

Readme File

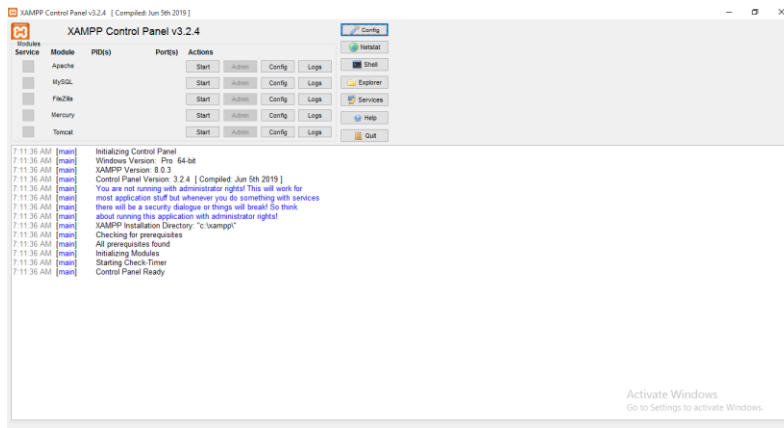
Software Security
OAuth 2.0
Assignment

Content

Starting Server-----	3
Open xampp – For this system we have used xampp as web server therefore first we need to run the server first.-----	3
Register Function-----	4
How to get client id and secret -----	5
How to change main.js path and upload.js path -----	9
Enabling APIs-----	11
Login-----	12
Upload files to google drive-----	12

Starting Server

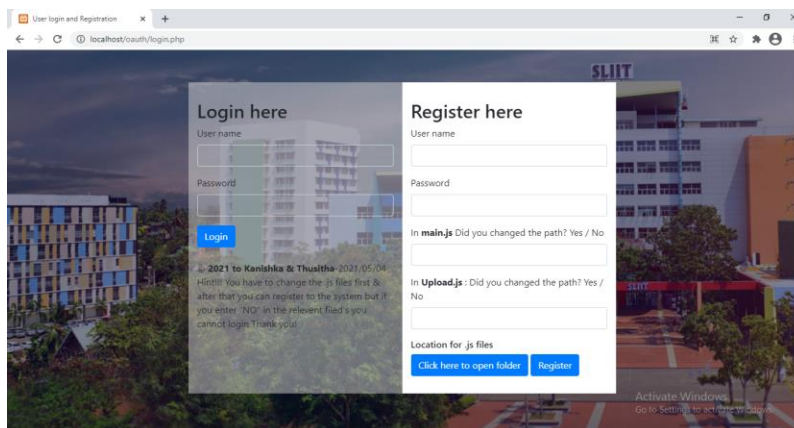
Open xampp – For this system we have used xampp as web server therefore first we need to run the server first.



Start apache and mysql and make sure that no errors occurred and services are running if not troubleshoot and solve it (in most cases program does not work due to used/blocked ports)



Open your favorite web browser and go to <http://localhost/oauth/login.php>



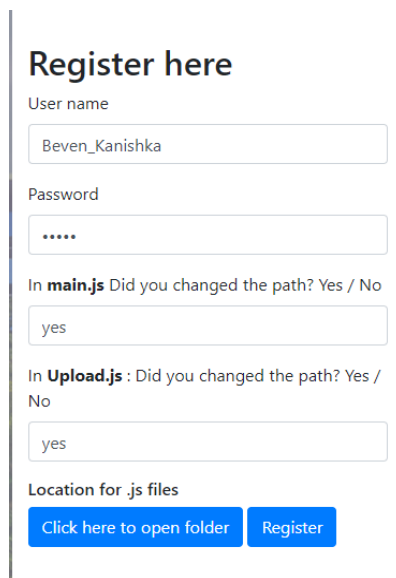
In here user have two options to choose from where, login method for users who have already registered for applicating by using registering function and added the relevant tokens to the paths, whereas register functions is for users who are new to the system and they need to fill up the relevant information (i.e., username and password and add the token paths to code) and create account.

