OAuth 2.0 Authorization Framework

Software Security – Assignment 2

Ishadi Malshani Gulawita

Faculty of Computing, Sri Lanka Institute of Information Technology,

New Kandy Rd, Malabe 10115, Sri Lanka

[ishadimg@outlook.com](mailto:ishadimg@outlook.com), ms20907266@my.sliit.lk

**Introduction**

OAuth is an authorization method to provide access to resources over the HTTP protocol. It can be used for authorization of various applications as well as for the manual user access. The general way it works is allowing an application to have an access token (which represents a user’s permission for the client to access their data) which it can use to authenticate a request to an API endpoint. There are two versions of OAuth authorization OAuth 1 (using HMAC-SHA signature strings) and OAuth 2 (using tokens over HTTPS).

In any OAuth 2.0 flow we can find the following roles:

1. **The Third-Party Application: "Client"**

The client is the application that is trying to get access to the user's account. It needs to get permission from the user before it can do so.

1. **The API: "Resource Server"**

The resource server is the API server used to access the user's data.

1. **The Authorization Server**

This is the server that presents the interface where the user approves or denies the request. In smaller implementations, this may be the same server as the API server, but larger scale deployments will often build this as a separate component.

1. **The User: "Resource Owner"**

The resource owner is the person who is giving access to some portion of their account.

The [OAuth 2.0 Authorization Framework specification](https://tools.ietf.org/html/rfc6749) defines four flows to get an Access Token. These flows are called grant types. Deciding which one is suited for the case depends mostly on the type of the application. The first one is [Authorization Code](https://auth0.com/docs/flows/concepts/auth-code). It is used by Web Apps executing on a server. This is also used by mobile apps, using the [Proof Key for Code Exchange (PKCE) technique](https://auth0.com/docs/flows/concepts/auth-code-pkce). The second on is [Implicit](https://auth0.com/docs/flows/concepts/implicit). It is used by JavaScript-centric apps (Single-Page Applications) executing on the user's browser. [Resource Owner Password Credentials](https://auth0.com/docs/api-auth/grant/password) is the third one and it is used by trusted apps. Finally, [Client Credentials](https://auth0.com/docs/flows/concepts/client-credentials): used for machine-to-machine communication. Also, when considering OAuth 2.0 framework, it is very important to understand **Authorization Code Flow**.





**User**

**Resource Owner**

**Authorization Server**

**Resource Server**

**Web Application/**

**Client Application**

1. Show me the Web Site Details
2. You have to be authenticated
3. Authenticate me with Google





**ID 111**

**Secret aaa**

1. http direct to Google login page for ID 111

1. Authentication for ID 111



1. “Allow access to your profile?”

**ID 111 Secret aaa**

**ID 222 Secret bbb**

**ID 333 Secret ccc**

1. “Yes”

8. http directs with Google Oauth code





9. Google Oauth code

10. Google Oauth code



11. Access Token

12. Give me profile details

13. User profile Details



14. You are authenticated

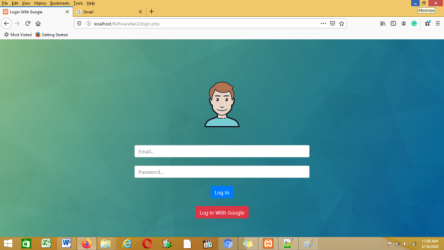
First the client application requests authorization to access service resources from the user. If the user has not already authenticated with the authorization server, the user will be required to login and consent at the server. After successful authentication, user has to authorize the client or user at the authorization server. The server responds to the request with an authorization code.Then the client can exchange the authorization code with an access token by using client secret. In return the authorization server will issue an access token and a refresh token in addition to that,with the acquired access token, the client can send requests to the resource server and access resource server API and user data presenting the access token as the means of authentication.

