

Sri Lanka Institute of Information Technology (SLIIT)



Software Security – Assignment 2

Lecturer: Dr. Dharshana Kasthurirathna

Group Members:

MS20920340 – T.M.T.S.B.P. Tennakoon

MS20919764 – B.K. Liyanage

Content

1. Overview	4
2. Work Flow of system	4
2.1. Flowchart	6
2.2. Sequence Diagram	7
2.2.1. Sequence diagram for login process.....	7
2.2.2. Sequence diagram for file uploading	8
3. Front end and the backend of the System.....	9
3.1. OAuth 2.0	10
4. Appendix A	12
5. Appendix B	13
6. Appendix C	17
6.1. User manual	17
6.1.1. Staring Server	19
6.1.2. Register Function	20
6.1.3. How to change main.js path and upload.js path	25
6.1.4. Enabling APIs	26
6.1.5. Login.....	27
6.1.6. Upload files to google drive	28
7. Appendix D	31
7.1.1. INDEX PAGE.....	31
7.1.2. LOGIN PAGE	32
7.1.3. LOGOUT PAGE.....	34
7.1.4. MAIN .JS PAGE	37
7.1.5. REGISTRATION .PHP PAGE	38
7.1.6. STYLE .CSS PAGE.....	39
7.1.7. UPLOAD.CSS PAGE	40
7.1.8. UPLOAD .HTML PAGE.....	41
7.1.9. UPLOAD.JS PAGE	43
7.1.10. VALIDATION PAGE.....	45

List of Figures

Figure 1.0 – Cross functional flow chart of web system (Authour developed)	6
Figure 1.1 – Sequence digaram of login process	7
Figure 1.2 – Sequence digaram of file upload process	8
Figure 1.3 – how Oauth works [1].....	10

1. Overview

The developed system “SLIIT OAuth 2.0 file upload” is a web-based file uploading system which was developed according to client and server architecture, which allows clients to upload files to google drive through the application in addition it also allows users to view the uploaded files which can be accessed through link provided in the file upload page. Moreover, the application has included primary security mechanism where username and password are required in order to access the file upload process which is handled by local database developed using SQL. In apart form that valid google account is also essential requirement of this application. Furthermore, the application was developed using a variety of server and client-side scripting and programming languages, as well as the OAuth 2.0 authorization, which allows a third-party web site or application to access the user's protected resources without exposing the user's credentials and their identity. The system is well developed using different type of graphics and colors with images which facilitate easy to use while maintaining the simplicity of the system.

2. Work Flow of system

This explains the work flow of the system for aditinal details regarding fungtanality and fetures of the system plese reffer the User manual at the appendix.

1.	The web application was develop as a file upload system which can upload files to the google drive. In the system at first client need to open web brower and type localhost address (since the server have’nt hosted yet this is how it should run in the client side).
2.	Once the client do that it web browser will open the login potal of the system, wherer client can either login or create a account using register function.
	<p>2.1 – Register function – In the register function user can choose username and password fortheir convineance, once it is done user need to change the path of the JavaScript file by entering client ID and Client Secret from google and ansers the path change quistion. And submit. Then the user data will be added to the local SQL server and create an entry.</p> <p>2.2 - Login – Once the user has registered for system, user can enter Username and password.</p>
3.	Validation (Username/password/js paths) – If the user account information or Js path information make any false value the system will return back to login page. Else the user wil be redirected to file upload page.
4..	Granting permissin for web application to access google drive – in here web application is requesting permisson through the user to access google drive therefore user has to log his/her google account using gmail and password.
	4.1 If - the user gmail and password is incorrecr user will be re-directed to google login again.

	<p>4.2 Else – Google will validate information and passes the message to the web app and the web app sends the authorization request which need to be approved by the user, then google will ask the user to allow or deny the authorization.</p> <p>4.2.1 If - the user refuse to allow user will be re-directed to login and session will be expired.</p> <p>4.2.2 Else – User allows the web application to use the google drive behalf, then google will provide a token to the application which is an authorization token for users google services (which included file uploading to drive). Then the user will be redirected to File upload page.</p>
5.	Browse/ Select / Choose files – In here user can browse files from local storage medium.
6.	Upload files - Once the files are selected upload function can be used and the progress bar represents the upload progress in the system. Once it is 100%. Files can be viewed from the Google Drive.

2.1. Flowchart

A flowchart is a formalized graphic representation of information about steps or a sequence of events which consist of series of different symbols in addition it illustrate the flow direction of the process [10]. Moreover, purpose of a flow chart is to provide people with a common language or reference point when dealing with a project or process.

Where as a cross-functional flowchart diagram illustrate relationships between functions in a grid-like diagram which consist sections phases in a process [9].

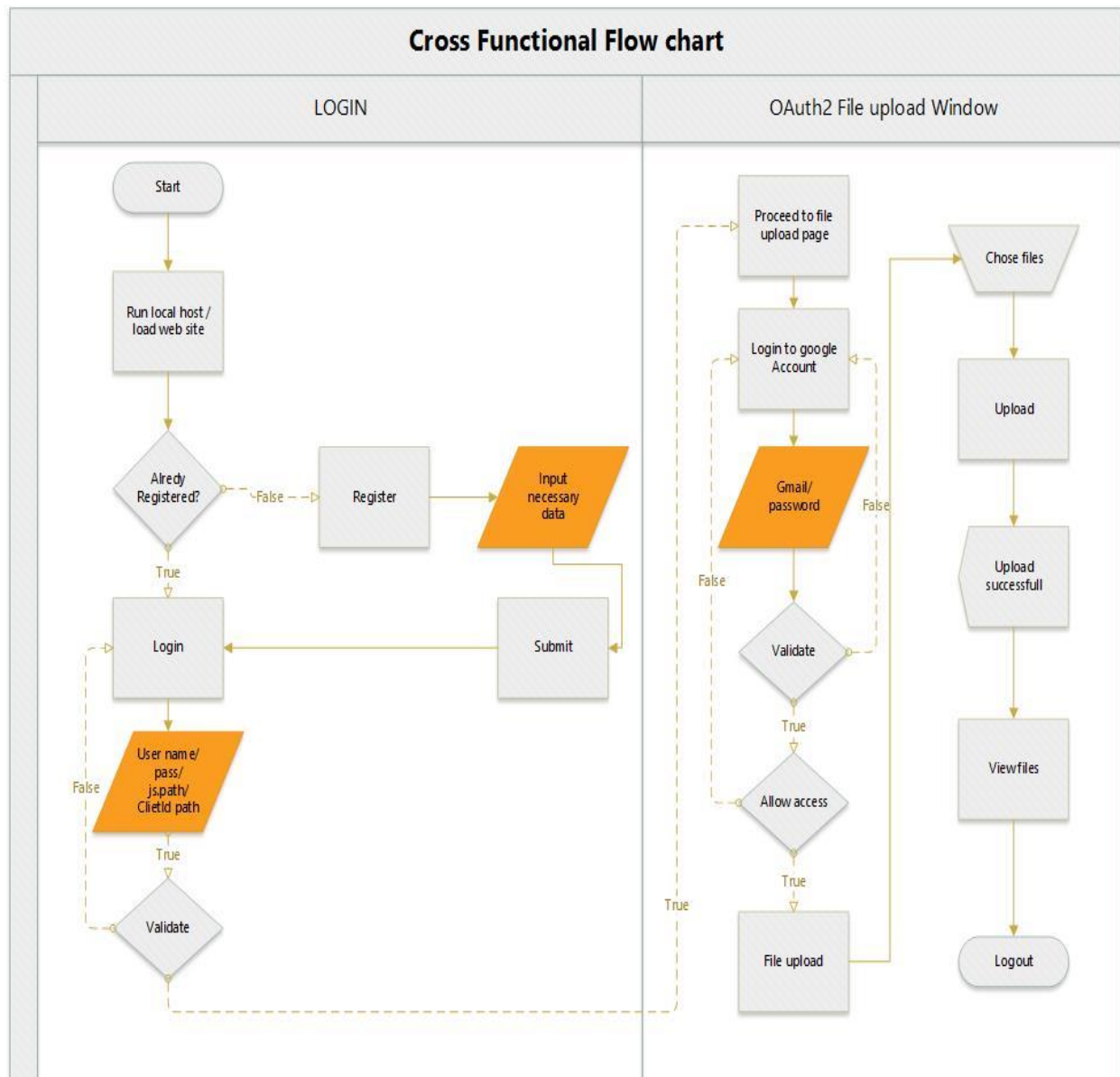


Figure 1.0 – Cross functional flow chart of web system (Authour developed)

2.2. Sequence Diagram

A sequence diagram is a type of conceptual diagram that represents the sequence of interactions between objects. Sequence diagrams are often used in specification and design to describe a software system, for example as part of UML activity diagrams. They show the order and timing relationships between messages passed by objects within a system.

A sequence diagram can be valuable in understanding all the possible behaviors within an object oriented program without need to examine its source code [8] [7].