Note: *one user at a time is allowed in a system, in an event of switching the user or the account, token paths need to be added manually at each time since the paths does not manage by the local SQL database.*

Register Function

In here user gets to choose username and password for the account which is maintain by SQL database.

- Username – The username should be meaning full name which can contain alpha numeric characters and the user's name should not contain any spaces in between.
- Password – The password can contain any number of characters there are no restrictions for that, however it is recommended to use strong password which contains at least 6 digits
- Main.js path and Upload.js – the paths need to configure manually at each time user accounts are switching.



Register here

User name

Password

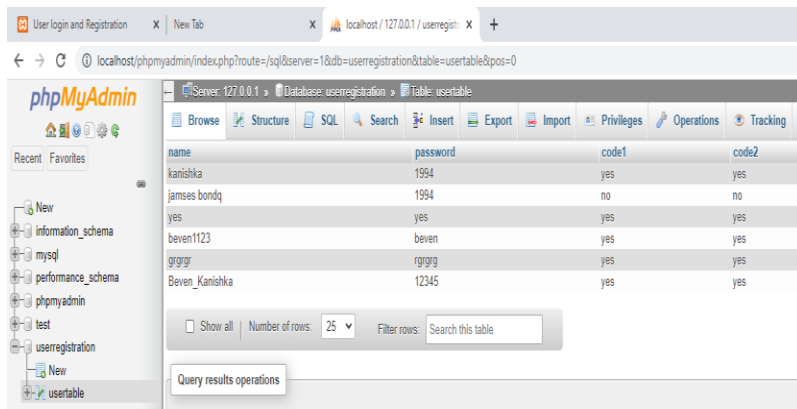
In **main.js** Did you changed the path? Yes / No

In **Upload.js** : Did you changed the path? Yes / No

Location for .js files

[Click here to open folder](#) [Register](#)

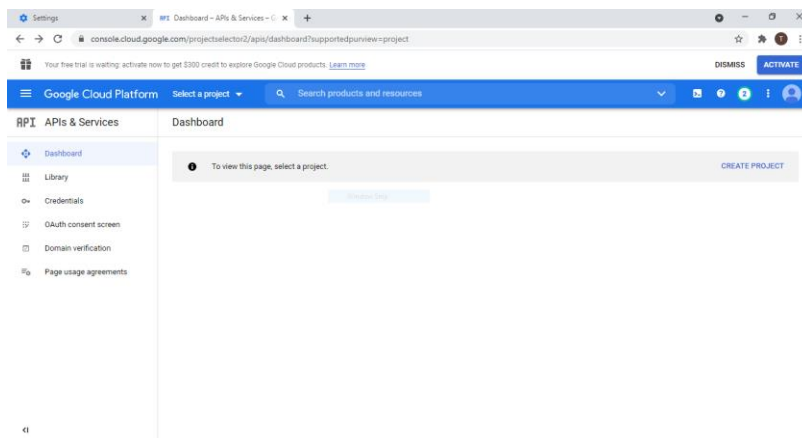
Once the user click register data is being stored in the database. As shown in the bellow figure



However, system does not allow any users haven't change main.js path and upload.js path and filed "yes" as in the form to pass to login form and they will be redirected to login page again, which was implemented in order to reduce server and network traffic.

How to get client id and secret

Go to <https://console.cloud.google.com/projectselector2/apis/dashboard?supportedpurview=project>



And then select create project option and give a project name and click create option and then it will take while to create project.

Google Cloud Platform Search products and resources

New Project

You have 11 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
Oauth 2.0 ss ?

