**DAILY ASSESSMENT 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 06-06-2020 | **Name:** | Sheela Golasangi |
| **Course:** | Python | **USN:** | 4AL16EC068 |
| **Topic:** | Application 10: Project Exercise on Building a Geocoder Web Service. | **Semester & Section:** | VIII  ‘B’ |
| **Github Repository:** | Sheela - Course |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of the session** |
| **REPORT**  Creating an API or Web application using python has been made easy with Flask. It is a micro web framework written in Python. Here you will create a python server using Flask, create database with PostgreSQL and deploy it on Heroku. We will create a simple application to store details of books and get stored data to demonstrate database transactions with our python server here.  Alexa is Amazon’s cloud-based voice service and the brain behind tens of millions of devices including the Echo family of devices, FireTV, Fire Tablet, and third-party devices with Alexa built-in. You can build voice experiences, or skills, that make everyday tasks faster, easier, and more delightful for customers.  Tens of thousands of developers have built skills using the Alexa Skills Kit (ASK), a collection of self-service APIs, tools, documentation, and code samples. With ASK, anyone can leverage Amazon’s knowledge in voice design to build quickly and easily. Start building today to reimagine your customer experience for voice and reach customers where they are. Streamlit Basics:  * The web app uses Streamlit. Streamlit is an easy to use web app building library purely in Python. * 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 from geopy.geocoders * Import Nominate from geopy.extra.rate\_limiter import Rate Limiter import plotly\_expressas 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.mark down (“Upploada CSV File with address columns (Street name &number, Postcode, City)”) * Reamlit 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 these condline,wealsoshowatest 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 (Seebelow 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 menuas weprogress building the App. * You can peak the final code forth is 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 Data Frame columns. * Thesecondfunctionbelowsimplychoosesaprobablyformattedcolumntouseasanaddress 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.    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 menuas weprogress building the App. * You can peak the final code forth is 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 Data Frame columns. * Thesecondfunctionbelowsimplychoosesaprobablyformattedcolumntouseasanaddress column.  Geocode:  * We can start now geocoding, and below function uses Nominatimgeocoder. * 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. |