

# *Web Application with OAuth 2.0 framework for secure access delegation*

**Students:**      1) B.K.G.P.N.Weerawardana      - MS20901226  
                         2) N.P.A.M.Reeshan                      - MS20912420

## **Project Description**

### **PROJECT OVERVIEW**

A Java based web application project to test OAuth 2.0 Framework by consuming OAuth Authorization Server and an OAuth Resource Server for services such as Facebook, Google.

## **Purpose of the project**

To demonstrate the use of OAuth 2.0 framework in web development.

## **Introduction**

OAuth is an open standard for access delegation. It allows applications to obtain limited access to user's data without providing the user credentials. OAuth provides access tokens, rather than credentials for APIs, authorized devices, Servers and Applications it works over HTTP. It decouples authentication from authorization, a secure delegated access to server resources on behalf of a resource owner is provided to a client via OAuth. This project demonstrates the use of OAuth authorization framework in web development with a java based web application.

### **OAUTH ROLES**

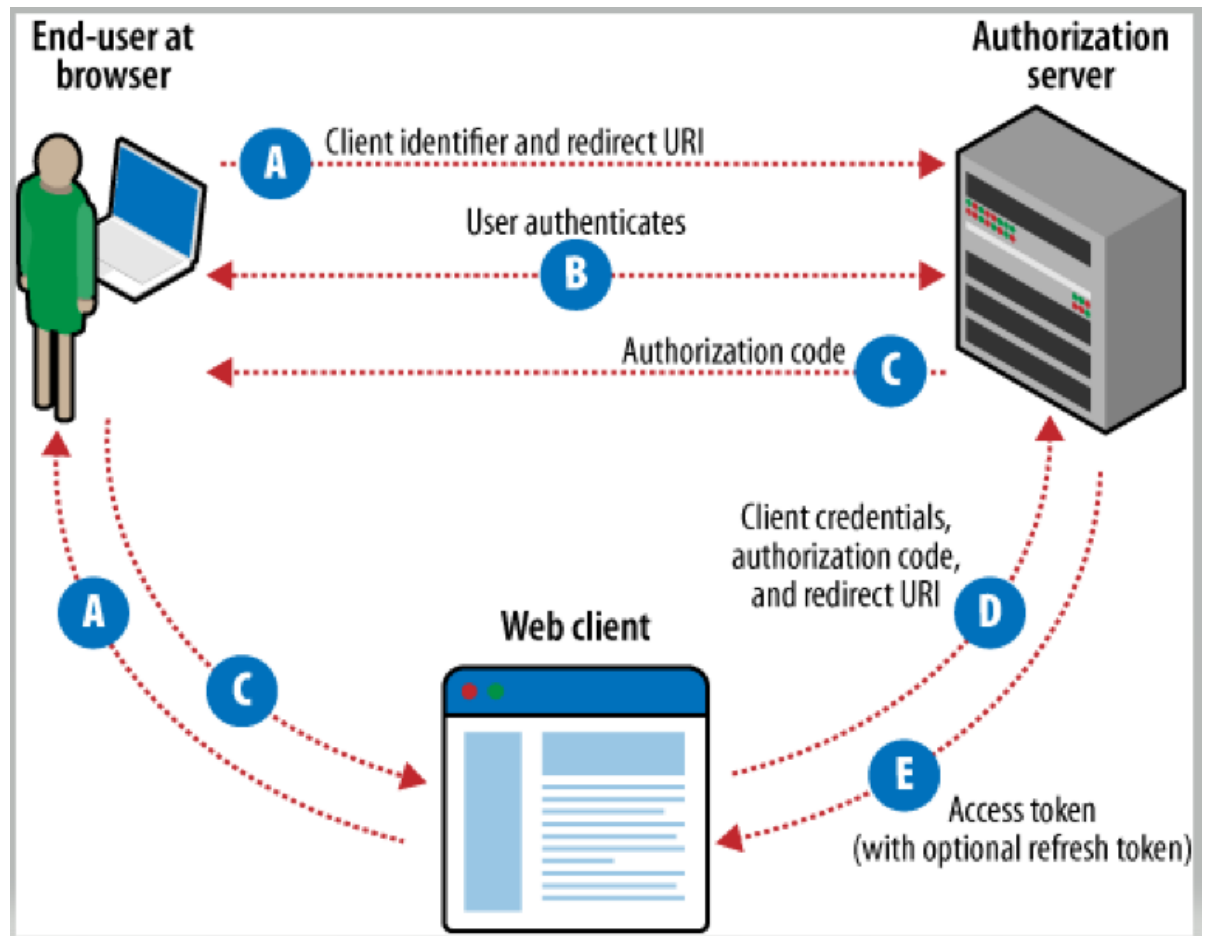
**Resource Owner:** The end user to whom the resource belongs to or the entity which can grant access to the secure resource.

**Resource Server:** The server hosting the protected resources. This is the API you want to access.

**Client:** The application requesting access to a protected resource on behalf of the Resource Owner.

**Authorization Server:** The server that authenticates the Resource Owner, and issues Access Tokens after getting proper authorization.

