```
# Name: Kulpreet Singh
# Roll No: 101803186
# Group: 3COE9
# Assignment-1 UCS531 (Cloud Computing)

# --------------------------------------------------------------------------------
# QUESTION
# Write a python program to create a backup of any folder from your PC to the cloud.
# This backup can be done in one of the following ways:
# 1) Backing up the files from source to destination after a fixed time period (e.g.
# 24 hours).
# 2) Comparing the contents/file (for eg. size of a file, last_modified_time etc.)
# of source and destination and if there are any changes in source folder, it will
# trigger the backup process.
# Also maintain the logs of a backup in a separate file (name it log.txt) in the
# following format:
# dd/mm/yyyy hh:mm:ss backup_status file_location/ issue (in case of backup failure)
# 01/01/2021 12:10:30 backup_successful C:\Users\debianPC\Desktop\bkup\sample.txt'
# 01/01/2021 12:10:30 backup_failed Network Issue/ File not Found or any other
# ……………………………...
# ……………………………...

# ----------------------------------------------------------------------------------
# APPROACH USED
# Here we are following the first way, i.e. backing up from source to destination
# after a fixed time period (24 hours) by using windows task scheduler

# Using this python script we create the backup on google drive as a cloud backup
# I have referred official docs from google. Following is the link to same:
# https://developers.google.com/drive/api/v3/quickstart/python

# ----------------------------------------------------------------------------------
# HOW TO CREATE A NEW PROJECT OVER GOOGLE CLOUD AND ENABLE DRIVE ACCESS
# - In order to run this script, we first need to create a new project over the GOOGLE
#   CLOUD PLATFORM, enter the required details in the project. One can refer the
#   following link: https://developers.google.com/workspace/guides/create-project
# - Then enable the GOOGLE DRIVE API for that project
# - Then we need to create and download authorization credentials for a desktop
#   application. Place the credentials.json in the working directory.

# ----------------------------------------------------------------------------------
# HOW TO RUN
# - Firstly, change the path to your source folder and also change the destination
#   drive folder id in the main()
# - Run the following command:
# pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib
# - Also import any other necessary library
# - Then run this script. You will be redirected to your browser where in you need to
#   sign in to your google drive account and allow access to your drive.
# - Finally, one can set up this script to run at a particular time of the day using
#   WINDOWS TASK SCHEDULER

# ----------------------------------------------------------------------------------
```

Roll No: 101803186 **Name: Kulpreet Singh** **Group: 3COE9**

```python
# RESULTS
# - Backup is created over the users drive in the given folder whose ID is entered by the
#   user in this script.
# - log.txt file is created in this projects working directory which contains the logs in
#   desired format

# -----------------------------------------------------------------------------------------
# CODE
# All necessary imports
import pickle
import os.path
from googleapiclient.discovery import build
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request
from apiclient.http import MediaFileUpload
import sys
import os
import datetime


class MyDrive():
    def __init__(self):
        """
        initialises instance of MyDrive class with user's drive for the
         read/write access.
        """
        # scopes signify the access permissions (read/write) to the users drive
        # If modifying these scopes, delete the file token.pickle.
        SCOPES = ['https://www.googleapis.com/auth/drive']
        # creds store the cloud credentials of user (e.g. client_id, secret_key)
        creds = None
        # The file token.pickle stores the user's access and refresh tokens, and is
        # created automatically when the authorization flow completes for the first
        # time.
        # If token.pickle file already exists, then load the credentials from it
        if os.path.exists('token.pickle'):
            with open('token.pickle', 'rb') as token:
                creds = pickle.load(token)
        # If there are no (valid) credentials available, let the user log in.
        if not creds or not creds.valid:
            if creds and creds.expired and creds.refresh_token:
                creds.refresh(Request())
            else:
                flow = InstalledAppFlow.from_client_secrets_file(
                    'KulpreetSingh_101803186_credentials.json', SCOPES)
                creds = flow.run_local_server(port=0)
            # Save the credentials for the next run
            with open('token.pickle', 'wb') as token:
                pickle.dump(creds, token)
        # service object is used to access the files in the drive
        self.service = build('drive', 'v3', credentials=creds)
    # We didn't use list_files function in this script but can be useful for
    # checking in case of failure in backup due to network failure
    def list_files(self, page_size=10):
        """
        Prints the names and ids of the first 10 files the user has access to.
        """
        # Call the Drive v3 API to list the file names
        results = self.service.files().list(
            pageSize=page_size,
            fields="nextPageToken, files(id, name)"
        ).execute()
        items = results.get('files', [])

        if not items:
            print('No files found.')
        else:
            print('Files:')
            for item in items:
                print(u'{0} ({1})'.format(item['name'], item['id']))


    # upload_file uploads file to the drive folder whose folder_id
    # is specified inside it and updates the logs
    def upload_file(self, filename, path, folder_id):
        """
        Uploads a file from user's local system to the drive folder.
        Updates the logs in the log.txt file
        """
        # open log.txt
        f = open('log.txt', 'a')

        try:
            # file stored as a media object to be uploaded
            media = MediaFileUpload(f"{path}{filename}")

            # query to find the files in the given directory with same name
            # and same parent folder
            response = self.service.files().list(
                q=f"name='{filename}' and parents='{folder_id}'",
                spaces='drive',
                fields='nextPageToken, files(id, name)',
                pageToken=None
            ).execute()

            # if the file doesn't already exist
            if len(response['files']) == 0:
                file_metadata = {
                    'name': filename,
                    'parents': [folder_id]
                }
                # create a new file in the drive folder
                file = self.service.files().create(
                    body=file_metadata,
                    media_body=media,
                    fields='id'
                ).execute()
                # update log.txt for successful execution
                x = datetime.datetime.now()
                timestamp = x.strftime("%d/%m/%Y %X")
                f.write(f"{timestamp} backup_successful {path}{filename}\n")
                print(f"{timestamp} backup_successful {path}{filename}")
            # if file already exists
            else:
                for file in response.get('files', []):
                    # update the file contents
                    update_file = self.service.files().update(
                        fileId=file.get('id'),
                        media_body=media,
                    ).execute()
                    # update log.txt for successful execution
                    x = datetime.datetime.now()
                    timestamp = x.strftime("%d/%m/%Y %X")
                    f.write(f"{timestamp} backup_successful {path}{filename}\n")
                    print(f"{timestamp} backup_successful {path}{filename}")
        except:
            # if there is an error/exception, update the log.txt for backup_failed
            x = datetime.datetime.now()
            timestamp = x.strftime("%d/%m/%Y %X")
            f.write(f"{timestamp} backup_failed {sys.exc_info()[0].__name__}\n")
            print(f"{timestamp} backup_failed {sys.exc_info()[0].__name__}")
        finally:
            # close log.txt
            f.close()


def main():
    # Enter the path of the folder whose backup you want to create
    path = "C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/"
    files = os.listdir(path)

    # creating a MyDrive object
    my_drive = MyDrive()

    # drive folder's id which is used for backup
    # Create a folder in your drive and enter the folder id here
    folder_id = "1f85g6l5-9bKgVv-nFK4rLl1qulpplQwU"

    # uploading files
    for item in files:
        my_drive.upload_file(item, path, folder_id)


if __name__ == '__main__':
    main()
```

