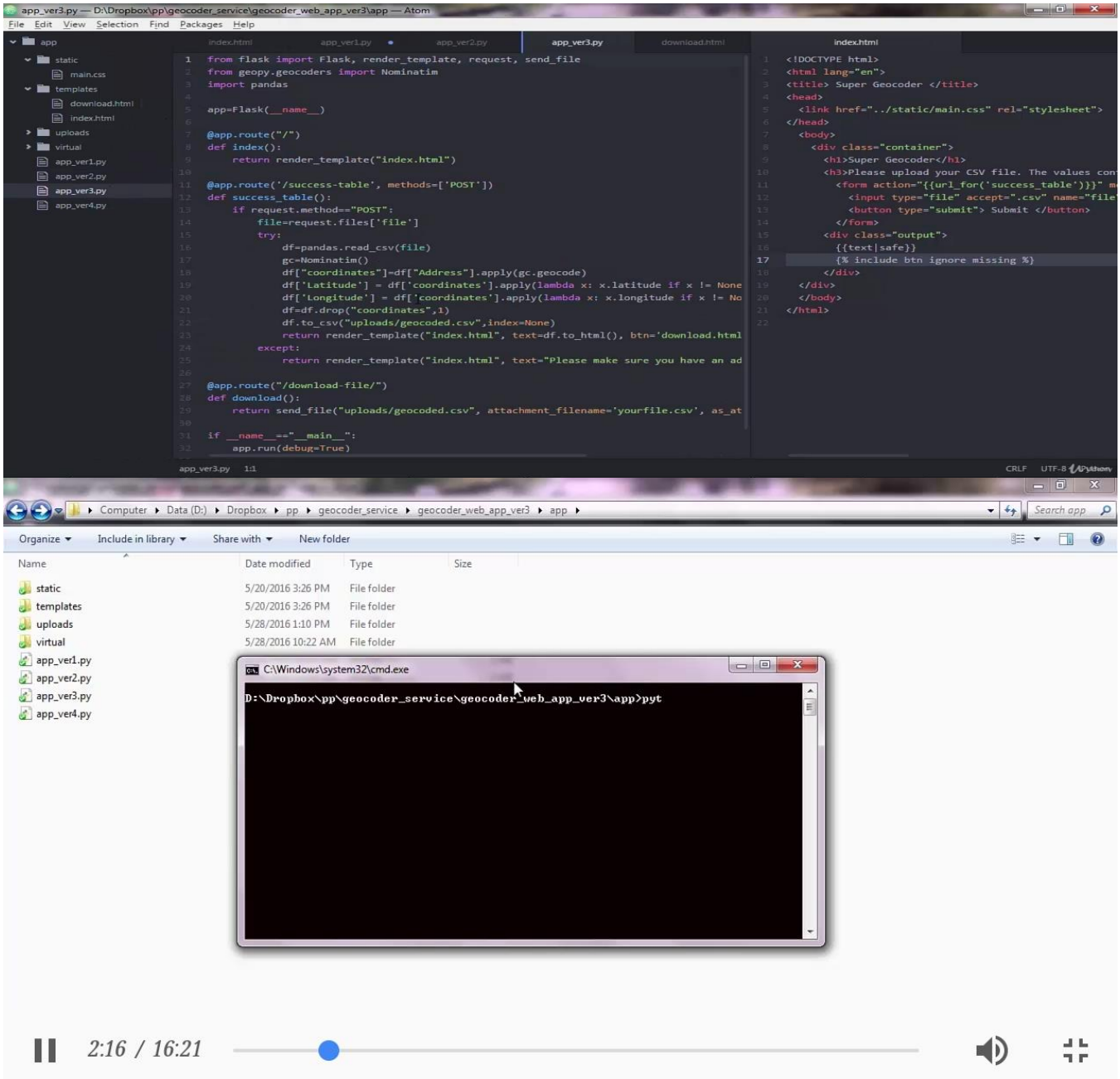
DAILY ASSESSMENT FORMAT

Date:	5th June 2020	**Name:**	Soundarya NA
Course:	UDEMY	**USN:**	4AL16EC077
Topic:	PYTHON: Application 11: Project Exercise on Building a Geocoder Web Service	**Semester & Section:**	8th - B

FORENOON SESSION DETAILS

Image:



Report:

Datasets are rarely complete and often require pre-processing. Imagine some datasets have only an address column without latitude and longitude columns to represent your data geographically. In that case, you need to convert your data into a geographic format. The process of converting addresses to geographic information — Latitude and Longitude — to map their locations is called Geocoding.