## *OAuth PROCESS*



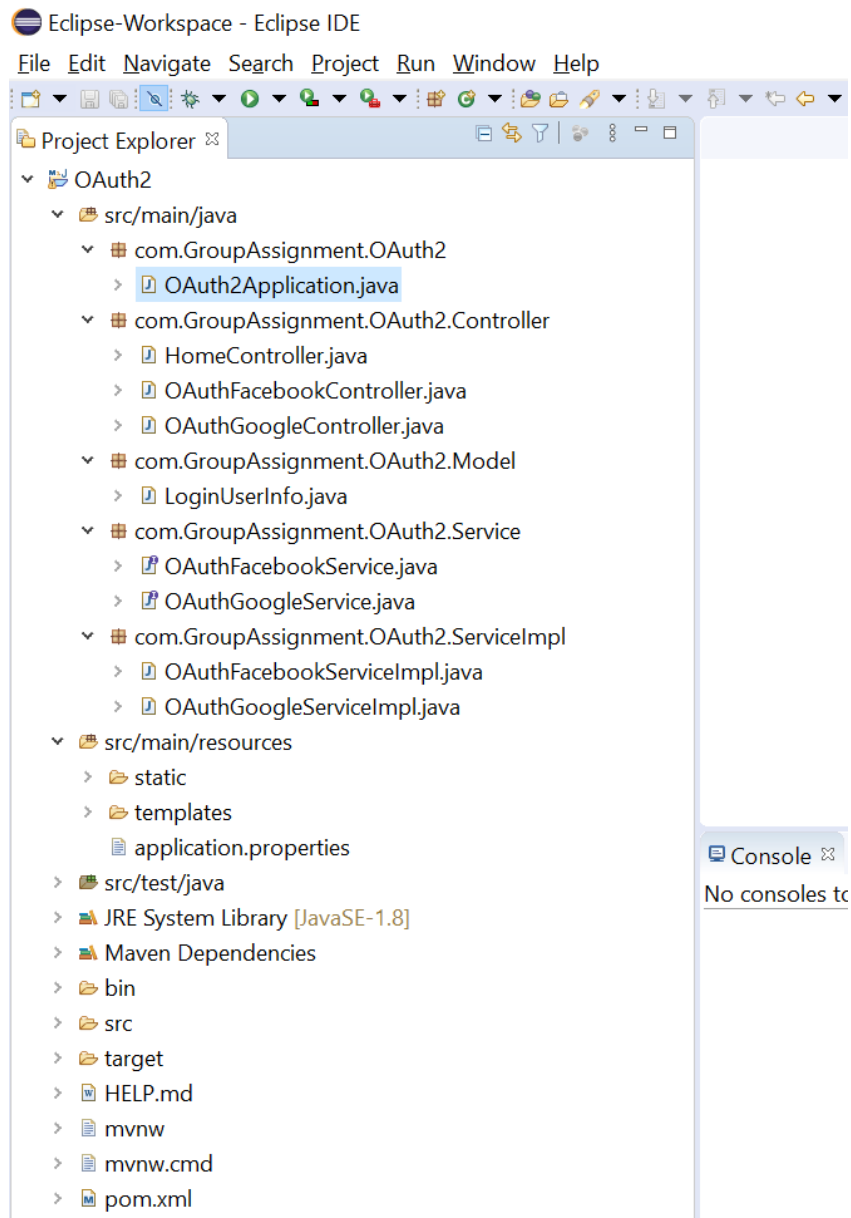
### **Methodology**

A web application created to demonstrate OAuth2. An index page which is designed with buttons to login to Facebook, Google and upon clicking, user authorization is requested (OAuth) once access is granted to specific scope by the user (Resource owner) First Name, Last Name and Email Id are fetched through relevant API (Resource Server) and displayed on the browser.

### *SOFTWARE HARDWARE USED*

spring-boot-starter-parent 2.2.6.RELEASE  
Apache Maven 4.0  
Java version 1.8  
spring-security-oauth2 2.1.5.RELEASE  
spring-boot-starter-oauth2-client  
spring-boot-starter-thymeleaf  
spring-boot-maven-plugin  
jquery 2.1.3  
bootstrap 3.2.0

## MAVEN PROJECT



## SCREENSHOTS

### Facebook Application

Creating a new application through developers portal

## Create a New App ID

Get started integrating Facebook into your app or website

Display Name

Contact Email

This email address is used to contact you about potential policy violations, app restrictions or steps to recover the app if it's been deleted or compromised.

By proceeding, you agree to the [Facebook Platform Policies](#)

CancelCreate App ID

facebook for developers

DocsToolsSupportMy Apps

SS\_OAuth2

APP ID: 720388762065557

In development

Reminder: Graph API v2.12 will be deprecated on May 05, 2020. Please use the [API Upgrade Tool](#) to understand how this might