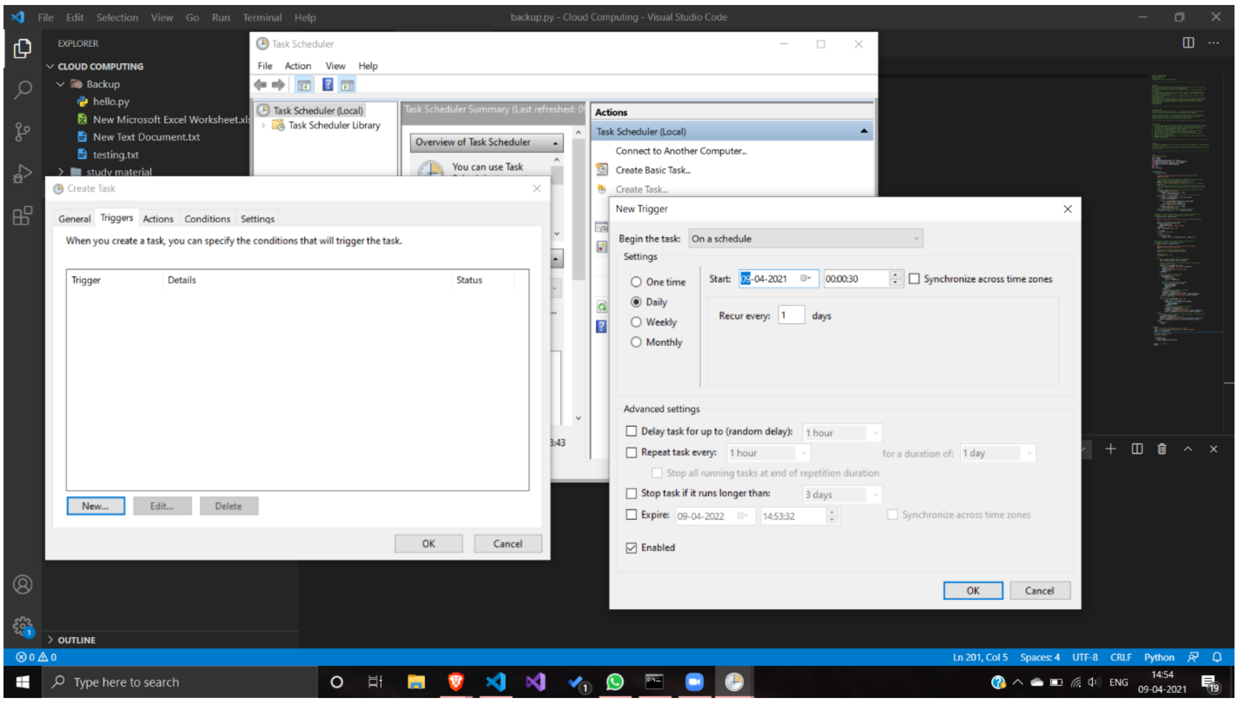
Roll No: 101803186 **Name: Kulpreet Singh** **Group: 3COE9**

HOW TO RUN

- Firstly, change the path to your source folder and also change the destination drive folder id in the main(). (For example: https://drive.google.com/drive/u/2/folders/1DFwVl3O0ABEeXu3eQouuv-4zuWkxs5O7 here 1DFwVl3O0ABEeXu3eQouuv-4zuWkxs5O7 is the folder_id.)
- Run the following command:

```
pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib
```

- Also import any other necessary library
- Then run this script. You will be redirected to your browser where in you need to sign in to your google drive account and allow access to your drive.
- Finally, one can set up this script to run at a particular time of the day using WINDOWS TASK SCHEDULER



