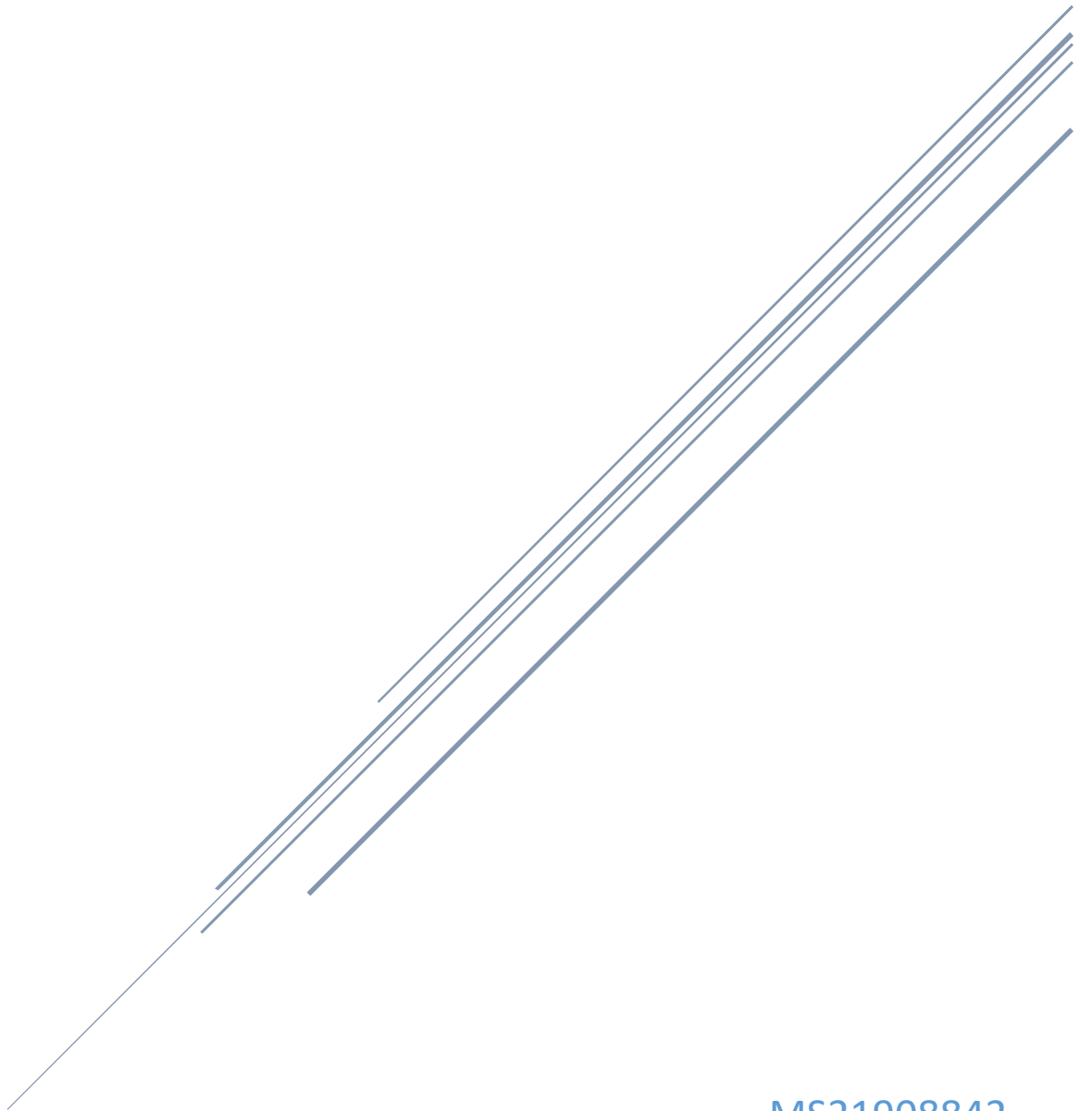


SOFTWARE SECURITY

Assignment 02



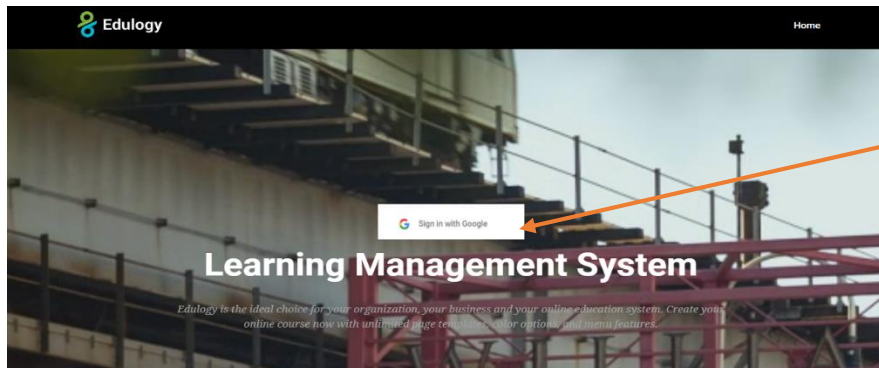
MS21908842
G.A.P Madubashana

1. Introduction

This report describes about the public oauth server details and other functionalities of the Learning Management System which is implemented to demonstrate the oauth authorization server. This, system is use google oauth server to validate the users and provide access to the system to perform their tasks. The system allows users to upload assignment to the google drive, view course details and register for courses after successfully login to the system using google authorizations.