Dashboard

Settings

Roles

Alerts

App Review

PRODUCTS +

Add a Product

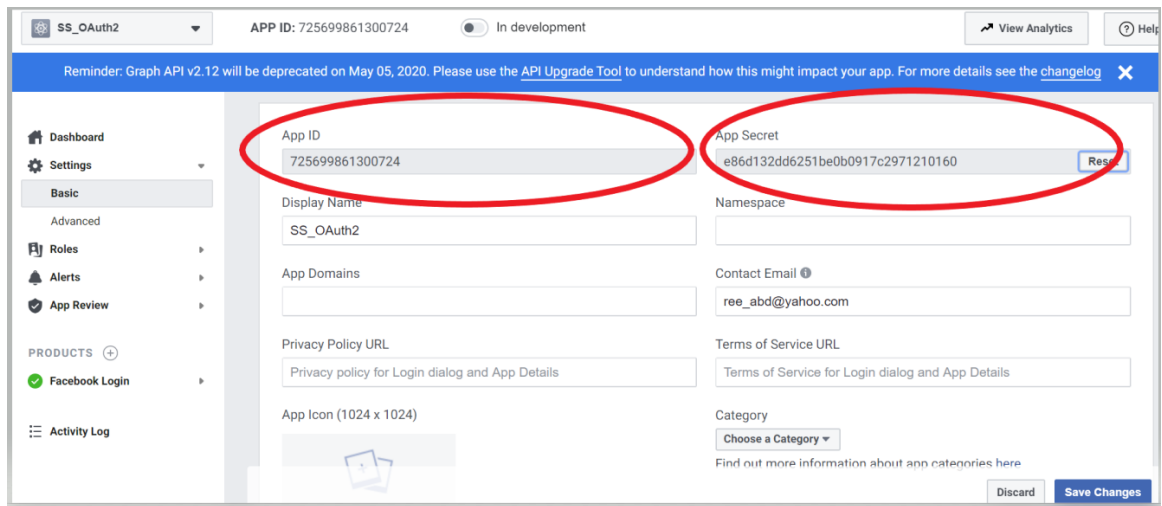
Facebook Login

The world's number one social login product.

Audience Network

Monetize your mobile app or website with n  
ads from 3 million Facebook advertisers

Note the App ID and App Secret



The image shows the 'Basic' tab of the Facebook App Settings for an application named 'SS\_OAuth2'. The App ID is 725699861300724 and the App Secret is e86d132dd6251be0b0917c2971210160. Both the App ID and App Secret fields are circled in red. The interface includes a left sidebar with navigation options like Dashboard, Settings, Roles, Alerts, App Review, and Activity Log. A top banner provides a reminder about Graph API v2.12 deprecation. The main content area contains various fields for app configuration, including Display Name, App Domains, Privacy Policy URL, App Icon, Namespace, Contact Email, Terms of Service URL, and Category. A 'Save Changes' button is located at the bottom right.

SS\_OAuth2 APP ID: 725699861300724 In development View Analytics Help

Reminder: Graph API v2.12 will be deprecated on May 05, 2020. Please use the [API Upgrade Tool](#) to understand how this might impact your app. For more details see the [changelog](#)

Dashboard Settings Basic Advanced Roles Alerts App Review PRODUCTS Facebook Login Activity Log

App ID: 725699861300724 App Secret: e86d132dd6251be0b0917c2971210160 [Reset]

Display Name: SS\_OAuth2 Namespace:

App Domains: Contact Email: ree\_abd@yahoo.com

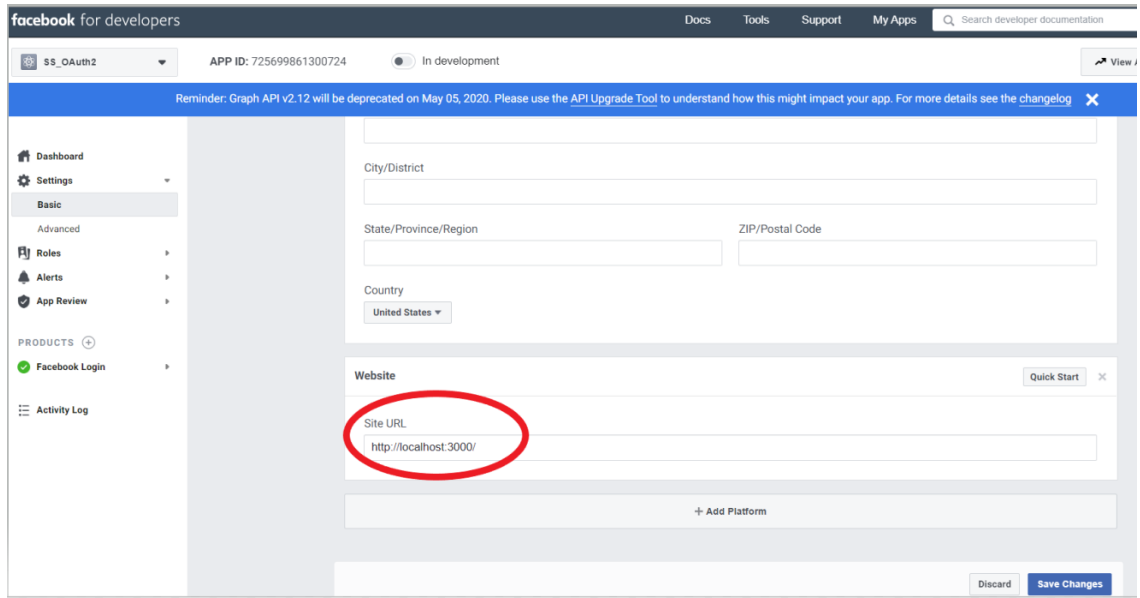
Privacy Policy URL: Privacy policy for Login dialog and App Details Terms of Service URL: Terms of Service for Login dialog and App Details

App Icon (1024 x 1024): [Image Placeholder]

Category: Choose a Category Find out more information about app categories [here](#)

Discard Save Changes

Set Site URL



The image shows the 'Website' tab of the Facebook App Settings for the same application. The 'Site URL' field is circled in red and contains the value 'http://localhost:3000/'. The interface is similar to the previous screenshot, with a left sidebar and a top banner. The main content area includes fields for City/District, State/Province/Region, ZIP/Postal Code, Country, and Website. A 'Quick Start' button is visible next to the Website field. A '+ Add Platform' button is at the bottom. A 'Save Changes' button is at the bottom right.

facebook for developers Docs Tools Support My Apps Search developer documentation

