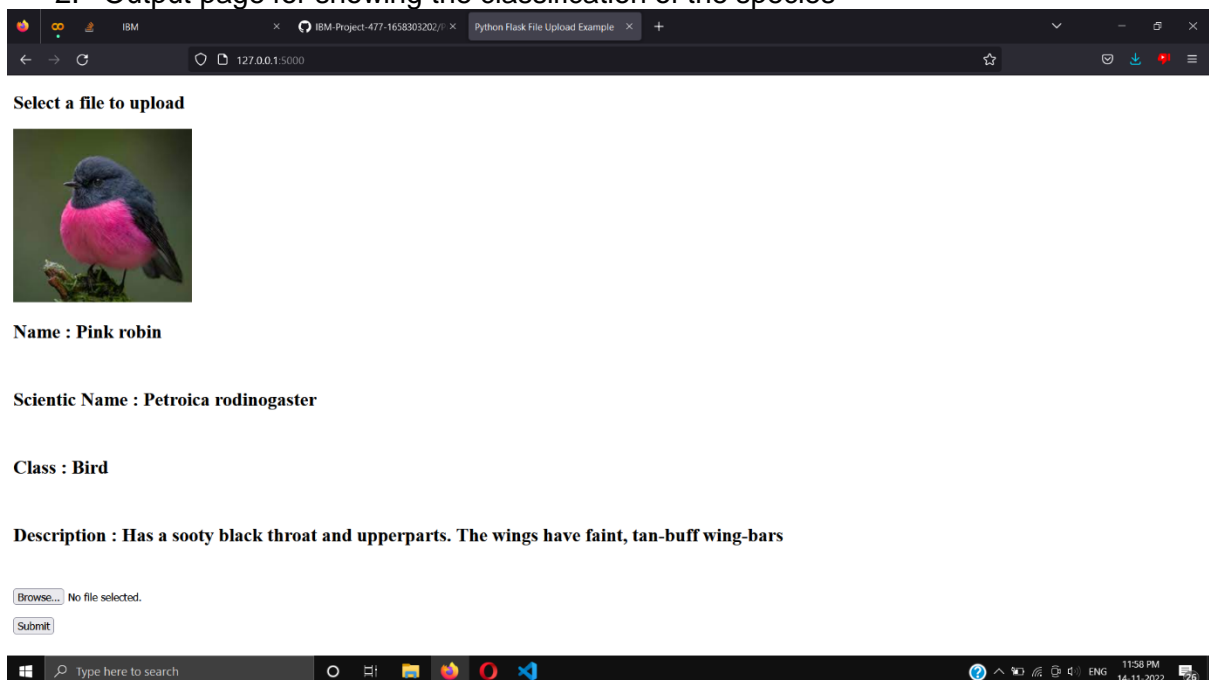
Date	12 November 2022
Team ID	PNT2022TMID35899
Project Name	Digital Naturalist - AI Enabled Tool For Biodiversity Researchers
Maximum Marks	4 Marks

Front-End:

1. Home page for uploading input

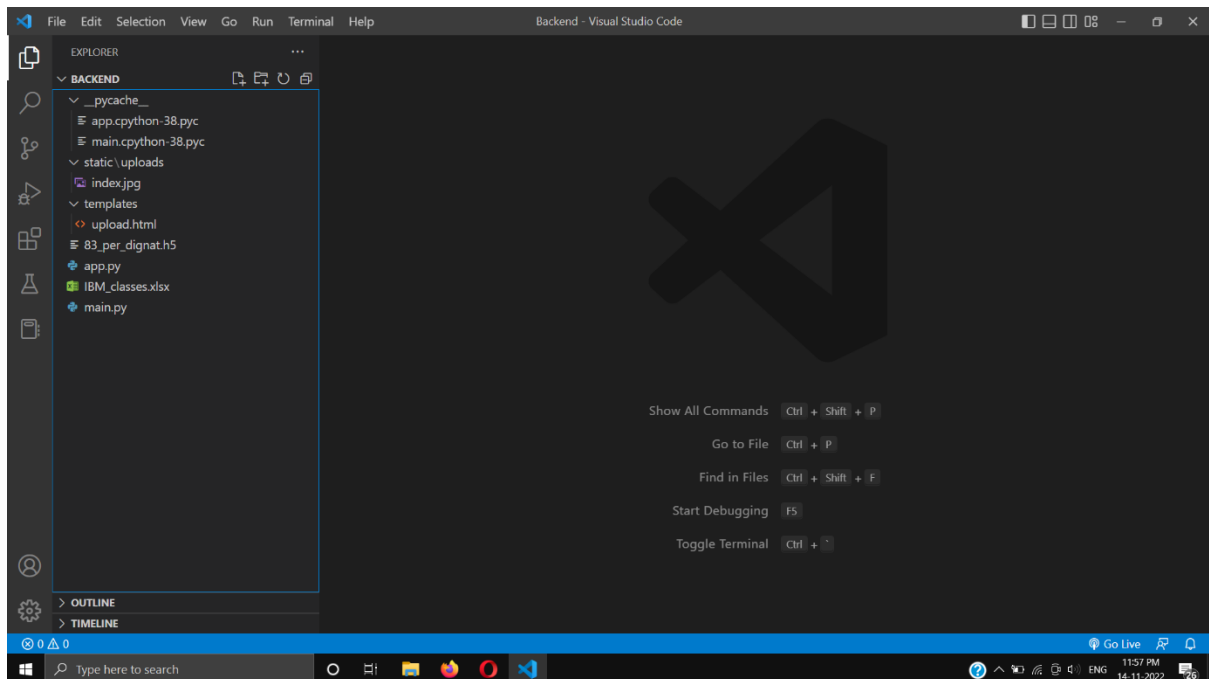


2. Output page for showing the classification of the species

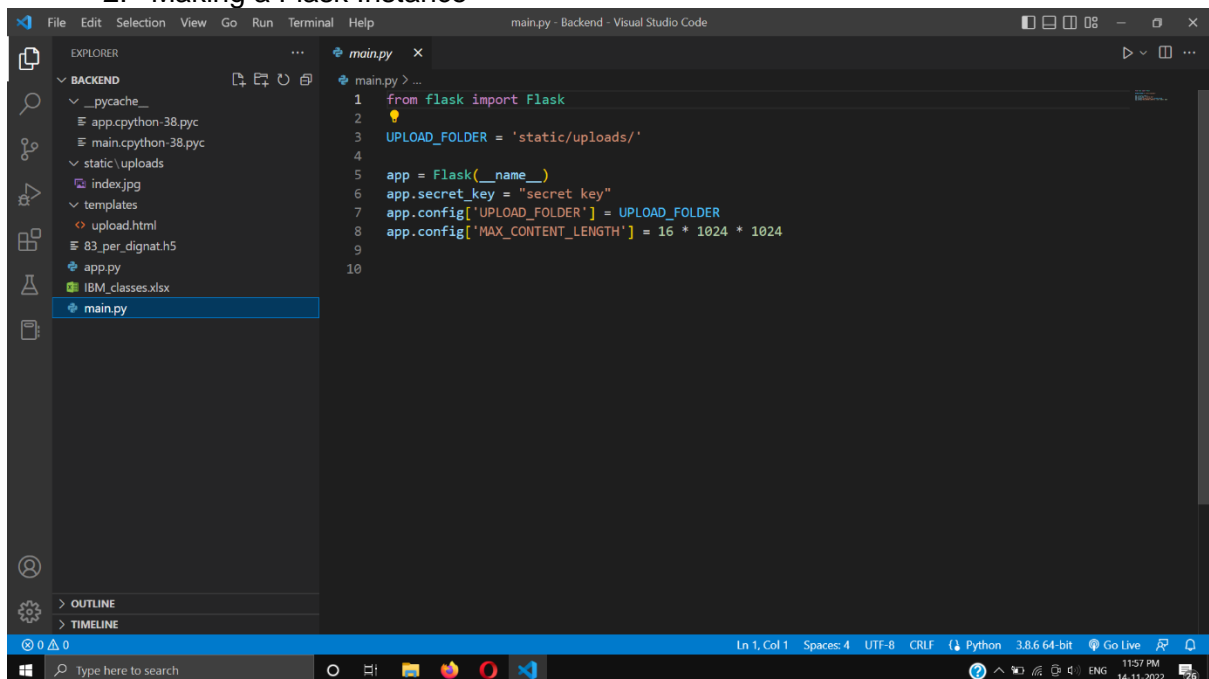


Backend using Flask

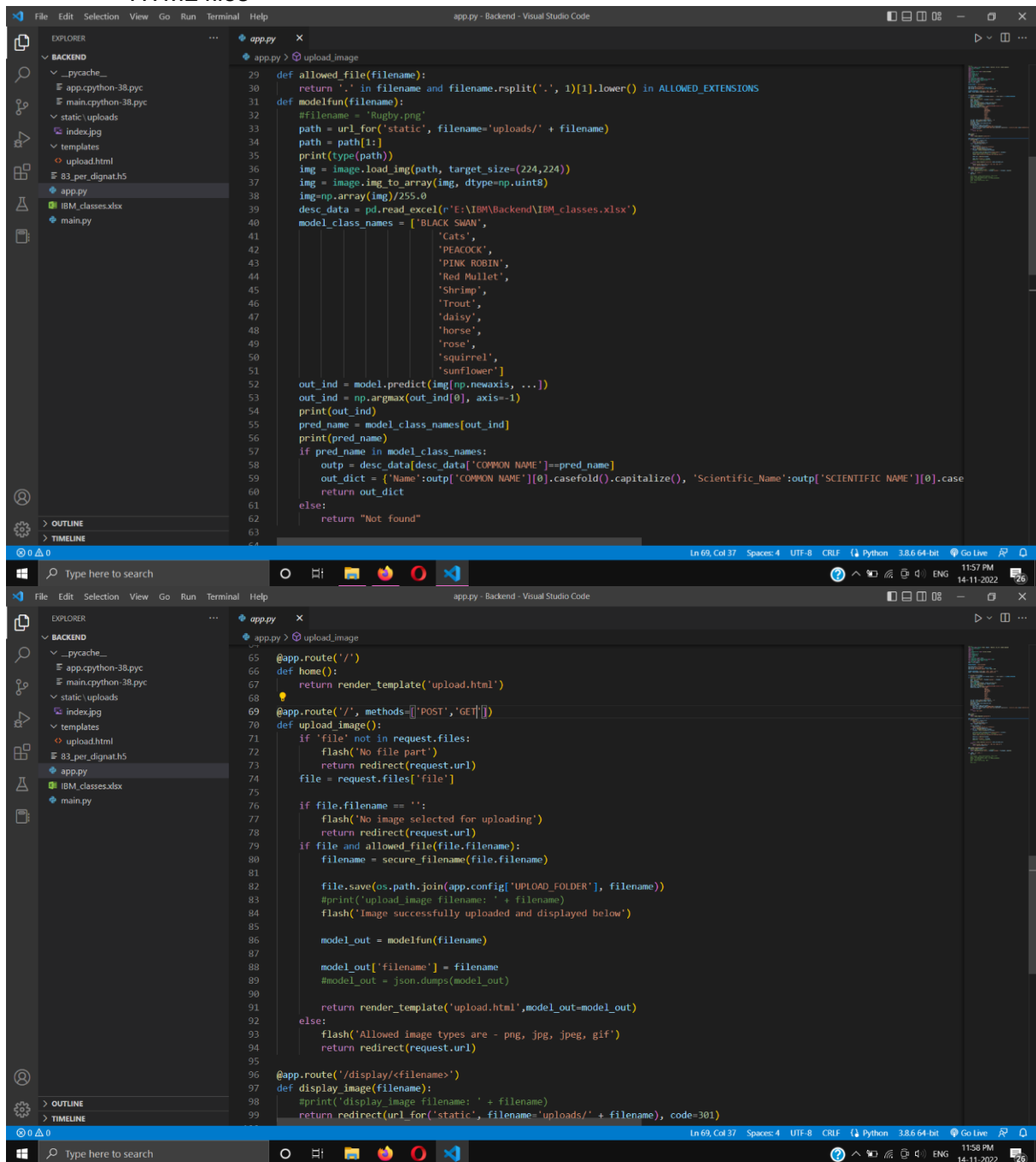
1. Application directory



2. Making a Flask Instance



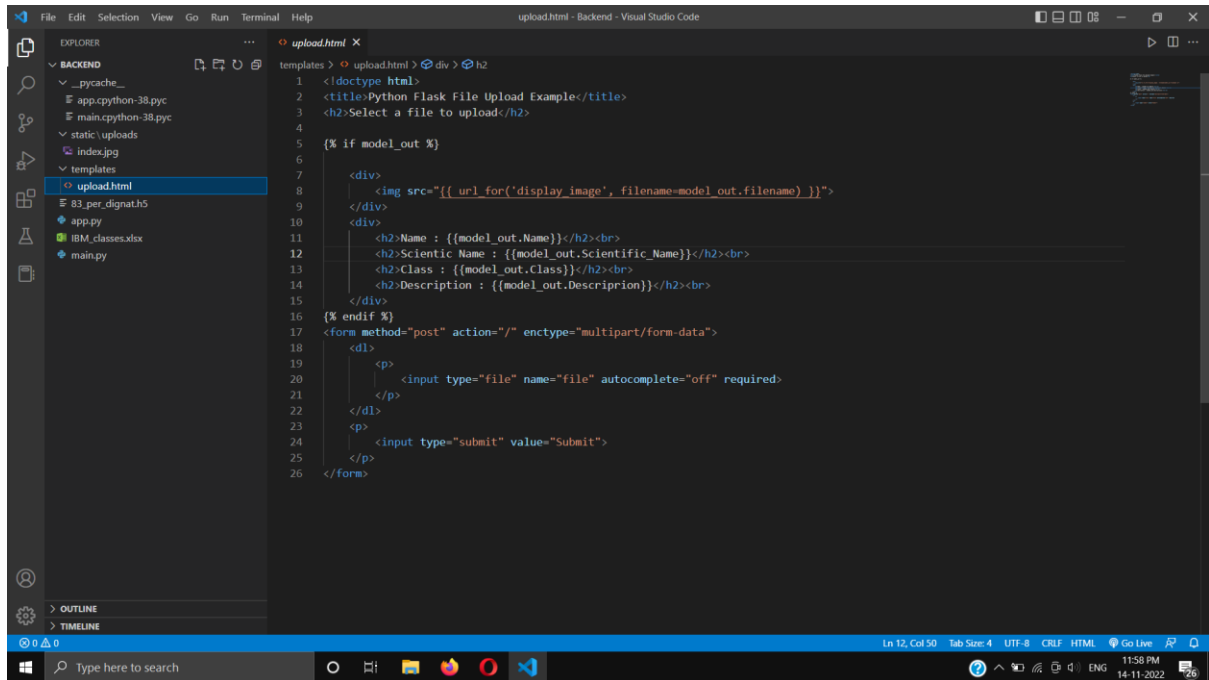
3. App.py for handling the requests and response. This contains the route to the HTML files



```
29 def allowed_file(filename):
30     return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
31 def modelfun(filename):
32     #filename = 'Rugby.png'
33     path = url_for('static', filename='uploads/' + filename)
34     path = path[1:]
35     print(type(path))
36     img = image.load_img(path, target_size=(224,224))
37     img = image.img_to_array(img, dtype=np.uint8)
38     img=np.array(img)/255.0
39     desc_data = pd.read_excel(r'E:\IBM\Backend\IBM_classes.xlsx')
40     model_class_names = ['BLACK SWAN',
41                          'Cats',
42                          'PEACOCK',
43                          'PINK ROBIN',
44                          'Red Mullet',
45                          'Shrimp',
46                          'Trout',
47                          'daisy',
48                          'horse',
49                          'rose',