2. System Structure

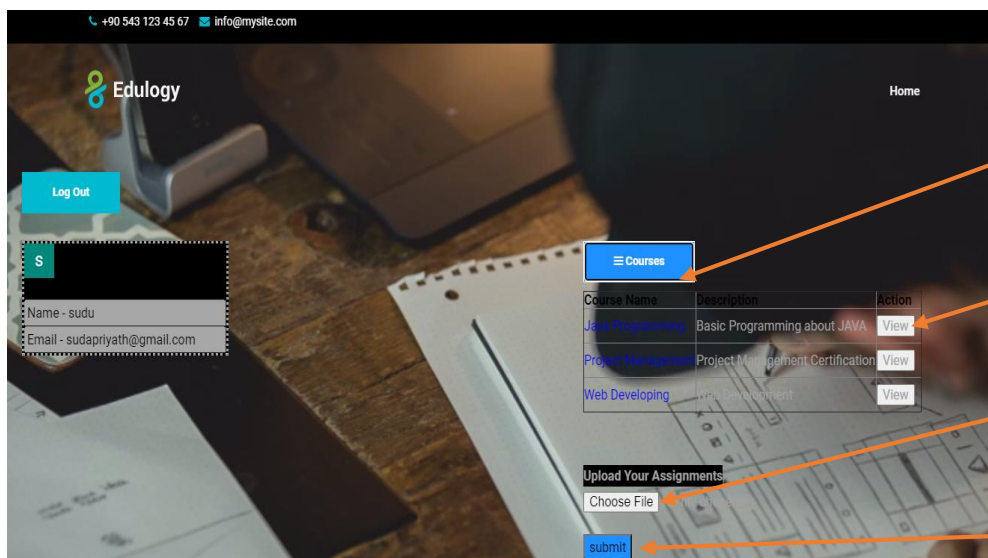
Home page



Click on 'Sign in' button to login to the web application

Figure 1: Home Page

Registered users' view



Click to view courses

Click to view course details

Click to choose a file

Click to submit the file to the google drive of the institute

Figure 2: Registered User's View

Unregistered users' view

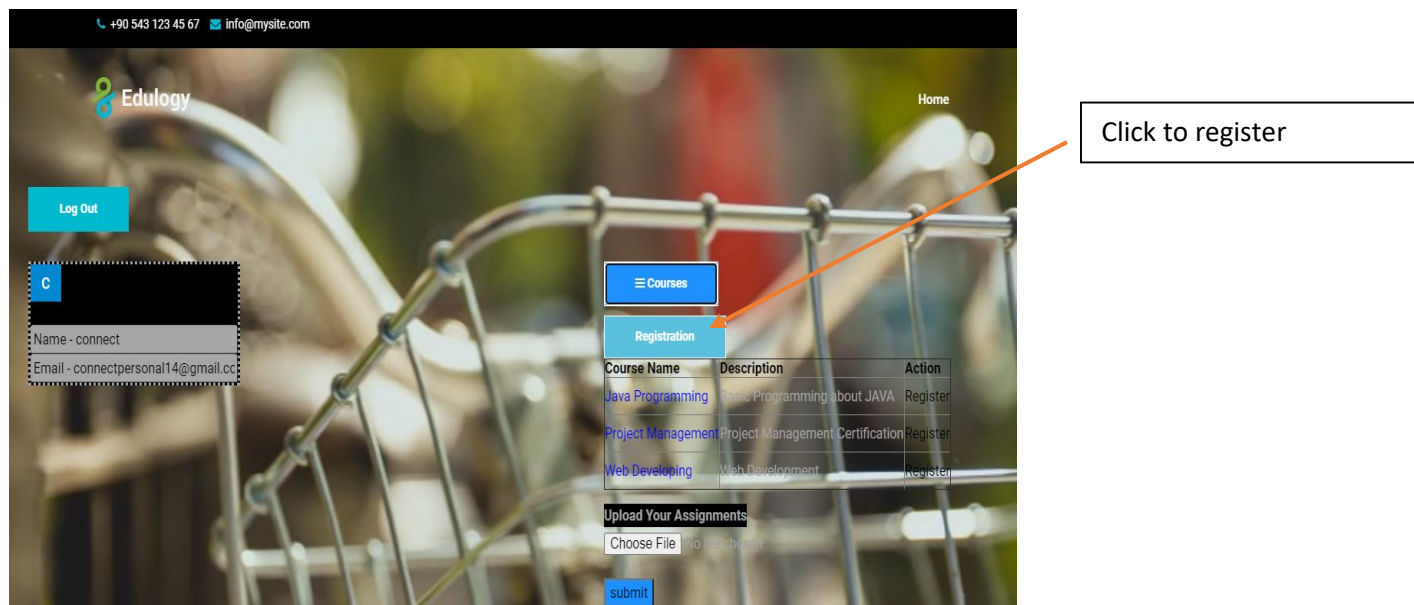


Figure 3: Unregistered User's View

3. Integrating Google Sign-in

