Software Security

Assignment 2 (2022)

FACULTY OF
GRADUATE STUDIES
& RESEARCH

SLIIT

Submitted to
Sri Lanka Institute of Information Technology

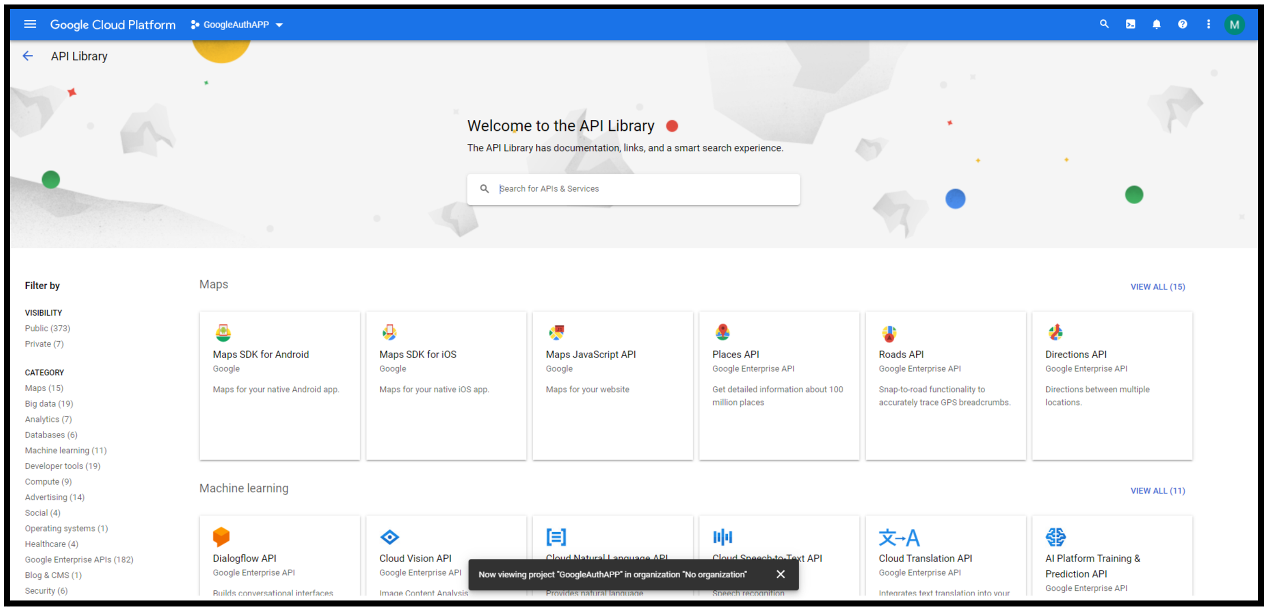
MS21928406 - W.B.M Kularatne
MS22904478 - M.A.L.C Siriwardena

In partial fulfillment of the requirements for the MSc in Information Systems

1. Introduction

The OAuth (open authorization) protocol was developed by the Internet Engineering Task Force and enables secure delegated access. It lets an application access a resource that is controlled by someone else (end user). This kind of access requires Tokens, which represent delegated right of access.

In the large-scale deployments, there are more resource servers available. For example, In Google's services have many resource servers like google maps, YouTube, Google Cloud Platform, Google drive, Google + and so many other services.



Welcome to the API Library

The API Library has documentation, links, and a smart search experience.

Search for APIs & Services

Filter by

VISIBILITY
Public (373)
Private (7)

CATEGORY
Maps (15)
Big data (19)
Analytics (7)
Databases (6)
Machine learning (11)
Developer tools (19)
Compute (9)
Advertising (14)
Social (4)
Operating systems (1)
Healthcare (4)
Google Enterprise APIs (182)
Blog & CMS (1)
Security (6)

Maps — VIEW ALL (15)

Maps SDK for Android
Google
Maps for your native Android app.

Maps SDK for iOS
Google
Maps for your native iOS app.

Maps JavaScript API
Google
Maps for your website

Places API
Google Enterprise API
Get detailed information about 100 million places

Roads API
Google Enterprise API
Snap-to-road functionality to accurately trace GPS breadcrumbs.

Directions API
Google Enterprise API
Directions between multiple locations.

Machine learning — VIEW ALL (11)

Dialogflow API
Google Enterprise API
Builds conversational interfaces

Cloud Vision API
Google Enterprise API
Image Content Analysis

Cloud Natural Language API
Provides natural language

Cloud Speech-to-Text API
Speech recognition

Cloud Translation API
Google Enterprise API
Integrates text translation into your

AI Platform Training & Prediction API
Google Enterprise API

Now viewing project "GoogleAuthAPP" in organization "No organization"

These all-resource servers work as a separate resource but they all use same authentication server for all their needs and smaller development have only one resource server and it build part of the same code or same deployment as an authentication server.