2.2.1. Sequence diagram for login process

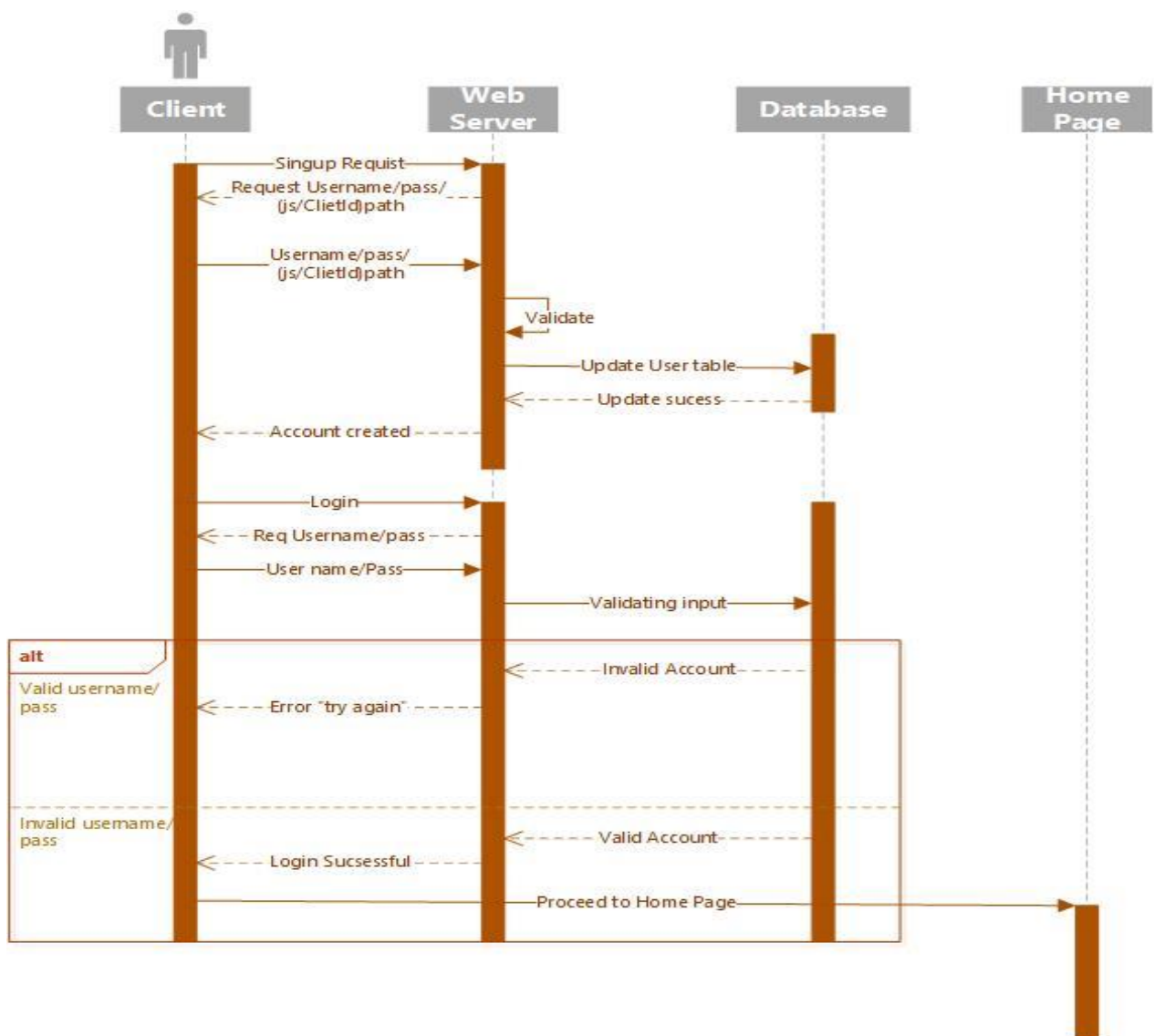


Figure 1.1 – Sequence digaram of login process

2.2.2. Sequence diagram for file uploading

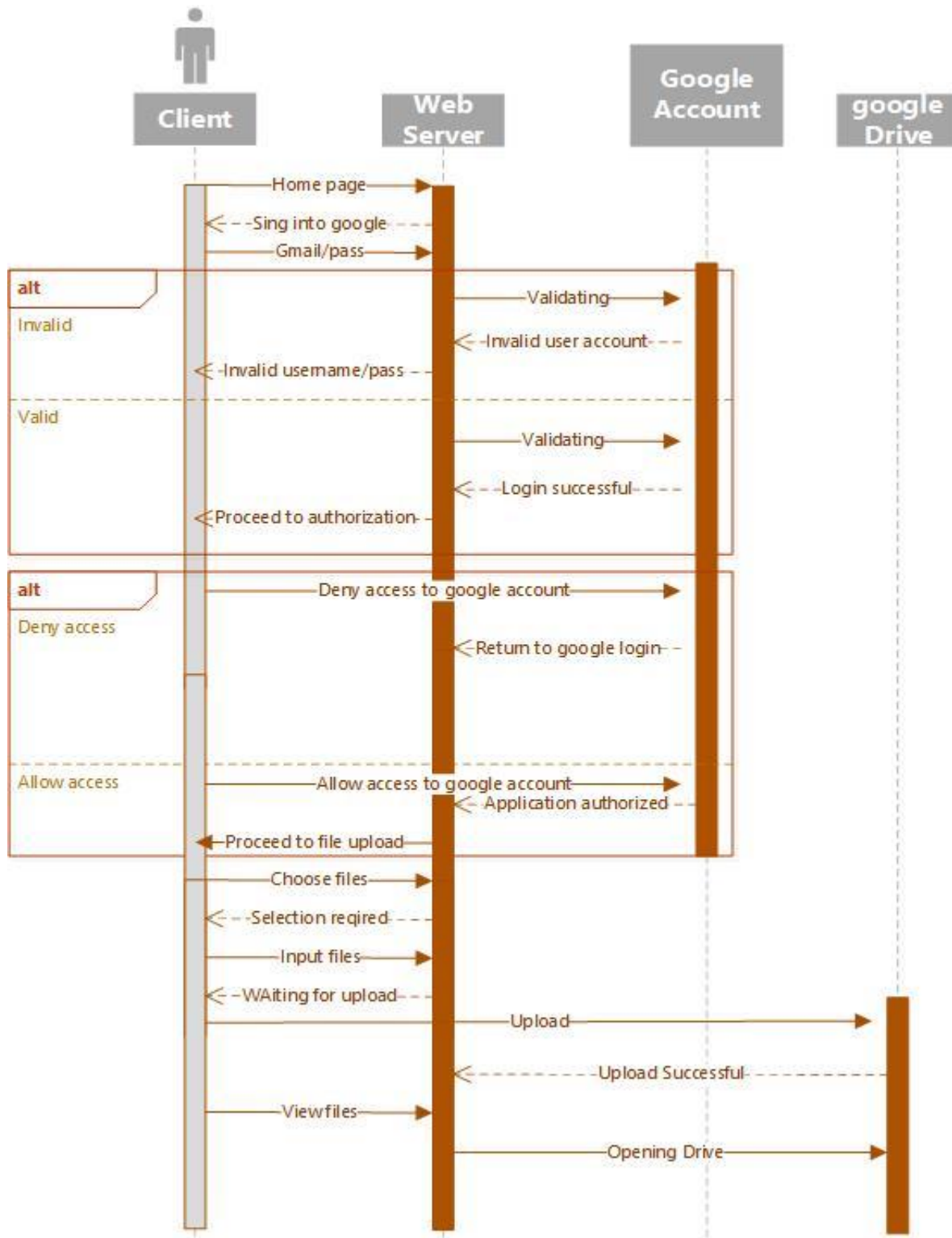


Figure 1.2 – Sequence diagram of file upload process

3. Front end and the backend of the System

The front end of the system, in this scenario the file upload system was developed using HTML, CSS and JavaScript. Where a range of color schemes, images and many more graphics are being included for the design of the web application. On the other hand, backend of the system where the database, HTML, PHP and SQL is being used to power up the web application, this is the place where all the gadgets in the frontend making them functional in order to fulfill the client's needs and wants. The following are the programming languages and data base languages which have been used in the system.

HTML	HTML stands for Hyper Text Markup Language, which is a standardized markup language used to format Web pages and web applications. It also included elements that guide the browser to display the content (e.g., "this is a heading", "this is a paragraph", "this is a link", etc.) [1] Moreover, it can also benefit from innovations like Cascading Style Sheets and scripting languages like JavaScript.[2]. Furthermore, there are several versions of HTML and HTML5 is the newest version.
CSS	CSS stands for Cascading Style Sheets, and it's a style sheet language for describing the presentation and formatting of a document written in markup language. It adds to the functionality of HTML. It's commonly combined with HTML to improve the appearance of web pages and user interfaces. However, most websites use CSS in combination with HTML and JavaScript to build user interfaces for web apps and many mobile applications. [3].
Java Script	According to JavaScript also known as JS, which was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. This is client-side scripting language which is being developed to interact with the user and make dynamic pages. Moreover, it is an interpreted programming language with object-oriented capabilities. Furthermore, JavaScript allows users to build modern web applications to interact directly without reloading the page every time which reduces the server load as well as immediate feedback to clients[4].
PHP	PHP is an open-source server-side scripting language which was initially invented in year 1994 by Danish-Canadian programmer Rasmus Lerdorf. PHP was previously referred to as Personal Home Page however, at present it is referring as the Hypertext Preprocessor. Furthermore, PHP is a server-side scripting language that is used for developing static and dynamic websites. Moreover, PHP also can be used develop web applications. [5] In addition PHP is simple and easy to understand language which does not require prior knowledge of HTML. Apart from that since PHP is a server-side scripting language, it is executed on the server; hence, client machines do not need to have PHP installed [6].
SQL	This stands for structured query language which is a common language is used variety of applications. It helps you to browse and manipulate data as well as perform database record insertion, scan, update, and deletion.