**Selected Scenario**

**Use OAuth 2.0 protocol to access Google API and authenticate user to display some of user details in a website**.

In this snenario, A web application is created for get some details(Eg : Name, First name, Last Name Email and etc) of a user in to a second web page .So there is first page login page(login.php) and login button. It asks user email password. I want that file is only accessible for the people who are logged in.So now at the moment people can login to this page only if they have valid account data.But I also want to enable those people to sign in with the Google account.So what I’m going to do is I will just add one button here “Log in with google” and once someone press the button we will redirect the person to the Google account and accuire account details. And now on the Google they will have to accept the permission that we are asking for and once they accept google will redirect them to the one page that I’m going to call Google callback page and now the key thing will happen inside this page we are going to check few details from the Google and if the user has given all the permissions and all data we are looking for we will redirect him to the Second page(index.php) and if he is not we will just redirect to the login page.

1. Need some user details from
2. You have to be authenticated
3. Authenticate me with Google



1. http direct to Google login page for ID

**My application**

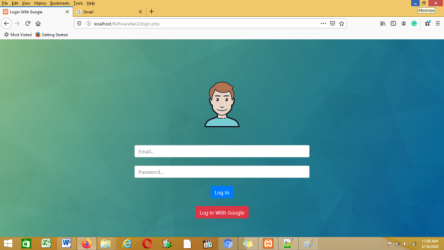
ID : 337920838897-hm70l5optfd0bsfqe8ocv8sd34sfu1bs.apps.googleusercontent.com

1. Authentication for ID



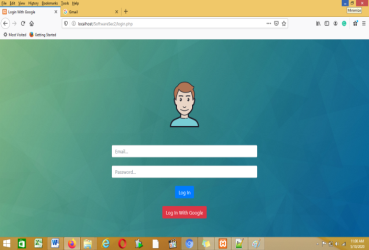
1. “Allow access to your profile?”
2. “Yes”

8. http directs with Google Oauth code



9. Google Oauth code

10. Google Oauth code

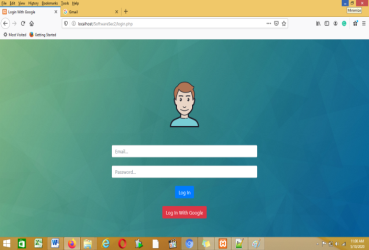


11. Access Token

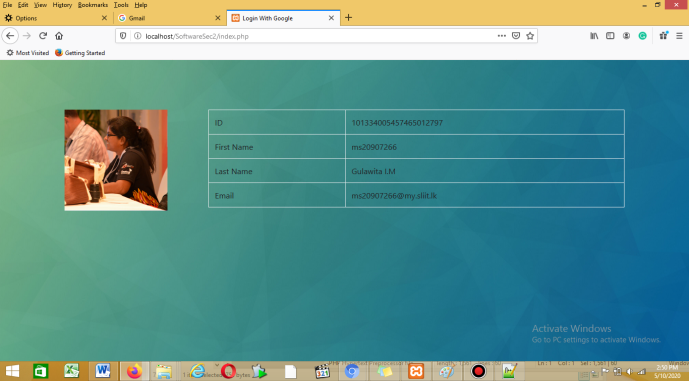
12. Give me profile details

13. User profile Details





14. You are authenticated



Finally, if the user has given all the permissions and all data we are looking for, we will redirect him to the Second page(index.php)

Here I’m describing all the steps I followed for this activity. The process of Creating the Client ID and Client Secret Code is attached as Appendix inthis report.

* Desing My login page 9 ( login.php)
* Downloaded Google API Library
* Creating “config.php”
* Get Client ID and Client Secret Code (Appendix)
* Get permission for Google Account
* Create google php file and get code. (g-callback.php)
* Generate Google OAuth 2.0 Code
* Generate Access Token
* Display Google Account data inside my websit

1. Design my login page

|  |  |
| --- | --- |
| <http://localhost/SoftwareSec2/login.php>  C:\Users\gulawita\Desktop\Software Security\Screenshots\1.png  C:\Users\gulawita\Desktop\Software Security\Screenshots\2.png  Here my main objective is implementing the funtion of “Login with Google” Buttun. | <!doctype html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"  content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  <meta http-equiv="X-UA-Compatible" content="ie=edge">  <title>Login With Google</title>  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous">  </head>  <style>  body {  background-image: url('images/logo1.jpg');  background-repeat: no-repeat;  background-attachment: fixed;  background-size: cover;  }  </style>  <body>  <div class="container" style="margin-top: 100px">  <div class="row justify-content-center">  <div class="col-md-6 col-offset-3" align="center">  <img src="images/logo2.png"><br><br>  <form >  <input placeholder="Email..." name="email" class="form-control"><br>  <input type="password" placeholder="Password..." name="password" class="form-control"><br>  <input type="submit" value="Log In" class="btn btn-primary"><br><br>  <input type="button" onclick="window.location = '<?php echo $loginURL ?>';" value="Log In With Google" class="btn btn-danger">  </form>  </div>  </div>  </div>  </body>  </html> |

2. Then downloaded Google API Library (https://developers.google.com/api-client-library/). Named the that File as “Google API” and added it in my local server folder.