50                          'squirrel',
51                          'sunflower']
52     out_ind = model.predict(img[np.newaxis, ...])
53     out_ind = np.argmax(out_ind[0], axis=-1)
54     print(out_ind)
55     pred_name = model_class_names[out_ind]
56     print(pred_name)
57     if pred_name in model_class_names:
58         outp = desc_data[desc_data['COMMON NAME']==pred_name]
59         out_dict = {'Name':outp['COMMON NAME'][0].casefold().capitalize(), 'Scientific_Name':outp['SCIENTIFIC NAME'][0].case
60         return out_dict
61     else:
62         return "Not found"
```

```
65 @app.route('/')
66 def home():
67     return render_template('upload.html')
68
69 @app.route('/', methods=['POST','GET'])
70 def upload_image():
71     if 'file' not in request.files:
72         flash('No file part')
73         return redirect(request.url)
74     file = request.files['file']
75
76     if file.filename == '':
77         flash('No image selected for uploading')
78         return redirect(request.url)
79     if file and allowed_file(file.filename):
80         filename = secure_filename(file.filename)
81
82         file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
83         #print('upload_image filename: ' + filename)
84         flash('Image successfully uploaded and displayed below')
85
86         model_out = modelfun(filename)
87
88         model_out['filename'] = filename
89         #model_out = json.dumps(model_out)
90
91         return render_template('upload.html',model_out=model_out)
92     else:
93         flash('Allowed image types are - png, jpg, jpeg, gif')
94         return redirect(request.url)
95
96 @app.route('/display/<filename>')
97 def display_image(filename):
98     #print('display image filename: ' + filename)
99     return redirect(url_for('static', filename='uploads/' + filename), code=301)
```

4. Added Flask Jinja 2 tags in HTML Pages to get and display the data from backend



```
1 <!doctype html>
2 <title>Python Flask File Upload Example</title>
3 <h2>Select a file to upload</h2>
4
5 {% if model_out %}
6
7     <div>
8         
9     </div>
10    <div>
11        <h2>Name : {{model_out.Name}}</h2><br>
12        <h2>Scientific Name : {{model_out.Scientific_Name}}</h2><br>
13        <h2>Class : {{model_out.Class}}</h2><br>
14        <h2>Description : {{model_out.Description}}</h2><br>
15    </div>
16 {% endif %}
17 <form method="post" action="/" enctype="multipart/form-data">
18     <dl>
19         <p>
20             <input type="file" name="file" autocomplete="off" required>
21         </p>
22     </dl>
23     <p>
24         <input type="submit" value="Submit">
25     </p>
26 </form>
```