3.1. OAuth 2.0

OAuth is a framework for authorization that is used as a standard protocol. It operates by granting third-party applications access to the user account and delegating user authentication to the provider that hosts the account [1]. In this web-application OAuth 2.0 has been used in order to access clients google drive. The below diagrams show the workflow of OAuth.

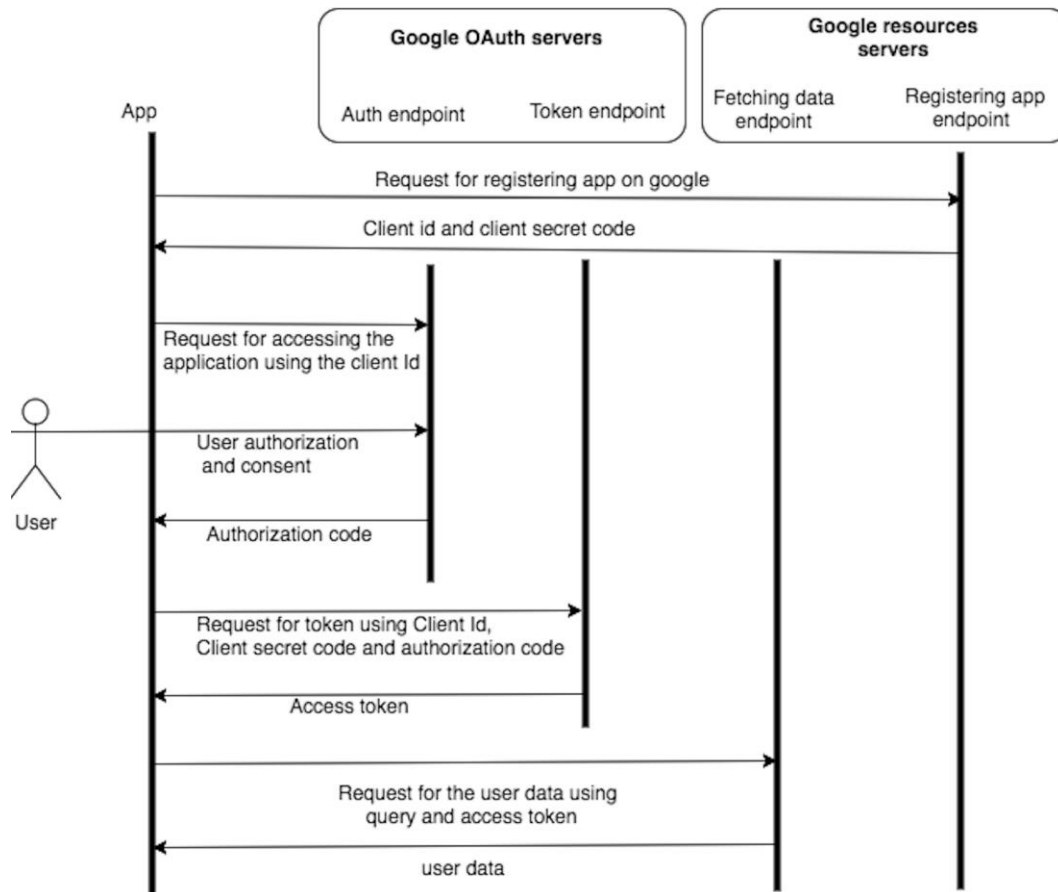


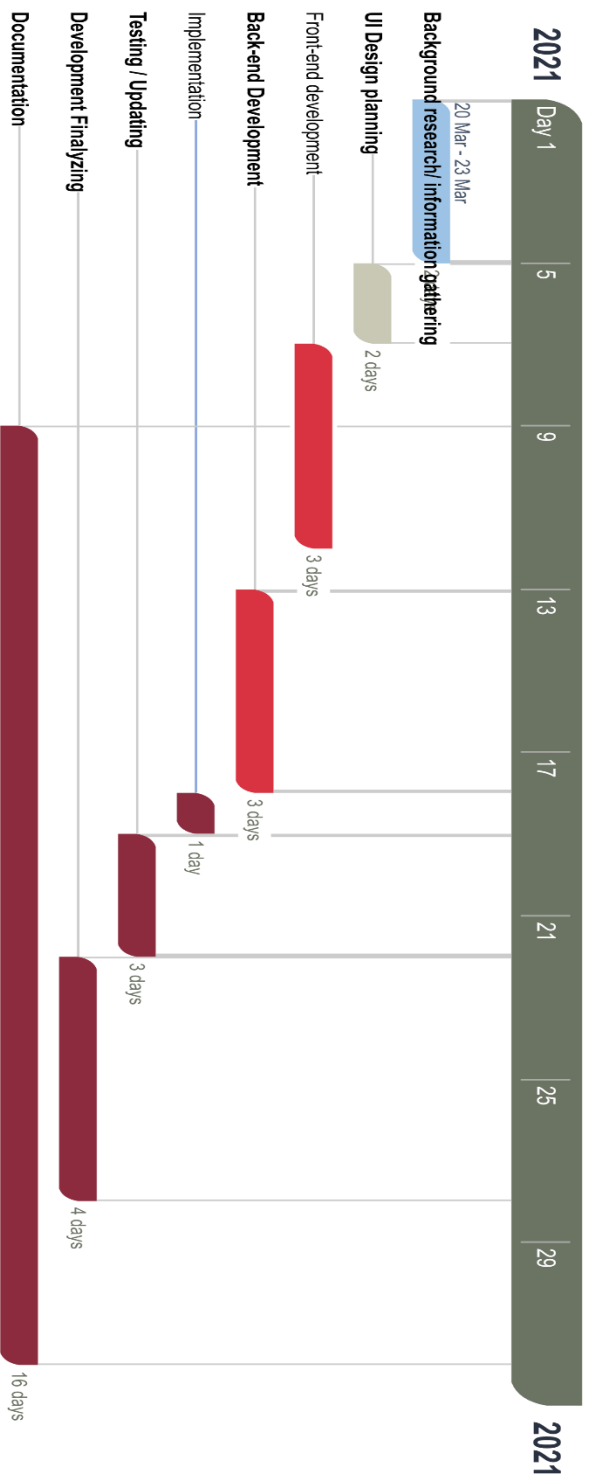
Figure 1.3 – how OAuth works [1]

As illustrated in the figure 1.3, the web-application makes the request to auth server in this case google auth server, then the server sends a response request to user by mentioning the authorization privileges, once the user allows the authorization, auth server sends the authorization code to the application then the application is sending token request to auth server and then access token is granted by the auth server. This is one time process where token is being issued once after that application can use the authorized token to access services according to provided privileges.

References

- [1] N. Hebb, "what-is-a-flow-chart," Breezetre, 2018. [Online]. Available: <https://www.breezetre.com/articles/what-is-a-flow-chart>. [Accessed 28 03 2021].
- [2] T. Contributor, "flowchart," Techtarget, 2021. [Online]. Available: <https://whatis.techtarget.com/definition/flowchart>. [Accessed 28 03 2021].
- [3] www.techopedia.com, "Sequence Diagram," techopedia, 2021. [Online]. Available: <https://www.techopedia.com/definition/27490/sequence-diagram-uml>. [Accessed 28 03 2021].
- [4] www.geeksforgeeks.org, "Unified Modeling Language (UML) | Sequence Diagrams," Geeksforgeeks, 12 02 2018. [Online]. Available: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>. [Accessed 28 03 2021].
- [5] www.w3schools.com, "HTML Introduction," w3schools, 2021. [Online]. Available: https://www.w3schools.com/html/html_intro.asp. [Accessed 28 03 2021].
- [6] www.javatpoint.com, "What is HTML," javatpoint, 2021. [Online]. Available: <https://www.javatpoint.com/what-is-html>. [Accessed 28 03 2021].
- [7] www.javatpoint.com, "what is CSS," javatpoint, 2021. [Online]. Available: <https://www.javatpoint.com/what-is-css>. [Accessed 28 03 2021].
- [8] www.javatpoint.com, "javascript-tutorial," javatpoint, 2021. [Online]. Available: <https://www.javatpoint.com/javascript-tutorial>. [Accessed 28 03 2021].
- [9] www.guru99.com, "What is PHP? Write your first PHP Program," Guru99, 2021. [Online]. Available: <https://www.guru99.com/what-is-php-first-php-program.html>. [Accessed 28 03 2021].
- [10] www.educative.io, "what is php," Educative, 2021. [Online]. Available: <https://www.educative.io/edpresso/what-is-php>. [Accessed 28 03 2021].
- [11] A. Parecki. [Online]. Available: <https://oauth.net/2/>. [Accessed 04 04 2021].
- [12] M. Anicas, "An Introduction to OAuth 2," 21 06 2014. [Online]. Available: <https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>. [Accessed 04 04 2021].

