This folder consists of the code for the given assignment:

1.**function1.py**:

Input:url given the task

This file consists code of the function 1 given in the task.

In this code we are taking the url and reading the content in the url using **beautifulsoup** library of python for reading html files.Later we have extracted the link from the table using '**url.get(href)**'

From that url we had extracted date

Later find the type of the file

Finally it stores all the details in the dataframe and saves that dataframe to a pickle file(**fun1.pickle**)

Output:fun1.pickle

2.**function2.py**:

Input : Output of the first function (**fun1.pickle**)

This file consists code of function 2 given in the task.

In this code input will be the output of first function

From that first we will exempt the exempt or nonexempt files according to the condition given.

Later we will find the most recent file or the files in the given period according to the given condition

Finally it stores the data frame into a pickle file (**fun2.pickle**)

Output:fun2.pickle

3.t**odo1.py**:

Assumptions:Input will be output from second function (**fun2.pickle**).Assume that the top managers are formed according to the assets on the last day of the final year. '1O' column.Top managers will be the company ID

Studied about AUM from '**https://www.investopedia.com/terms/a/aum.asp**'

Input : Output of the first file(**fun2.pickle**)

Output : Top 15 Managers according to AUM

4.todo2.py:

Assumptions Employers and Managers count are included in the same column and rows with no state names are named under 'other' state name.Distribution would be **'The graph of the number of employees in that state vs State'**

Input : Output from second function (**fun2.pickle**)

Output : **The graph of the number of employees in that state vs State**

5.url_download.py:

Input : ‘todo1.py or todo2.py’

This program takes the url and downloads the zip file and extracts the contents of the zip file and returns the location of the extract file.

Output: Location of the extracted file

6.Algorithm of 3 to 5 Todo’s:

First extract all the company id’s from the files of the second url

First read the json file given in the 4th point of Todo’s and then create a dataframe from it containing the attributes [‘bc_scorce_id’ ,’score’ ,’bc_firm_name’, ’bc_branches_count’, ’bc_sec_number’,’bc_ia_scope’,’bc_ia_address_details’,’bc_other_names’]

Extract all the rows which has score>0.4

Extract all rows which has ‘BLACKSTONE’ in it

Check if the company id’s extracted is present in the json file

Store the bc_firm_name and bc_score_id in a data frame