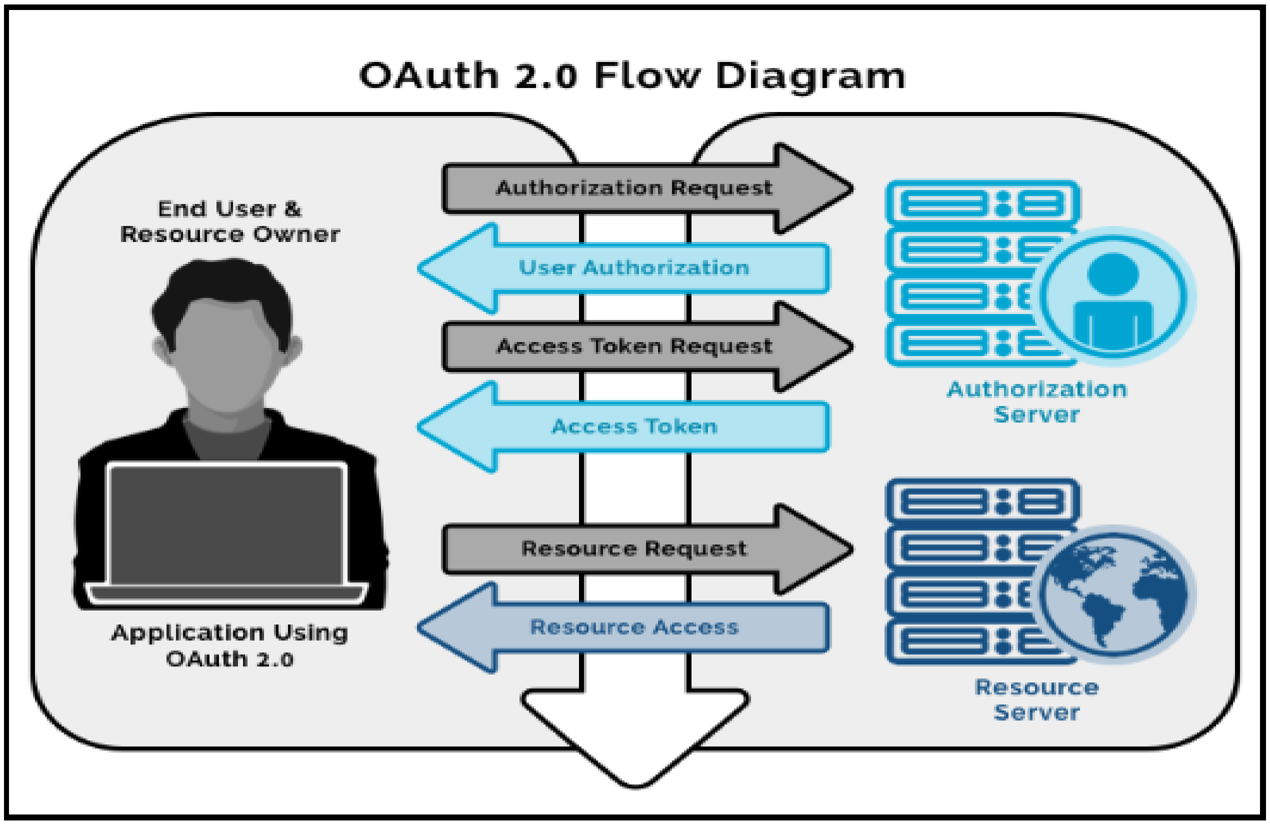
2. Token Based Security

The OAuth Access Token transaction needs the participation of three parties: the end-user, the application (API), as well as the resource. The transaction begins when the user expresses an intent to utilize the API.

1. The application or API (application programming interface) requests authorisation from the resource through giving verification of the user's confirmed identity.

2. After the permission has been validated, the resource gives an Access-Token to the API without revealing users or passwords.

3. To access the resource, the tokens come with API access authorisation. These are known as scopes, and that each token will have an approved scope for each API. The application can only access the resource to the degree permitted by the scope.



Resource Owner, Resource Server, Client, and Authorized Server are the roles of the OAuth framework. In this method, client need to show valid credentials to authorization server for generate an access token. In OAuth 2.0, it is selecting a suitable method for their application. That method becomes noticeably clear if you know the type of client that you have. Ex. web, mobile, etc.

3. THE APP

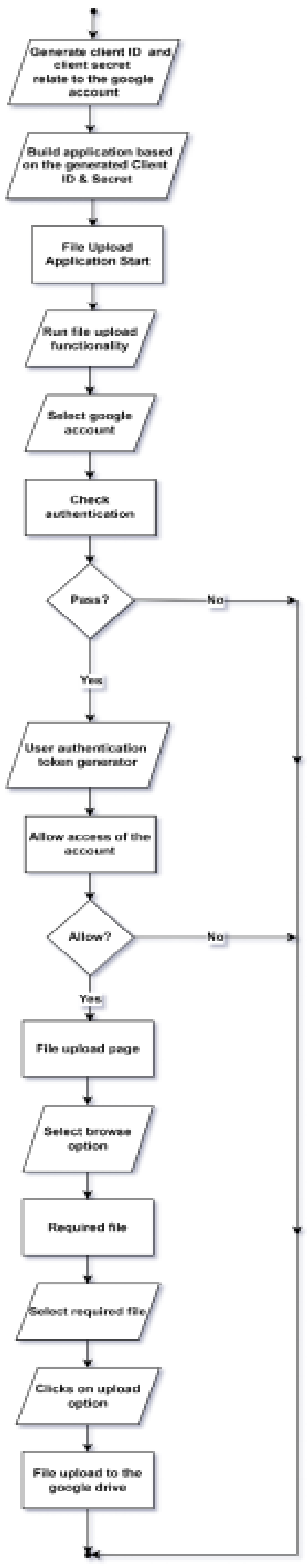
This development was mostly concerned with the setup and functioning of the OAuth server. The goal of the creating app was to upload a file to Google Drive from of the client computer utilizing OAuth 2.0 access token authorisation.

Both the Google cloud console and the Google Drive API were used to configure OAuth credentials, and html was utilized to create web app pages. Java Script is used to create the application's functionality.

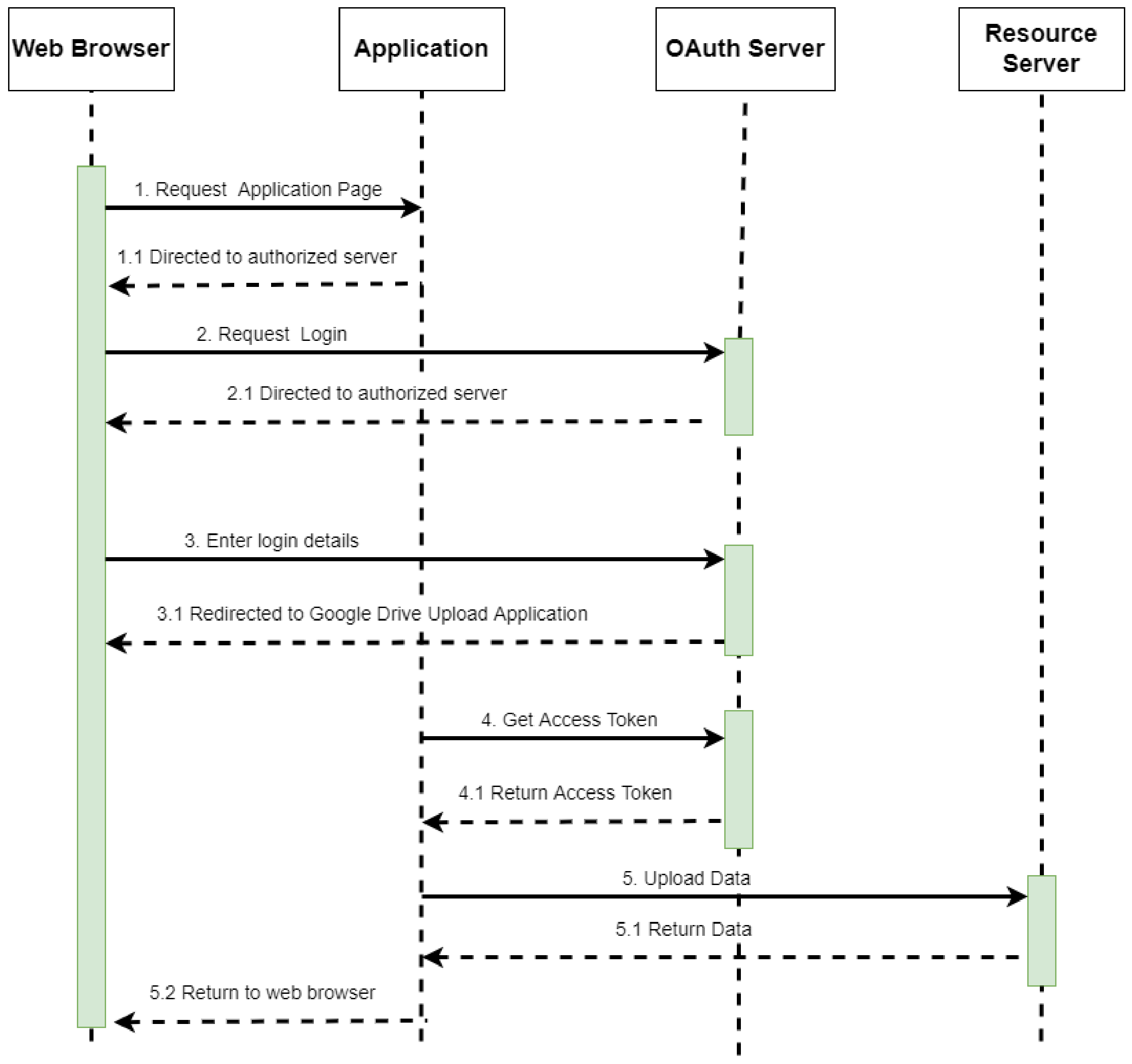
When developing a project, the initial step is to activate Google Drive API and then configure OAuth server as API credentials. It would then produce the Client ID, Client Security, Redirect URI, and Redirect URL. When creating the file upload app, these parameters are used. These attributes identify the Google account that we utilized for this application.