3. Creating “config.php”

|  |  |
| --- | --- |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\3.png | <?php  session\_start();  require\_once "GoogleAPI/vendor/autoload.php";  $gClient = new Google\_Client();  $gClient->setClientId("");  $gClient->setClientSecret("");  ?> |

4. Get ClientId and ClientSecret from Google API

- Added them in config.php and complete php file.

|  |  |
| --- | --- |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\13.png | <?php  session\_start();  require\_once "GoogleAPI/vendor/autoload.php";  $gClient = new Google\_Client();  $gClient->setClientId("337920838897-hm70l5optfd0bsfqe8ocv8sd34sfu1bs.apps.googleusercontent.com");  $gClient->setClientSecret("q0FkbMrpKUKrWtSrKT\_tzOCn");  $gClient->setApplicationName("SS Assignment OAUTH");  $gClient->setRedirectUri("http://localhost/SoftwareSec2/g-callback.php");  $gClient->addScope("https://www.googleapis.com/auth/plus.login https://www.googleapis.com/auth/userinfo.email");  ?> |

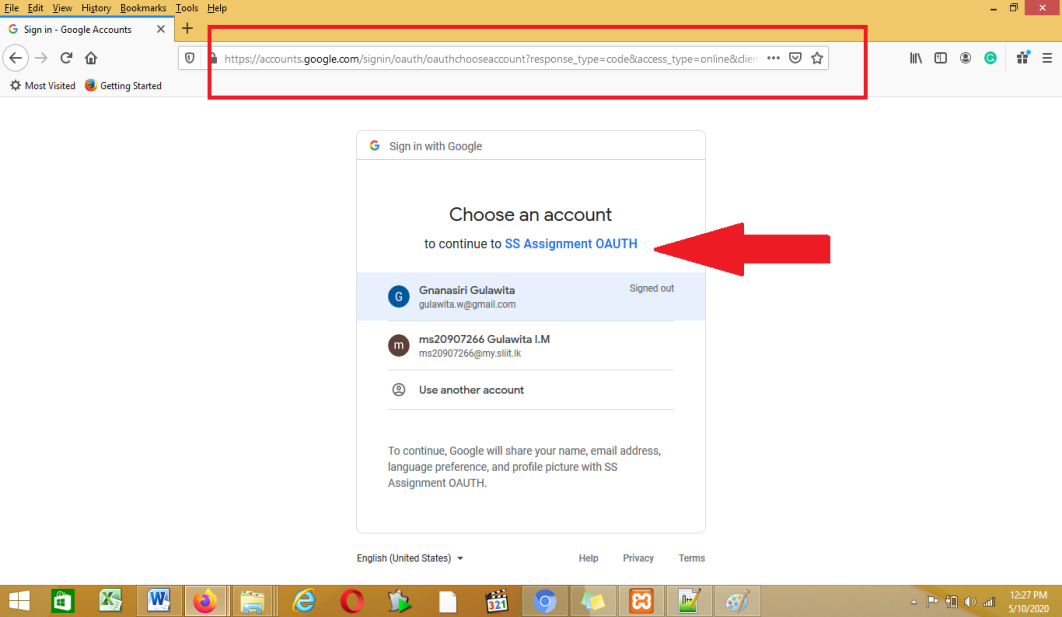
* Again update login.php (Changes are marked in differnet color)