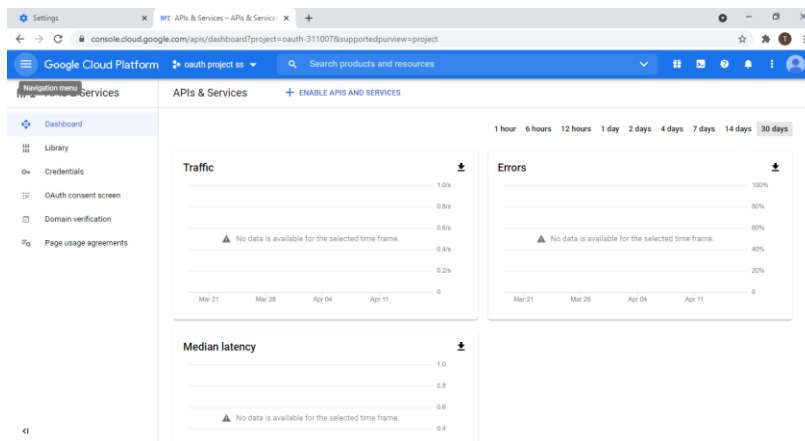
The name has invalid characters. Enter letters, numbers, single quotes, hyphens, spaces or exclamation points.

Location *
No organization BROWSE

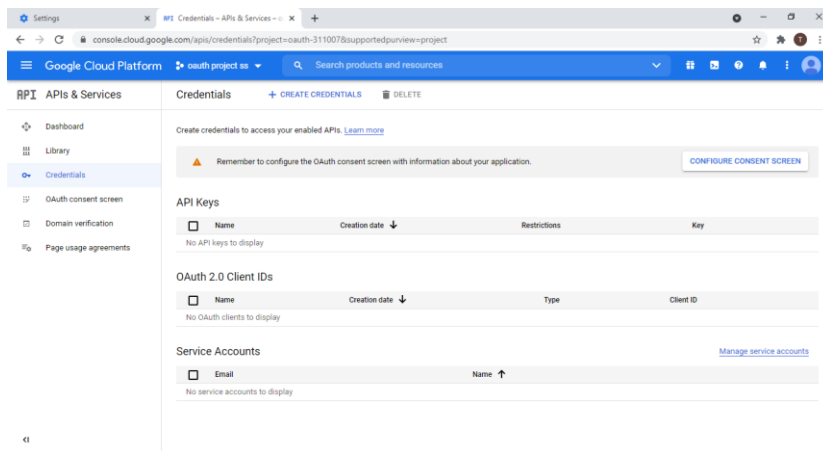
Parent organization or folder

CREATE CANCEL

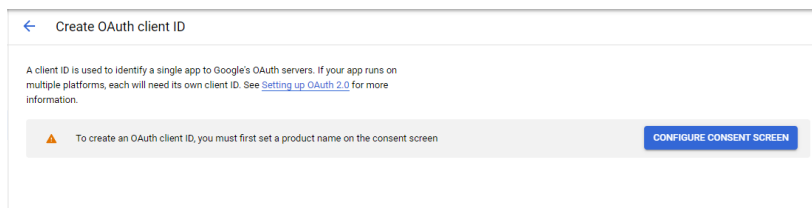
Once the creating project is done it will proceed to API & Services page.



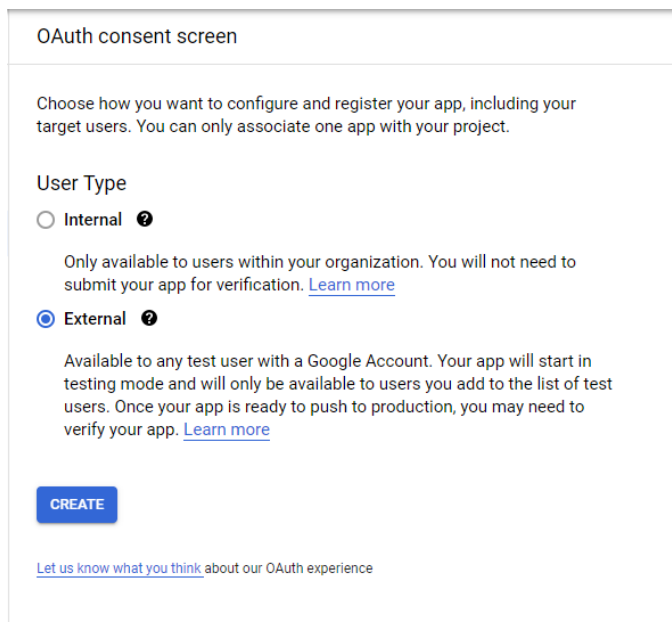
Then go to credentials and click on it and once the credential page is loaded click on create credentials. And under that Create OAuth client ID option need to be selected.