This Learning Management System use authorization credentials to provide google sign-in for user. Below are the steps that followed to create the google authorization credentials.

- Create a Project in the Credentials page.

Create a project using the credentials page and select that project in order to create credentials.

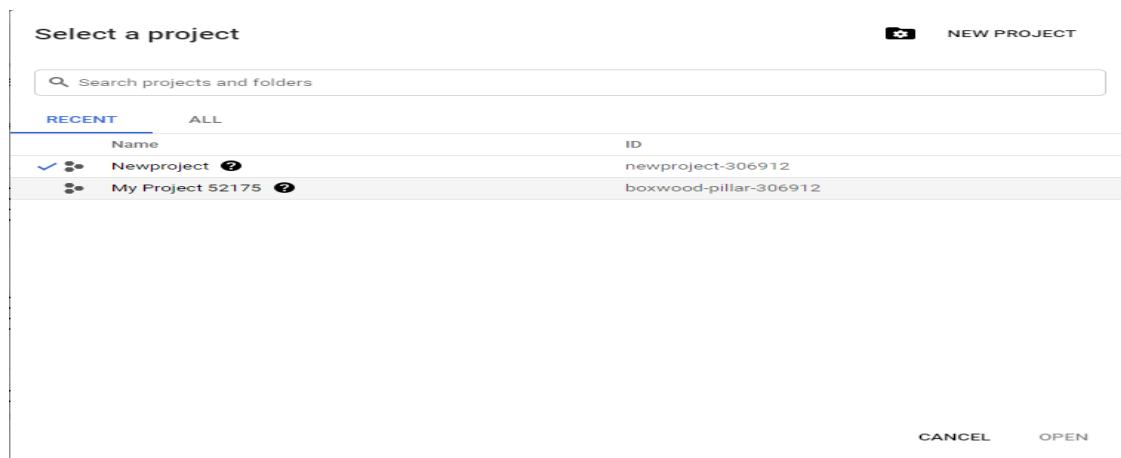


Figure 4: Creating a Project

- b. Click on Create Credentials

Click on create credentials and after that it will prompt a message box. Then click on 'OAuth client ID'.