To run this application locally we have used the “live server”. After installing it, follow the steps as given in the README file.

4. Flow Diagram


Generate client ID and client secret relate to the google account
Build application based on the generated Client ID & Secret
File Upload Application Start
Run file upload functionality
Select google account
Check authentication
Pass ?
No
Yes
User authentication token generator
Allow access of the account
Allow?
No
Yes
File upload page
Select browse option
Required file
Select required file
Clicks on upload option
File upload to the google drive

5. Sequential Diagram

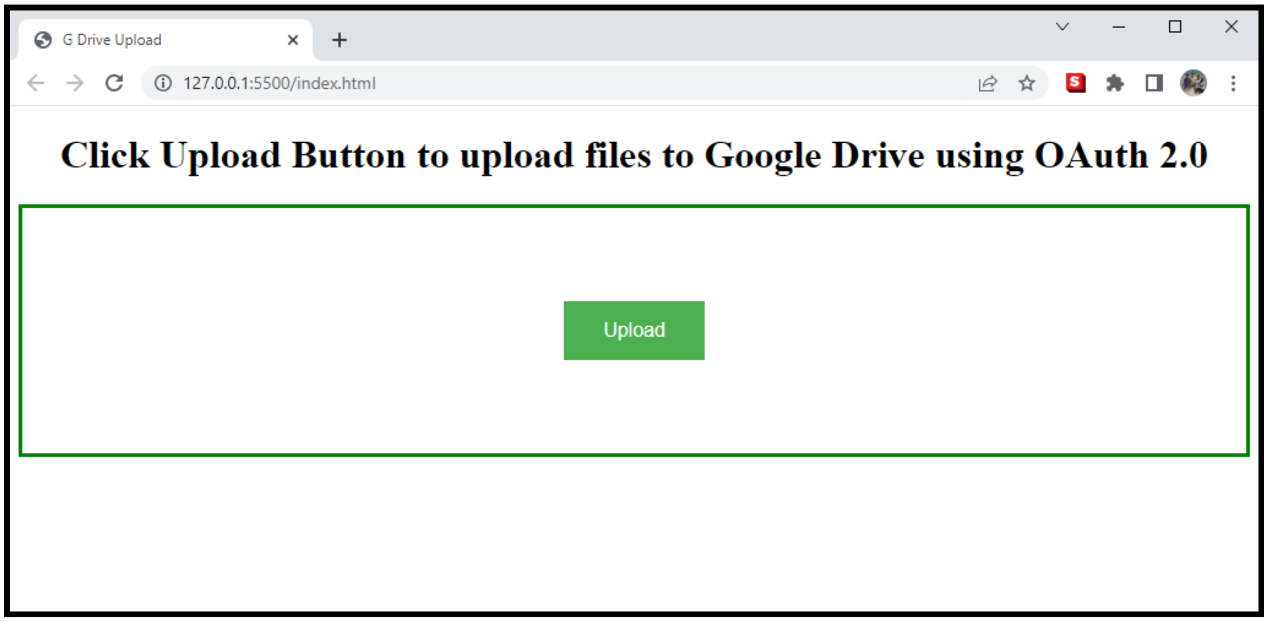

Web Browser
Application
OAuth Server
Resource Server
1. Request Application Page
1.1 Directed to authorized server
2. Request Login
2.1 Directed to authorized server
3. Enter login details
3.1 Redirected to Google Drive Upload Application
4. Get Access Token
4.1 Return Access Token
5. Upload Data
5.1 Return Data
5.2 Return to web browser

6. The Process

The developed application will help to upload any type of file to the authenticated google drive of the user.

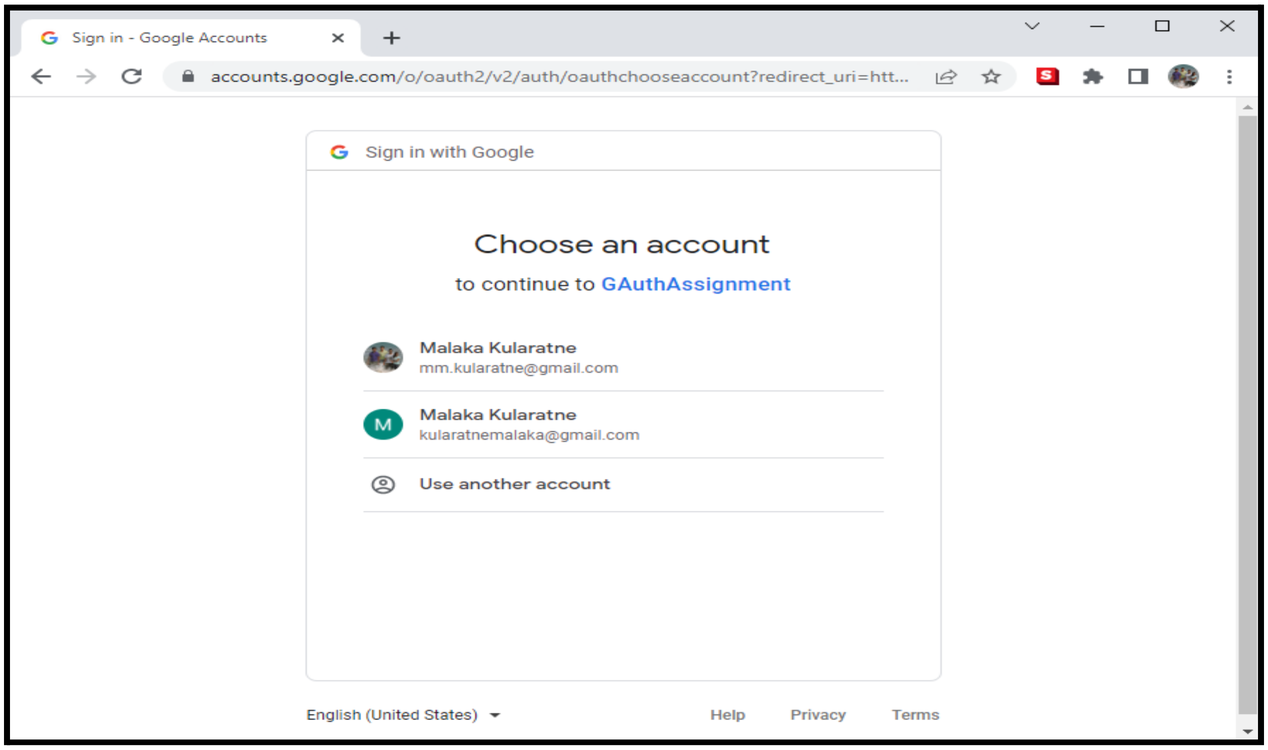
- The user ought to select the file upload feature to start the web application process. [Screen 01]
- Then the user will be redirected to the account selection window. [Screen 02]
- In the window the user can select the account for the google drive file upload.
- According to the selected Gmail account, the authentication process starts.
- If the account has identified as an authenticated account, the authentication token will be generated.
- Authentication token will be used during the active session of the user for the validation of the user action.
- For each login new authentication token will be generated.
- If the user is not authenticated, user will be existed from the process.
- Once the user authenticated, User will be directed to the file upload window
- In the file upload window, the user has a selection as ‘Browse’ and then can browse the required file in the device.
- Once the file is selected, click on the ‘Upload’ button.
- Then the file will be uploaded to the google drive.
- And user can check the G-drive and confirm the availability of uploaded file and finally exit from the application.