New Trigger

Begin the task: On a schedule

Settings

One time [✓ Daily] Weekly Monthly

Start: 09-04-2021 00:00:30 ☐ Synchronize across time zones

Recur every: 1 days

Advanced settings

☐ Delay task for up to (random delay): 1 hour

☐ Repeat task every: 1 hour for a duration of: 1 day

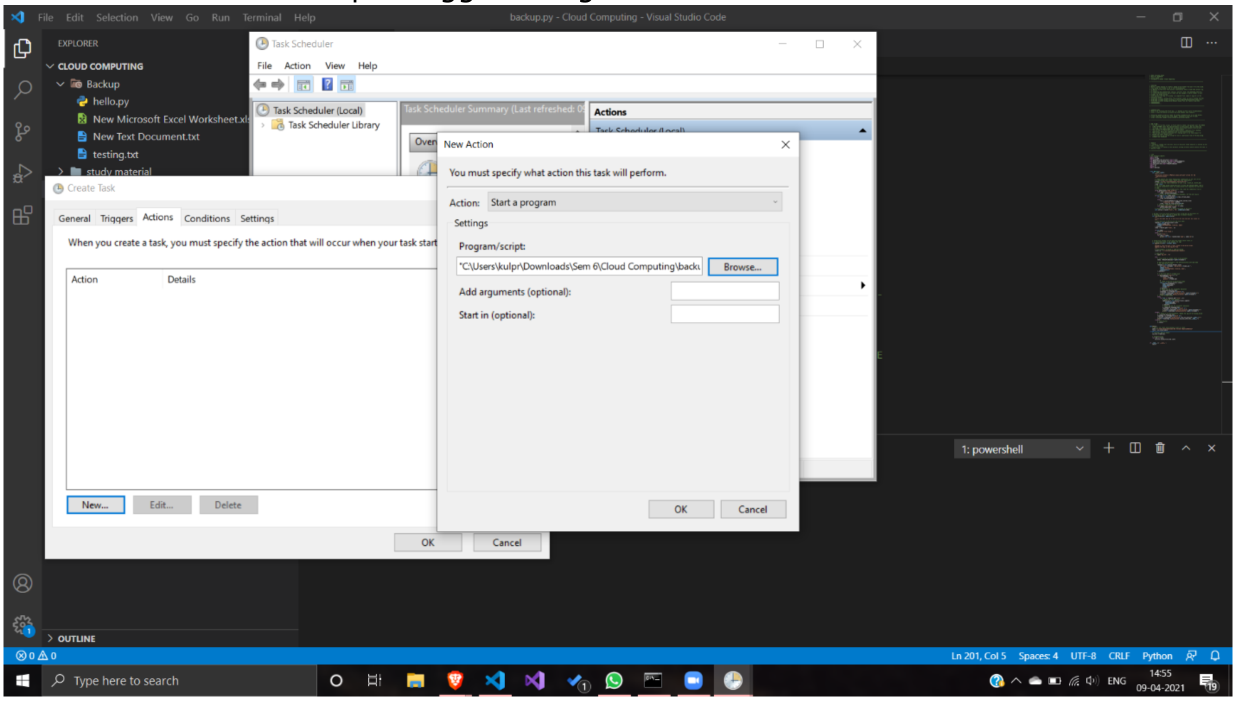
☐ Stop all running tasks at end of repetition duration

☐ Stop task if it runs longer than: 3 days

☐ Expire: 09-04-2022 14:53:32 ☐ Synchronize across time zones

☑ Enabled

- This is how we set up a trigger using windows task scheduler.



New Action

You must specify what action this task will perform.

Action: Start a program

Settings

Program/script: "C:\Users\kulpr\Downloads\Sem 6\Cloud Computing\backu

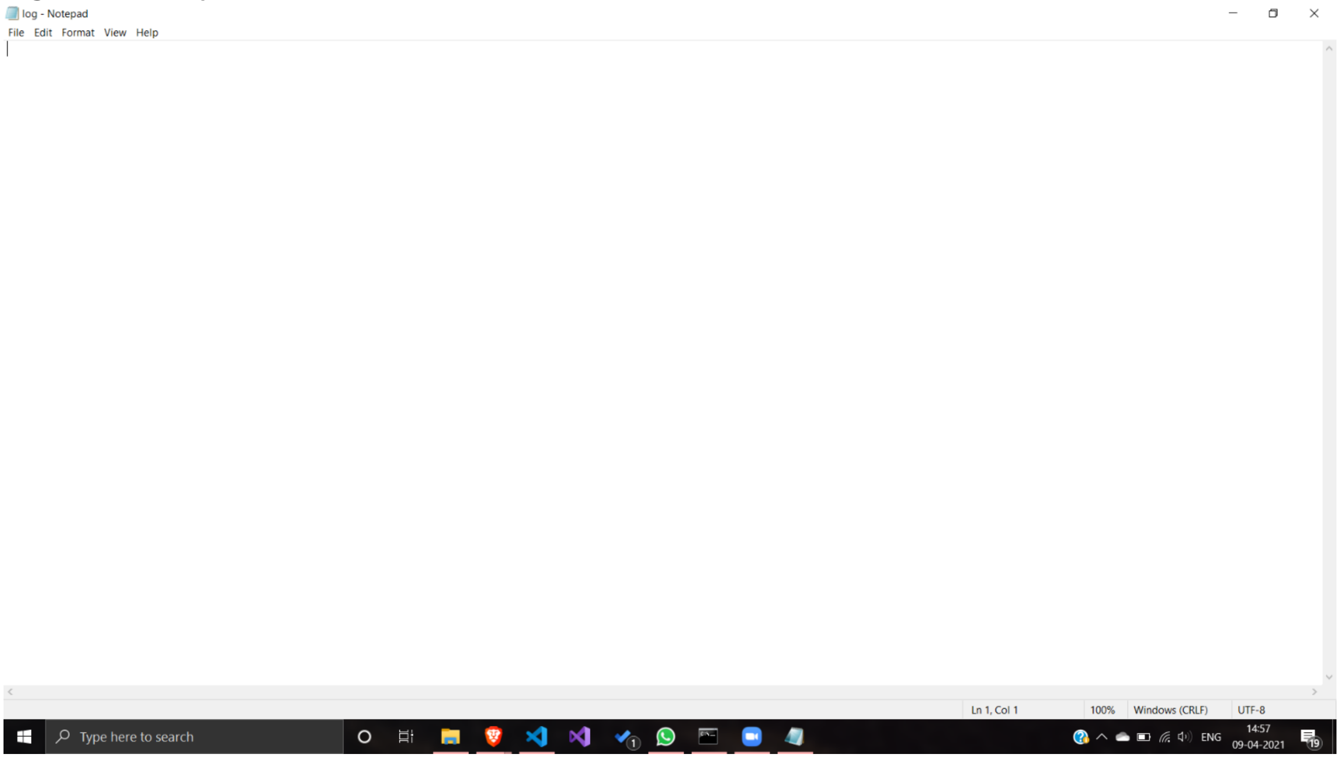
Add arguments (optional):

