PRACTICAL NO. 6
Implementation of Cloud Computing Services

LOB6: Understand the cloud computing services and identity management in the cloud.
LO6: Implement programs for cloud computing services and identity management.

As per Microsoft "cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping lower your operating costs, run your infrastructure more efficiently and scale as your business needs change."

Cloud Computing Service Models - Cloud computing service models indicate the type of service that is being offered. The three service models that are essential components of cloud computing standards are:

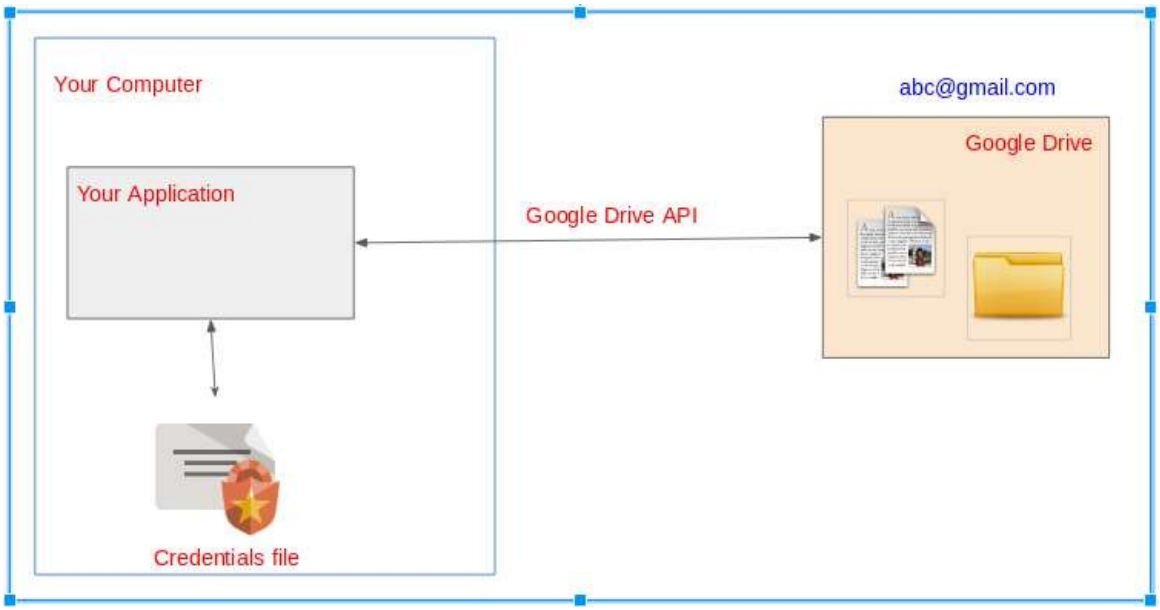
- **Software as a Service (SaaS)** Applications delivered as a service to end users over the Internet. The major players in this field are Oracle (with its CRM on Demand solution), Salesforce.com, and Google (with its Google Apps).
- **Platform as a Service (PaaS)** Application development and deployment platform (comprising application servers, databases, etc.) delivered as a service. Amazon Elastic Compute Cloud (EC2) and Savvis are the prominent providers of this model of cloud service.
- **Infrastructure as a Service (IaaS)** Server, storage, and network hardware and associated software delivered as a service. Amazon EC2 is the prominent provider of this model of cloud service.

Google Drive access using Java –

Google drive allows you to store, share, and collaborate on files and folders from your mobile device, tablet, or computer. Launched on April 24, 2012, Google Drive allows users to store files in the cloud (on Google's servers), synchronize files across devices, and share files.In addition to a web interface, Google Drive offers apps with offline capabilities for Windows and macOS computers, and Android and iOS smartphones and tablets.

Google Drive encompasses Google Docs, Google Sheets, and Google Slides, which are a part of the Google Docs Editors office suite that permits collaborative editing of documents, spreadsheets, presentations, drawings, forms, and more. Files created and edited through the Google Docs suite are saved in Google Drive.