7. Screens



Click Upload Button to upload files to Google Drive using OAuth 2.0

Figure 1 : Starting Page


Sign in - Google Accounts
accounts.google.com/o/oauth2/v2/auth/oauthchooseaccount?redirect_uri=htt...
Sign in with Google
Use another account
English (United States)
Help
Privacy
Terms

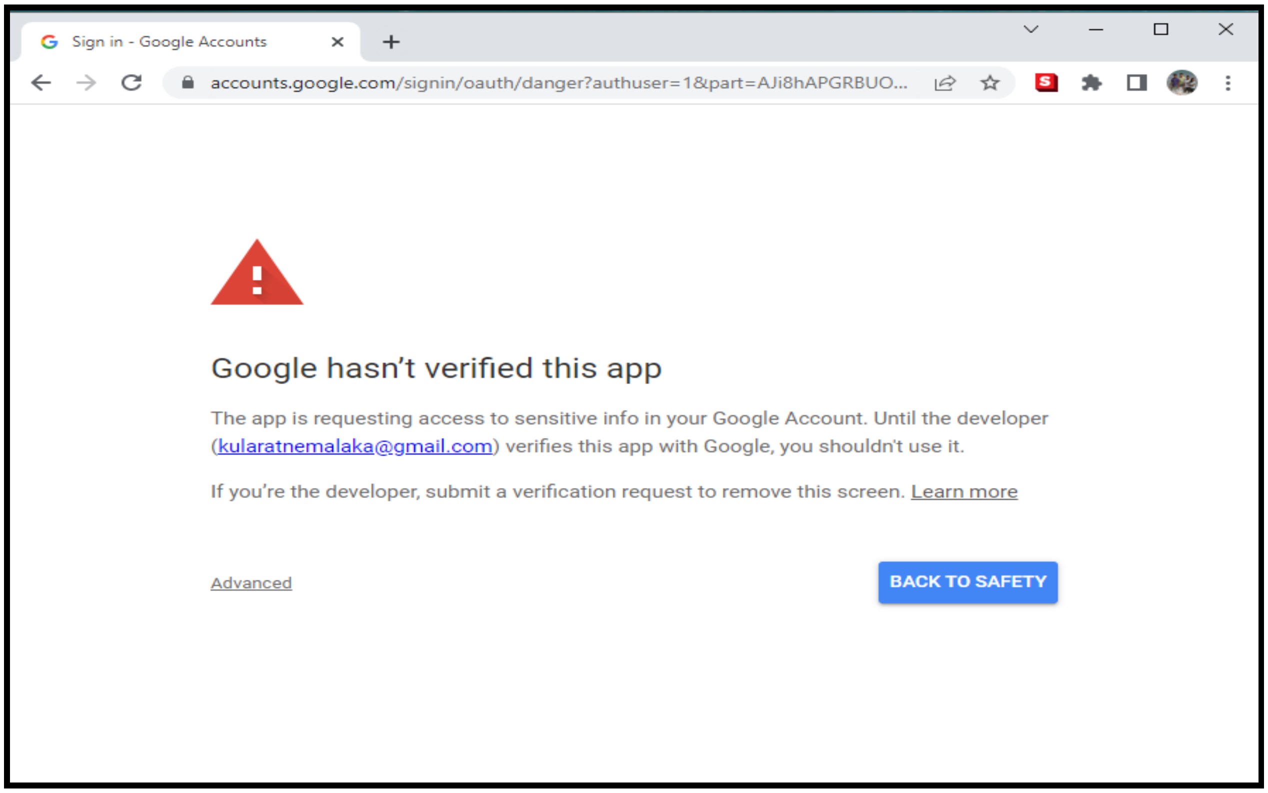
Choose an account

to continue to GAuthAssignment

Malaka Kularatne
mm.kularatne@gmail.com

Malaka Kularatne
kularatnemalaka@gmail.com

Figure 2 : Account selection for authentication

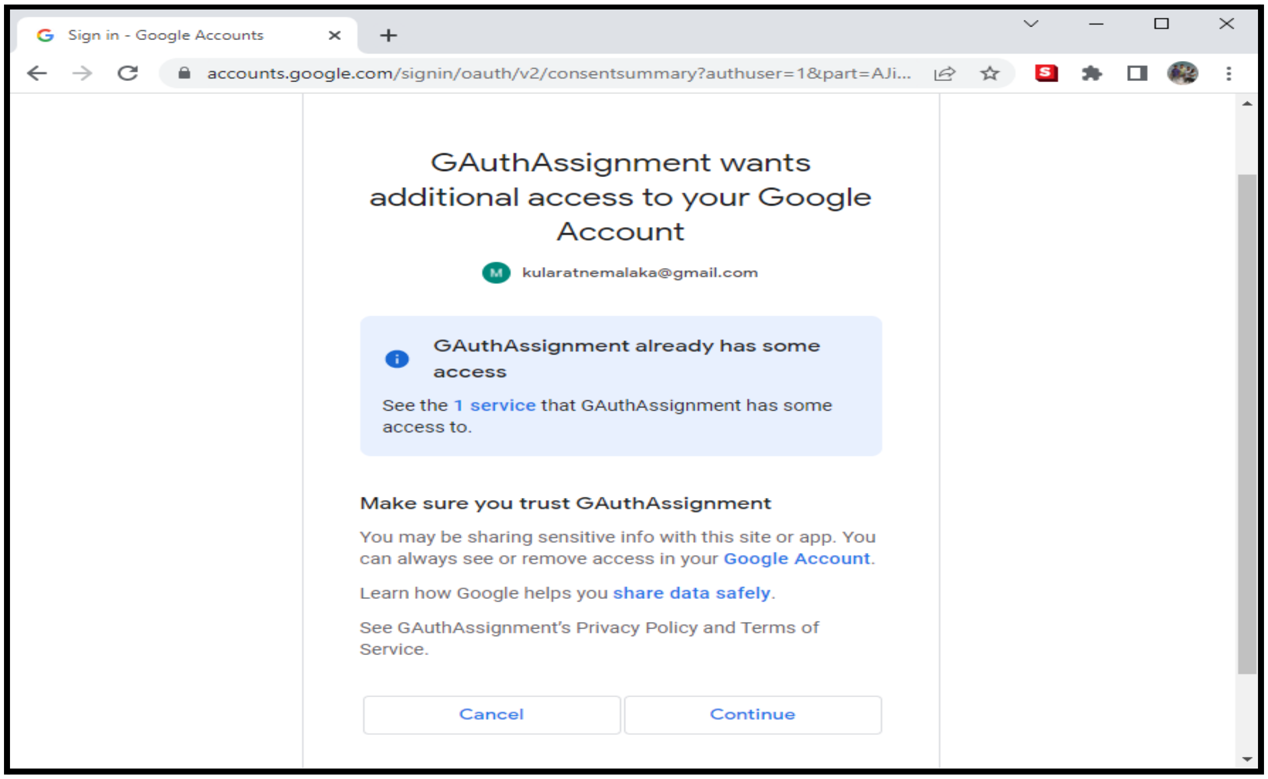

Sign in - Google Accounts
accounts.google.com/signin/oauth/danger?authuser=1&part=AJi8hAPGRBUO...

