

## DAILY ASSESSMENT FORMAT

Date:	30/6/2020	Name:	SAMRUDDI SHETTY
Course:	Python	USN:	4AL16IS047
Topic:	DAY 10	Semester & Section:	8 <sup>th</sup> sem
Github Repository:			

### FORENOON SESSION DETAILS

#### Demonstration of the Geocoding Web Service Application and Project Requirements



ID	Address	Name	Employees	Latitude	Longitude
0 1	3660 1st St San Francisco CA 94114 USA	Madeira	8	37.756489	-122.429343
1 2	735 Dolores St San Francisco CA 94119 USA	Bready Shop	15	NaN	NaN
2 3	332 Hill St San Francisco California 94114 USA	Super River	25	37.755725	-122.428601
3 4	3995 23rd St San Francisco CA 94114 USA	Ben's Shop	10	37.752965	-122.431714
4 5	1056 Sanchez St San Francisco California USA	Sanchez	12	37.752146	-122.429815

## Index.html

```
app
├── static
│   └── main.css
├── templates
│   ├── download.html
│   └── index.html
├── uploads
├── virtual
│   ├── app_ver1.py
│   ├── app_ver2.py
│   ├── app_ver3.py
│   └── app_ver4.py
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <title> Super Geocoder </title>
4 <head>
5   <link href="../../static/main.css" rel="stylesheet">
6 </head>
7 <body>
8   <div class="container">
9     <h1>Super Geocoder</h1>
10    <h3>Please upload your CSV file. The values containing addresses should be in a column named <em>address</em> or <em>Address</em></h3>
11    <form action="{{url_for('success_table')}}" method="POST" enctype="multipart/form-data">
12      <input type="file" accept=".csv" name="file" />
13      <button type="submit"> Submit </button>
14    </form>
15    <div class="output">
16      {{text|safe}}
17      {% include btn ignore missing %}
18    </div>
19  </div>
20 </body>
21 </html>
22
```

## Main.css

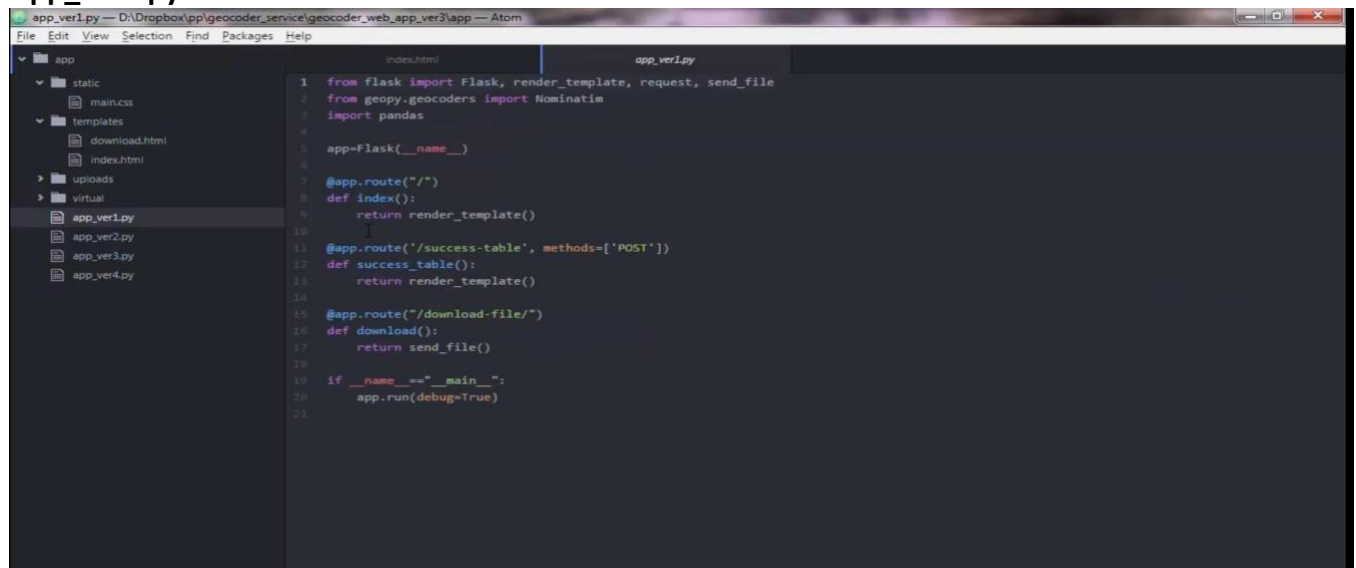
```
app
├── static
│   ├── main.css
│   └── index.html
├── templates
│   ├── download.html
│   └── index.html
├── uploads
├── virtual
│   ├── app_ver1.py
│   ├── app_ver2.py
│   ├── app_ver3.py
│   └── app_ver4.py
```

```
1 html, body {
2   height: 100%;
3   margin: 0;
4 }
5
6 .container {
7   margin: 0 auto;
8   width: 200px;
9   height: 100%;
10  background-color: #006666;
11  color: #a6ffff;
12  overflow: hidden;
13  text-align: center;
14 }
15
16 .container form {
17   margin: 20px;
18 }
19
20 .container h1 {
21   font-family: Arial, sans-serif;
22   font-size: 30px;
23   color: #00CCEE;
24   margin-top: 80px;
25 }
26
27 .container button {
28   width: 70px;
29   height: 30px;
30   background-color: steelblue;
31   margin: 3px;
32 }
```

```
app
├── static
│   ├── main.css
│   └── index.html
├── templates
│   ├── download.html
│   └── index.html
├── uploads
├── virtual
│   ├── app_ver1.py
│   ├── app_ver2.py
│   ├── app_ver3.py
│   └── app_ver4.py
```