Software Security OAuth 2.0

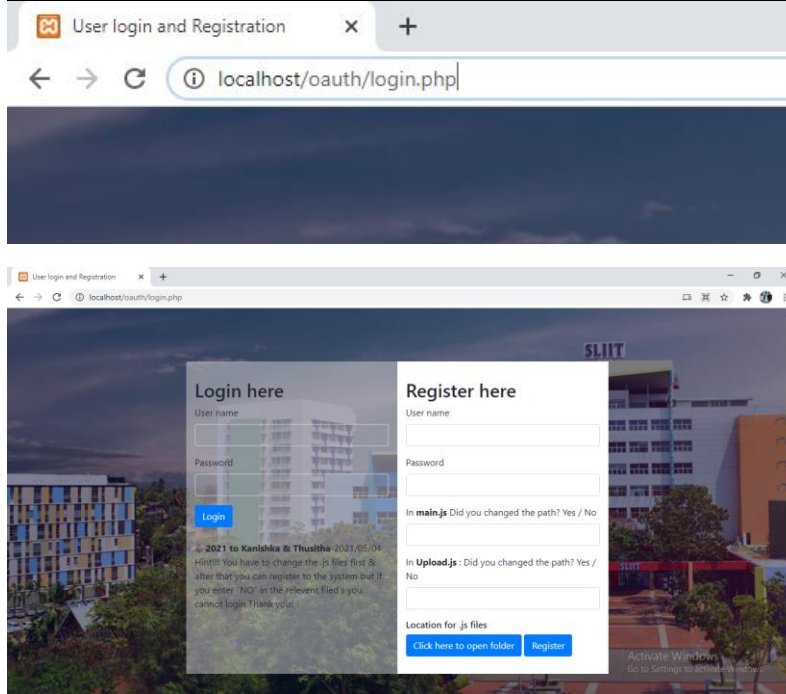


4. Appendix A

5. Appendix B

5.1. Test case 1 – Loading web application

Test no	Expected output	Status
1. Loading local host/Oauth	Display the login page	Pass



5.2. Test case 2 – “Register” function (Creating account)

Test no	Expected output	Status
1. Enter Username/password and main.js path “yes”/ Upload.js path “yes”	Page refreshed and data stored in database	Pass
2. Enter Username/password and main.js path “no”/ Upload.js path “no”	Page refreshed and data stored in database	Pass

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](#)

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

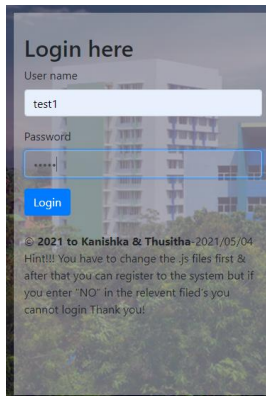
Location for .js files

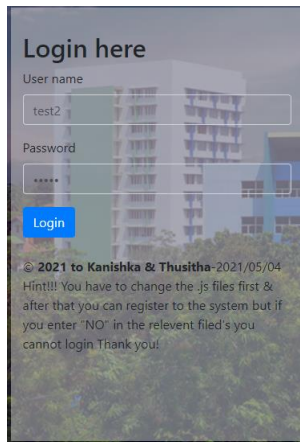
[Click here to open folder](#)
[Register](#)

username	password	main.js	upload.js
test1	test1	yes	yes
test2	test2	no	no

5.3. Test case 3 – Login using Username and Password

Test no	Expected output	Status
1. Enter valid Username/password of account which has main.js path “yes” and Upload.js path “yes”	Proceed to File transfer page	Pass
2. Enter valid Username/password of account which has main.js path “no” and Upload.js path “no”	Login failed and refresh the page back to login page	Pass
3. Enter Invalid Username/password	Login failed and refresh the page back to login page	Pass





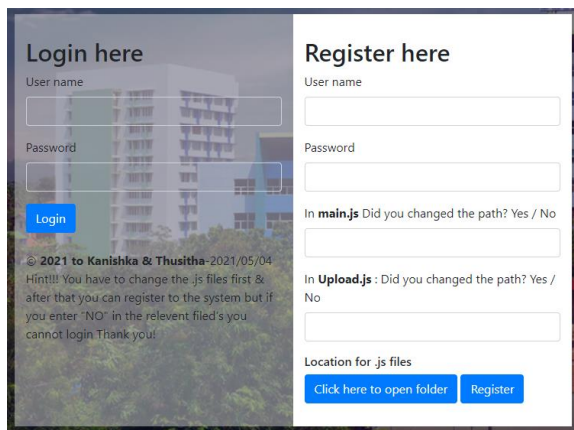
Login here

User name

Password

Login

© 2021 to Kanishka & Thusitha-2021/05/04
Hint!!!! You have to change the .js files first & after that you can register to the system but if you enter "NO" in the relevent filed's you cannot login Thank you!



Login here

User name

Password

Login

© 2021 to Kanishka & Thusitha-2021/05/04
Hint!!!! You have to change the .js files first & after that you can register to the system but if you enter "NO" in the relevent filed's you cannot login Thank you!

Register here

User name

Password

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

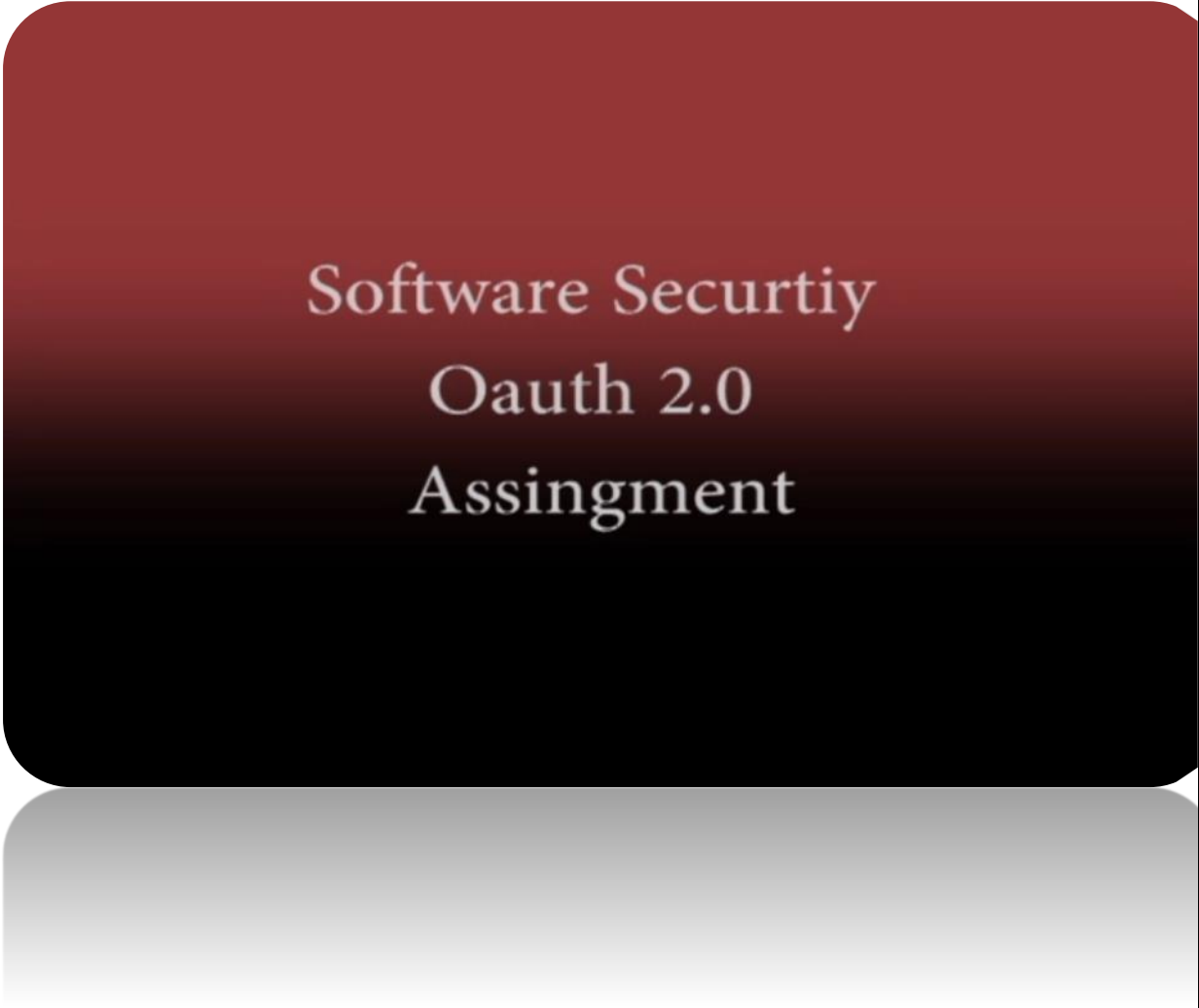
In **Upload.js** : Did you changed the path? Yes / No

Location for .js files

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

6. Appendix C

6.1. User manual



Software Security OAuth 2.0 Assignment

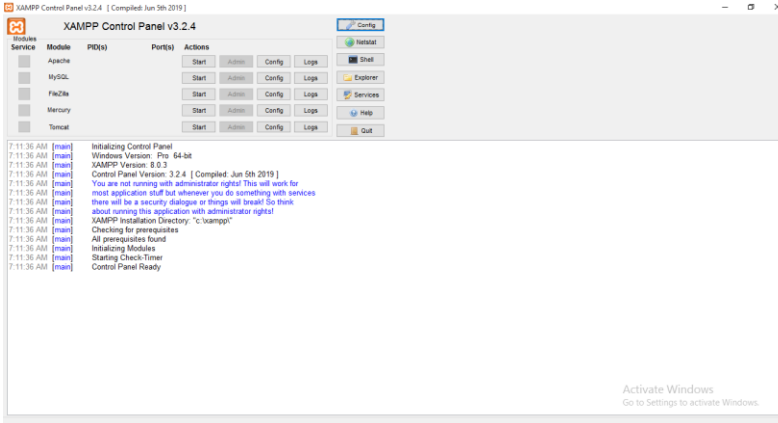
Copyright © 2021 B.K.Liyange & T.M.T.S.B.P.Tennakoon. All Rights Reserved