Start in (optional):

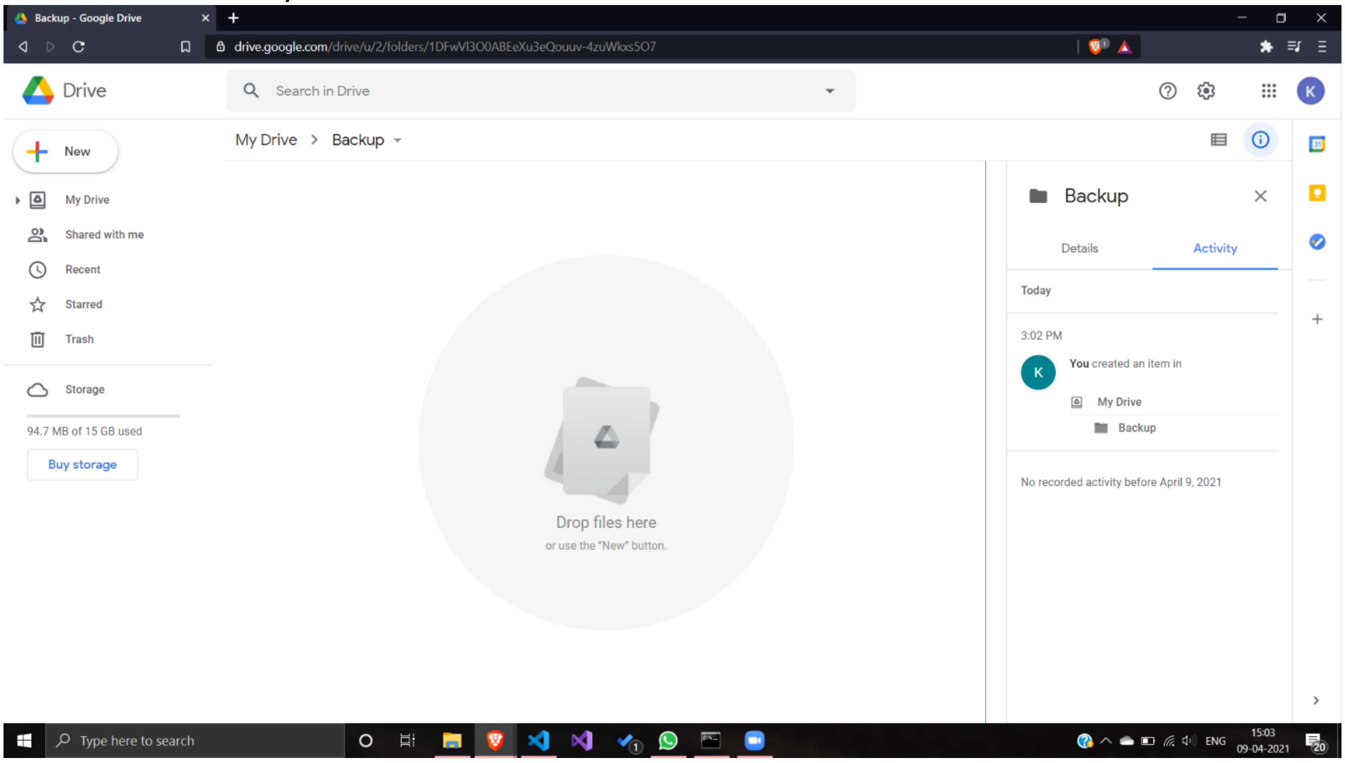
- This is where we specify what action is to be performed which in our case is to run the script backup.py.