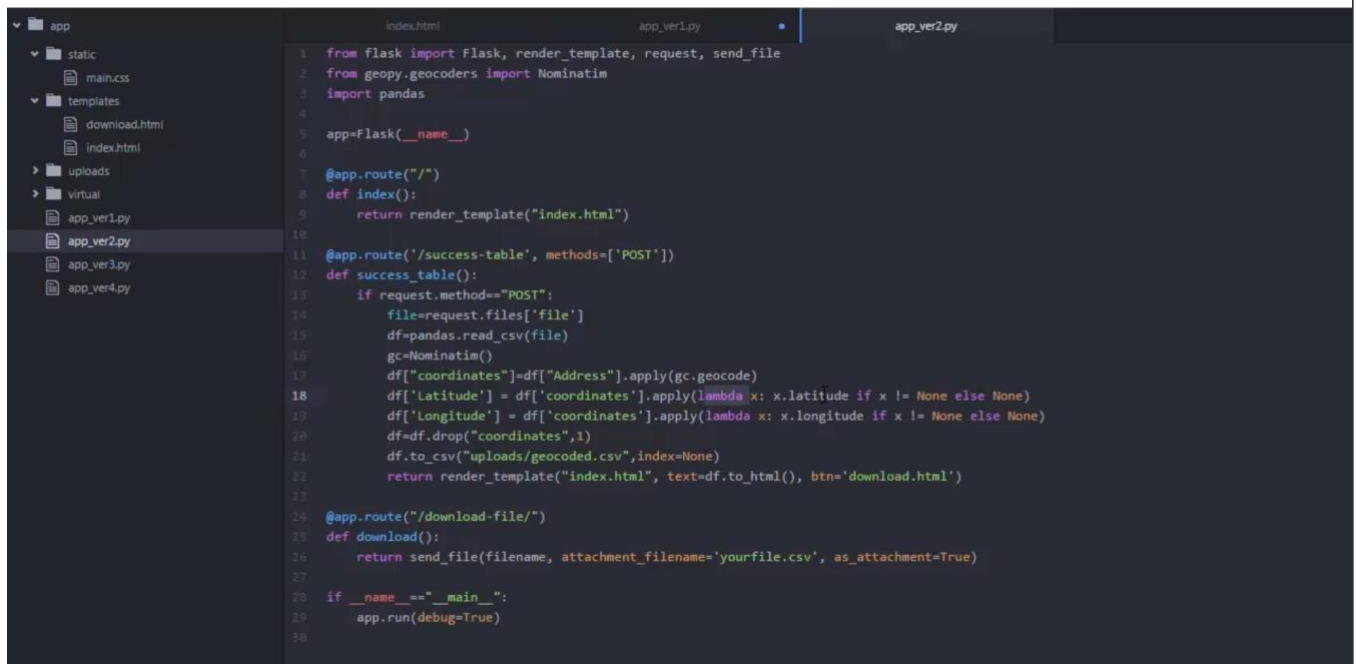
```
15
16 .container form {
17   margin: 20px;
18 }
19
20 .container h1 {
21   font-family: Arial, sans-serif;
22   font-size: 30px;
23   color: #00CCEE;
24   margin-top: 80px;
25 }
26
27 .container button {
28   width: 70px;
29   height: 30px;
30   background-color: steelblue;
31   margin: 3px;
32 }
33
34 .container input {
35   width: 200px;
36   height: 15px;
37   font-size: 15px;
38   margin: 2px;
39   padding: 5px;
40   transition: all 0.2s ease-in-out;
41 }
42
43 .output {
44   display: inline-block;
45 }
```