Content

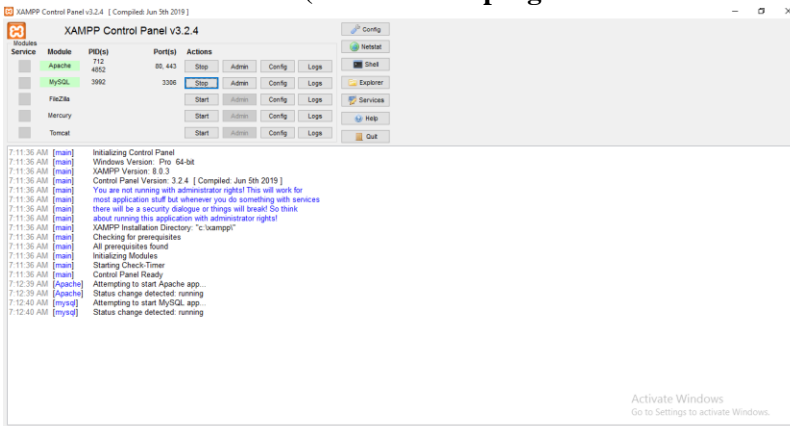
- Starting Server 19
- Open xampp – For this system we have used xampp as web server therefore first we need to run the server first. 19
- Register Function 20
- How to get client id and secret 21
- How to change main.js path and upload.js path 25
- Enabling APIs 26
- Login 27
- Upload files to google drive 28

6.1.1. Starting Server

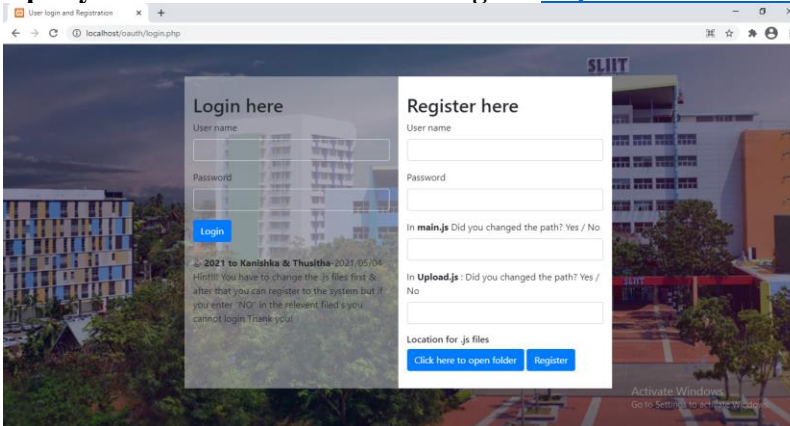
Open xampp – For this system we have used xampp as web server therefore first we need to run the server first.



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 applying 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

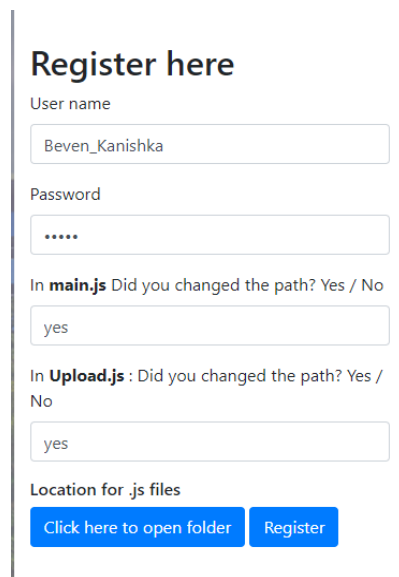
relevant information (i.e., username and password and add the token paths to code) and create account.

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.*

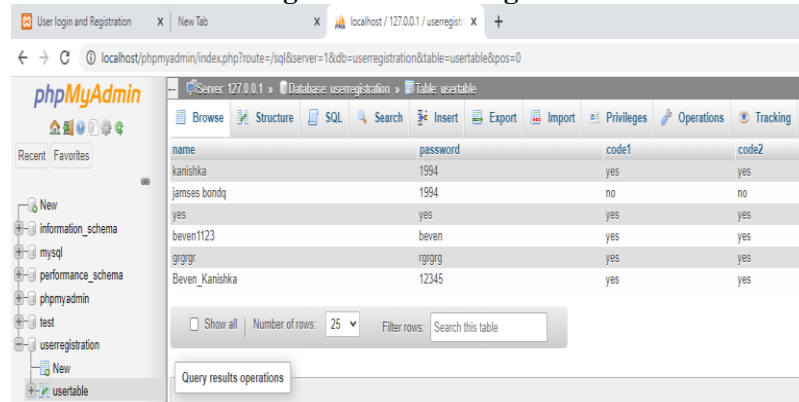
6.1.2.Register Function

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.



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

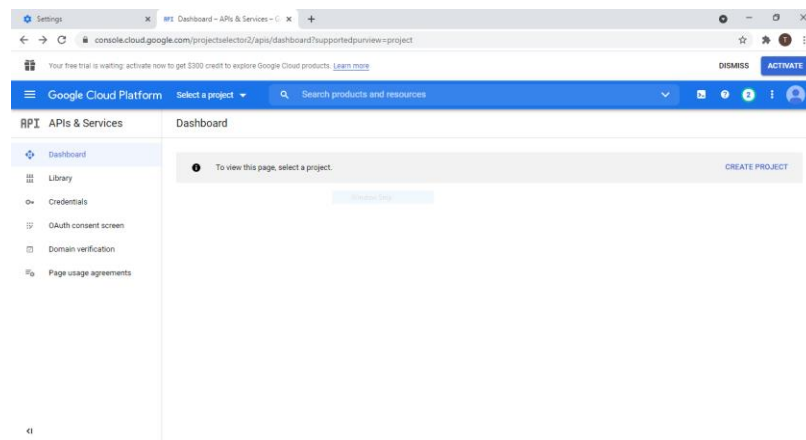


name	password	code1	code2
kanishka	1994	yes	yes
jameses bondq	1994	no	no
yes	yes	yes	yes
beven1123	beven	yes	yes
grgrgr	rgrgrg	yes	yes
Beven_Kanishka	12345	yes	yes

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.

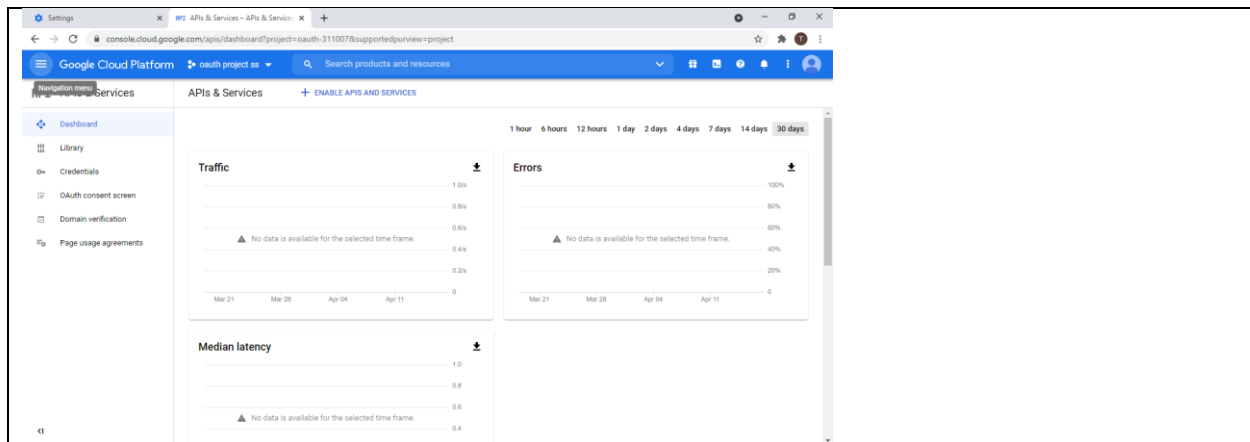
How to get client id and secret

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

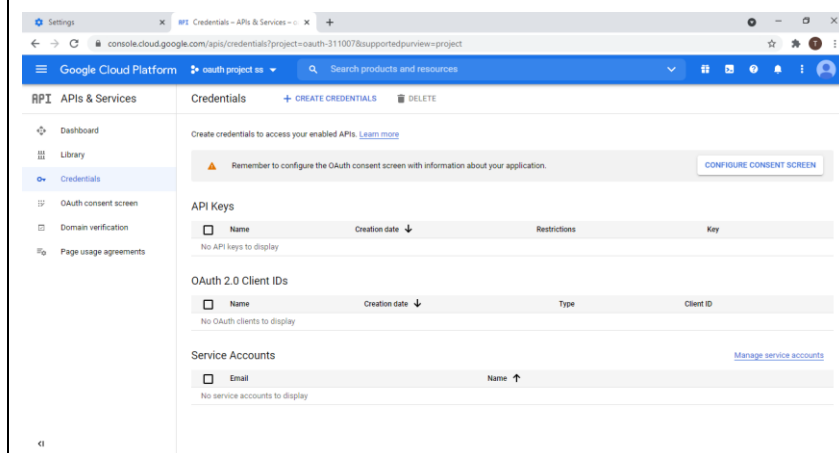


And then select create project option and give a project name and click create option and then it will take while to create project.