SS\_OAuth2 APP ID: 725699861300724 In development View

Reminder: Graph API v2.12 will be deprecated on May 05, 2020. Please use the [API Upgrade Tool](#) to understand how this might impact your app. For more details see the [changelog](#)

Dashboard Settings Basic Advanced Roles Alerts App Review PRODUCTS Facebook Login Activity Log

City/District:

State/Province/Region: ZIP/Postal Code:

Country: United States

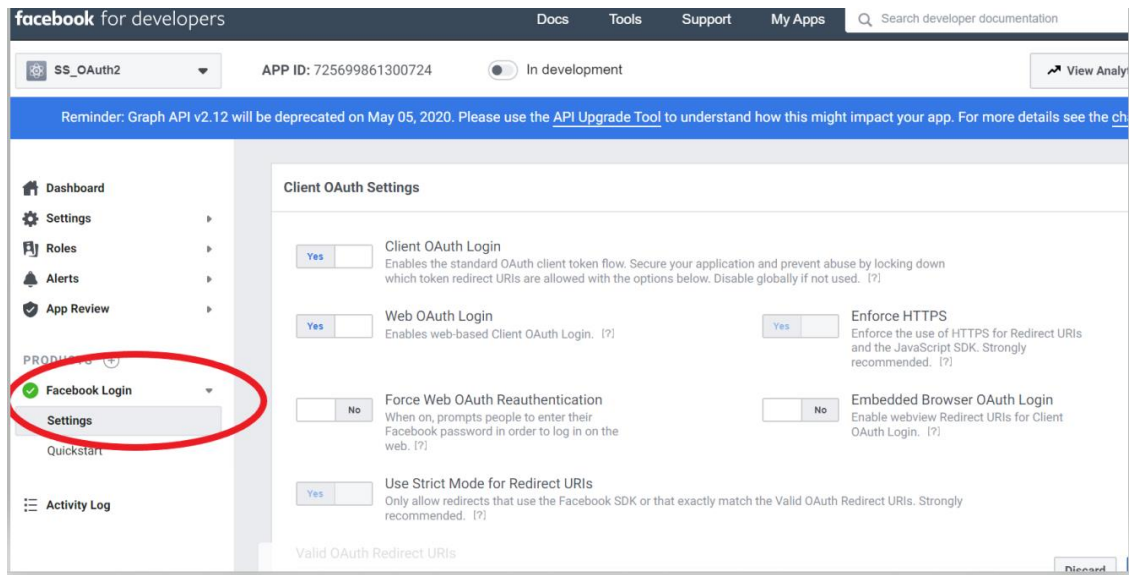
Website: [Quick Start]