Then user need to be click the “Congifure consent screen”



Once it is done, user type will be asked for this application user need to provide “external” as user type and click on create.



Once it is done user will be asked to fill up app information. Such as app name, email etc. however only the app name and the user support email are the compulsory therefore they need to filled in order to continue. Once it is done click on “save and continue” on the bottom of the page.

Then again navigate back to credentials page > create credentials > Create OAuth client ID. On that page from the drop-down list select the application type. (here for this case, we need chose web application).

And (you can edit client name in necessary) if not scroll down to bottom of the page and their user need to add the address of the page “Authorized JavaScript origins”. Since we are running it as local host field need to be filled as “http://localhost” , and then under “Authorized redirect URIs” <http://localhost/oauth/upload.html> this URL need to inserted, note if the names of the files were changed then the address names also need to be changed otherwise it will not work correctly.

URIs

http://localhost

+ ADD URI

Authorized redirect URIs ?

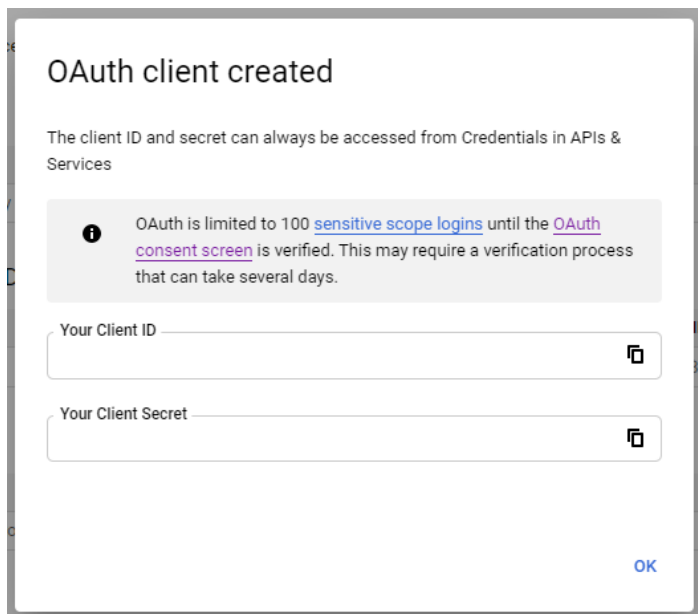
For use with requests from a web server

URIs

http://localhost/oauth/upload.html

+ ADD URI














And click the create. Once it is done Client id and Client secret will be displayed on the screen, it also allows users to copy the id and secret, therefore it is recommended to copy it and save it on notepad or any other text file will be useful for the next few steps.



How to change main.js path and upload.js path

We have used xampp server for this demonstration, therefore the exact locations and some paths maybe different for other application and errors might cause due to them. In order to change the path first need to locate the source file, in the Xampp server the source files are contain in the “htdocs” folder. In most cases location will be C:\xampp\htdocs.

Once you locate the htdocs you need to identify the folder which contain the sources file. Here in this case, it is C:\xampp\htdocs\OAuth. Then you need to open the folder. And this is how it should look like.

	google	9/29/2020 5:41 PM	JPG File	37 KB
	image	4/16/2021 8:35 AM	JPG File	306 KB
	index	4/17/2021 12:45 PM	PHP File	3 KB
	login	4/16/2021 9:14 AM	PHP File	3 KB
	logout	4/9/2021 8:57 PM	PHP File	1 KB
	main	4/16/2021 8:56 AM	CSS File	3 KB
	main	3/25/2021 1:31 PM	JavaScript File	1 KB
	registration	4/16/2021 9:19 AM	PHP File	2 KB
	style	4/15/2021 9:22 AM	CSS File	1 KB
	upload	4/16/2021 9:01 AM	CSS File	2 KB
	upload	4/16/2021 9:04 AM	HTML File	2 KB
	upload	3/28/2021 9:51 PM	JavaScript File	5 KB
	validation	4/16/2021 9:21 AM	PHP File	1 KB