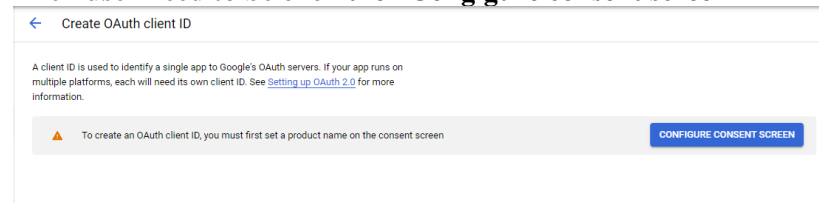
Once the creating project is done it will proceed to API & Services page.



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 “Configure 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.


OAuth consent screen

Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.

User Type

☐ Internal 

Only available to users within your organization. You will not need to submit your app for verification. [Learn more](#)

☒ External 

Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. [Learn more](#)

[CREATE](#)

[Let us know what you think](#) about our OAuth experience

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.

Edit app registration

1

 OAuth consent screen —

2

 Scopes —

3

 Test users —

4

 Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *

The name of the app asking for consent

User support email *

For users to contact you with questions about their consent

App logo

[BROWSE](#)

Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.

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).

Google Cloud Platform

oauth project ss

Search products and resources

APIs & Services

← Create OAuth client ID

Dashboard

Library

Credentials

OAuth consent screen

Domain verification

Page usage agreements

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information.

Application type *

Web application

Android

Chrome app

iOS

TVs and Limited Input devices

Desktop app

Universal Windows Platform (UWP)

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 ?

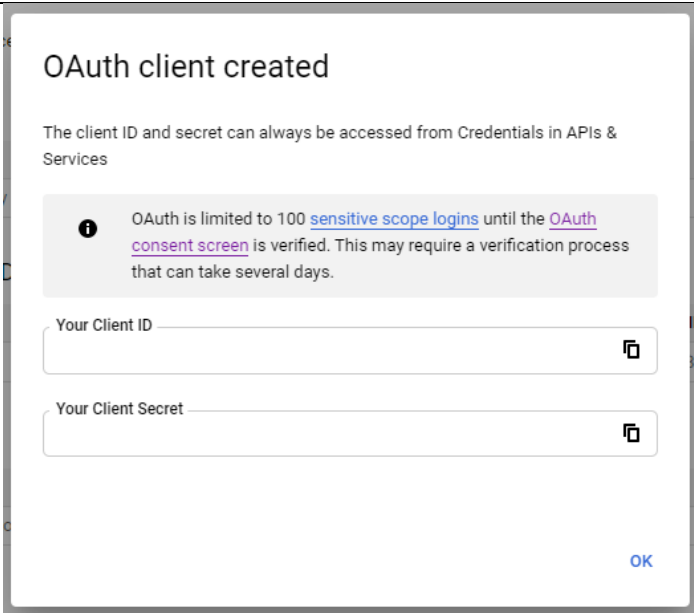
For use with requests from a web server

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.