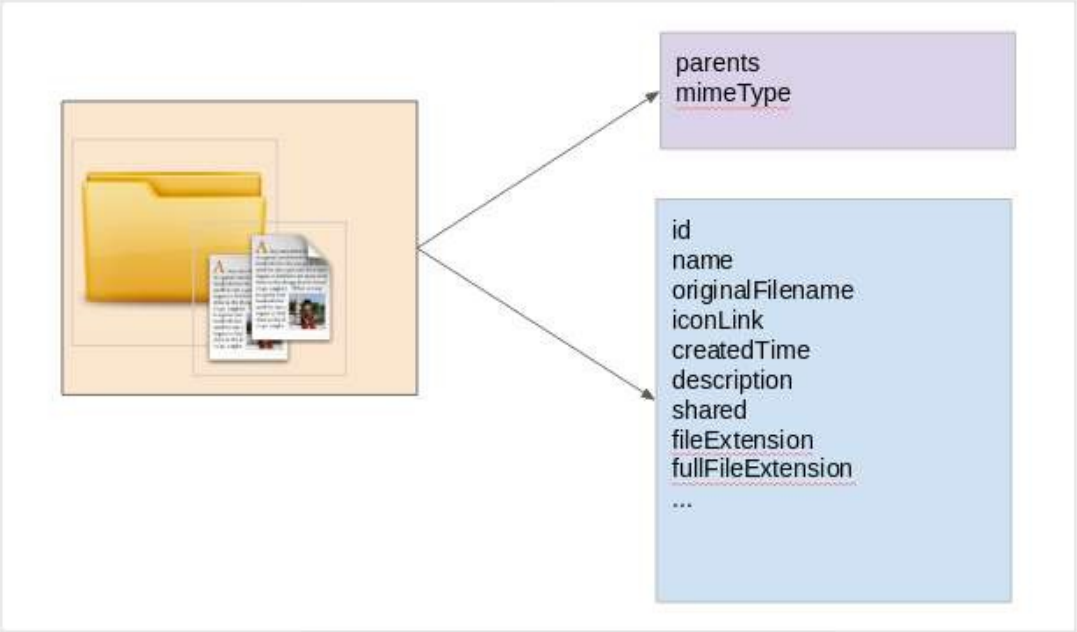
You can store your files on it. For your application to be able to manipulate with the files on **Google Drive**, it needs a **credentials**. The credentials are simply a file which will be placed on the computer where your application is being deployed like the following illustration:



Concept of Directory and File in Google Drive is a little different from the Directory & File concept in operating systems. Following are basic characteristics:

1. In Google Drive, one File/Directory can have one or more parent Directories.
2. In the same directory, the files can have the same name but differentID.
3. The com.google.api.services.drive.model.File class represents for both, File and Directory.

The com.google.api.services.drive.model.File class has a lot of fields as in the following illustration:



There are 2 important fields such as **mineType & parents**:

parents

The ID list of parent directories of present file (directory). Root directories or root file have parent directory with **ID = "root"**.

mineType

MIME Type	**Description**
application/vnd.google-apps.document	Google Docs
application/vnd.google-apps.file	Google Drive file
application/vnd.google-apps.folder	Google Drive folder
application/vnd.google-apps.site	Google Sites
application/vnd.google-apps.spreadsheet	Google Sheets
. . .	

Operators used for fields:

Field	**Value Type**	**Operators**	**Description**
Name	string	contains, =, !=	Name of the file.
fullText	string	contains	Full text of the file including name, description, content, and indexable text
mimeType	string	contains, =, !=	MIME type of the file
modifiedTime	date	<=, <, =, !=, >, >=	Date of the last modification of the file

Google provides Google APIs client library, to access protected data stored in Google services, such as documents, sheets, images, etc. we must use OAuth 2.0 authorization. Google APIs support OAuth 2.0 flows for various client applications. In all these flows, the client application requests an access token that is associated with only your client application and the owner of the protected data being accessed. The access token is also linked with a limited scope that defines the type of data your client application has access to. A crucial objective for OAuth 2.0 is to offer convenient and secure access to the protected data while reducing the potential impact if an access token is stolen.

Exercise:

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

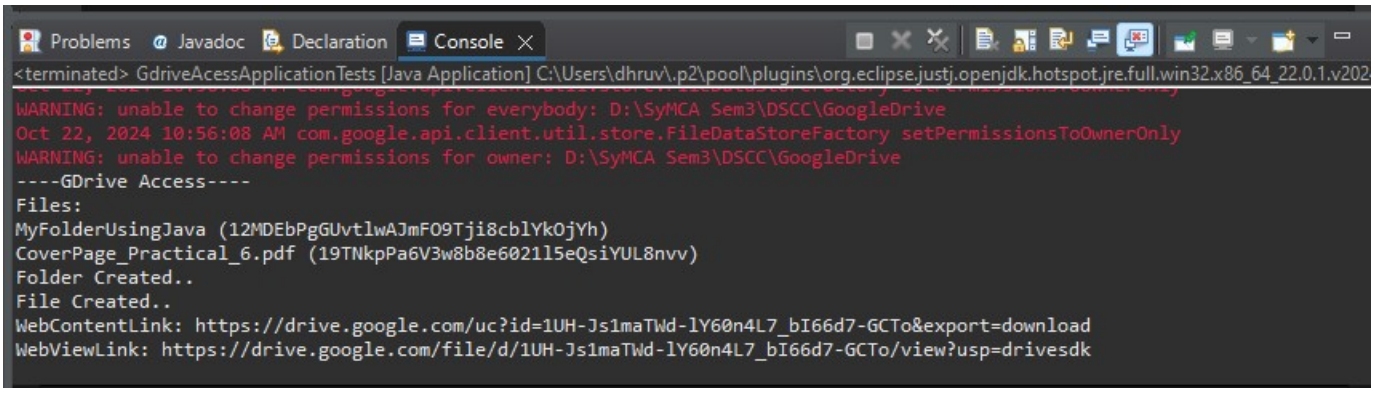
GDriveAccessApplicationTest.java:

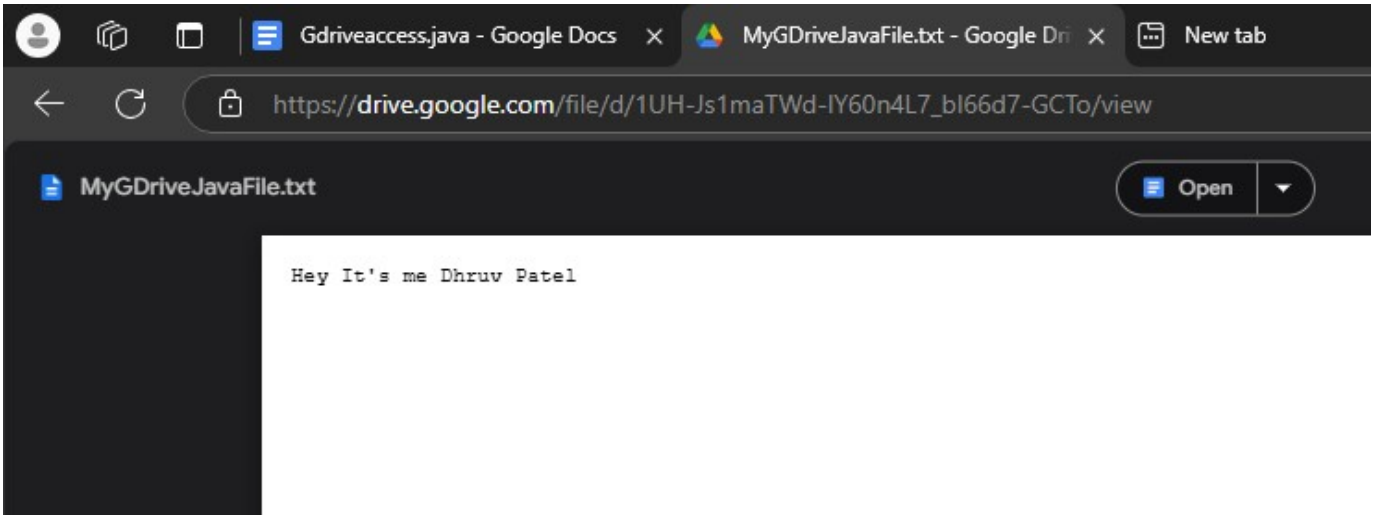
```
package in.ac.famt.GdriveAcess;

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;

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 java.util.Scanner;


//@SpringBootApplication
public class GdriveAcessApplicationTests {
        private static final JsonFactory JSON_FACTORY = JacksonFactory.getDefaultInstance();
```

