**Assignment**

**CSA0814 – Python Programming**

|  |  |
| --- | --- |
| **Register Number** | **192311316** |
| **Name** | **K.krishna charan** |

**Title: File search utility**

**Problem Statement:** Create a Python program that automatically backs up specified files or directories to a remote server or cloud storage provider, ensuring data redundancy and disaster recovery.

**Code:**

# Import the following modules

**import** shutil

**from** datetime **import** date

**import** os

**import** sys

# When there is need, just change the directory

os.chdir(sys.path[0])

# Function for performing the

# backup of the files and folders

**def** take\_backup(src\_file\_name,

                dst\_file\_name**=**None,

                src\_dir**=**'',

                dst\_dir**=**''):

**try**:

          # Extract the today's date

        today **=** date.today()

        date\_format **=** today.strftime("%d\_%b\_%Y\_")

        # Make the source directory,

        # where we wanna backup our files

        src\_dir **=** src\_dir**+**src\_file\_name

        # If user not enter any source file,

        # then just give the error message...

**if** **not** src\_file\_name:

            print("Please give atleast the Source File Name")

            exit()

**try**:

            # If user provides all the inputs

**if** src\_file\_name **and** dst\_file\_name **and** src\_dir **and** dst\_dir:

                src\_dir **=** src\_dir**+**src\_file\_name

                dst\_dir **=** dst\_dir**+**dst\_file\_name

            # When User Enter Either

            # 'None' or empty String ('')

**elif** dst\_file\_name **is** None **or** **not** dst\_file\_name:

                dst\_file\_name **=** src\_file\_name

                dst\_dir **=** dst\_dir**+**date\_format**+**dst\_file\_name

            # When user Enter an empty

            # string with one or more spaces (' ')

**elif** dst\_file\_name.isspace():

                dst\_file\_name **=** src\_file\_name

                dst\_dir **=** dst\_dir**+**date\_format**+**dst\_file\_name

            # When user Enter an a

            # name for the backup copy

**else**:

                dst\_dir **=** dst\_dir**+**date\_format**+**dst\_file\_name

            # Now, just copy the files

            # from source to destination

            shutil.copy2(src\_dir, dst\_dir)

            print("Backup Successful!")

**except** FileNotFoundError:

            print("File does **not** exists!,\

            please give the complete path")

