Practical No:05

Implementation of Cloud Computing Services

1. Write a java program to access the files from your Google drive account and read and write the file contents from your program. Code:

```
package com.avk.Gdrive;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.security.GeneralSecurityException;
import java.util.Collections;
import java.util.List;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.google.api.client.auth.oauth2.Credential;
import com.google.api.client.extensions.java6.auth.oauth2.AuthorizationCodeInstalledApp;
import com.google.api.client.extensions.jetty.auth.oauth2.LocalServerReceiver;
import com.google.api.client.googleapis.auth.oauth2.GoogleAuthorizationCodeFlow;
import com.google.api.client.googleapis.auth.oauth2.GoogleClientSecrets;
import com.google.api.client.googleapis.javanet.GoogleNetHttpTransport;
import com.google.api.client.http.AbstractInputStreamContent;
import com.google.api.client.http.InputStreamContent;
import com.google.api.client.http.javanet.NetHttpTransport;
import com.google.api.client.json.JsonFactory;
import com.google.api.client.json.jackson2.JacksonFactory;
import com.google.api.client.util.store.FileDataStoreFactory;
import com.google.api.services.drive.Drive;
import com.google.api.services.drive.DriveScopes;
import com.google.api.services.drive.model.File;
import com.google.api.services.drive.model.FileList;

@SpringBootApplication
public class GdriveApplication {
	private static final JsonFactory JSON_FACTORY =
JacksonFactory.getDefaultInstance();
	// Directory to store user credentials for this application.
//private static final java.io.File CREDENTIALS_FOLDER = new
java.io.File(System.getProperty("user.home"), "credentials");	private
static final java.io.File CREDENTIALS_FOLDER = new
java.io.File("C:\\Users\\Atharv Vinayak Kher\\Downloads");

	// Global instance of the scopes required by this program.	private
static final List<String> SCOPES =
```

```
        Collections.singletonList(DriveScopes.DRIVE);
        //https://developers.google.com/resources/api-
        libraries/documentation/drive/v2/java/latest/com/google/api/services/drive/DriveScop
        es.html

                    private static Credential getCredentials(final NetHttpTransport
        HTTP_TRANSPORT) throws IOException {                    java.io.File
clientSecretFilePath = new java.io.File("C:\\Users\\Atharv  Vinayak
Kher\\Downloads\\driveApiForJava.json");
                                        if
(!clientSecretFilePath.exists()) {
                            throw new FileNotFoundException("Please copy credentials.");

                        // Load client secrets.
                        InputStream in = new FileInputStream(clientSecretFilePath);
                        GoogleClientSecrets clientSecrets =
        GoogleClientSecrets.load(JSON_FACTORY, new InputStreamReader(in));

                        // Build flow and trigger user authorization request.
                        GoogleAuthorizationCodeFlow flow = new
        GoogleAuthorizationCodeFlow.Builder(HTTP_TRANSPORT, JSON_FACTORY,
clientSecrets,                                          SCOPES).setDataStoreFactory(new
FileDataStoreFactory(CREDENTIALS_FOLDER))
                                    .setAccessType("offline").build();
                        //System.out.println("Flow info - " + flow.toString());
                                        return new
AuthorizationCodeInstalledApp(flow, new
LocalServerReceiver()).authorize("user");
                    }
                public static void main(String[] args) throws IOException,
        GeneralSecurityException {
                                // 1: Build a new authorized API client service.
        final NetHttpTransport HTTP_TRANSPORT =
        GoogleNetHttpTransport.newTrustedTransport();

            // 2: Read client_secret.json file & create Credential object.
        Credential credential = getCredentials(HTTP_TRANSPORT);        // 3:
        Create Google Drive Service.
            Drive service = new Drive.Builder(HTTP_TRANSPORT, JSON_FACTORY,
        credential).setApplicationName("GDrive Access").build();
            System.out.println("----" + service.getApplicationName() + "----");        //
        Print the names and IDs for up to 10 files.
            FileList result = service.files().list().setPageSize(20).setFields("nextPageToken,
        files(id, name)").execute();
            String lastFile = "";
            List<File> files = result.getFiles();        if
        (files == null || files.isEmpty()) {
        System.out.println("No files found.");
            } else {
```

```
            System.out.println("Files:");
            for (File file : files) {
                System.out.printf("%s (%s)\n", file.getName(), file.getId());
lastFile = file.getId();
            }
        }

        //Downloading a file from GDrive          java.io.OutputStream output = new
java.io.FileOutputStream("output.pdf");
service.files().export("1PYbZPX7VkxXelaMIww-
0v3yAx8NOghjOFGeKqJd3uzA", "application/pdf").executeAndDownloadTo(output);
        System.out.println("File downloaded..")
        java.io.File uploadFileContent = new java.io.File("D:\\mca sem3\\demo.txt");
        String contentType = "text/plain";
//Create Folder on Google Drive         File
fileMetadata = new File();
fileMetadata.setName("MyFolderUsingJava");
        fileMetadata.setMimeType("application/vnd.google-apps.folder");
        //fileMetadata.setParents(folderIdParent);
        File file = service.files().create(fileMetadata).setFields("id, name").execute();
if(file != null)
         System.out.println("Folder Created..");
        //Creating a file on GDrive
        AbstractInputStreamContent uploadStreamContent = new
InputStreamContent(contentType,new FileInputStream(uploadFileContent));
fileMetadata = new File();
        fileMetadata.setName("MyGDriveJavaFile.txt");          file =
service.files().create(fileMetadata, uploadStreamContent).setFields("id,
webContentLink, webViewLink, parents").execute();
                if(file !=
null) {
         System.out.println("File Created..");
         System.out.println("WebContentLink: " + file.getWebContentLink() );
         System.out.println("WebViewLink: " + file.getWebViewLink() );
        }}}
```