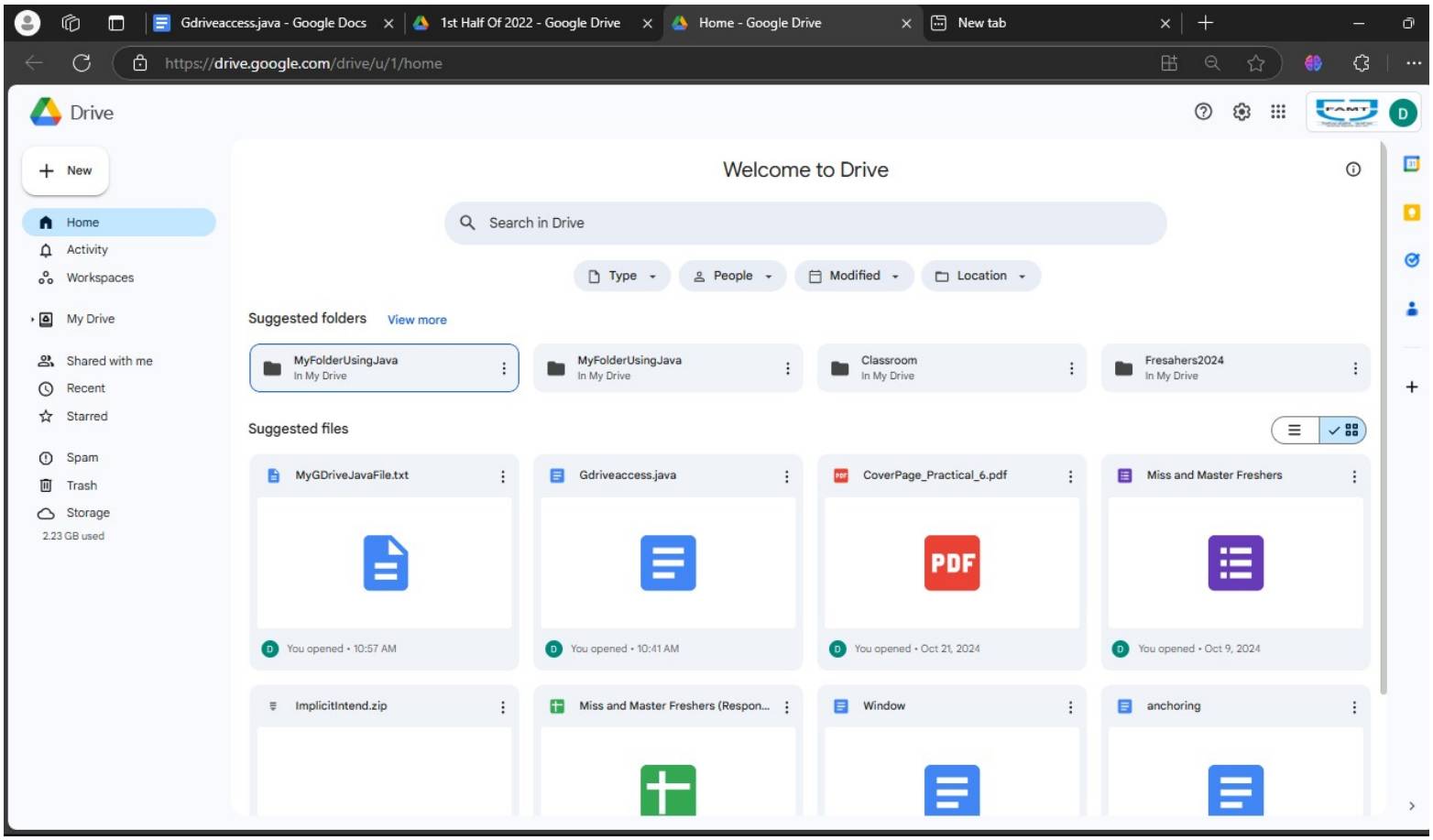
```
// Directory to store user credentials for this application.
//private static final java.io.File CREDENTIALS_FOLDER = new java.io.File(System.getProperty("user.home"),
"credentials");
private static final java.io.File CREDENTIALS_FOLDER = new java.io.File("D:/SyMCA Sem3/DSCC/GoogleDrive");

// 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/DriveScopes.html

private static Credential getCredentials(final NetHttpTransport HTTP_TRANSPORT) throws IOException {
   java.io.File clientSecretFilePath = new java.io.File("D:/SyMCA Sem3/DSCC/GoogleDrive/credits.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(2).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();
```

```
    }
}


//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
java.io.File uploadFileContent = new java.io.File("D:/SyMCA Sem3/DSCC/GoogleDrive/TestUploadFile.txt");
String contentType = "text/plain";

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:




Gdriveaccess.java - Google Docs
MyGDriveJavaFile.txt - Google Dri
New tab
https://drive.google.com/file/d/1UH-Js1maTWd-IY60n4L7_bI66d7-GCTo/view
MyGDriveJavaFile.txt
Open
Hey It's me Dhruv Patel


Gdriveaccess.java - Google Docs
1st Half Of 2022 - Google Drive
Home - Google Drive
New tab
https://drive.google.com/drive/u/1/home
Drive
New
Home
Activity
Workspaces
My Drive
Shared with me
Recent
Starred
Spam
Trash
Storage
2.23 GB used
Welcome to Drive
Search in Drive
Type
People
Modified
Location
Suggested folders
View more
MyFolderUsingJava
In My Drive
MyFolderUsingJava
In My Drive
Classroom
In My Drive
Fresahers2024
In My Drive
Suggested files
MyGDriveJavaFile.txt
You opened • 10:57 AM
Gdriveaccess.java
You opened • 10:41 AM
CoverPage_Practical_6.pdf
You opened • Oct 21, 2024
Miss and Master Freshers
You opened • Oct 9, 2024
ImplicitIntend.zip
Miss and Master Freshers (Respon...
Window
anchoring