    # When we need to backup the folders only...

**except** PermissionError:

        dst\_dir **=** dst\_dir**+**date\_format**+**dst\_file\_name

        # Copy the whole folder

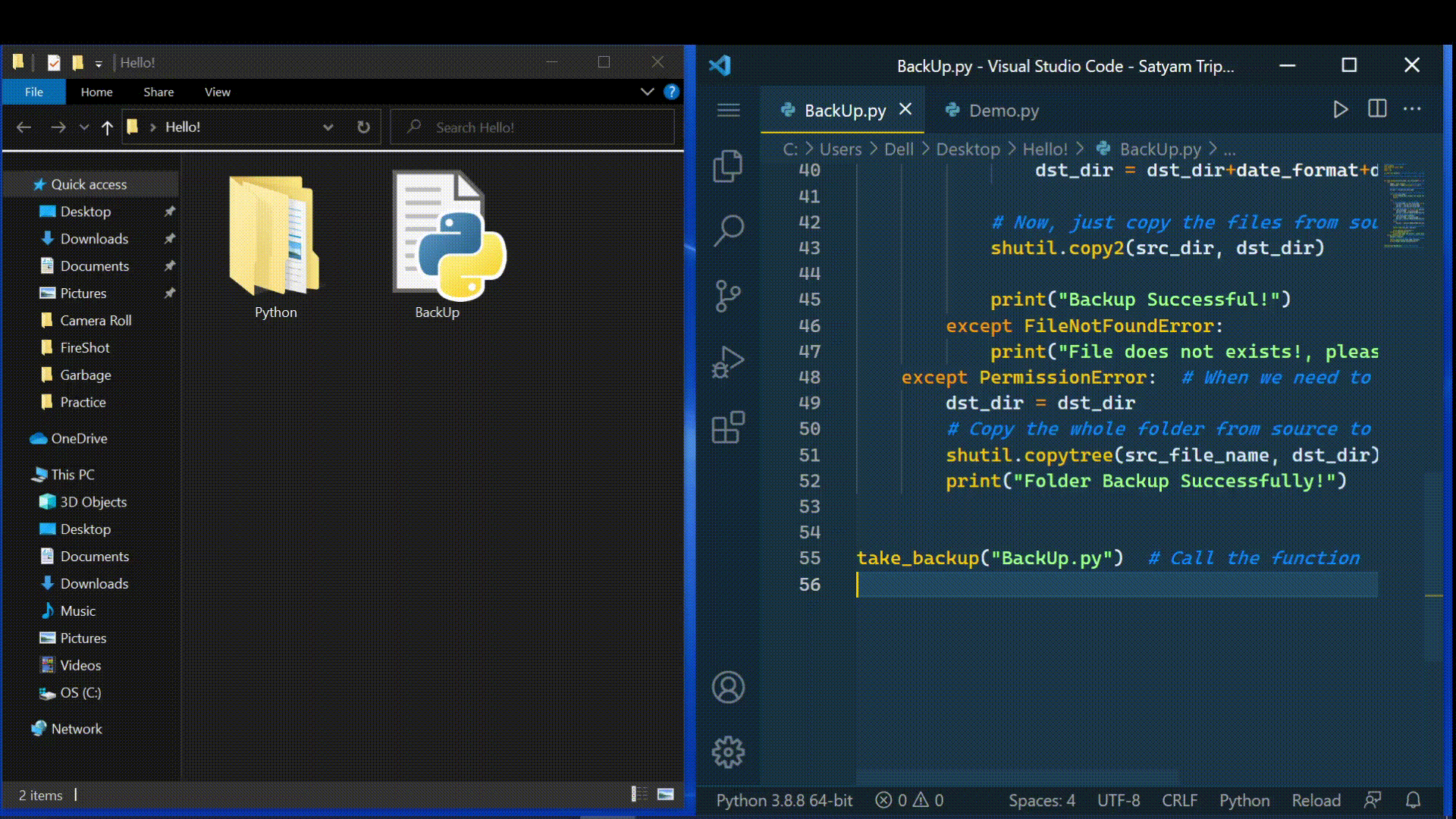
        # from source to destination

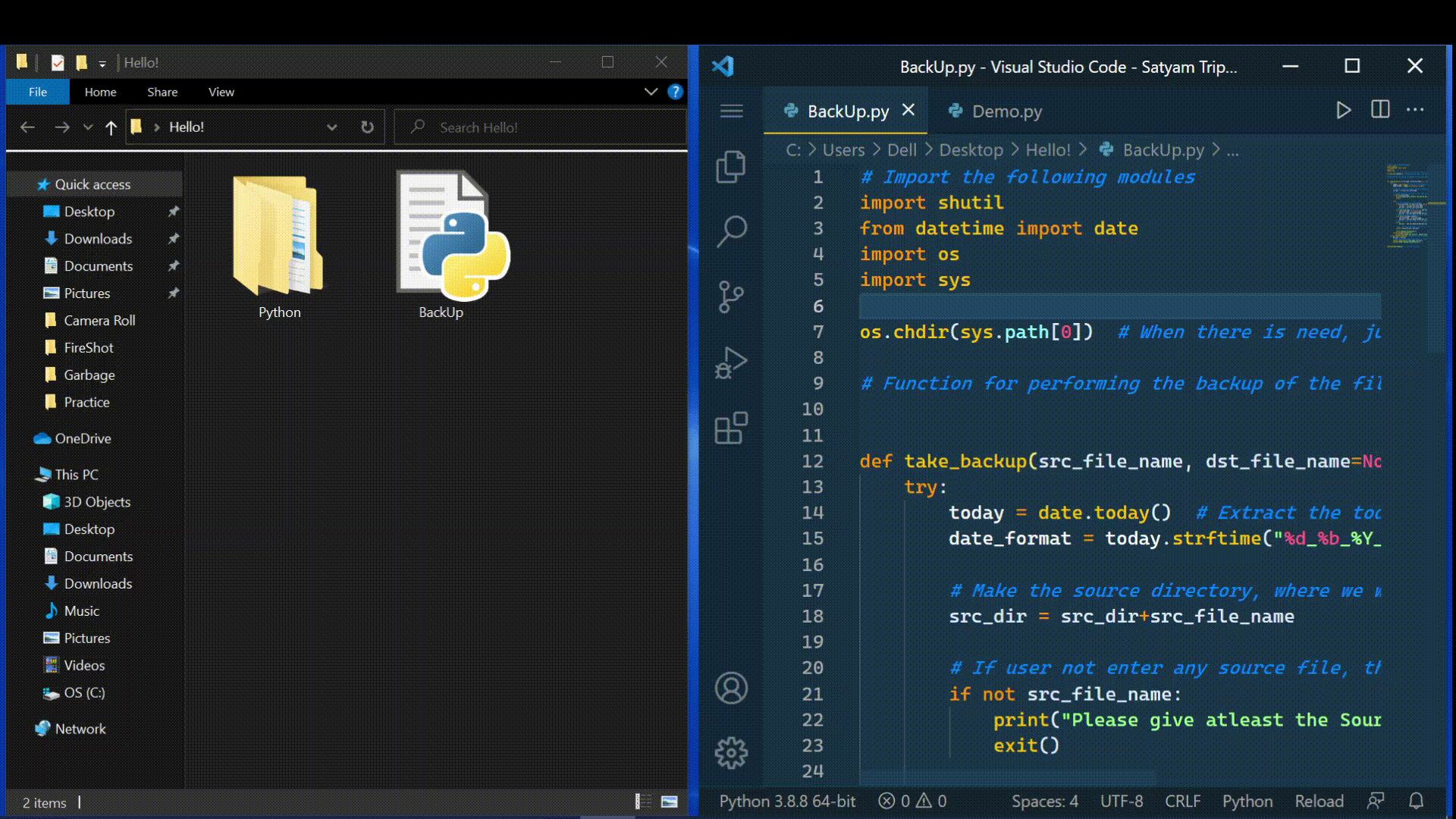
        shutil.copytree(src\_file\_name, dst\_dir)

# Call the function

take\_backup("GFG.txt")

**Output Screen Shots:**





**Conclusion:**

file backup tool enhances data protection through automation, efficient file management, and integration with cloud storage solutions. It provides a robust framework for ensuring data redundancy and effective disaster recovery, ultimately contributing to a more resilient data management strategy.