Google hasn't verified this app

The app is requesting access to sensitive info in your Google Account. Until the developer (kularatnemalaka@gmail.com) verifies this app with Google, you shouldn't use it.

If you're the developer, submit a verification request to remove this screen. Learn more

Figure 3 : Proceed by clicking on advanced


Sign in - Google Accounts
accounts.google.com/signin/oauth/v2/consentsummary?authuser=1&part=AJi...

GAuthAssignment wants additional access to your Google Account

kularatnemalaka@gmail.com

GAuthAssignment already has some access

See the 1 service that GAuthAssignment has some access to.

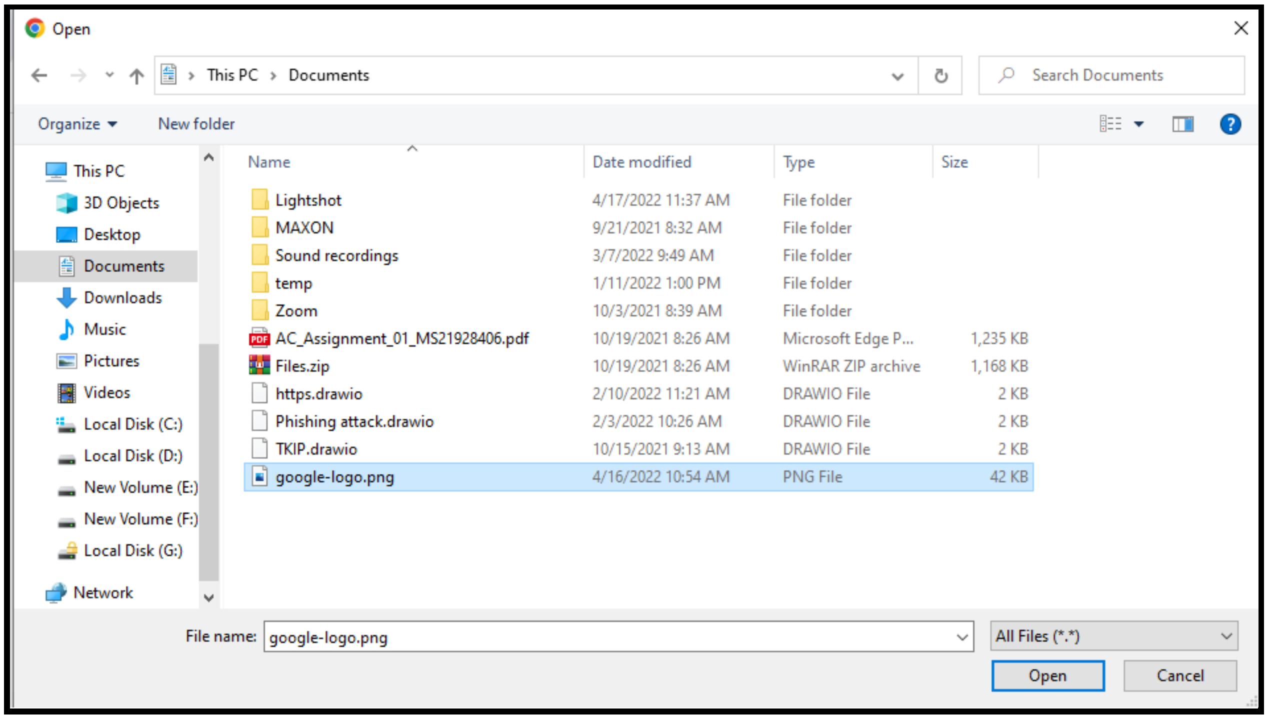
Make sure you trust GAuthAssignment

You may be sharing sensitive info with this site or app. You can always see or remove access in your Google Account.

Learn how Google helps you share data safely.

See GAuthAssignment's Privacy Policy and Terms of Service.

