

# DAILY ASSESSMENT FORMAT

Date:	06-06-2020	Name:	Jagadeesha Hegde
Course:	Logic Design	USN:	4AL17EC036
Topic:	Application 10: Project Exercise on Building a Geocoder Web Service	Semester & Section:	6th A-sec
Github Repository:	Jagadeesha-036		

## FORENOON SESSION DETAILS

Image of session

The screenshot shows the Atom code editor with two files open: `index.html` and `download.html`. The `index.html` file contains Python code for a Flask web application. The `download.html` file contains HTML code for a form to upload a CSV file.

```

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     if request.method=="POST":
14         file=request.files['file']
15         try:
16             df=pandas.read_csv(file)
17             gc=Nominatim()
18             df['coordinates']=df['Address'].apply(gc.geocode)
19             df['latitude'] = df['coordinates'].apply(lambda x: x.latitude if x != None
20             df['longitude'] = df['coordinates'].apply(lambda x: x.longitude if x != None
21             df=df.drop("coordinates",1)
22             df.to_csv("uploads/geocoded.csv",index=None)
23             return render_template("index.html", text=df.to_html(), btn='download.html')
24         except:
25             return render_template("index.html", text="Please make sure you have an ad
26
27 @app.route("/download-file/")
28 def download():
29     return send_file("uploads/geocoded.csv", attachment_filename='yourfile.csv', as_at
30
31 if __name__=="__main__":
32     app.run(debug=True)

```

```

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 con
11 <form action="{{url_for('success_table')}}" m
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>

```

**Report – Report can be typed or hand written for up to two pages.**

Geocoding request and response (latitude/longitude lookup) The following example requests the latitude and longitude of "1600 Amphitheatre Parkway, Mountain View, CA", and specifies that the output must be in JSON format. you can test this by entering the URL into your web browser (be sure to replace YOUR\_API\_KEY with your actual API key). The response includes the latitude and longitude of the address.

View the developer's guide for more information about building geocoding request URLs and available parameters and understanding the response.

Below is a sample geocoding response, in JSON:

```
{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "1600",
          "short_name" : "1600",
          "types" : [ "street_number" ]
        },
        {
          "long_name" : "Amphitheatre Parkway",
          "short_name" : "Amphitheatre Pkwy",
          "types" : [ "route" ]
        },
        {
          "long_name" : "Mountain View",
```

```
"short_name" : "Mountain View",
"types" : [ "locality", "political" ]
},
{
  "long_name" : "Santa Clara County",
  "short_name" : "Santa Clara County",
  "types" : [ "administrative_area_level_2", "political" ]
},
{
  "long_name" : "California",
  "short_name" : "CA",
  "types" : [ "administrative_area_level_1", "political" ]
},
{
  "long_name" : "United States",
  "short_name" : "US",
  "types" : [ "country", "political" ]
},
{
  "long_name" : "94043",
  "short_name" : "94043",
  "types" : [ "postal_code" ]
}
],
"formatted_address" : "1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA",
"geometry" : {
  "location" : {
    "lat" : 37.4267861,
    "lng" : -122.0806032
  },

```

```

    "location_type" : "ROOFTOP",
    "viewport" : {
      "northeast" : {
        "lat" : 37.4281350802915,
        "lng" : -122.0792542197085
      },
      "southwest" : {
        "lat" : 37.4254371197085,
        "lng" : -122.0819521802915
      }
    },
    "place_id" : "ChIJtYuu0V25j4ARwu5e4wwRYgE",
    "plus_code" : {
      "compound_code" : "CWC8+R3 Mountain View, California, United States",
      "global_code" : "849VCWC8+R3"
    },
    "types" : [ "street_address" ]
  },
  "status" : "OK"
}

```

Reverse geocoding request and response (address lookup) The following example requests the address corresponding to a given latitude/longitude in Brooklyn, NY, USA. It specifies that the output must be in JSON format. You can test this by entering the URL into your web browser (be sure to replace 'YOUR\_API\_KEY' with your actual API key). The response includes a human-readable address for the latitude and longitude location.

--