Then open main.js file using any text editor > then past the client id which was copied earlier in the process and save the file.

```
$(document).ready(function(){

    var clientId = "912398529236-aklhnqotm0ued5dfaakmcas8g4stee3.apps.googleusercontent.com" //ENTER YOUR CLIENT ID FROM THE GOOGLE DEVELOPER
    var redirect_uri="http://localhost:8080/OAuthproject/upload.html"
    var scope="https://www.googleapis.com/auth/drive"
    var url=""

    $("#login").click(function(){
        signIn(clientId,redirect_uri,scope,url);
    })

    function signIn(clientId,redirect_uri,scope,url){
        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";
        window.location = url;
    }
})
```

Then open Upload.js file using any text editor > then here you need to past both client id and client secret which was copied earlier in the process and save the file. That's it.

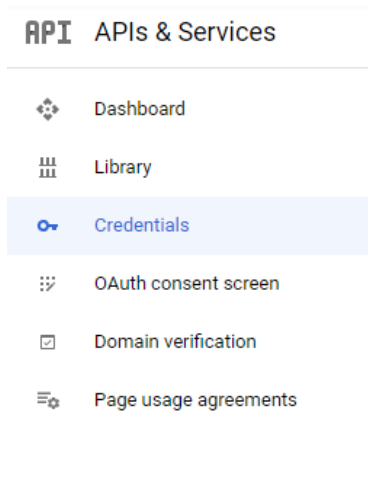
```
$(document).ready(function(){

    const urlParams = new URLSearchParams(window.location.search);
    const code = urlParams.get('code');
    const redirect_uri = "http://localhost/oauth/upload.html" // replace with your redirect_uri;
    const client_secret = "Ne93mkKkYt3dS2mbsvA5ZlnI"; // replace with your client secret
    const scope = "https://www.googleapis.com/auth/drive";
    var access_token = "";
    var client_id = "465458451916-m4mlpn93jbvumvubtmj09to777mpbu36.apps.googleusercontent.com" // replace it with your client id;

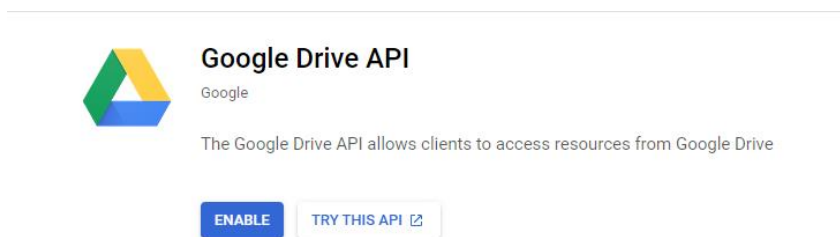
    $.ajax({
        type: 'POST',
        url: "https://www.googleapis.com/oauth2/v4/token",
        data: {code:code
            ,redirect_uri:redirect_uri,
            client_secret:client_secret,
            client_id:client_id,
            scope:scope,
            grant_type:"authorization_code"},
        dataType: "json",
        success: function(resultData) {
```

Enabling APIs

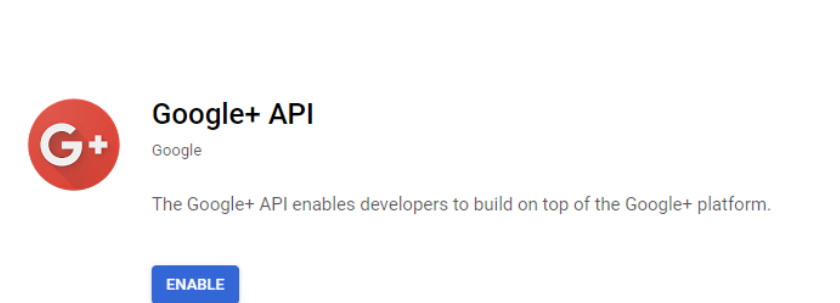
In order to enable APIs user need to navigate to library page.