Figure 4 : Giving necessary permissions

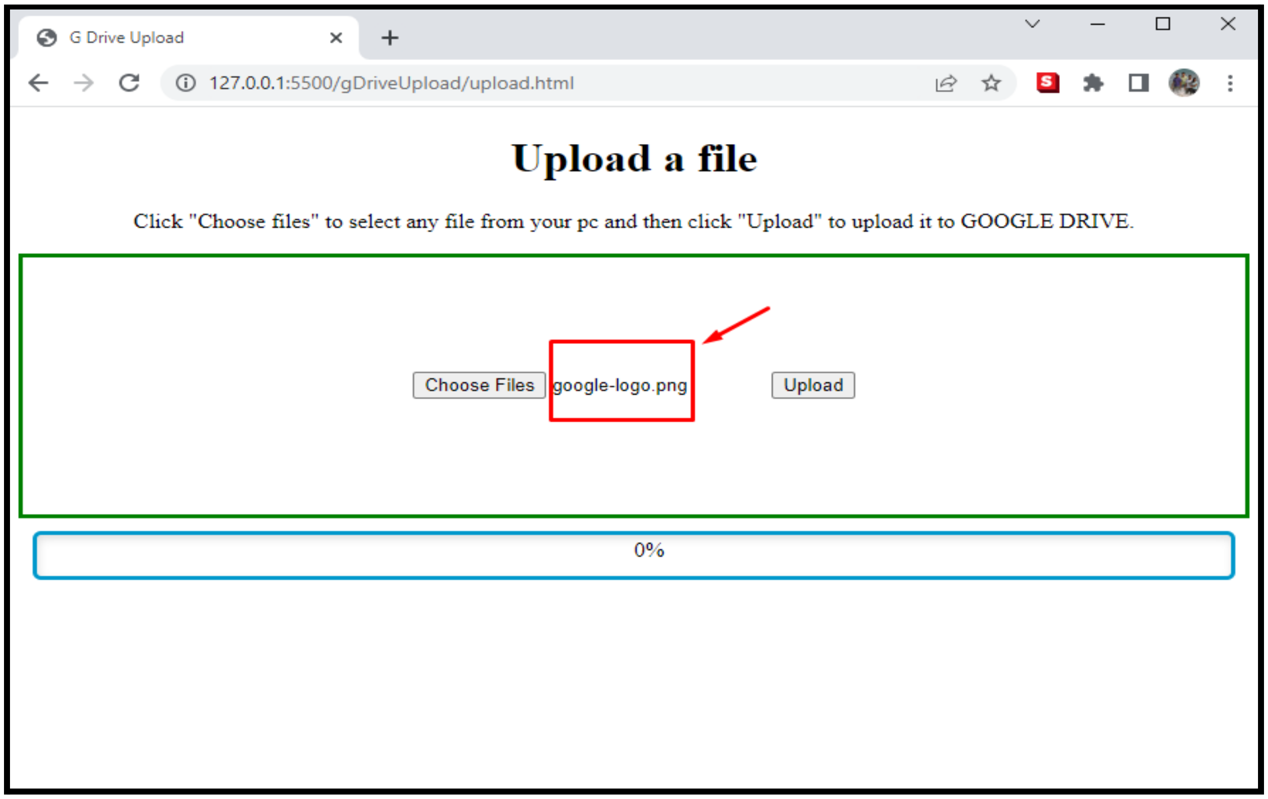


Name	Date modified	Type	Size
Lightshot	4/17/2022 11:37 AM	File folder	
MAXON	9/21/2021 8:32 AM	File folder	
Sound recordings	3/7/2022 9:49 AM	File folder	
temp	1/11/2022 1:00 PM	File folder	
Zoom	10/3/2021 8:39 AM	File folder	
AC_Assignment_01_MS21928406.pdf	10/19/2021 8:26 AM	Microsoft Edge P...	1,235 KB
Files.zip	10/19/2021 8:26 AM	WinRAR ZIP archive	1,168 KB
https.drawio	2/10/2022 11:21 AM	DRAWIO File	2 KB
Phishing attack.drawio	2/3/2022 10:26 AM	DRAWIO File	2 KB
TKIP.drawio	10/15/2021 9:13 AM	DRAWIO File	2 KB
google-logo.png	4/16/2022 10:54 AM	PNG File	42 KB

File name: google-logo.png

All Files (*.*)

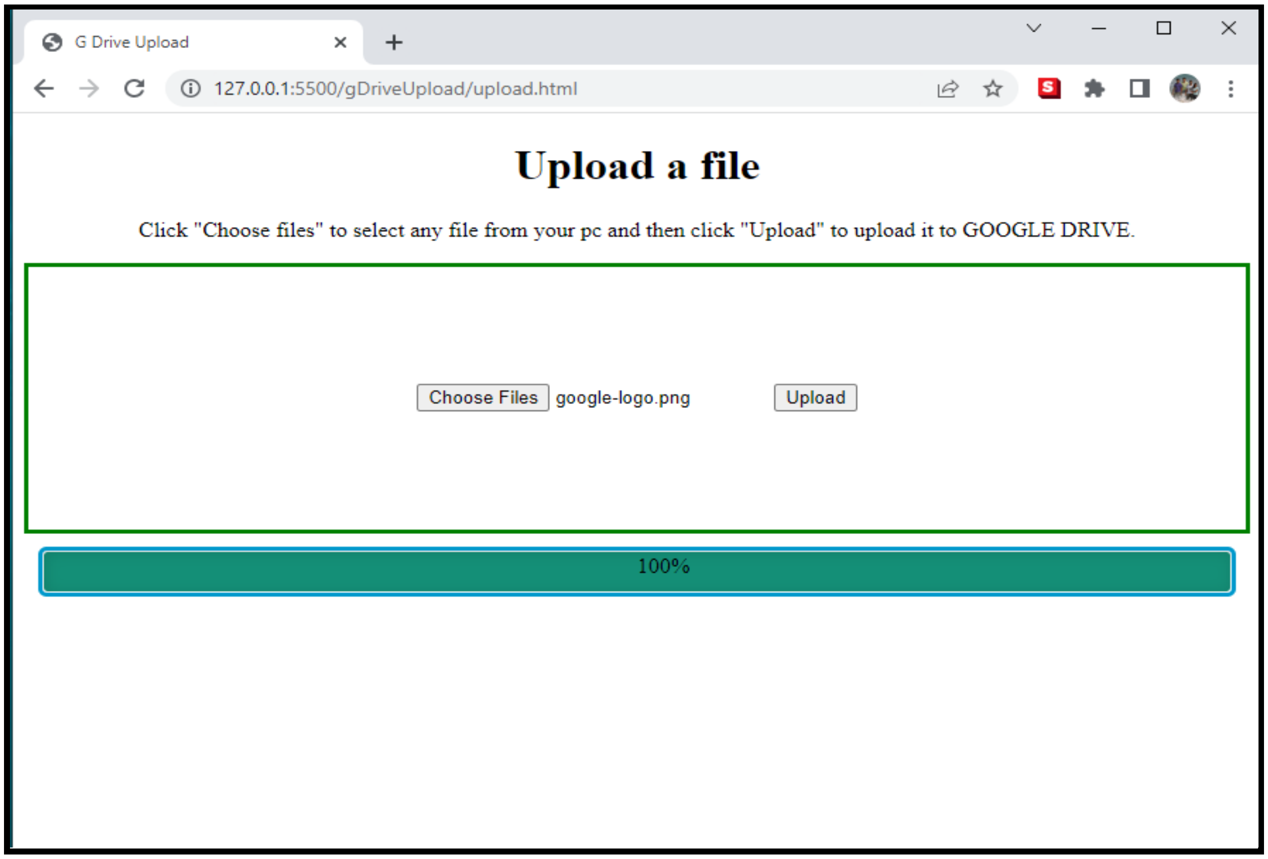
Figure 5 : Browse file for upload



Upload a file

Click "Choose files" to select any file from your pc and then click "Upload" to upload it to GOOGLE DRIVE.

Figure 6 : Selected file



G Drive Upload

127.0.0.1:5500/gDriveUpload/upload.html

Upload a file

Click "Choose files" to select any file from your pc and then click "Upload" to upload it to GOOGLE DRIVE.