## App\_ver1.py



```
1 from flask import Flask, render_template, request, send_file
2 from geopy.geocoders import Nominatim
3 import pandas
4
5 app=Flask(__name__)
6
7 @app.route("/")
8 def index():
9     return render_template("index.html")
10
11 @app.route('/success-table', methods=['POST'])
12 def success_table():
13     return render_template("index.html")
14
15 @app.route("/download-file/")
16 def download():
17     return send_file()
18
19 if __name__ == "__main__":
20     app.run(debug=True)
```

## App\_ver2.py



```
1 from flask import Flask, render_template, request, send_file
2 from geopy.geocoders import Nominatim
3 import pandas
4 import datetime
5
6 app=Flask(__name__)
7
8 @app.route("/")
9 def index():
10     return render_template("index.html")
11
12 @app.route('/success-table', methods=['POST'])
13 def success_table():
14     if request.method=="POST":
15         file=request.files['file']
16         df=pandas.read_csv(file)
17         gc=Nominatim()
18         df["coordinates"]=df["Address"].apply(lambda x: gc.geocode(x).latitude if x != None else None)
19         df["Longitude"] = df["coordinates"].apply(lambda x: x.longitude if x != None else None)
20         df=df.drop("coordinates",1)
21         df.to_csv("uploads/geocoded.csv",index=None)
22         return render_template("index.html", text=df.to_html(), btn='download.html')
23
24 @app.route("/download-file/")
25 def download():
26     return send_file(filename, attachment_filename='yourfile.csv', as_attachment=True)
27
28 if __name__ == "__main__":
29     app.run(debug=True)
```

```
from flask import Flask, render_template, request, send_file
from geopy.geocoders import Nominatim
import pandas
import datetime

app=Flask(__name__)

@app.route("/")
```

```

def index():
    return render_template("index.html")

@app.route('/success-table', methods=['POST'])
def success_table():
    global filename
    if request.method=="POST":
        file=request.files['file']
        try:
            df=pandas.read_csv(file)
            gc=Nominatim(scheme='http')
            df["coordinates"]=df["Address"].apply(gc.geocode)
            df['Latitude'] = df['coordinates'].apply(lambda x: x.latitude if x != None else None)
            df['Longitude'] = df['coordinates'].apply(lambda x: x.longitude if x != None else None)
            df=df.drop("coordinates",1)
            filename=datetime.datetime.now().strftime("sample_files/%Y-%m-%d-%H-%M-%S-%f"+"%.csv")
            df.to_csv(filename,index=None)
            return render_template("index.html", text=df.to_html(), btn='download.html')
        except Exception as e:
            return render_template("index.html", text=str(e))

@app.route("/download-file/")
def download():
    return send_file(filename, attachment_filename='yourfile.csv', as_attachment=True)

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

```