And search google drive API and enable it.

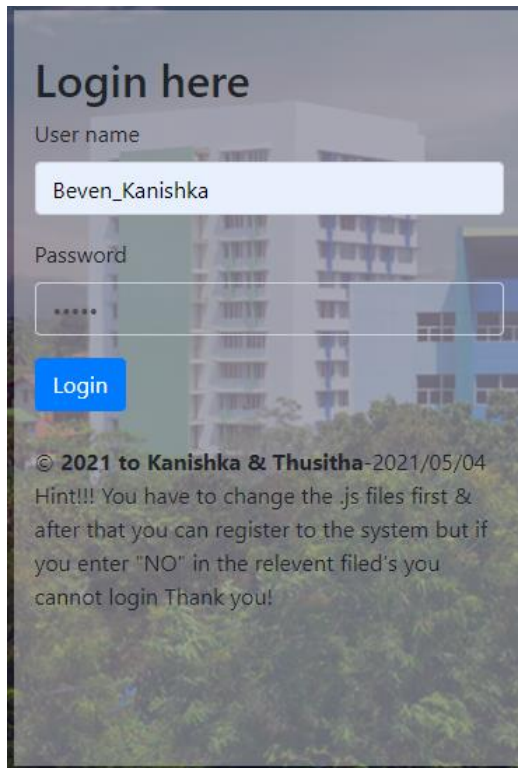


Then again navigate to library and search for google+ API and enable it, that's it.



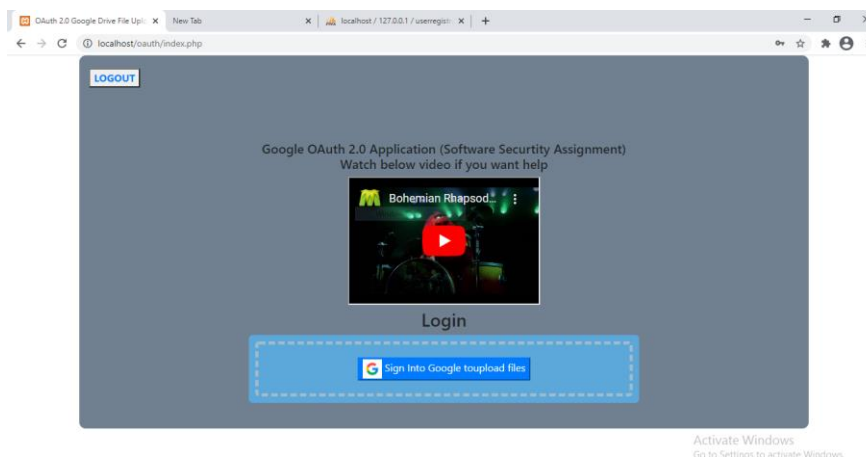
Login

Once the register process is completed user can login to the system using username and password. If by any chance user try to login with bogus credentials user will be re-directed to the login page again.



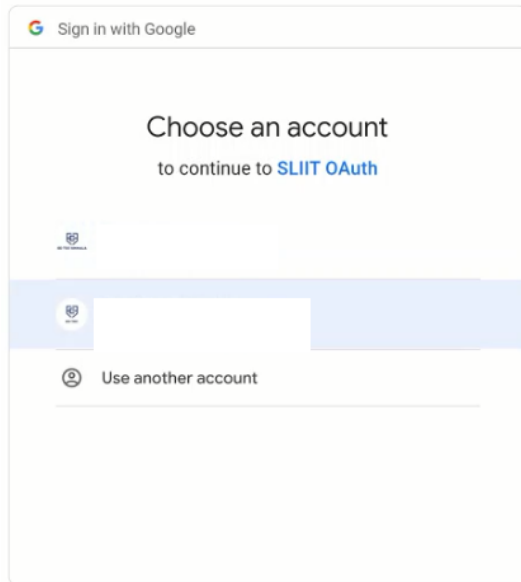
Upload files to google drive

Once user input valid credentials and click on “login” user will be redirected to file upload process where users have to login with google account credentials.