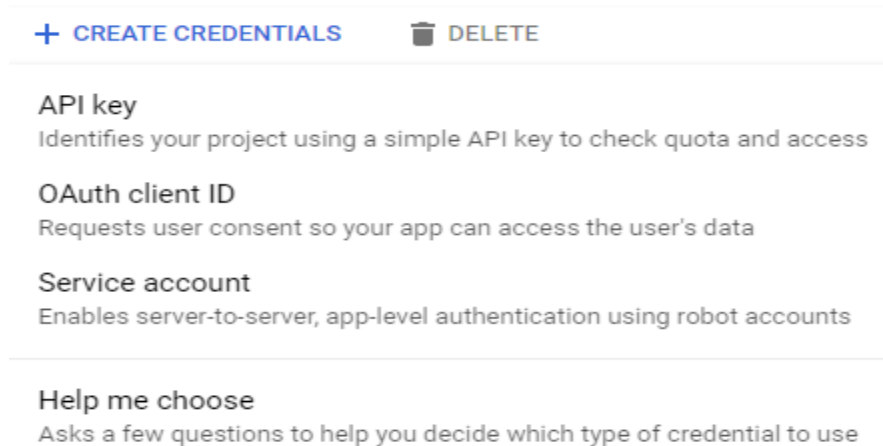


Figure 5: Creating Credentials - 1

- c. Select application type and adding necessary information.

After come in to the 'create credentials page', select the type of application that need to create the credentials and provide a name for that OAuth 2.0 client.