Roll No: 101803186 **Name: Kulpreet Singh** **Group: 3COE9**

- log.txt initially

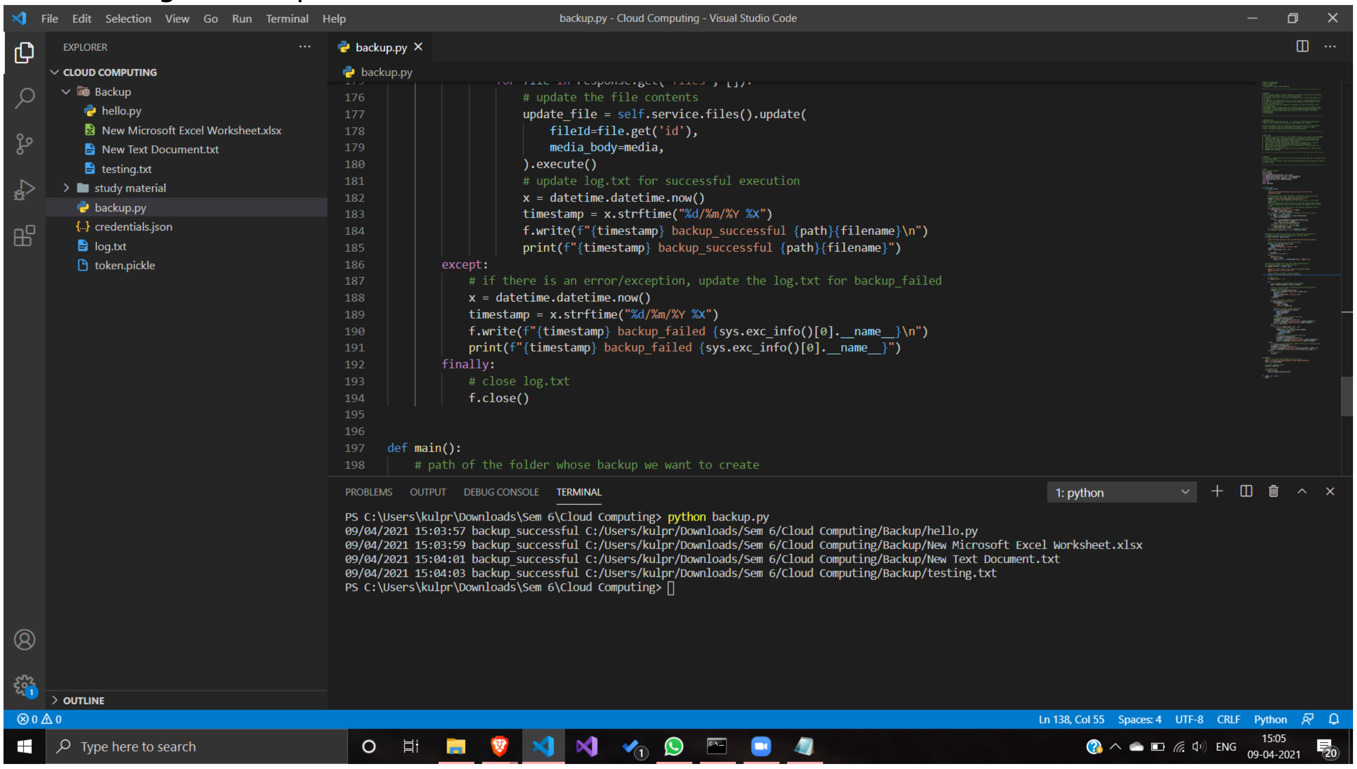


- Drive Folder initially



Roll No: 101803186 **Name: Kulpreet Singh** **Group: 3COE9**

- On Running the script



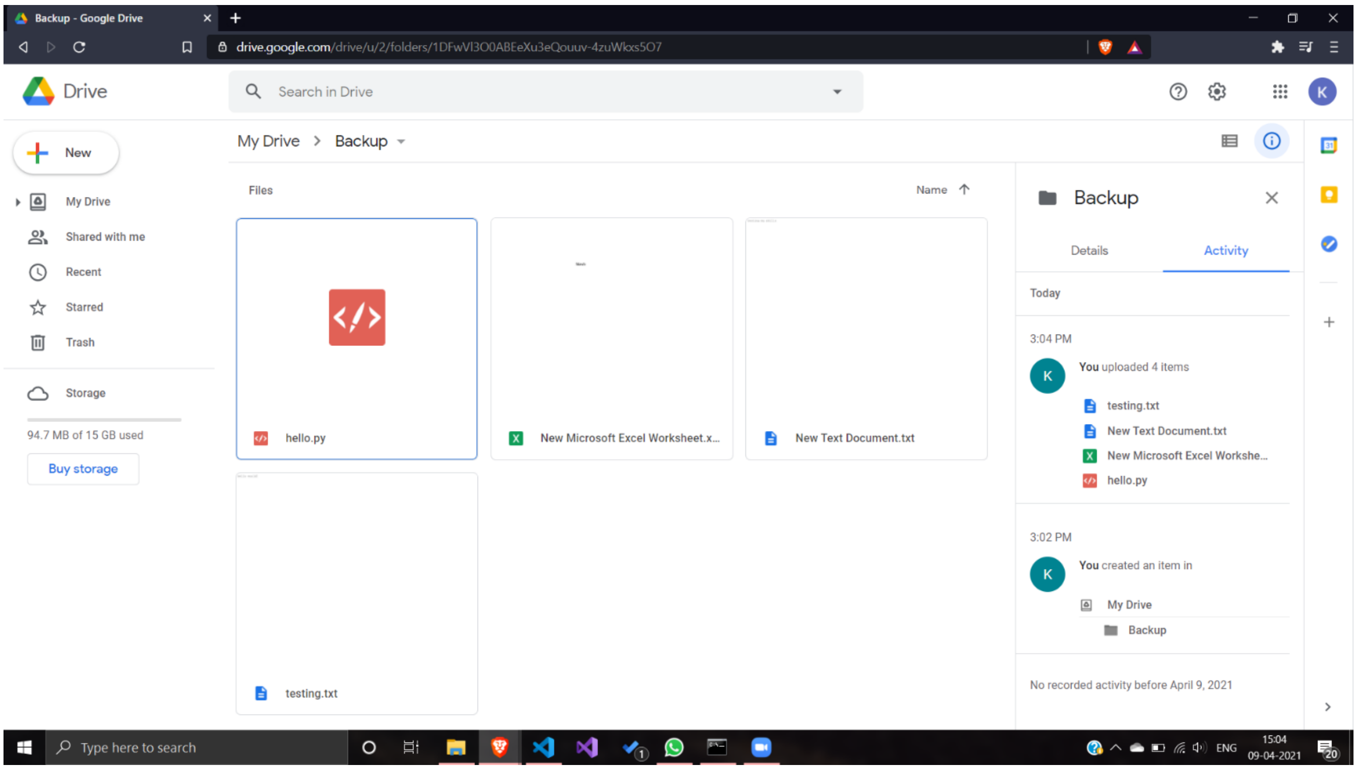
```
        # update the file contents
                    update_file = self.service.files().update(
                        fileId=file.get('id'),
                        media_body=media,
                    ).execute()
                    # update log.txt for successful execution
                    x = datetime.datetime.now()
                    timestamp = x.strftime("%d/%m/%Y %X")
                    f.write(f"{timestamp} backup_successful {path}{filename}\n")
                    print(f"{timestamp} backup_successful {path}{filename}")
        except:
            # if there is an error/exception, update the log.txt for backup_failed
            x = datetime.datetime.now()
            timestamp = x.strftime("%d/%m/%Y %X")
            f.write(f"{timestamp} backup_failed {sys.exc_info()[0].__name__}\n")
            print(f"{timestamp} backup_failed {sys.exc_info()[0].__name__}")
        finally:
            # close log.txt
            f.close()


def main():
    # path of the folder whose backup we want to create
```

```
PS C:\Users\kulpr\Downloads\Sem 6\Cloud Computing> python backup.py
09/04/2021 15:03:57 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/hello.py
09/04/2021 15:03:59 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/New Microsoft Excel Worksheet.xlsx
09/04/2021 15:04:01 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/New Text Document.txt
09/04/2021 15:04:03 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/testing.txt
PS C:\Users\kulpr\Downloads\Sem 6\Cloud Computing> 
```

backup.py - Cloud Computing - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
CLOUD COMPUTING
Backup
hello.py
New Microsoft Excel Worksheet.xlsx
New Text Document.txt
testing.txt
study material
backup.py
credentials.json
log.txt
token.pickle
OUTLINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: python
Ln 138, Col 55 Spaces: 4 UTF-8 CRLF Python
Type here to search
15:05
09-04-2021