Site URL: http://localhost:3000/

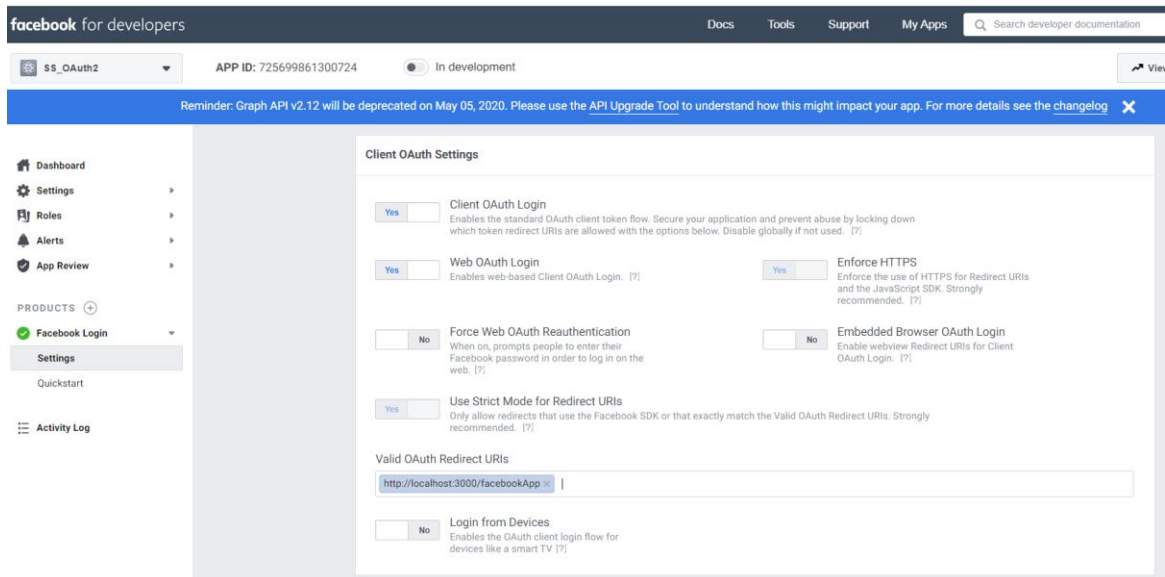
+ Add Platform

Discard Save Changes

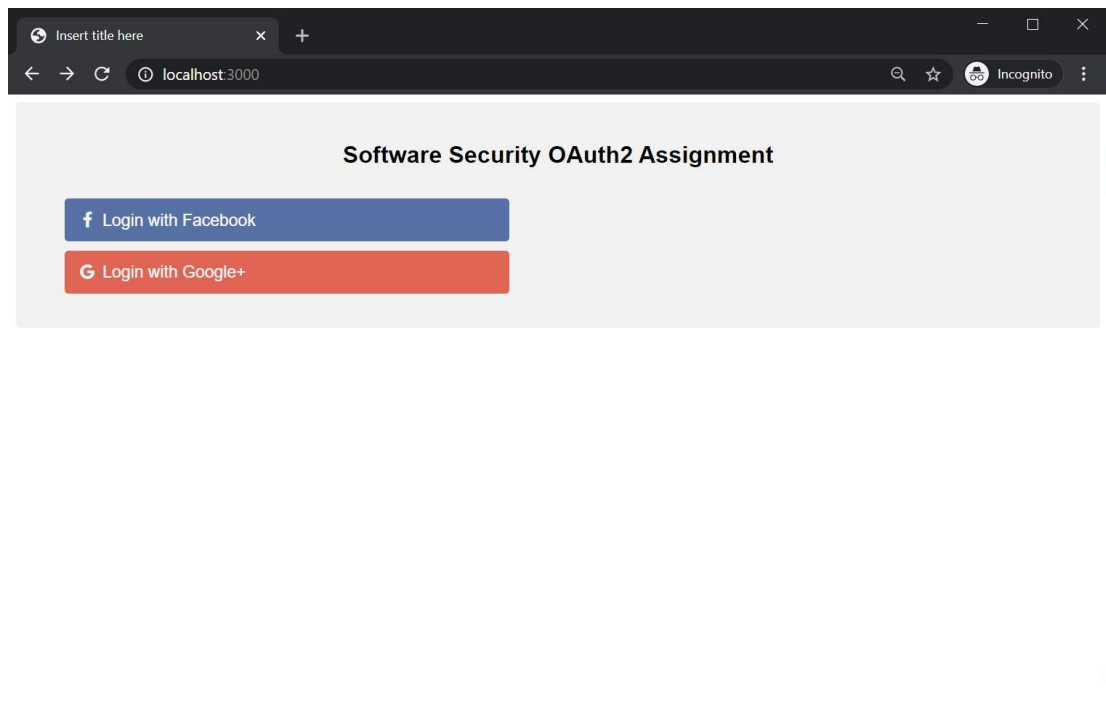
## Set OAuth Settings



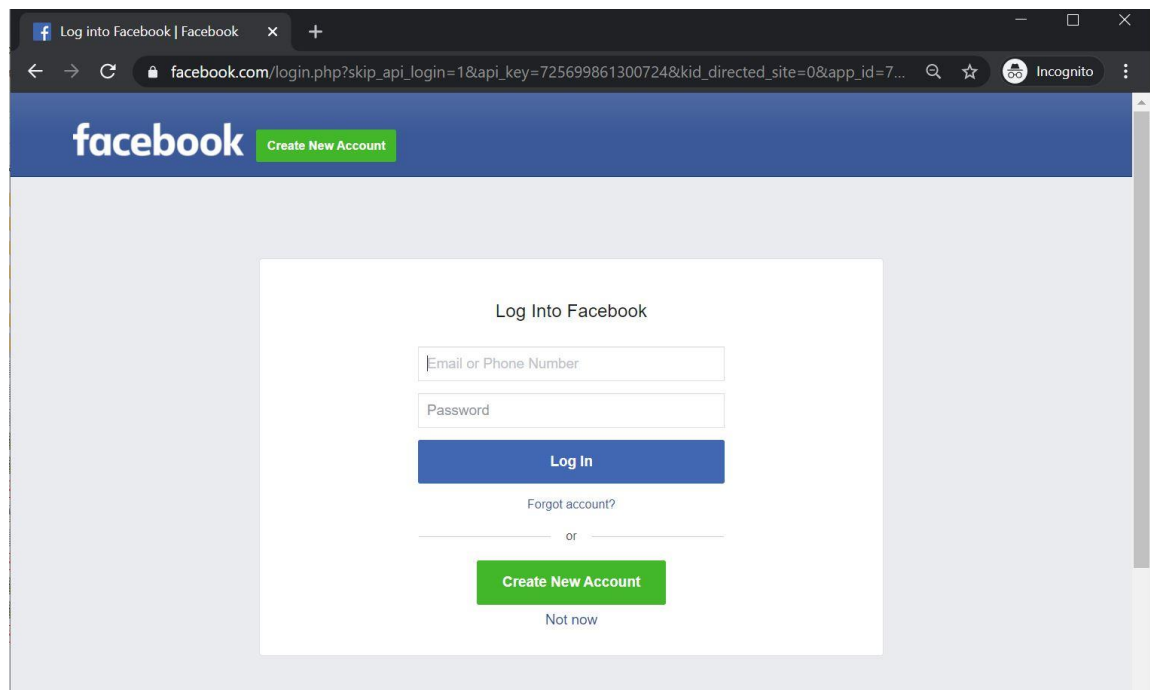
## Set redirect URLs, (For local host it is not needed)