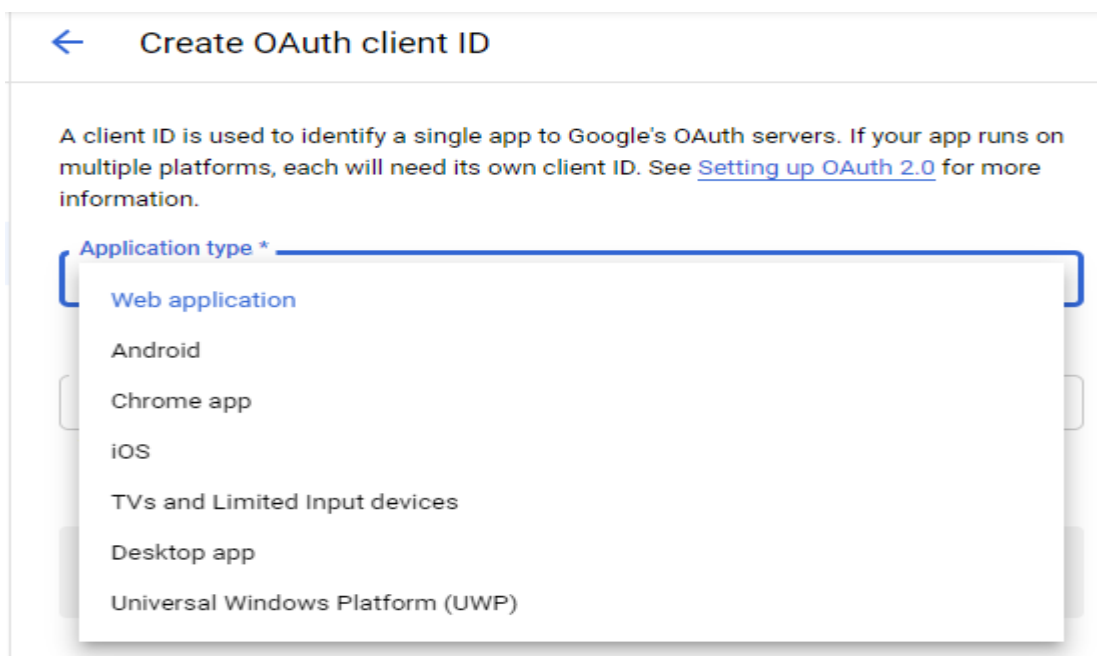


Figure 6: Creating Credentials - 2

Fill out the redirect URIs or JavaScript origins if any and click on create.

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins ?

For use with requests from a browser

+ ADD URI

Authorized redirect URIs ?

For use with requests from a web server

+ ADD URI

CREATE

CANCEL

Figure 7: Creating Credentials - 3

After creating the credentials, those details are appearing in the OAuth 2.0 client IDs section.






OAuth 2.0 Client IDs					
<input type="checkbox"/>	Name	Creation date ↓	Type	Client ID	
<input type="checkbox"/>	Web client 1	Mar 7, 2021	Web application	281893744936-mhu9...	   


Figure 8: OAuth Client IDs


- d. Download the JSON file and add it to the application folder.

←

Client ID for Web application

 DOWNLOAD JSON

 RESET SECRET

 DELETE

Name *

Web client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

Client ID

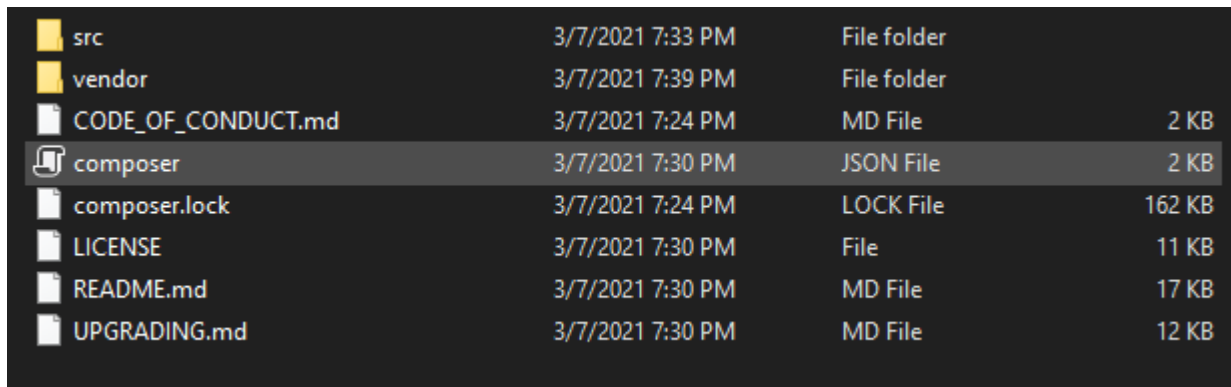
281893744936-mhu99ofaqnmc4t67ca4spe9p0ec3eh3.apps.googleusercontent.com

Client secret

pEBzVwuHUyvDOYyou51uhBAdf

Figure 9: Download JSON File

- e. Get the API files downloaded and add those to the application folder.



src	3/7/2021 7:33 PM	File folder	
vendor	3/7/2021 7:39 PM	File folder	
CODE_OF_CONDUCT.md	3/7/2021 7:24 PM	MD File	2 KB
composer	3/7/2021 7:30 PM	JSON File	2 KB
composer.lock	3/7/2021 7:24 PM	LOCK File	162 KB
LICENSE	3/7/2021 7:30 PM	File	11 KB
README.md	3/7/2021 7:30 PM	MD File	17 KB
UPGRADING.md	3/7/2021 7:30 PM	MD File	12 KB

Figure 10: API Files

- f. After do the above steps successfully, then backend of the application need to be developed. So, the codes that used to implement the google sign-in is included in the appendix section.

4. Google OAuth 2.0 Process

Mainly, this authorization process needed some key components such as client id, client secret and access token.