Figure 7 : Upload Completed

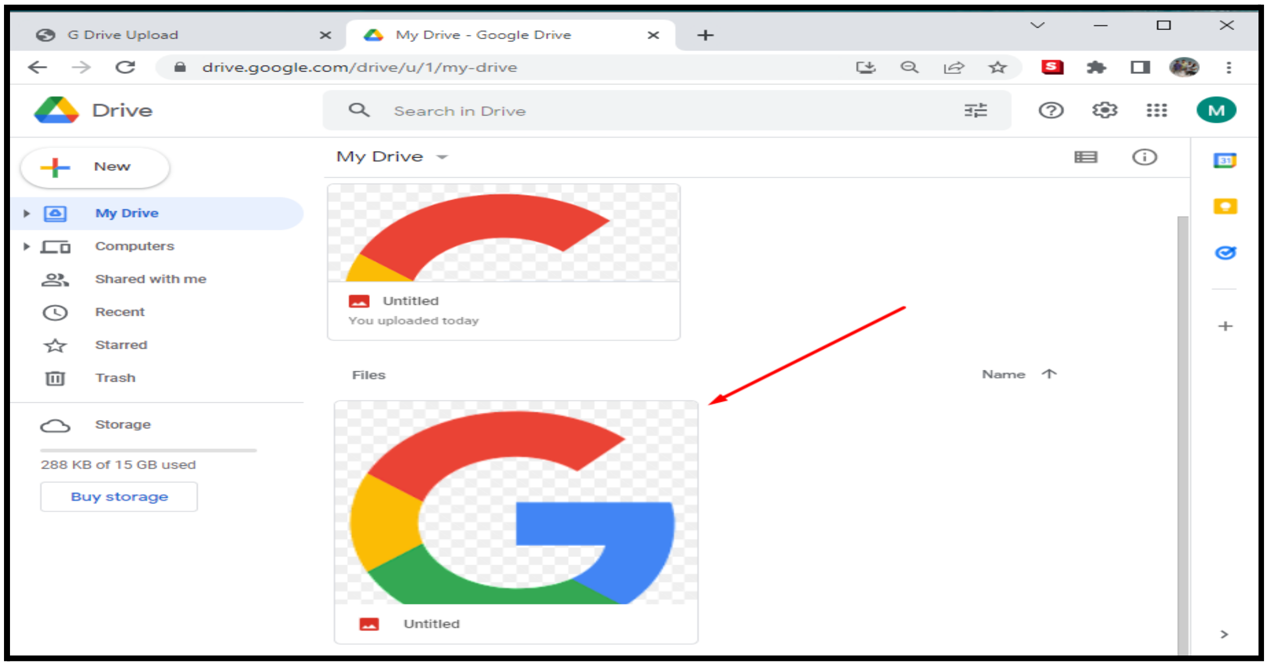
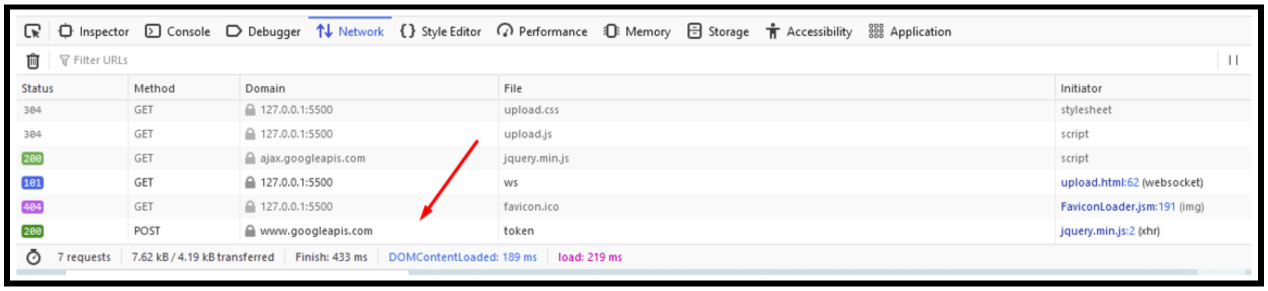


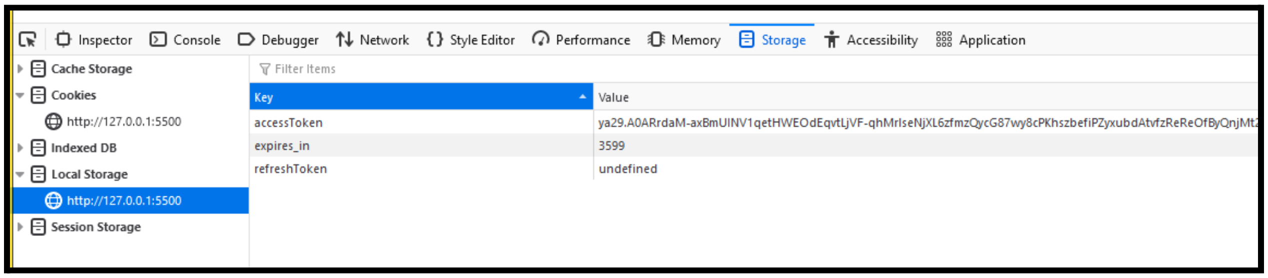
Figure 8 : Uploaded file in Google Drive



Status	Method	Domain	File	Initiator
304	GET	127.0.0.1:5500	upload.css	stylesheet
304	GET	127.0.0.1:5500	upload.js	script
200	GET	ajax.googleapis.com	jquery.min.js	script
101	GET	127.0.0.1:5500	ws	upload.html:62 (websocket)
404	GET	127.0.0.1:5500	favicon.ico	FaviconLoader.jsm:191 (img)
200	POST	www.googleapis.com	token	jquery.min.js:2 (xhr)

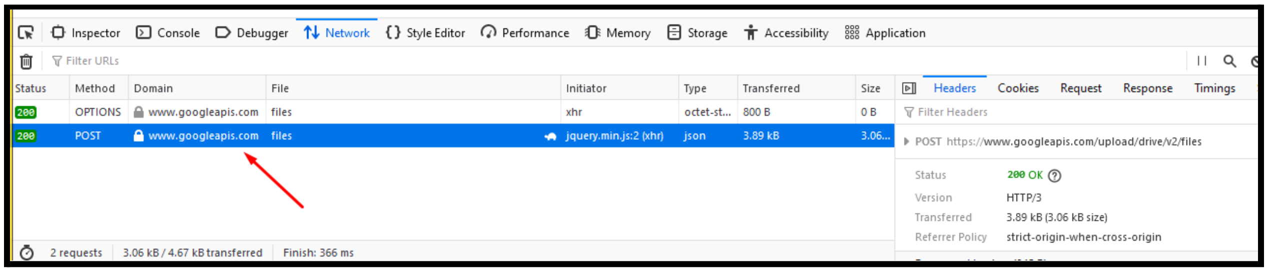
7 requests | 7.62 kB / 4.19 kB transferred | Finish: 433 ms | DOMContentLoaded: 189 ms | load: 219 ms

Figure 9 : Token that got generated



Key	Value
accessToken	ya29.A0ARrdaM-axBmUINV1qetHWEOdEqvtLjVF-qhMrlseNjXL6zfmzQycG87wy8cPKhszbefiPZyxubdAtvfzReReOfByQnjMt
expires_in	3599
refreshToken	undefined

Figure 10 : Token that got generated



Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	OPTIONS	www.googleapis.com	files	xhr	octet-st...	800 B	0 B
200	POST	www.googleapis.com	files	jquery.min.js:2 (xhr)	json	3.89 kB	3.06...

2 requests | 3.06 kB / 4.67 kB transferred | Finish: 366 ms

POST https://www.googleapis.com/upload/drive/v2/files

Status	200 OK
Version	HTTP/3
Transferred	3.89 kB (3.06 kB size)
Referrer Policy	strict-origin-when-cross-origin

Figure 11 : Uploaded file

8. References

[1] K. Dodanduwa and I. Kaluthanthri, “Role of trust in OAuth 2.0 and OpenID Connect,” arXiv [cs.CR], 2018.

[2] “OAuth 2.0 for client-side web applications,” Google Developers. [Online]. Available: https://developers.google.com/identity/protocols/oauth2/javascript-implicit-flow.

[3] T. Kawasaki, “The simplest guide to OAuth 2.0 - Takahiko Kawasaki,” Medium, 01-Aug-2017. [Online]. Available: https://darutk.medium.com/the-simplest-guide-to-oauth-2-0-8c71bd9a15bb.