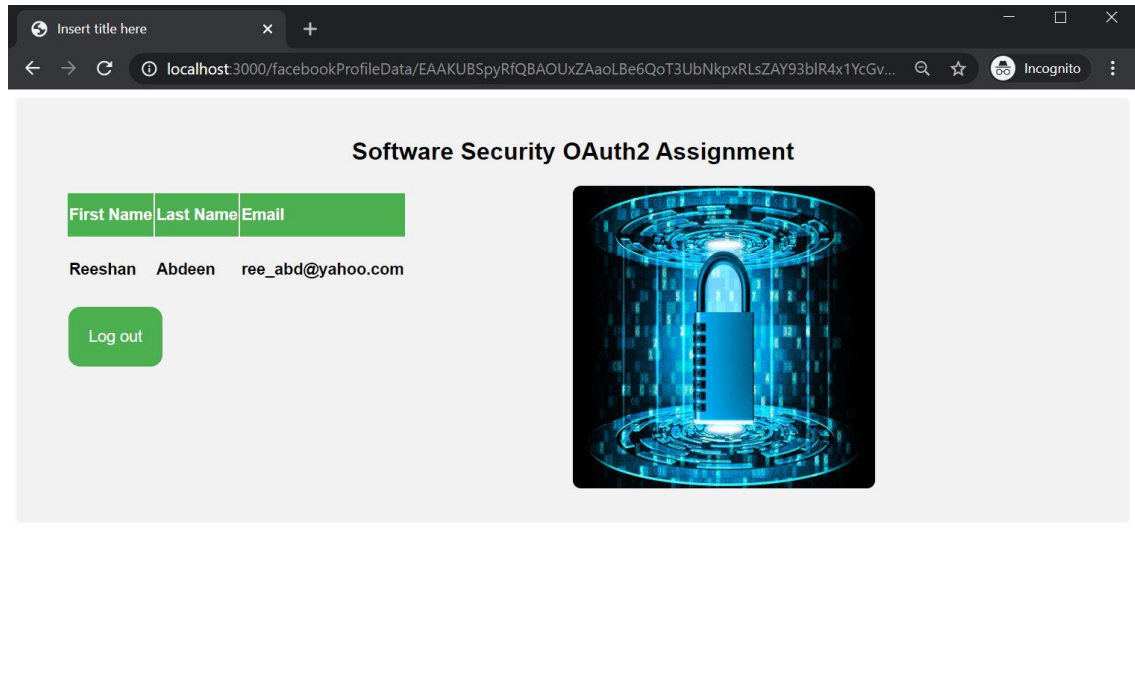
## 1) Index.html



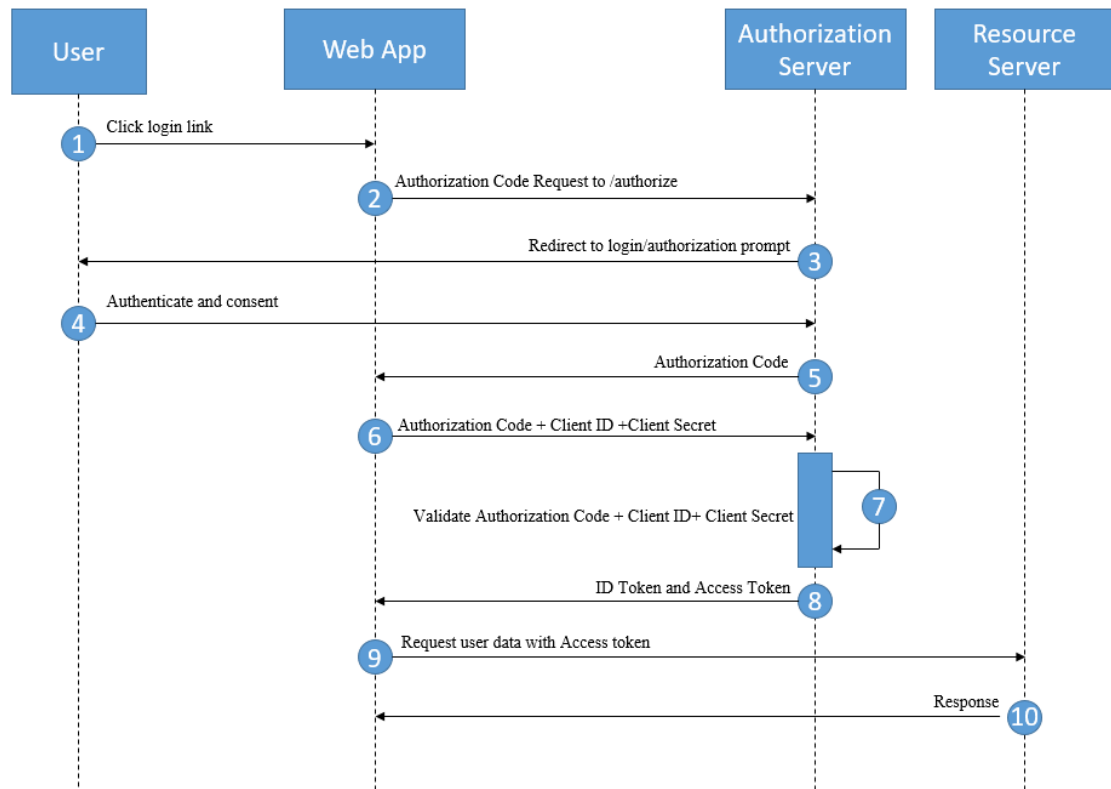
## 2) Facebook Login



### 3) Redirected page after authorization



### ***OAUTH MESSAGE FLOW***





## MESSAGES

### **Authorization code request to Authorization from Web Application:**

GET

/v2.5/dialog/oauth?client\_id=725699861300724&response\_type=code&redirect\_uri=http%3A%2F%2Flocalhost%3A3000%2FfacebookApp&scope=public\_profile%2Cemail%2Cuser\_photos  
HTTP/1.1