- Client ID – Identifier for the particular application and a unique string which is representing the registration details provided by the client.
- Client Secret – This client password uses to authenticate with the google server.
- Access Token – The token which is issued to the client after the authorization and the life time of this token to limited to a particular time period.

When a user tries to login to the Learning Management System, following are the steps that occurring till the authorization.

- a. Create authorization request and set parameters.

In this step authorization request will be created.

- b. In order to authenticate, particular user redirect to OAuth 2.0 server.

- c. At this step, Google prompts a consent window to the user and it request access.

- d. Handle the OAuth 2.0 server response

At this stage, if the user accepts the particular request, authorization code will be received and if not an error message will be received as the response.

- e. After receive the authorization code to the web server, it can exchange the authorization code for an access token.

Following diagram depicts the requests send to OAuth and google servers, and responses received to authorize the website to access the relevant data.

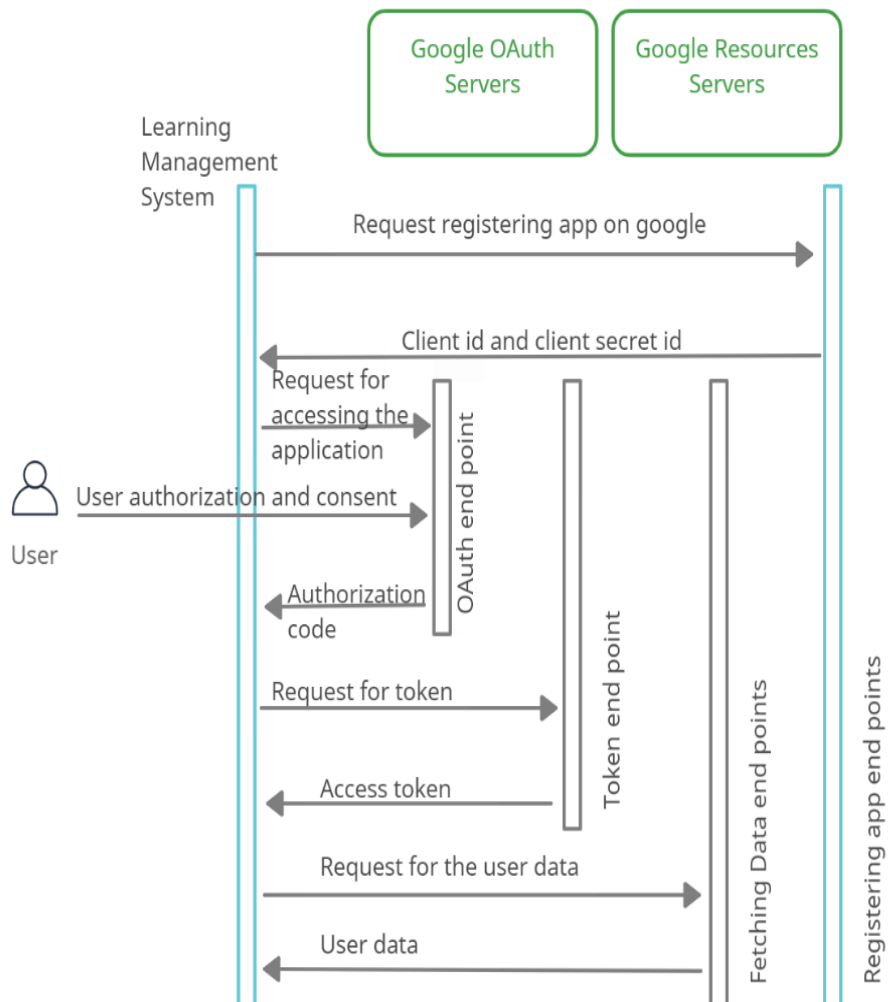


Figure 11: OAuth and google server Requests

5. Integrating Google Drive API to Upload Files

a) Enable Google Drive API

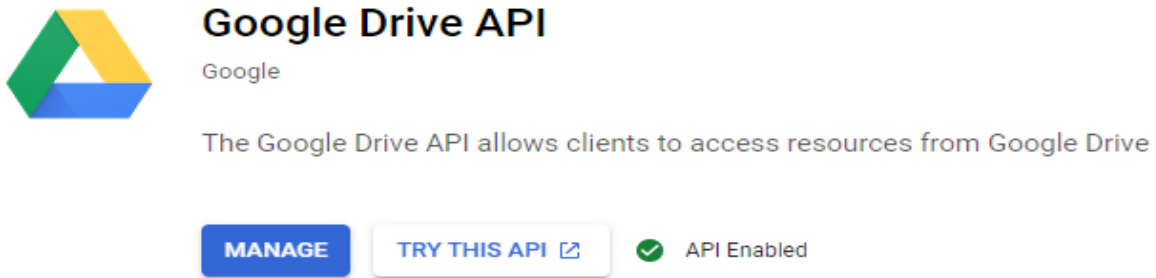


Figure 12: Enable Google Drive API

b) Configure Google client library in the project

Install the library using following command to the project folder

```
composer require google/apiclient:^2.0
```

6. Other Google Resources

- Google map widget was integrated with this web application to indicate the location. That was integrated using a google API key.

API Keys




<input type="checkbox"/>	Name	Creation date ↓	Restrictions	Key		
<input type="checkbox"/>	▲ API key 1	Mar 10, 2021	None	AIzaSyAVFI...EKMqq85B-g		 

Figure 13: Google API Keys

Appendix

Main PHP codes use to perform the functionalities

```
<script>
    function myFunction()
    {
        var a = document.getElementById("myDIV");
        if (a.style.display === "none") {
            a.style.display = "block";
        } else {
            a.style.display = "none";
        }
    }

    function reply_click(clicked_id)
    {
        if(clicked_id == '1'){

            document.getElementById("courseid").value = clicked_id;
            document.getElementById("Price").value = "RS.30000.00";
            document.getElementById("date").value = "Monday - Friday / 3 pm - 5 pm";
        } else if(clicked_id == '2'){
            document.getElementById("courseid").value = clicked_id;
            document.getElementById("Price").value = "RS.36000.00";
            document.getElementById("date").value = "Monday - Wednesday / 3 pm - 5 pm";
        };
        } else if(clicked_id == '3'){
            document.getElementById("courseid").value = clicked_id;
            document.getElementById("Price").value = "RS.40000.00";