- Drive Folder



Backup - Google Drive
drive.google.com/drive/u/2/folders/1DFwVl3O0ABEeXu3eQouuv-4zuWkxs5O7

My Drive > Backup

Files

hello.py
New Microsoft Excel Worksheet.x...
New Text Document.txt
testing.txt

Backup

Details Activity

Today

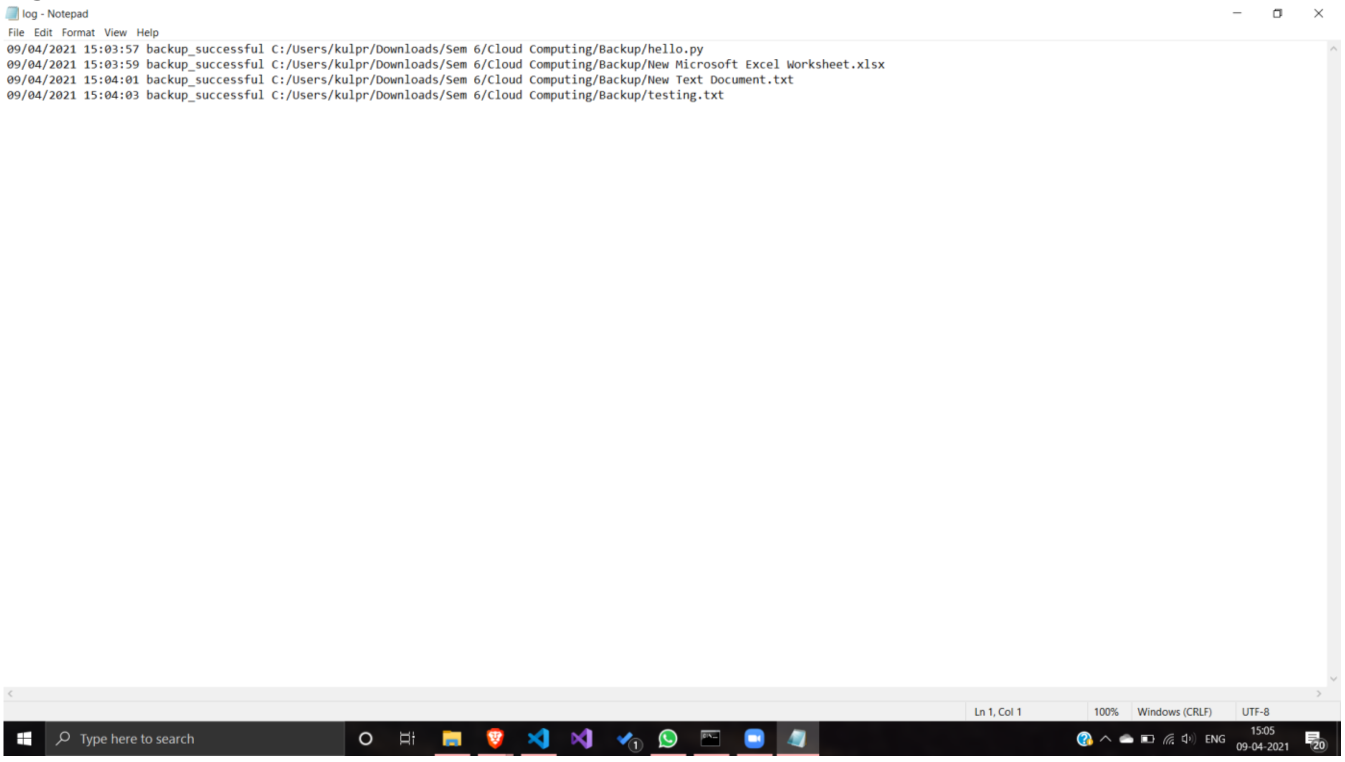
3:04 PM
You uploaded 4 items
testing.txt
New Text Document.txt
New Microsoft Excel Workshe...
hello.py