Then user can select account or add a new account by providing valid email and password. Once it is done the if the invputs are valid then it will proceed to the next page.

Note: Google account should be matched with client id and client secret.



Note: In sometimes error might occur as shown in the bellow diagram. Yet there is no hart or malfunction of the system it is just a warring. In order to soleve it. Go to Advanced > go to slit oauth (unsafe)



Google hasn't verified this app

The app is requesting access to sensitive info in your Google Account. Until the developer (edtec.thusitha@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](#)

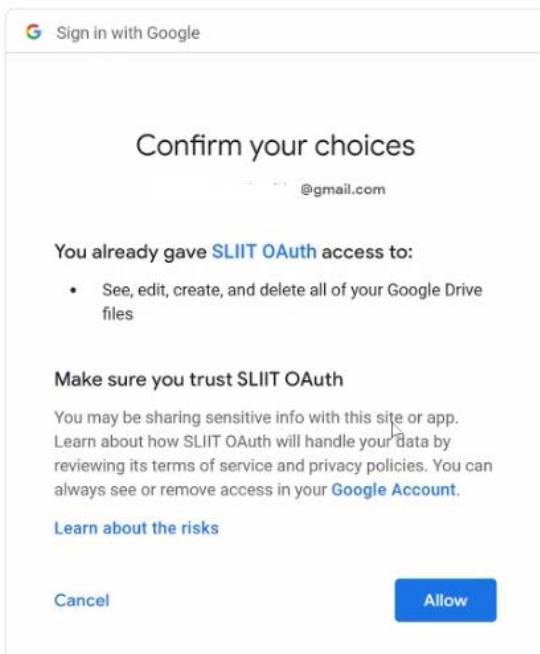
[Hide Advanced](#)

[BACK TO SAFETY](#)

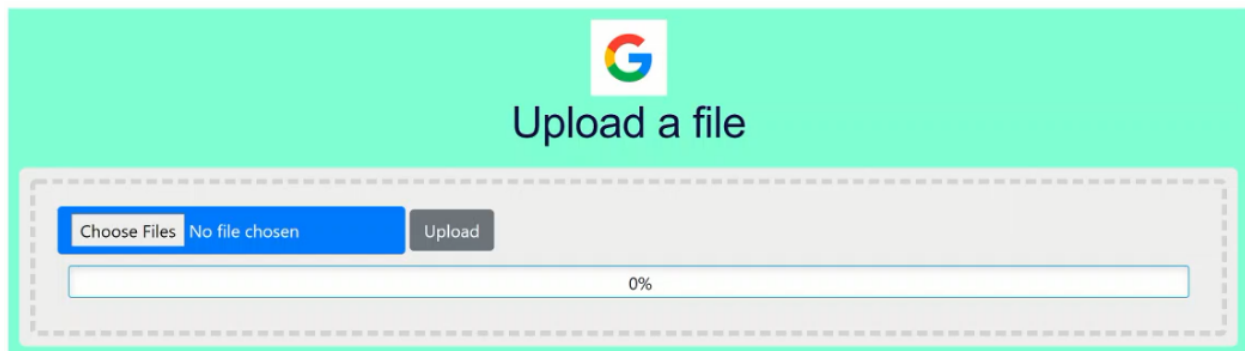
Continue only if you understand the risks and trust the developer (edtec.thusitha@gmail.com).

[Go to SLIIT OAuth \(unsafe\)](#)

Then user need to provide access by clicking “allow” in case if the user deny the allow request it will lead back to google login page.



Once the permission is granted the user will proceed to file upload page where user gets to choose/browse files which need to be uploaded. From there user can select files which are locally available in the pc or any other storage medium.



Once the files are selected then user can proceed with upload function by clicking the upload button then the loading bar will display the upload progress once it reached 100% the uploading process is completed. And user can view files via their google drive.