|  |
| --- |
| <?php  require\_once "config.php";  $loginURL = $gClient->createAuthUrl();  ?>  <!doctype html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"  content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  <meta http-equiv="X-UA-Compatible" content="ie=edge">  <title>Login With Google</title>  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous">  </head>  <style>  body {  background-image: url('images/logo1.jpg');  background-repeat: no-repeat;  background-attachment: fixed;  background-size: cover;  }  </style>  <body>  <div class="container" style="margin-top: 100px">  <div class="row justify-content-center">  <div class="col-md-6 col-offset-3" align="center">  <img src="images/logo2.png"><br><br>  <form >  <input placeholder="Email..." name="email" class="form-control"><br>  <input type="password" placeholder="Password..." name="password" class="form-control"><br>  <input type="submit" value="Log In" class="btn btn-primary"><br><br>  <input type="button" onclick="window.location = '<?php echo $loginURL ?>';" value="Log In With Google" class="btn btn-danger">  </form>  </div>  </div>  </div>  </body>  </html> |

5. Get permission for Google Account

Go to the <http://localhost/SoftwareSec2/login.php> and clicled Log in with Google

- Generate Access Token



<https://accounts.google.com/signin/oauth/oauthchooseaccount?response_type=code&access_type=online&client_id=337920838897-hm70l5optfd0bsfqe8ocv8sd34sfu1bs.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%2FSoftwareSec2%2Fg-callback.php&state&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fplus.login%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email&approval_prompt=auto&o2v=1&as=6Qxsis7hyQ_mX0A4wGbDGg&flowName=GeneralOAuthFlow>

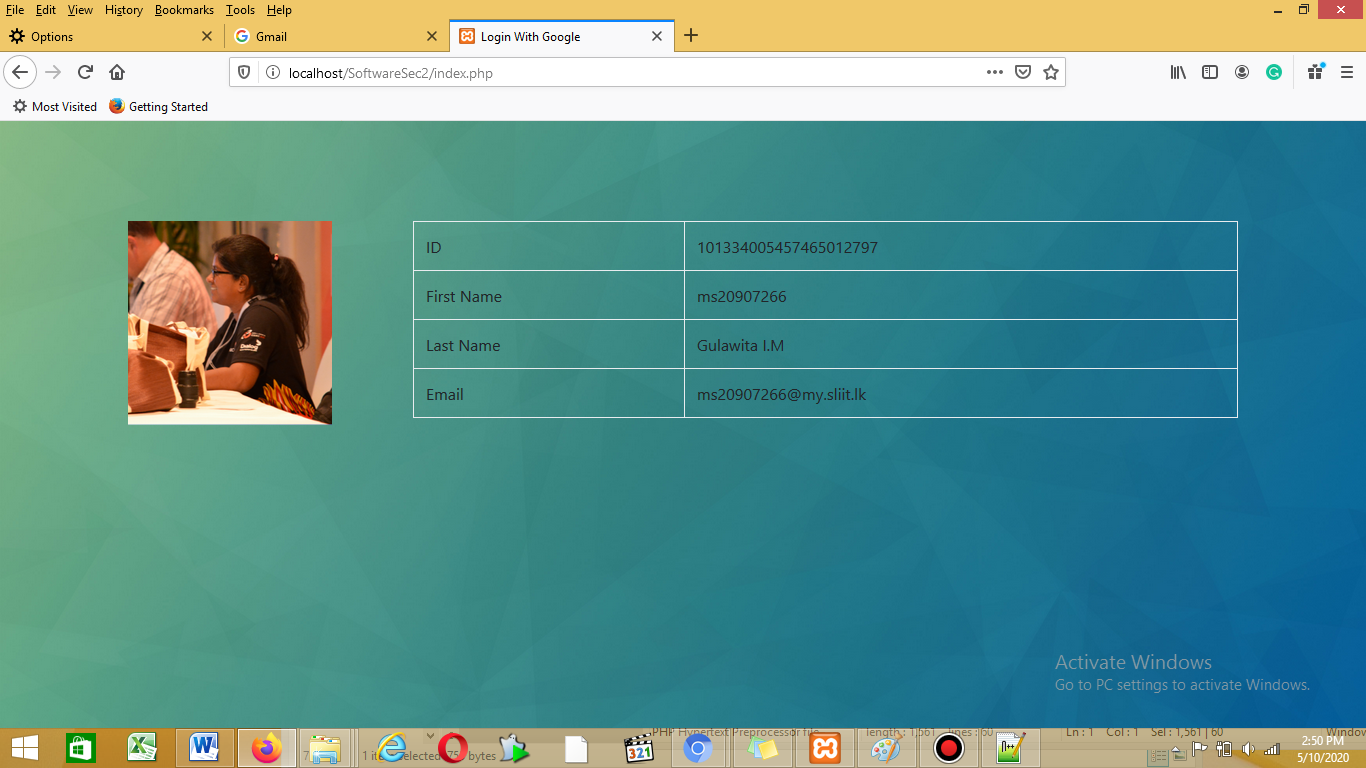
6. Create google php file and get code. (g-callback.php)