### **Redirected Authorization Request to Resource owner:**

GET

/login.php?skip\_api\_login=1&api\_key=725699861300724&kid\_directed\_site=0&app\_id=725699861300724&signed\_next=1&next=https%3A%2F%2Fwww.facebook.com%2Fv2.5%2Fdialog%2Foauth%3Fclient\_id%3D725699861300724%26response\_type%3Dcode%26redirect\_uri%3Dhttp%253A%252F%252Flocalhost%253A3000%252FfacebookApp%26scope%3Dpublic\_profile%252Cemail%252Cuser\_photos%26ret%3Dlogin%26fbapp\_pres%3D0%26logger\_id%3D6ba50edb-0c5c-452b-b0b6-f04f0ed50849&cancel\_url=http%3A%2F%2Flocalhost%3A3000%2FfacebookApp%3Ferror%3Daccess\_denied%26error\_code%3D200%26error\_description%3DPermissions%2Berror%26error\_reason%3Duser\_denied%23\_%3D\_&display=page&locale=en\_US&pl\_dbl=0 HTTP/1.1

### **Authorization code from Authorization server**

GET /facebookApp?code=AQDgtQ9LxLohvMIHENij0Yd4XS8mqTZj0uFe2x7EwombPg3VrQ\_F-cRAvik1rJhjQ0XMO5AExVy3L9peDw7SWMKD2\_ZbR-ekBj85qAFkxWZif7c8\_cbclmWqOe09i1jFsDI5V6OuCFJtv674LgdVMTyyrs3x-0D6vAB1T8Bmhn8hw3EGZKANuc\_ttEms9DpHdAC4pv8qWTSafifNIXdvXLykw3WpsE87TrsohbS3fG1mB2XtWBEhAkCiI70RLVqVdcXgiwaJ0r7ITkWjEiAlZ4L-6TACbLGYf9xAwShrvLYlo4p532U40RiOTHWVzcSbBY HTTP/1.1

### **Access Token**

GET

/facebookProfileData/EEAKUBSpyRfQBAlefZBJDW9aI1dMuuHoQ5T2UkSi4pQQu5Ph65nDRjF505LMpUfldAS09NTXDBQcyBwPypODVkuUhhGxIxrD0QjZATGq9hsTvbCI71HuTSCQAPO6IIVSr2AdICoEIJEH0i9fcV5KoSBCOnwpNgZD HTTP/1.1

### **Request user data with Access Token**

http://localhost:3000/facebookProfileData/EEAKUBSpyRfQBAlefZBJDW9aI1dMuuHoQ5T2UkSi4pQQu5Ph65nDRjF505LMpUfldAS09NTXDBQcyBwPypODVkuUhhGxIxrD0QjZATGq9hsTvbCI71HuTSCQAPO6IIVSr2AdICoEIJEH0i9fcV5KoSBCOnwpNgZD HTTP/1.1#\_=\_

## Appendix –A

### 1) Main Class

```
package com.GroupAssignment.OAuth2;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
```

```
public class OAuth2Application {
```

```
    //main class
```

```
    public static void main(String[] args) {
        SpringApplication.run(OAuth2Application.class, args);
    }
```

```
}
```

### 2) Pom.xml (Dependency)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
```

```
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-
4.0.0.xsd">
```

```
    <modelVersion>4.0.0</modelVersion>
```

```
    <parent>
```

```
        <groupId>org.springframework.boot</groupId>
```

```
        <artifactId>spring-boot-starter-parent</artifactId>
```

```
        <version>2.2.6.RELEASE</version>
```

```

        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.GroupAssignment</groupId>
    <artifactId>OAuth2</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>OAuth2</name>
    <description>Software Security OAuth2 Assignment</description>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>

        <!-- https://mvnrepository.com/artifact/com.h2database/h2 -->
    <dependency>
        <groupId>com.h2database</groupId>
        <artifactId>h2</artifactId>

        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

```

```

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
    <exclusions>
        <exclusion>
            <groupId>org.junit.vintage</groupId>
            <artifactId>junit-vintage-engine</artifactId>
        </exclusion>
    </exclusions>
</dependency>

```

```

<!--
https://mvnrepository.com/artifact/org.springframework.social/spring-social-core -->

```

```

<dependency>
    <groupId>org.springframework.social</groupId>
    <artifactId>spring-social-core</artifactId>
    <version>1.1.6.RELEASE</version>
</dependency>

```

```

<!-- https://mvnrepository.com/artifact/org.springframework.social/spring-social-web -->

```

```

<dependency>
    <groupId>org.springframework.social</groupId>

```

```
<artifactId>spring-social-web</artifactId>

<version>1.1.6.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework.social/spring-social-config -->
<dependency>
  <groupId>org.springframework.social</groupId>
  <artifactId>spring-social-config</artifactId>
  <version>1.1.0.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework.social/spring-social-facebook -->
<dependency>
  <groupId>org.springframework.social</groupId>
  <artifactId>spring-social-facebook</artifactId>
  <version>2.0.3.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework.social/spring-social-linkedin -->
<dependency>
  <groupId>org.springframework.social</groupId>
  <artifactId>spring-social-linkedin</artifactId>
  <version>1.0.0.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework.social/spring-social-google -->
<dependency>
  <groupId>org.springframework.social</groupId>
  <artifactId>spring-social-google</artifactId>
  <version>1.0.0.RC1</version>
```

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

### 3) application.properties

**server.port=3000**

spring.social.facebook.app-id=691419415009768

spring.social.facebook.app-secret=d050626848573ef79a83b12b909caae5

spring.social.github.app-id=8b0e03b365d0d0e4f6e1

spring.social.github.app-secret=27bbced66e0dc6a4700f2c44b8d0c0261410aee9

spring.social.linkedin.app-id=

spring.social.linkedin.app-secret=

spring.social.google.app-id=1032610895721-  
7mfa0s5046qt6da3sgrhmg054j49nsm7.apps.googleusercontent.com

spring.social.google.app-secret=tEMz8xNb5b7LWmk949gVXy1Y

#### 4) index.html

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css" />

<link rel="stylesheet" th:href="@{/css/style.css}" />

</head>

<body>

<div class="container">


<div class="row">

<h2 style="text-align:center">Software Security OAuth2 Assignment</h2>


<div class="col">

<a th:href="@{/facebookloginOAuth}" class="fb btn">

<i class="fa fa-facebook fa-fw"></i> Login with Facebook

</a>


<a th:href="@{/googleloginOAuth}" class="google btn"><i class="fa fa-google fa-fw">

</i> Login with Google+

</a>

</div>

</div>

</div>

</div>
```

