**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **June 6, 2020** | **Name:** | **Persis P** |
| **Course:** | **Python** | **USN:** | **4AL17EC069** |
| **Topic:** | **Application 10: Project Exercise on Building a Geocoder Web Service** | **Semester & Section:** | **6th sem & B sec** |
| **GitHub Repository:** |  |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **from flask import Flask, render\_template, request, send\_file**  **from geopy.geocoders import ArcGIS**  **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=ArcGIS(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)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **June 6, 2020** | **Name:** | **Persis P** | |
| **Course:** | **Python** | **USN:** | **4AL17EC069** | |
| **Topic:** | **Section 33: legacy exercise** | **Semester & Section:** | **6th sem & B sec** | |
| **Git hub repository** |  |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**  **# Python 3 code to find sum**  **# of elements in given array**  **def \_sum(arr,n):**    **# return sum using sum**  **# inbuilt sum() function**  **return(sum(arr))**    **# driver function**  **arr=[]**  **# input values to list**  **arr = [12, 3, 4, 15]**    **# calculating length of array**  **n = len(arr)**    **ans = \_sum(arr,n)**    **# display sum**  **print ('Sum of the array is ', ans)**  **# Python3 code to demonstrate**  **# clearing a list using**  **# clear and Reinitializing**    **# Initializing lists**  **list1 = [1, 2, 3]**  **list2 = [5, 6, 7]**    **# Printing list1 before deleting**  **print ("List1 before deleting is : "**  **+ str(list1))**    **# deleting list using clear()**  **list1.clear()**    **# Printing list1 after clearing**  **print ("List1 after clearing using clear() : "**  **+ str(list1))**    **# Printing list2 before deleting**  **print ("List2 before deleting is : "**  **+ str(list2))**    **# deleting list using reinitialization**  **list2 = []**    **# Printing list2 after reinitialization**  **print ("List2 after clearing using reinitialization : "**  **+ str(list2))** | | | |