|  |  |
| --- | --- |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\16.png | <?php  require\_once "config.php";  if (isset($\_SESSION['access\_token']))  $gClient->setAccessToken($\_SESSION['access\_token']);  else if (isset($\_GET['code'])) {  $token = $gClient->fetchAccessTokenWithAuthCode($\_GET['code']);  $\_SESSION['access\_token'] = $token;  } else {  header('Location: login.php');  exit();  }  $oAuth = new Google\_Service\_Oauth2($gClient);  $userData = $oAuth->userinfo\_v2\_me->get();  $\_SESSION['id'] = $userData['id'];  $\_SESSION['email'] = $userData['email'];  $\_SESSION['gender'] = $userData['gender'];  $\_SESSION['picture'] = $userData['picture'];  $\_SESSION['familyName'] = $userData['familyName'];  $\_SESSION['givenName'] = $userData['givenName'];  header('Location: index.php');  exit();  ?> |

7. Display Google Account Data inside web application

* Create the redirecting page (index.php)

|  |  |
| --- | --- |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\18.png | <?php  session\_start();  if (!isset($\_SESSION['access\_token'])) {  header('Location: login.php');  exit();  }  ?>  <!doctype html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"  content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  <meta http-equiv="X-UA-Compatible" content="ie=edge">  <title>Login With Google</title>  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous">  </head>  <style>body {  background-image: url('images/logo1.jpg');  background-repeat: no-repeat;  background-attachment: fixed;  background-size: cover;  }</style>  <body><div class="container" style="margin-top: 100px">  <div class="row">  <div class="col-md-3">  <img style="width: 80%;" src="<?php echo $\_SESSION['picture'] ?>">  </div>  <div class="col-md-9">  <table class="table table-hover table-bordered">  <tbody><tr><td>ID</td>  <td><?php echo $\_SESSION['id'] ?></td>  </tr><tr> <td>First Name</td>  <td><?php echo $\_SESSION['givenName'] ?></td>  </tr><tr><td>Last Name</td>  <td><?php echo $\_SESSION['familyName'] ?></td>  </tr><tr><td>Email</td>  <td><?php echo $\_SESSION['email'] ?></td></tr>  </tbody>  </table>  </div>  </div>  </div>  </body>  </html> |

- Display required data inside my application.

8. Finally creating logout.php file as follows

|  |  |
| --- | --- |
|  | <?php  require\_once "config.php";  unset($\_SESSION['access\_token']);  $gClient->revokeToken();  session\_destroy();  header('Location: login.php');  exit();  ?> |

**Appendix**

Following are the steps followed for Gogle API

To use Google identity platform with OAuth, it requires to create new project in Google API console. (<https://console.developers.google.com/>)

- Create new Project

|  |
| --- |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\4.pngCreate new project in Google API console. |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\5.pngCreate new project as “SS Assignment OAUTH” |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\6.png  SS Assignment OAUTH new project is created |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\7.png  Go to the API Library and Enable Google+ API |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\10.png  OAuth consent screen – Selected User Type as External |
| OAuth consent screen – Given required details C:\Users\gulawita\Desktop\Software Security\Screenshots\11.png  C:\Users\gulawita\Desktop\Software Security\Screenshots\9.pngGo to credentials and Create credentioals with OAuth Cliend ID |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\12.png  OAuth credentials – Given required details  C:\Users\gulawita\Desktop\Software Security\Screenshots\13.png  Get My Client ID and Client Secret |
| C:\Users\gulawita\Desktop\Software Security\Screenshots\14.png  Added Client ID and Client Secret in to config.php |