Geocoding is the computational process of transforming a physical address description to a location on the Earth's surface.

pip install geopandas

pip install geopy

Geocoding single address:

To geolocate a single address, you can use Geopy python library. Geopy has different Geocoding services that you can choose from, including Google Maps, ArcGIS, AzureMaps, Bing, etc. Some of them require API keys, while others do not need.

Code:

```
locator = Nominatim(user_agent="myGeocoder")
location = locator.geocode("Champ de Mars, Paris, France")
print("Latitude = {}, Longitude = {}".format(location.latitude, location.longitude))
```

Output:

Latitude = 48.85614465, Longitude = 2.29782039332223

Geocoding addresses for pandas:

```
df = pd.read_csv("addresses.csv")
df.head()
```

The following table provides the first five rows of the DataFrame table. As we can see, there are no latitude and longitude columns to map the data.

	Typ	Nr	Namn	Address1	Address3	Address4	Address5	Telefon
0	Butik	102	Fältöversten	Karlaplan 13	115 20	STOCKHOLM	Stockholms län	08/662 22 89
1	Butik	104	NaN	Nybrogatan 47	114 39	STOCKHOLM	Stockholms län	08/662 50 16
2	Butik	106	Garnisonen	Karlavägen 100 A	115 26	STOCKHOLM	Stockholms län	08/662 64 85
3	Butik	110	NaN	Hötorgshallen	111 57	STOCKHOLM	Stockholms län	08/56849241
4	Butik	113	Sergel	Drottninggatan 45	111 21	STOCKHOLM	Stockholms län	08/21 47 44

We concatenate address columns into one that is appropriate for geocoding. For example, the first address is:

Karlaplan 13,115 20,STOCKHOLM,Stockholms län, Sweden

Once we create the address column, we can start geocoding as below code snippet.

Code:

```
from geopy.extra.rate_limiter import RateLimiter
# 1 - conveneint function to delay between geocoding calls
geocode = RateLimiter(locator.geocode, min_delay_seconds=1)
# 2- - create location column
df['location'] = df['ADDRESS'].apply(geocode)
# 3 - create longitude, laatitude and altitude from location column (returns tuple)
df['point'] = df['location'].apply(lambda loc: tuple(loc.point) if loc else None)
# 4 - split point column into latitude, longitude and altitude columns
df[['latitude', 'longitude', 'altitude']] = pd.DataFrame(df['point'].tolist(), index=df.index)
```

	Typ	Nr	Namn	latitude	longitude	altitude
0	Butik	102	Fältöversten	59.338315	18.089960	0.0
1	Butik	104	NaN	59.337207	18.079098	0.0
2	Butik	106	Garnisonen	59.335380	18.100626	0.0
3	Butik	110	NaN	59.334327	18.062604	0.0
4	Butik	113	Sergel	59.332481	18.062809	0.0