6.1.3.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 = "Ne93mKkYt3dS2mbsvA5ZlnI"; // 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) {

```

6.1.4.Enabling APIs

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

API	APIs & Services
	Dashboard
	Library
	Credentials
	OAuth consent screen
	Domain verification
	Page usage agreements

And search google drive API and enable it.



Google Drive API

Google

The Google Drive API allows clients to access resources from Google Drive

ENABLE

TRY THIS API [↗](#)

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



Google+ API

Google

The Google+ API enables developers to build on top of the Google+ platform.

ENABLE

6.1.5.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.

Login here

User name

Beven_Kanishka

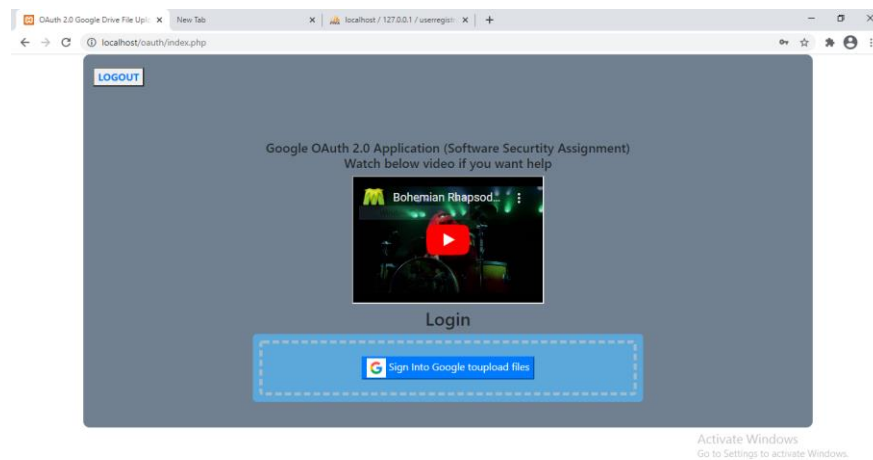
Password

Login

© 2021 to Kanishka & Thusitha-2021/05/04
Hint!!! You have to change the .js files first & after that you can register to the system but if you enter "NO" in the relevent filed's you cannot login Thank you!

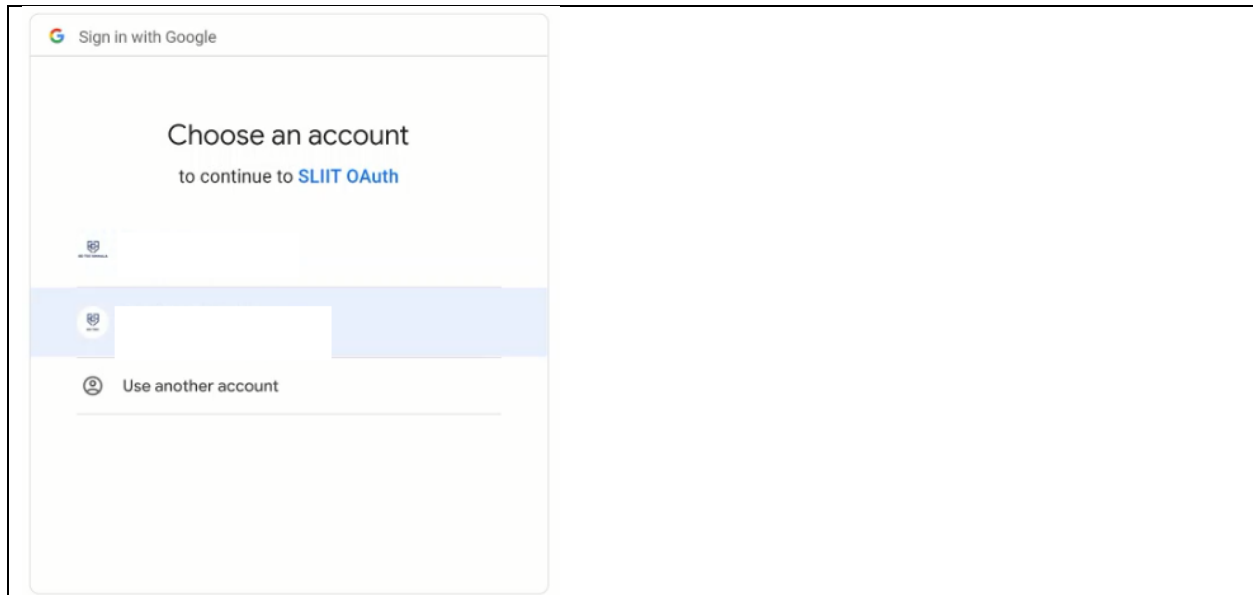
6.1.6.Upload files to google drive

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.

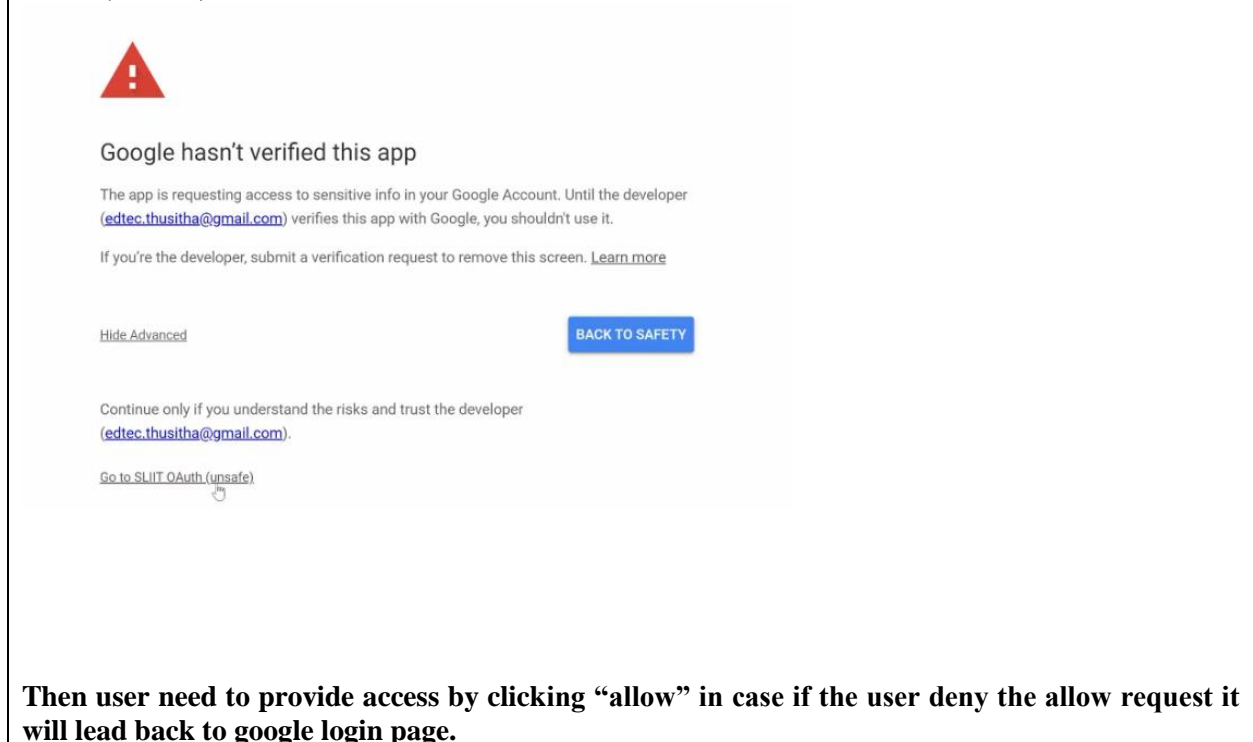


Then user can select account or add a new account by providing valid email and password. Once it is done the if the invputs are valid then it will proceed to the next page.

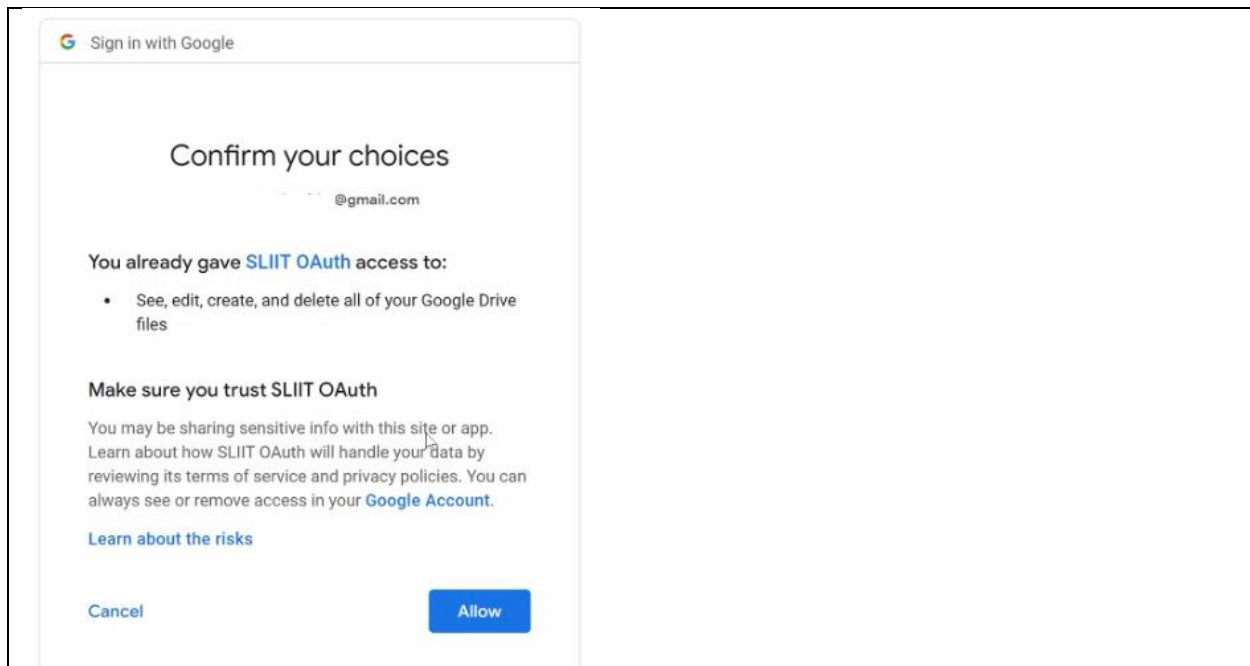
Note: Google account should be matched with client id and client secret.



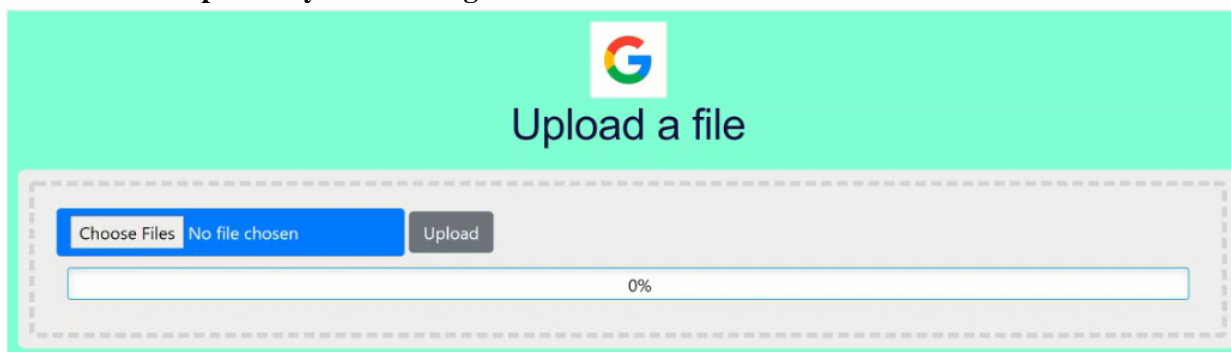
Note: In sometimes error might occur as shown in the bellow diagram. Yet there is no hart or malfunction of the system it is just a warring. In order to soleve it. Go to Advanced > go to slit oauth (unsafe)



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.

7. Appendix D

7.1.1.INDEX PAGE

<!by using this code we can enhance the security of the system because if some one tries to access the website without giving login credentials , automatically redirects to the login page >

```
<?php
session_start();

if(!isset($_SESSION['username'])){
header('location:login.php');

}
?>
```

<!This is the index page of the system user can sign in to thier google drive using their credentials & if user wants some help regrading steps , there is a youtube video embeded to this page >

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<title>OAuth 2.0 Google Drive File Upload (Software Securitty Assignment)</title>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet" type="text/css" media="screen" href="main.css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="main.js"></script>
<link
rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous"
/>
</head>
<body>
<div class="container">
```

<!This is logout button - you can logged out from the system and it redirects to the login page>

```
<button><a class="float-right" href="logout.php"><b>LOGOUT</b></a></button>
<div class="wrapper">
<div class="container">
```

<h5>Google OAuth 2.0 Application (Software Securitty Assignment) Watch below video if you want help</h5>

<!Embedded youtube video link>

```

    <iframe width="300" height="200"
src="https://www.youtube.com/embed/tgbNymZ7vqY">
</iframe>
<h3>Login</h3>
    <div class="upload-container">
        <div class="border-container">
            <button class="btn btn-primary" id="login">
                
                Sign Into Google toupload files
            </button>
        </div>
    </div>
</div>
</div>
</div>
</body>
</html>

```

7.1.2.LOGIN PAGE

```

<!login .php - after you run the program frirst time you are riderected to this page>
<!in here you can register using and name and password >
<!in order to login you must change the paths in upload.js & main.js>

<!DOCTYPE html>
<html>
<head>
    <title> User login and Registration</title>
    <link rel="stylesheet" type="text/css" href="style.css">

    <link                                rel="stylesheet"                                type="text/css"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>

    <h1>Software Security OAuth 2.0 Assignment<br></h1>

    <!this is the login form you can use your registerd login credentials here>

    <div class="Container">
        <div class="login-box">
    <div class="row">
    <div class="col-md-6 login-left">
        <h2>Login here</h2>
        <form action="validation.php" method="post">

```



```

<div class="form-group">
    <label>User name</label>
    <input type="text" name="user" class="form-control" required>
</div>
<div class="form-group">
    <label>Password</label>
    <input type="Password" name="password2" class="form-control" required>
</div>
<button type="submit" class="btn btn-primary">Login </button>
<br>
<br>
&copy; <b>2021 to Kanishka & Thusitha</b>-<?php echo date("2021/05/04");?>
<p> Hint!!!
You have to change the .js files first & after that you can register to the system but if you enter "NO" in
the relevent filed's you cannot login

Thank you!
</p>

<!--this is the Register form you can use this form to register new user to the system and if some one
already in the system it will not be processed-->
    </form>

</div>
<div class="col-md-6 login-right">
    <h2>Register here</h2>
    <form action="registration.php" method="post">

<div class="form-group">
    <label>User name</label>
    <input type="text" name="user" class="form-control" required>
</div>
<div class="form-group">
    <label>Password</label>
    <input type="Password" name="password1" class="form-control" required>
</div>

<div class="form-group">
    <label>In <b>main.js </b>Did you changed the path? Yes / No</label>
    <input type="Text" name="code1" class="form-control" required>
</div>
<div class="form-group">
    <label>In<b> Upload.js</b> : Did you changed the path? Yes / No </label>
    <input type="Text" name="code2" class="form-control" required>
</div>

<h6>Location for .js files</h6>