[4] “Setting up Google OAuth 2.0 authentication,” Bettyblocks.com. [Online]. Available: https://docs.bettyblocks.com/en/articles/1012838-setting-up-google-oauth-2-0-authe ntication.

9. Appendix

index.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>G Drive Upload</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="main.js"></script>

    <style>
        .center {
         display: flex;
         justify-content: center;
         align-items: center;
         height: 200px;
         border: 3px solid green;
        }

       h1 {text-align: center;}

       .button  {
          background-color :  #4CAF50;
          border : none;
          color : white;
          padding : 15px 32px;
          text-align : center;
          text-decoration: none;
          display : inline-block;
          font-size: 17px;
          margin : 5px 3px;
          cursor : pointer;
       }
    </style>

 </head>
 <body>

    <h1>Click Upload Button to upload files to Google Drive using OAuth 2.0</h1>

  <div class="center">
       <button class="button" id="login">
          Upload
       </button>
  </div>
</body>
</html>
```

main.js

```
$(document).ready(function ()  {
   // client-id of this project
   var clientId = "279888638943-o33dgjbqfp5bp8otbqcsu1j2pfd1oe5m.apps.googleusercontent.com";


   // redirect_uri specifies the redirect path after the authentication
   var redirect_uri = "http://127.0.0.1:5500/upload.html";

   // The scope  of this project
   var scope  = "https://www.googleapis.com/auth/drive";

   //  this is the url which the user is redirected to (Currently not necessary)
   var url = "";

   //  This will take the clicking event of this button
   $("#login").click(function () {
      signIn(clientId, redirect_uri, scope, url);
   });

   function signIn(clientId ,  redirect_uri ,  scope, url) {
      // This is the actual URL that the user will get redirected to
      url = "https://accounts.google.com/o/oauth2/v2/auth?redirect_uri=" + redirect_uri
         + "&prompt=consent&response_type=code&client_id=" + clientId + "&scope=" + scope
         + "&access_type=offline";

      // redirecting to the URL
      window.location = url;
   }
});
```

upload.html

```
 <!DOCTYPE  html>
<html>
 <head>
     <meta charset="utf-8" />
     <meta http-equiv="X-UA-Compatible" content="IE=edge">
     <title>G Drive Upload</title >
     <meta name="viewport" content ="width=device-width, initial-scale=1">
     <link rel="stylesheet" type="text/css" media="screen" href="upload.css" />
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
     <script src="upload.js"></script>

    <style>
        .center {
        display: flex;
        justify-content :  center;
        align-items : center;
        height: 200px;
        border: 3px solid green;
        }

    </style>

</head>
<body >
    <h1><Center>Upload a file</Center></h1></t></t>
    <p><Center>Click "Choose files" to select any file from your pc and then click "Upload" to upload it to
GOOGLE DRIVE.</Center></p></t></t>
   <div >
    <div class="center">

     <input id="files" type="file" name="files[]" multiple/>
     <button id="upload" >Upload</button>

     </div>
     <div id="progress-wrp">
        <div class="progress-bar"></div>
        <div class="status">0%</div>
    </div>
    </div>
   <div id="result"></div>
</body>
</html>
```

upload.js

```
 $(document).ready(function () {
    const urlParams = new URLSearchParams(window.location.search ) ;
    const code = urlParams.get('code') ;
    const redirect_uri = "http://127.0.0.1:5500/upload.html" ; // The redirect uri
    const client_secret = "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"; // Project Client Secret
    const scope = "https://www.googleapis.com/auth/drive";
    var access_token = "";
    var client_id = "279888638943-o33dgjbqfp5bp8otbqcsu1j2pfd1oe5m.apps.googleusercontent.com"; // Project
Client ID


   $.ajax({
        type: 'POST',
       url : "https://www.googleapis.com/oauth2/v4/token",
       data: {
            code: code,
            redirect_uri : redirect_uri,
            client_secret : client_secret,
            client_id: client_id,
            scope : scope ,
            grant_type: "authorization_code"
       },
       dataType: "json",
        success: function (resultData) {
            localStorage.setItem("accessToken", resultData.access_token);
            localStorage.setItem("refreshToken", resultData.refreshToken);
            localStorage.setItem("expires_in", resultData.expires_in);
            window.history.pushState( {}, document.title, "/gDriveUpload/" + "upload.html");
        }
   } ) ;

   function stripQueryStringAndHashFromPath(url)  {
       return  url.split("?")[0].split("#")[0];
    }

   var Upload = function (file) {
       this.file = file;
   };

   Upload.prototype.getType = function () {
       localStorage.setItem("type", this.file.type);
       return this.file.type;
   };

   Upload.prototype.getSize = function () {
        localStorage.setItem("size", this.file.size);
        return this.file.size;
    } ;

   Upload.prototype.getName = function () {
       return this.file.name;
    } ;

   Upload.prototype.doUpload  = function ()  {
       var that = this;
       var formData = new FormData();

       // add assoc key values, this will be posts values
       formData.append("file", this.file, this.getName());
       formData.append("upload_file", true);

       $.ajax( {
           type:  "POST",
           beforeSend: function (request)  {
               request.setRequestHeader("Authorization",  "Bearer" + " " + localStorage.getItem("accessToken"));

           } ,
           url : "https://www.googleapis.com/upload/drive/v2/files",
           data: {
               uploadType : " media"
           },
            xhr : function () {
               var myXhr = $.ajaxSettings.xhr();
                if (myXhr.upload) {
                   myXhr.upload.addEventListener('progress', that.progressHandling, false);
                }
                return myXhr ;
           } ,
           success : function  (data) {
               console.log(data);
           },
            error : function (error) {
                console.log(error);
           } ,
            async: true,
            data: formData,
            cache: false,
            contentType : false,
            processData: false ,
            timeout : 60000
       });
   };

   Upload.prototype.progressHandling = function (event) {
       var percent =  0;
       var position =  event.loaded || event.position;
       var total = event.total;
       var progress_bar_id  =  "#progress-wrp";
       if  (event.lengthComputable) {
           percent = Math.ceil( position/ total* 100);
       }
       // update progress-bars classes so it fits the code
       $(progress_bar_id + " .progress-bar").css("width", +percent + "%");
       $(progress_bar_id + " .status").text(percent + "%");
   };

   $("#upload").on("click",  function (e) {
           var file = $("#files")[0].files[0];
           var upload = new  Upload(file);


           // executing the upload
           upload.doUpload();
       });

   });
```

upload.css

```
#progress-wrp  {
     border : 3px solid #0098CC;
     padding:  2px;
     position: relative;
     height:  31px;
     border-radius : 5px;
     margin :  11px;
     text-align : left;
     background : #fff;
     box-shadow : inset 1px 3px 6px rgba(0, 0, 0, 0.13);
  }

  #progress-wrp .progress-bar {
     height : 100%;
     border-radius : 4px;
     background-color:  #148F78;
     width : 0;
     box-shadow: inset 2px 1px 10px rgba(0, 0, 0, 0.10);
  }

  #progress-wrp .status {
    top : 4px;
    left : 51%;
    position : absolute;
    display : inline-block;
    color: #000000;
  }
```