**Source Codes**

**Login.php**

|  |
| --- |
| <?php  require\_once "config.php";  if (isset($\_SESSION['access\_token'])) {  header('Location: index.php');  exit();  }  $loginURL = $gClient->createAuthUrl();  ?>  <!doctype html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"  content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  <meta http-equiv="X-UA-Compatible" content="ie=edge">  <title>Login With Google</title>  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous">  </head>  <style>  body {  background-image: url('images/logo1.jpg');  background-repeat: no-repeat;  background-attachment: fixed;  background-size: cover;  }  </style>  <body>  <div class="container" style="margin-top: 100px">  <div class="row justify-content-center">  <div class="col-md-6 col-offset-3" align="center">  <img src="images/logo2.png"><br><br>  <form >  <input placeholder="Email..." name="email" class="form-control"><br>  <input type="password" placeholder="Password..." name="password" class="form-control"><br>  <input type="submit" value="Log In" class="btn btn-primary"><br><br>  <input type="button" onclick="window.location = '<?php echo $loginURL ?>';" value="Log In With Google" class="btn btn-danger">  </form></div>  </div> </div>  </body>  </html> |

**Config.php**

|  |
| --- |
| <?php  session\_start();  require\_once "GoogleAPI/vendor/autoload.php";  $gClient = new Google\_Client();  $gClient->setClientId("337920838897-hm70l5optfd0bsfqe8ocv8sd34sfu1bs.apps.googleusercontent.com");  $gClient->setClientSecret("q0FkbMrpKUKrWtSrKT\_tzOCn");  $gClient->setApplicationName("SS Assignment OAUTH");  $gClient->setRedirectUri("http://localhost/SoftwareSec2/g-callback.php");  $gClient->addScope("https://www.googleapis.com/auth/plus.login https://www.googleapis.com/auth/userinfo.email");  ?> |

**g-callback.php**

|  |
| --- |
| <?php  require\_once "config.php";  if (isset($\_SESSION['access\_token']))  $gClient->setAccessToken($\_SESSION['access\_token']);  else if (isset($\_GET['code'])) {  $token = $gClient->fetchAccessTokenWithAuthCode($\_GET['code']);  $\_SESSION['access\_token'] = $token;  } else {  header('Location: login.php');  exit();  }  $oAuth = new Google\_Service\_Oauth2($gClient);  $userData = $oAuth->userinfo\_v2\_me->get();  $\_SESSION['id'] = $userData['id'];  $\_SESSION['email'] = $userData['email'];  $\_SESSION['gender'] = $userData['gender'];  $\_SESSION['picture'] = $userData['picture'];  $\_SESSION['familyName'] = $userData['familyName'];  $\_SESSION['givenName'] = $userData['givenName'];  header('Location: index.php');  exit();  ?> |

**index.php**

|  |
| --- |
| <?php  session\_start();  if (!isset($\_SESSION['access\_token'])) {  header('Location: login.php');  exit();  }  ?>  <!doctype html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport"  content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  <meta http-equiv="X-UA-Compatible" content="ie=edge">  <title>Login With Google</title>  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous">  </head>  <style>  body {  background-image: url('images/logo1.jpg');  background-repeat: no-repeat;  background-attachment: fixed;  background-size: cover;  }  </style>  <body>  <div class="container" style="margin-top: 100px">  <div class="row">  <div class="col-md-3">  <img style="width: 80%;" src="<?php echo $\_SESSION['picture'] ?>">  </div>  <div class="col-md-9">  <table class="table table-hover table-bordered">  <tbody>  <tr>  <td>ID</td>  <td><?php echo $\_SESSION['id'] ?></td>  </tr>  <tr>  <td>First Name</td>  <td><?php echo $\_SESSION['givenName'] ?></td>  </tr>  <tr>  <td>Last Name</td>  <td><?php echo $\_SESSION['familyName'] ?></td>  </tr>  <tr>  <td>Email</td>  <td><?php echo $\_SESSION['email'] ?></td>  </tr>  </tbody></table></div>  </div>  </div>  </body>  </html> |

**logout.php**

|  |
| --- |
| <?php  require\_once "config.php";  unset($\_SESSION['access\_token']);  $gClient->revokeToken();  session\_destroy();  header('Location: login.php');  exit();  ?> |