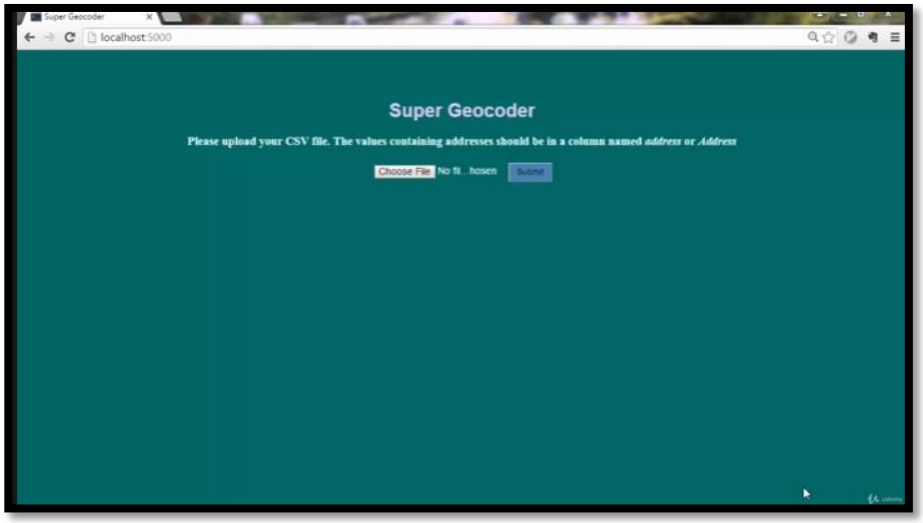
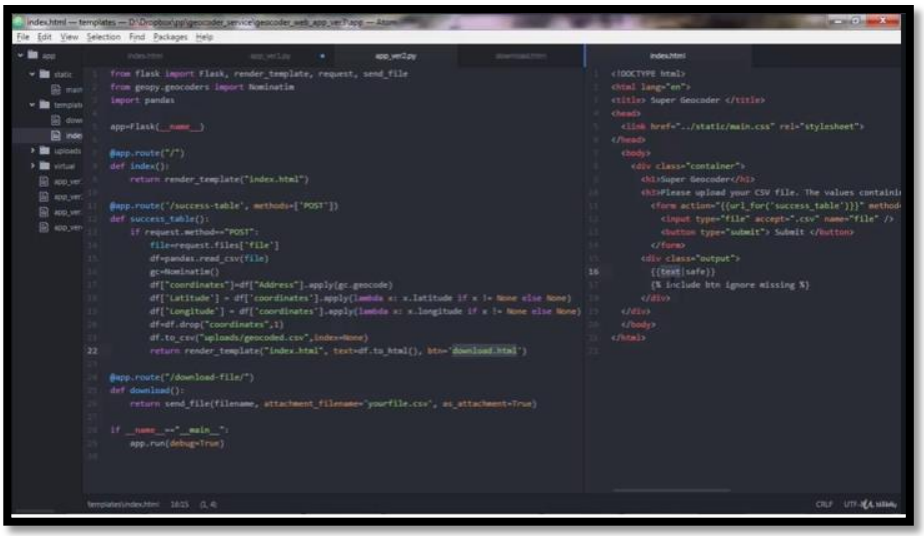
DAILY ASSESSMENT

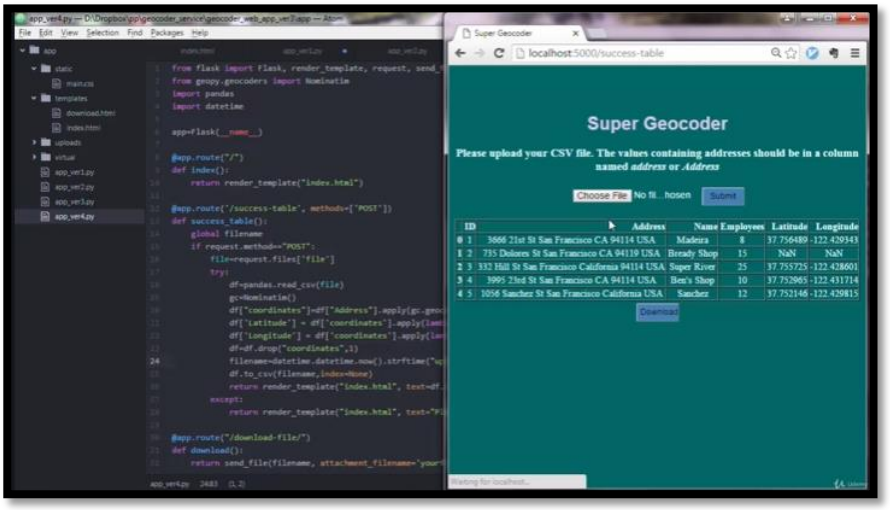
Date:	**6/06/2020**	**Name:**	**Davis S. Patel**
Course:	**Python Course**	**USN:**	**4AL16EC045**
Topic:	**Application 10: Project Exercise on Building a Geocoder Web Service**	**Semester & Section:**	**8th - A**
GitHub Repository:	**Davis**		

FORENOON SESSION DETAILS

Image of session







REPORT –

We often need to convert addresses to geographic locations (latitude and longitude), and this is called geocoding. There are several free geocoding API (with a limit of course) that you can use. In this tutorial, I will show you how to create the free geocoding application that you can drag and drop CSV files with address and get (download) a geocoded addresses as CSV.