Output:

```
Console
<terminated> GdriveApplication [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe  (01-Dec-2022, 12:22:59 am – 12:23:06 am) [pid: 2300]
----GDrive Access----
Files:
DSCC_Practical_Manual_06.docx (18r3jfrBlmPdj2-EKxGguvZGKscnqBuaE)
MC LAB (2nd half of 2022) MC LAB (2nd half of 2022) (15tPifmXolyLnyLzkos0P4ZzQbIMpfFJTVk9DCrxq4PF5LVnVuFZTjYmVMBzD7OWFSdL6p3jB)
Practical No. 10.docx (1v3Pl93iPOkOBm0lmsd9ZAWafW4XU2W4c)
Practical No. 11.docx (1CYmgTCgSgDiaJiYMGpppvluKVJo5hTcf)
Practical No. 9.docx (1ujcd6M688Rloj7b_VvGAgkMcyTc01t1Z)
Practical No. 8.docx (1ypxdJXluttP26e7eBZAhZi-b4GpFcSZp)
Practical No. 5.docx (1lZCxPB073G5Xpt1HERNHJfloPVUk5mun)
Practical No. 6.docx (1oFFlYCaaqY6tL_twsh-VIb0ljc999SZT)
Practical No. 2.docx (1rM16Bfag0Moh4MoV_YT5PGChQtGPvu_q)
Practical_04_Ethereum.pdf (1Um8S0lejAx4kOAIbUDLk3Y2WQN_9enF9)
05_Ethereum.pdf (1Z1urYlYQn61Qw1gqTAxSQgfj154hWbsy)
FAMT 2022 - React Assignment  list (16FxJsnWYhRvXF2LkOJWxbNQThbwDX0YqzHyvaE-VM5M)
27_Atharva Kher - Practical No_02 (1jb5Ajz3g88lQsMYOoyeedrxrGFHEFrlVQBlU-5vcyzZovpxqoHB72HRbZR0wDC5CpL_OKGY)
27_Atharva Kher - Practical No_02 (1HrrTNSS6pb-X6QdhhM36PdEQMdRSPX-0gUoJ94CE_w4)
Actions Class in Selenium Exercises.pdf (1y4q0glkhd57ZmxsvwQqv_dpxf3hp9K44)
ERD.png (1Edu4M6Eu4mjDZFrb0KgPndYiaCmA6Uy-)
origin (1doeWo2ZfhS3QgGGmbJ1IAn9ONT_cPUw3)
heads (18MUTtCcGvhwg0jCRd3ZZyZXtKPkb470D)
remotes (1F8uFL8diP-nirPltCb4itnDNNdELpfrU)
refs (1vimGwlO-llMeykaQl2WJmzMXsKL8Kqjg)
File downloaded..
Folder Created..
File Created..
```