            document.getElementById("date").value = " Saturday - Sunday / 3 pm - 5 pm";
        };
    }
}
</script>

<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);

$google_redirect_url = 'http://localhost:10080/auth/redirect.php';
```

```
session_start();

// google api files
include_once 'lib/vendor/autoload.php';

// New google client
$gClient = new Google_Client();
$gClient->setApplicationName('ApplicationName');
$gClient->setAuthConfigFile('client_secret_281893744936-
mhu99ofaqcnmc4t67ca4spe9p0ec3eh3.apps.googleusercontent.com.json');
$gClient->addScope(Google_Service_Oauth2::USERINFO_PROFILE);
$gClient->addScope(Google_Service_Oauth2::USERINFO_EMAIL);

// New Google Service
$google_oauthV2 = new Google_Service_Oauth2($gClient);

// LOGOUT?
if (isset($_REQUEST['logout']))
{
    unset($_SESSION["auto"]);
    unset($_SESSION['token']);
    $gClient->revokeToken();
    header('Location: ' . filter_var($google_redirect_url, FILTER_SANITIZE_URL));
}

// GOOGLE CALLBACK?
if (isset($_GET['code']))
{
    $gClient->authenticate($_GET['code']);
    $_SESSION['token'] = $gClient->getAccessToken();
    header('Location: ' . filter_var($google_redirect_url, FILTER_SANITIZE_URL));
    return;
}

// PAGE RELOAD?
if (isset($_SESSION['token']))
{
    $gClient->setAccessToken($_SESSION['token']);
}

// Autologin?
if(isset($_GET["auto"]))
{
    $_SESSION['auto'] = $_GET["auto"];
}
```

```

}

// LOGGED IN?
if ($gClient->getAccessToken())
{

    try {
        $user = $google_oauthV2->userinfo->get();

        $_SESSION['token']    = $gClient->getAccessToken();
        $arr = json_decode(json_encode($_SESSION['token']));
        $to = $arr->access_token;
        file_put_contents('token.json',  '{"access_token": "'.$to.'"', "expires_in": 3599,"refresh_token":"1\\\/\\\/0gjBBJrwz540vCgYIARAAGBASNwF-L9Ir6NhuzuAKDcvGNK28IuaSpHkLzgHXnb8lwunZMatfUzBxtNaIfXDvfC7z4WkgjD_3QaU","scope":"https:\\\/\\\/www.googleapis.com\\\/auth\\\/drive","token_type":"Bearer","created":1618817979}' );

        //(View course details / Course registration / Assignment Submission)
        echo ' <section id="home" class="video-section js-height-full">
        <br /><br /><br /><br /><br /><a style = "position:absolute; left:30px; top:200px;" href="?logout=1" class="btn btn-primary wow slideInLeft">Log Out</a><br />
        <div style = "position:absolute; left:30px; top:280px;border-style: dotted;background-color: black;">
        <a href="'. $profile_url.'" style = "" target="_blank"></a><br /><br />
        <input type="text" value="Name - ' . $user_name.'" disabled style= "width:270px; color: black; " /><br />
        <input type="text" value="Email - ' . $email.'" disabled style= "width:270px; color: black; "/>
        </div>

        <div style = "position:absolute; left:780px; top:540px;">
            <form action="submit.php" method="post" enctype="multipart/form-data" >
                <label for="" style = "background-color: Black;">Upload Your Assignments</label>
                <input type="file" name="file" >
                <br>