We build the geocoding App with Python using Geopandas and Streamlit. Optionally you need an IDE like Visual studio code to run the App. Let us get started. We import first the libraries.

This GIF below shows a glimpse of what we are going to build. It will allow users to upload files and interact by choosing the right columns.

Streamlit Basics

The web app uses Streamlit. Streamlit is an easy to use web app building library purely in Python. I create a python file (app.py) which we are going to write our code.

Let us first importing the libraries we need

```
import time
import base64import streamlit as stimport pandas as pd
import geopandas as gpdimport geopy
from geopy.geocoders import Nominatim
from geopy.extra.rate_limiter import RateLimiterimport
plotly_express as px
```

We create first the headlines and run the App to test if it is working.

```
st.image("geocoding.jpg")
st.title("Geocoding Application in Python")
st.markdown("Uppload a CSV File with address columns (Street name &
number, Postcode, City)")
```

Streamlit uses a well-defined API which you can simply start using immediately. In the first line of the code, we display an image using st.image() . In the second line, we also show a test as tittle using st.tittle() . And finally, we show text using st.markdown() . Now, let us run the App.

Running Streamlit is as simple as writing on a terminal:

```
streamlit run app.py
```

Running the App will spin up a browser, and you can see the App is running if there are no errors. The image, the title and the text are there (See below image). We will continue working on this interface.

Upload CSV Files

To upload files, we can use st.file_upoader() . We create a function that allows us to interact with the local data using the st.file_upoader().

We create the main function, and inside it, we upload a CSV file. Once the CSV is uploaded, we can use Pandas to read the data and display the first few rows of the data. We will edit this main menu as we progress building the App. You can peak the final code for this function in the last section — the App.

Create or Choose Address columns

We need a probably formatted address column, and in this App, therefore we design so that it can accept a well-formatted column and geocode or create the address column from columns in the data. Here is an example of a properly formatted address. It has street name and number, postcode, the city and the country.

```
Karlaplan 13,115 20,STOCKHOLM, Sweden
```

The below two functions allow the user to select which option they want and later process the choice under the main menu function. The first one formats and creates an address column from DataFrame columns.

The second function below simply chooses a probably formatted column to use as an address column.

Geocode

We can start now geocoding, and below function uses Nominatim geocoder. The function returns a geocoded data frame with Latitude and Longitude columns.

Once we geocode the data, we can display it in a map. This below function uses the Plotly Express. To pass a figure to Streamlit, you can use st.plotly_chart() . Keep in mind also that you can use other libraries to plot your data.

Download Geocoded CSV File

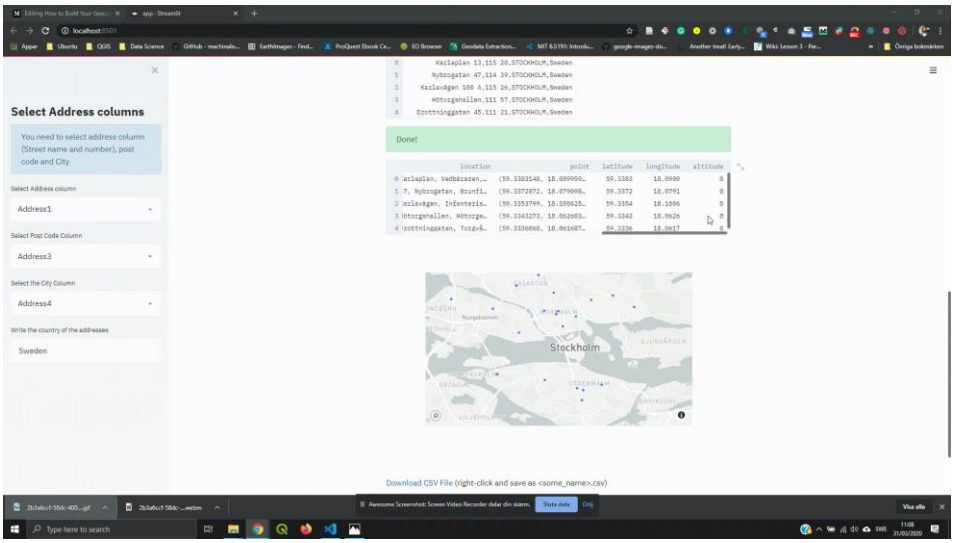
Once the data is geocoded, the App shows the data frame again with Latitudes and Longitudes. It would be nice also to be able to download the geocoded data.

To download the file, we can write the function below, and it allows us to right-click and save the file with a given name.

The App

Putting together all the code, the geocoding application code looks like this.

We can add some more functionality and build on top of this to allow other use cases if we want. Here is a glimpse of how to download the geocoded file in the App.



CERTIFICATE –