```

```

<a href="C:\xampp\htdocs\Oauth"><button type="submit" class="btn btn-primary">
    Click here to open folder</button>

</a>

<button type="submit" class="btn btn-primary">Register </button>


</form>
</div>

</div>
</div>

</div>

</body>
</html>

```

7.1.3.LOGOUT PAGE

```

<?php
session_start();
session_destroy();

```

```

header('location:login.php');
?>

```

MAIN .CSS PAGE

```

<! css part >
* {
    box-sizing: border-box;
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
}

body {

```

```

font-family: "Montserrat", sans-serif;
background: #ffffff;
width: 100%;
height: 100%;
}

.wrapper {
margin: auto;
max-width: 640px;
padding-top: 60px;
text-align: center;
}

.container {
background-color: #708090;
padding: 20px;
border-radius: 10px;
}

h1 {
color: #130f40;
font-family: "Varela Round", sans-serif;
letter-spacing: -0.5px;
font-weight: 700;
padding-bottom: 10px;
}

.upload-container {
background-color: rgb(93, 168, 218);
border-radius: 6px;
padding: 10px;
}

.border-container {
border: 5px dashed rgba(198, 198, 198, 0.65);
border-radius: 6px;
padding: 20px;
}

.border-container p {
color: #130f40;
font-weight: 600;
font-size: 1.1em;
letter-spacing: -1px;
margin-top: 30px;
margin-bottom: 0;
opacity: 0.65;
}

#file-browser {
text-decoration: none;

```

```

color: rgb(22, 42, 255);
border-bottom: 3px dotted rgba(22, 22, 255, 0.85);
}

#file-browser:hover {
color: rgb(0, 0, 255);
border-bottom: 3px dotted rgba(0, 0, 255, 0.85);
}

.icons {
color: #95afc0;
opacity: 0.55;
}

.new_footer_area {
background: #fbfbfd;
}

.new_footer_top {
padding: 120px 0px 270px;
position: relative;
overflow-x: hidden;
}

.new_footer_area .footer_bottom {
padding-top: 5px;
padding-bottom: 50px;
}

.footer_bottom {
font-size: 14px;
font-weight: 300;
line-height: 20px;
color: #7f88a6;
padding: 27px 0px;
}

.new_footer_top .company_widget p {
font-size: 16px;
font-weight: 300;
line-height: 28px;
color: #6a7695;
margin-bottom: 20px;
}

.new_footer_top .company_widget .f_subscribe_two .btn_get {
border-width: 1px;
margin-top: 20px;
}

.btn_get_two:hover {
background: transparent;
color: #5e2ced;
}

.btn_get:hover {
color: #fff;
}

```

```
background: #6754e2;
border-color: #6754e2;
-webkit-box-shadow: none;
box-shadow: none;
}
a:hover,
a:focus,
.btn:hover,
.btn:focus,
button:hover,
button:focus {
text-decoration: none;
outline: none;
```

7.1.4.MAIN .JS PAGE

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

    var clientId = "912398529236-aklhnqhqt0ued5dfaakmcas8g4stee3.apps.googleusercontent.com"
    //ENTER YOUR CLIENT ID FROM THE GOOGLE DEVELOPER CONSOLE

    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;

    }

})
```

```
});
```

7.1.5.REGISTRATION .PHP PAGE

```
<?php
session_start();
// data base connection user name is root and no password
header('location:login.php');
$con=mysqli_connect('localhost','root','');

mysqli_select_db($con,'userregistration');
//columns in data base
$name=$_POST['user'];
$password=$_POST['password1'];
$code1=$_POST['code1'];
$code2=$_POST['code2'];

//in this section we check if there is already a member registered in database as the name you entered if
//registered a message will pop up saying "User name already taken"
$query="select * from usertable where name ='$name'";

$result = mysqli_query($con,$query);

$num = mysqli_num_rows($result);

if ($num==1)
{

echo "User name already taken";

}

else {
//if not registered user name and password will be registered in to the database

$query="insert into usertable(name,password,code1,code2)values('$name','$password','$code1','$code2')";
mysqli_query($con,$query);

// if successful registration happens this message will pop up
echo "Registration Success full ";

}

?

```

7.1.6.STYLE .CSS PAGE

```
body{
background:linear-gradient(rgba(0,0,50,0.5),rgba(0,0,50,0.5)),url(image.jpg);
background-size:cover;
background-position: center;
}

.login-box{

    max-width: 700px;
    float: none;
    margin: 150px auto;
}

.login-left{
background: rgba(211,211,211,0.5);
padding: 30px;
}

.login-right{
background: #fff;
padding: 30px;
}

.form-control{

    background-color: transparent !important;
}

a{

    color: #fff !important;
    margin-top:-200px !important;
}

h1{

color: #fff !important;
margin-top:200px !important ;
text-align: center;
text-transform: uppercase;
}
```

7.1.7.UPLOAD.CSS PAGE

```
#progress-wrp {
  border: 1px solid #0099CC;
  padding: 1px;
  position: relative;
  height: 30px;
  border-radius: 3px;
  margin: 10px;
  text-align: left;
  background: #fff;
  box-shadow: inset 1px 3px 6px rgba(0, 0, 0, 0.12);
}

#progress-wrp .progress-bar {
  height: 100%;
  border-radius: 3px;
  background-color: #9adef3;
  width: 0;
  box-shadow: inset 1px 1px 10px rgba(0, 0, 0, 0.11);
}

#progress-wrp .status {
  top: 3px;
  left: 50%;
  position: absolute;
  display: inline-block;
  color: #000000;
}

* {
  box-sizing: border-box;
  -moz-box-sizing: border-box;
  -webkit-box-sizing: border-box;
}

h1 {
  color: #130f40;
  font-family: "Varela Round", sans-serif;
  letter-spacing: -0.5px;
  font-weight: 700;
  padding-bottom: 10px;
}

.upload-container {
  background-color: rgb(239, 239, 239);
  border-radius: 6px;
  padding: 10px;
}
```



```

}

.border-container {
border: 5px dashed rgba(202, 208, 212, 0.65);
border-radius: 6px;
padding: 20px;
}

.border-container p {
color: #130f40;
font-weight: 600;
font-size: 1.1em;
letter-spacing: -1px;
margin-top: 30px;
margin-bottom: 0;
opacity: 0.65;
}

#file-browser {
text-decoration: none;
color: rgb(202, 208, 212);
border-bottom: 3px dotted rgba(22, 22, 255, 0.85);
}

#file-browser:hover {
color: rgb(0, 0, 255);
border-bottom: 3px dotted rgba(0, 0, 255, 0.85);
}

.icons {
color: #95afc0;
opacity: 0.55;
}

```

7.1.8.UPLOAD .HTML PAGE

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<title>OAuth 2.0 Google Drive File 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>
<link
rel="stylesheet"

```

```

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous"
/>
</head>
<body>
  <div class="container">

    <a class="float-right" href="logout.php">LOGOUT</a>

    <div class="wrapper" style="padding: 10px;">
      <div class="container" style="background-color: aquamarine;padding: 10px;">
        <center></center>
        <h1><center>Upload a file</center></h1>

        <div class="upload-container">
          <div class="border-container">

            <input
              class="btn btn-primary"
              id="files"
              type="file"
              name="files[]"
              multiple
            />
            <button class="btn btn-secondary" id="upload">Upload</button>
            <p>Hint !!! In here you can select file and upload it to the google drive. . . . </p>

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

```

7.1.9.UPLOAD.JS PAGE

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

    const urlParams = new URLSearchParams(window.location.search);
    const code = urlParams.get('code');
    const redirect_uri = "http://localhost:8080/OAuthproject/upload.html" // replace with your
    redirect_uri;
    const client_secret = "Bws3KUipe0y6LZQbFhPEqfAB"; // replace with your client secret
    const scope = "https://www.googleapis.com/auth/drive";
    var access_token= "";
    var client_id = "912398529236-aklhnhqotm0ued5dfaakmcas8g4stee3.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) {

            localStorage.setItem("accessToken",resultData.access_token);
            localStorage.setItem("refreshToken",resultData.refreshToken);
            localStorage.setItem("expires_in",resultData.expires_in);
            window.history.pushState({}, document.title, "/GoogleLoginApp/" + "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);
    }

    $(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);

    upload.doUpload();
});

});

```

7.1.10. VALIDATION PAGE

```

<?php
session_start();

// validation page database connection
$con= mysqli_connect('localhost','root','');

mysqli_select_db($con,'userregistration');

$name=$_POST['user'];
$pass=$_POST['password2'];
$code1=$_POST['code1'];
$code2=$_POST['code2'];

// only allows user to login if name and password registered in database and yes yes as the answers for the
registration form questions

```

```
$s="select * from usertable where name ='$name' && password ='$pass'&& code1='yes'&& code2='yes'";

$result = mysqli_query($con,$s);

$num = mysqli_num_rows($result);

if ($num == 1){

    $_SESSION['username']= $name;

    header('location:index.php');
    echo "welcome";
    // if not the page will be redirected to login page
}else {
    header('location:login.php');
}

?>
```