3:02 PM
You created an item in
My Drive
Backup

No recorded activity before April 9, 2021

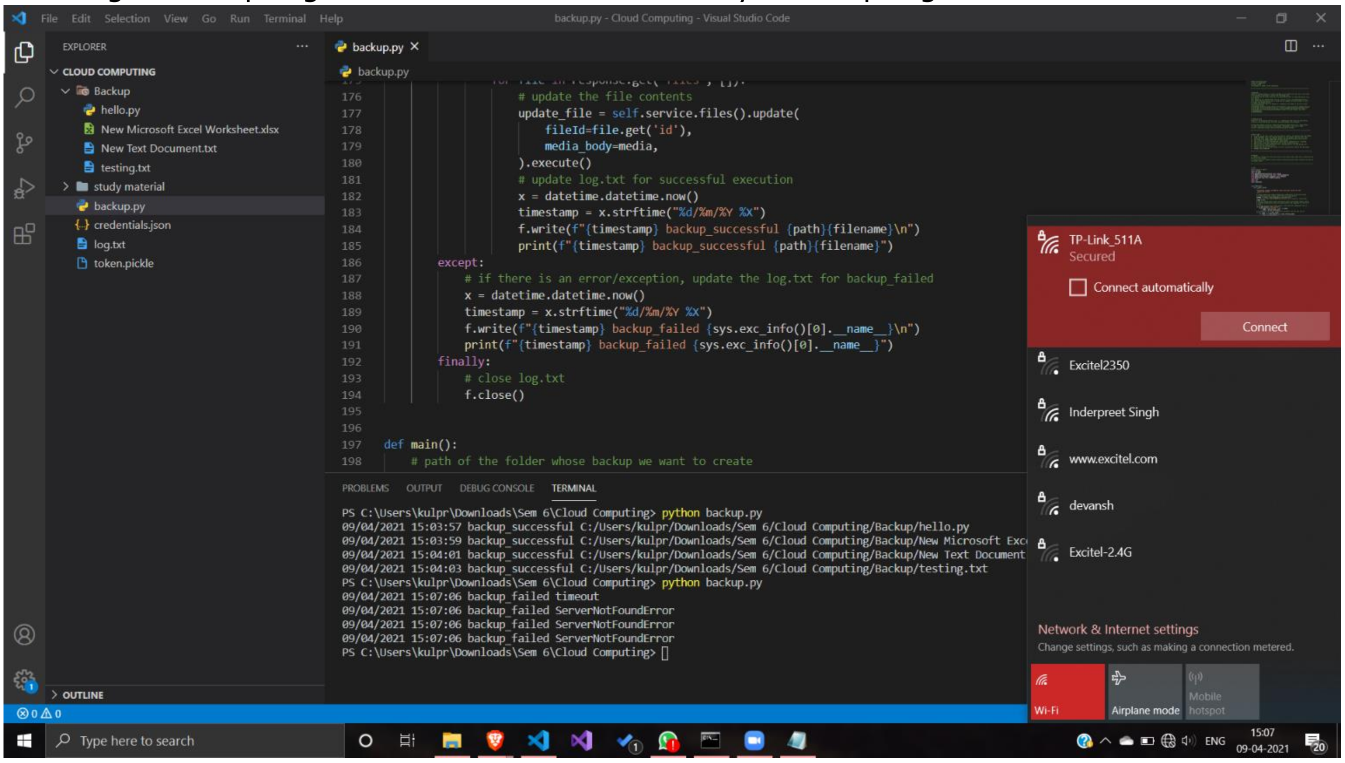
Roll No: 101803186 **Name: Kulpreet Singh** **Group: 3COE9**

- log.txt



```
09/04/2021 15:03:57 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/hello.py
09/04/2021 15:03:59 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/New Microsoft Excel Worksheet.xlsx
09/04/2021 15:04:01 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/New Text Document.txt
09/04/2021 15:04:03 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/testing.txt
```

- Running the script again but this time intentionally interrupting the network



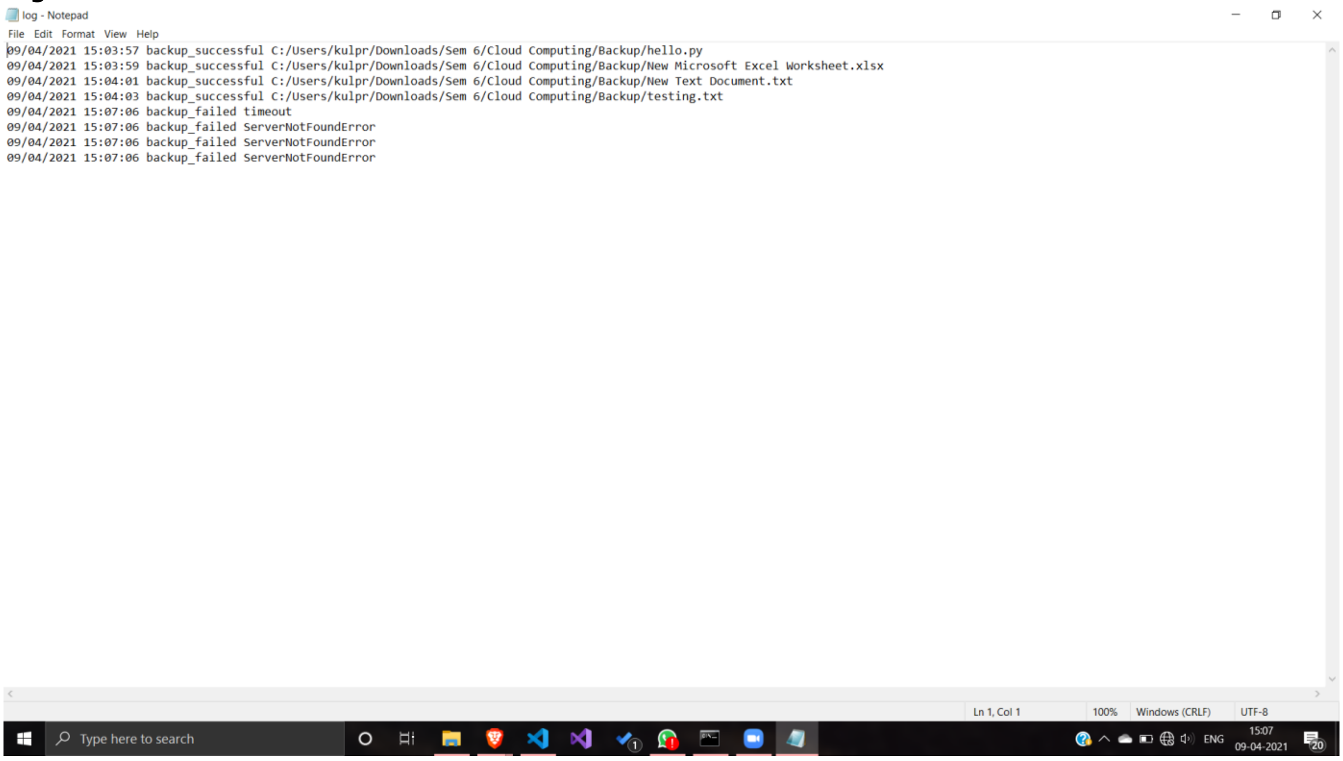
```python
                    # update the file contents
                    update_file = self.service.files().update(
                        fileId=file.get('id'),
                        media_body=media,
                    ).execute()
                    # update log.txt for successful execution
                    x = datetime.datetime.now()
                    timestamp = x.strftime("%d/%m/%Y %X")
                    f.write(f"{timestamp} backup_successful {path}{filename}\n")
                    print(f"{timestamp} backup_successful {path}{filename}")
        except:
            # if there is an error/exception, update the log.txt for backup_failed
            x = datetime.datetime.now()
            timestamp = x.strftime("%d/%m/%Y %X")
            f.write(f"{timestamp} backup_failed {sys.exc_info()[0].__name__}\n")
            print(f"{timestamp} backup_failed {sys.exc_info()[0].__name__}")
        finally:
            # close log.txt
            f.close()


def main():
    # path of the folder whose backup we want to create
```

```
PS C:\Users\kulpr\Downloads\Sem 6\Cloud Computing> python backup.py
09/04/2021 15:03:57 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/hello.py
09/04/2021 15:03:59 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/New Microsoft Exc
09/04/2021 15:04:01 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/New Text Document
09/04/2021 15:04:03 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/testing.txt
PS C:\Users\kulpr\Downloads\Sem 6\Cloud Computing> python backup.py
09/04/2021 15:07:06 backup_failed timeout
09/04/2021 15:07:06 backup_failed ServerNotFoundError
09/04/2021 15:07:06 backup_failed ServerNotFoundError
09/04/2021 15:07:06 backup_failed ServerNotFoundError
PS C:\Users\kulpr\Downloads\Sem 6\Cloud Computing> 
```