Table with longitude and latitude

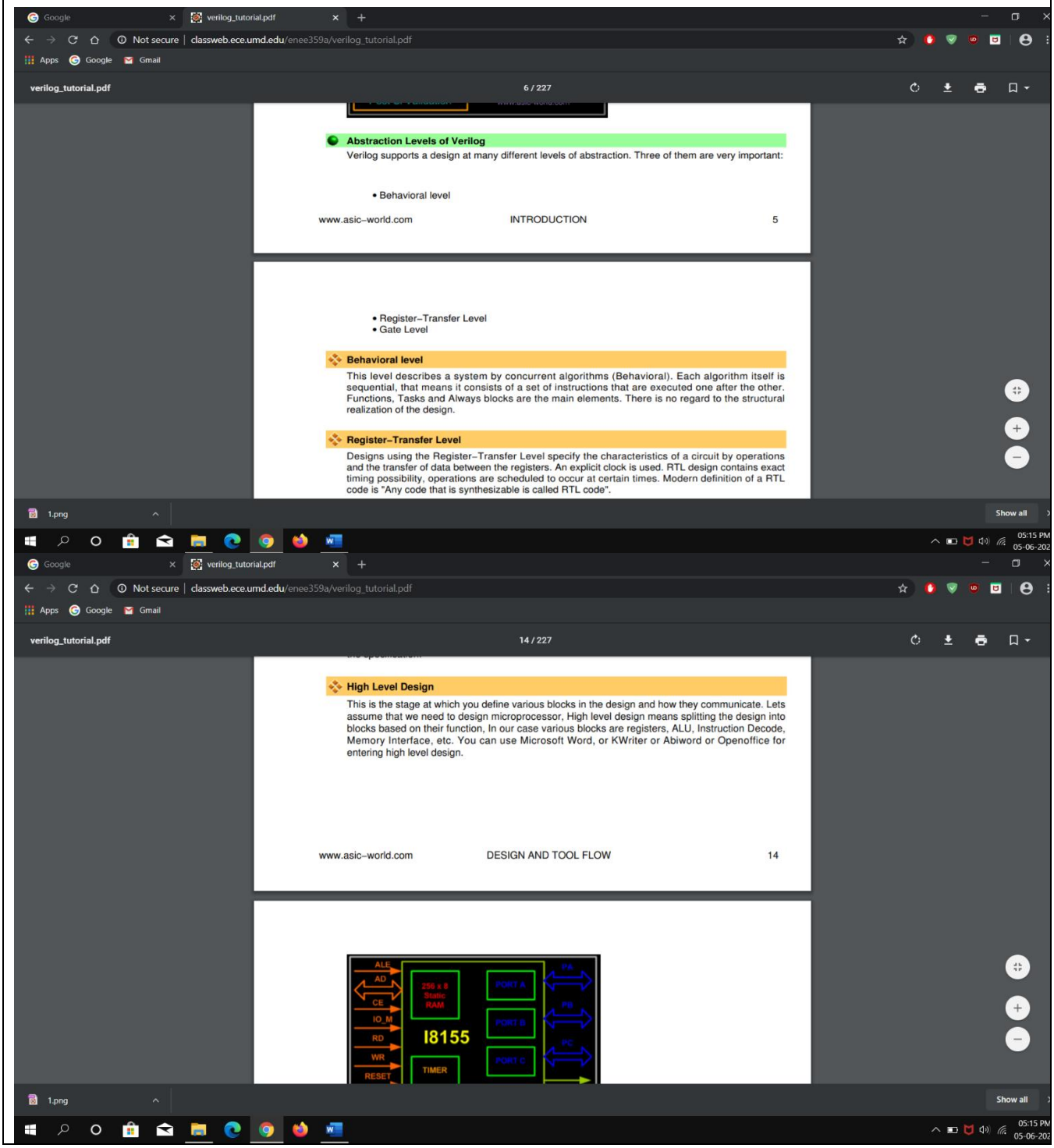
Code:

```
<html>
 <head>
 <meta name="viewport" content="initial-scale=1.0, width=device-width" />
 <script src="https://js.api.here.com/v3/3.1/mapsjs-core.js"type="text/javascript" charset="utf-8"></script>
 <script src="https://js.api.here.com/v3/3.1/mapsjs-service.js"type="text/javascript" charset="utf-8"></script>
 </head>
 <body style='margin: 0'>
 <div style="width: 100vw; height: 100vh" id="mapContainer"></div>
 <script>
  // Initialize the platform object:
  var platform = new H.service.Platform({
  'apikey': '{{apikey}}'
  });

        const lng = {{longitude}};
        const lat = {{latitude}};

// Obtain the default map types from the platform object
        var defaultLayers = platform.createDefaultLayers();

// Instantiate (and display) a map object:
var map = new H.Map(
  document.getElementById('mapContainer'),
  defaultLayers.vector.normal.map,
  {
   zoom: 10,
   center: { lat: lat, lng: lng }
  });
```

```
        const marker = new H.map.Marker({lat: lat, lng: lng});
        map.addObject(marker);
  </script>
  </body>
</html>
```

Date:	5th June 2020	**Name:**	Soundarya NA
Course:	Digital Design using HDL	**USN:**	4AL16EC077
Topic:	Verilog example programs	**Semester & Section:**	8th - B

Image:



Report:

Implement a verlog module to count number of 0's in a 16 bit number in compiler:

Code:

```
module num_ones_for(
   input [15:0] A,
   output reg [4:0] ones
   );

integer i;

always@(A)
begin
   ones = 0;  //initialize count variable.
   for(i=0;i<16;i=i+1)   //check for all the bits.
      if(A[i] == 1'b1)    //check if the bit is '1'
         ones = ones + 1;    //if its one, increment the count.
end

endmodule
```

Code:

```
module num_ones_for(
   input [15:0] A,
   output reg [4:0] ones
   );

integer i;

always@(A)
```

```
begin
   ones = 0;  //initialize count variable.
   for(i=0;i<16;i=i+1)   //for all the bits.
      ones = ones + A[i]; //Add the bit to the count.
end
```

Testbench:

```
module tb;

   // Inputs
   reg [15:0] A;

   // Outputs
   wire [4:0] ones;

   // Instantiate the Unit Under Test (UUT)
   num_ones_for uut (
      .A(A),
      .ones(ones)
   );

   initial begin
      A = 16'hFFFF;   #100;
      A = 16'hF56F;   #100;
      A = 16'h3FFF;   #100;
      A = 16'h0001;   #100;
      A = 16'hF10F;   #100;
      A = 16'h7822;   #100;
      A = 16'h7ABC;   #100;
   end
endmodule
```