```

```

        <input style = "background-
color: DodgerBlue;color: Black;" type="submit" name="submit" value="submit" >
    </form>
</div>

    <div style = "position:absolute; left:780px; top:280px;">
        <button onclick="myFunction()" class="btn" style = "background-
color: DodgerBlue;"><i class="fa fa-bars"></i> Courses</button>
        <p> </p>
        <div id="myDIV" style="display:none">
            <table class="data" cellpadding="8" border="1">
                <thead>
                    <tr>
                        <th class="first" style="width:120px;color: Black
; "> Course Name </th>
                        <th class="" style = "color: Black;"> Description
                    </th>
                        <th class="" style = "color: Black;">Action </th>
                    </tr>
                </thead>
                <tbody>';

//Check whether user is registered or not
$db = mysqli_connect('localhost', 'root', '', 'test');
$sql = "SELECT * FROM registrations WHERE email = '$email'";
$result = mysqli_query($db, $sql);
$rows = mysqli_num_rows($result);

if ($rows > 0) {

} else {

}

echo '

    <div class="modal fade" id="myModal" role="dialog">
        <div class="modal-dialog">

```

[illegible]

[illegible]

```

        //Save user details in the database
        $name = "";
        $email = "";
        $id = 0;
        $update = false;

        if (isset($_POST['save'])) {
            $name = $_POST['name'];
            $email = $_POST['email'];

            mysqli_query($db, "INSERT INTO registrations (email, name) VALUES ('$email', '$name')");
            header('location: redirect.php');
        }

        } catch (Exception $e) {
            // The user revoke the permission for this App. Therefore reset session token

            unset($_SESSION["auto"]);
            unset($_SESSION['token']);
            header('Location: ' . filter_var($google_redirect_url, FILTER_SANITIZE_URL));
        }
    }
    else // Sign up
    {

        $authUrl = $gClient->createAuthUrl();

        // Fast access or manual login button?
        if(isset($_GET["auto"]))
        {

        }
        else
        {
            echo ' <section id="home" class="video-section js-height-full">

                <!-- <div class="overlay"></div> -->
                <div class="home-text-wrapper relative container">
                    <div class="home-message">

```