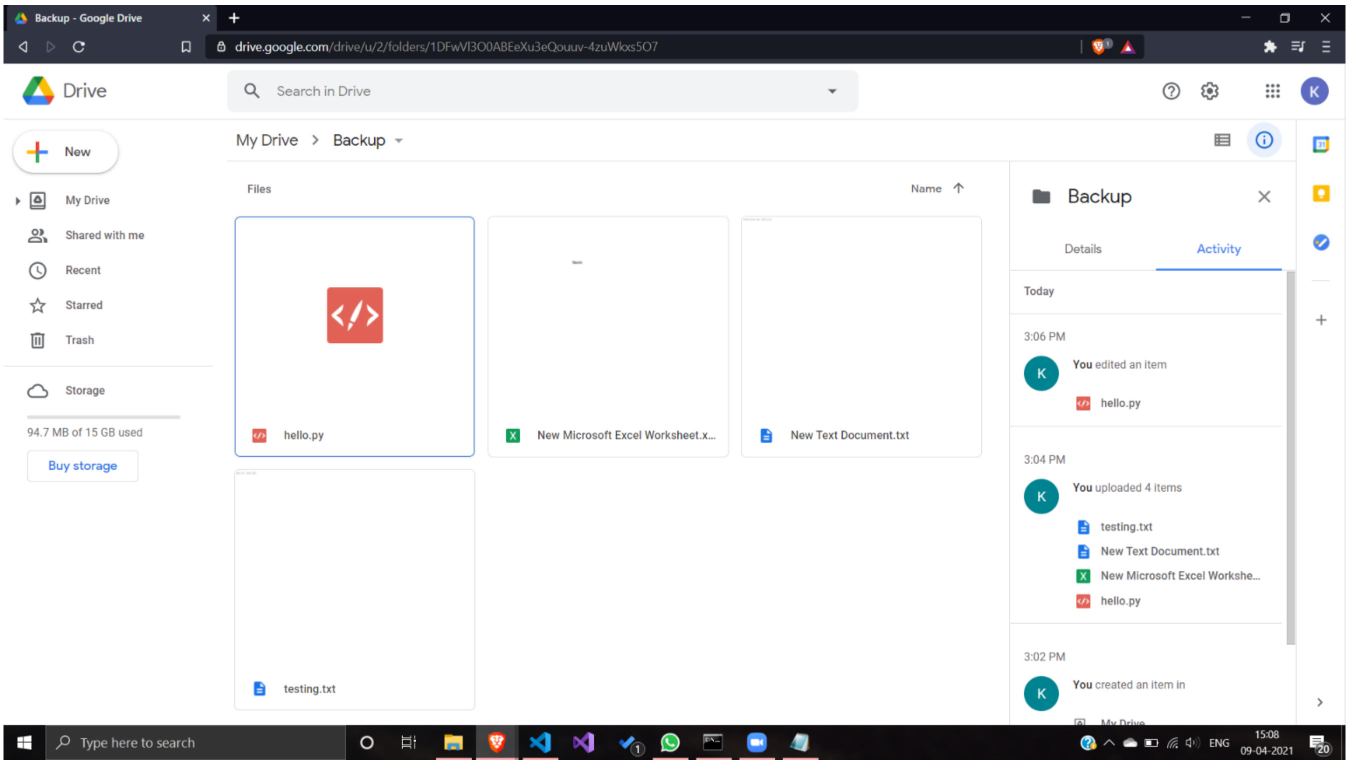
Roll No: 101803186 **Name: Kulpreet Singh** **Group: 3COE9**

- log.txt



```
09/04/2021 15:03:57 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/hello.py
09/04/2021 15:03:59 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/New Microsoft Excel Worksheet.xlsx
09/04/2021 15:04:01 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/New Text Document.txt
09/04/2021 15:04:03 backup_successful C:/Users/kulpr/Downloads/Sem 6/Cloud Computing/Backup/testing.txt
09/04/2021 15:07:06 backup_failed timeout
09/04/2021 15:07:06 backup_failed ServerNotFoundError
09/04/2021 15:07:06 backup_failed ServerNotFoundError
09/04/2021 15:07:06 backup_failed ServerNotFoundError
```

- Drive Folder



drive.google.com/drive/u/2/folders/1DFwVl3O0ABEeXu3eQouuv-4zuWlxs5O7

My Drive > Backup

Files

hello.py
New Microsoft Excel Worksheet.x...
New Text Document.txt
testing.txt

Backup

Details Activity

Today

3:06 PM
You edited an item
hello.py

3:04 PM
You uploaded 4 items
testing.txt
New Text Document.txt
New Microsoft Excel Workshe...
hello.py

3:02 PM
You created an item in