</body>

</html>

## 5) UserProfile.index

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css" />

<link rel="stylesheet" th:href="@{/css/style.css}" />

</head>

<body>

<div class="container">

<div class="row">

<h2 style="text-align:center">Software Security OAuth2 Assignment</h2>

<div class="col">

<table id="customers">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

<tr>

<td><h4 th:text="\${ user.firstName }"></h4></td>



```
<td> <h4 th:text="${user.lastName}"></h4></td>
```

```
<td> <h4 th:text="${user.imageUrl}"></h4></td>
```

```
</tr>
```

```
</table>
```

```
<a th:href="@{'/logout'}"> <button class="button button4"> Log out</button> </a>
```

```
</div>
```

```

```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

## 6) Style.css

```
body {  
    font-family: Arial, Helvetica, sans-serif;  
}
```

```
* {  
    box-sizing: border-box;
```

```
}
```

```
/* style the container */
```

```
.container {  
    position: relative;  
    border-radius: 5px;  
    background-color: #f2f2f2;  
    padding: 20px 0 30px 0;  
}
```

```
/* style inputs and link buttons */
```

```
input,  
.btn {  
    width: 100%;  
    padding: 12px;  
    border: none;  
    border-radius: 4px;  
    margin: 5px 0;  
    opacity: 0.85;  
    display: inline-block;  
    font-size: 17px;  
    line-height: 20px;  
    text-decoration: none; /* remove underline from anchors */  
}
```

```
input:hover,
```

```
.btn:hover {  
    opacity: 1;  
}
```

```
/* add appropriate colors to fb, twitter and google buttons */
```

```
.fb {  
    background-color: #3B5998;  
    color: white;  
}
```

```
.twitter {  
    background-color: #55ACEE;  
    color: white;  
}
```

```
.google {  
    background-color: #dd4b39;  
    color: white;  
}
```

```
/* style the submit button */  
input[type=submit] {  
    background-color: #4CAF50;  
    color: white;  
    cursor: pointer;  
}
```

```
input[type=submit]:hover {  
    background-color: #45a049;  
}
```

```
/* Two-column layout */  
.col {  
    float: left;  
    width: 50%;  
    margin: auto;
```

```
padding: 0 50px;
margin-top: 6px;
}

/* Clear floats after the columns */
.row:after {
    content: "";
    display: table;
    clear: both;
}

/* vertical line */
.vl {
    position: absolute;
    left: 50%;
    transform: translate(-50%);
    border: 2px solid #ddd;
    height: 175px;
}

/* text inside the vertical line */
.vl-innertext {
    position: absolute;
    top: 50%;
    transform: translate(-50%, -50%);
    background-color: #f1f1f1;
    border: 1px solid #ccc;
    border-radius: 50%;
    padding: 8px 10px;
}
```

```
/* hide some text on medium and large screens */
```

```
.hide-md-lg {  
    display: none;  
}
```

```
/* bottom container */
```

```
.bottom-container {  
    text-align: center;  
    background-color: #666;  
    border-radius: 0px 0px 4px 4px;  
}
```

```
/* Responsive layout - when the screen is less than 650px wide, make the two columns stack on top of each other instead of next to each other */
```

```
@media screen and (max-width: 650px) {
```

```
.col {  
    width: 100%;  
    margin-top: 0;  
}
```

```
/* hide the vertical line */
```

```
.vl {  
    display: none;  
}
```

```
/* show the hidden text on small screens */
```

```
.hide-md-lg {  
    display: block;  
    text-align: center;  
}
```

```
}
```

```
.button {
```

```
    background-color: #4CAF50; /* Green */  
    border: none;
```

```
color: white;
padding: 20px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
margin: 4px 2px;
cursor: pointer;
}
```

```
.button4 {border-radius: 12px;}
```

```
font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
border-collapse: collapse;
width: 100%;
}
```

```
#customers td, #customers th {
border: 1px solid #ddd;
padding: 8px;
}
```

```
#customers tr:nth-child(even){background-color: #f2f2f2;}
```

```
#customers tr:hover {background-color: #ddd;}
```

```
#customers th {
padding-top: 12px;
padding-bottom: 12px;
text-align: left;
background-color: #4CAF50;
```

```
        color: white;
    }
    img {
        border-radius: 8px;
    }
```

## **@Model**

### 1) UserInfoModel

```
package com.GroupAssignment.OAuth2.Model;
```

```
public class LoginUserInfo {

    //user data model

    public String firstName;
    public String lastName;
    public String imageUrl;

    public LoginUserInfo() {}

    public LoginUserInfo(String firstName, String lastName, String imageUrl) {
        super();
        this.firstName = firstName;
        this.lastName = lastName;
        this.imageUrl = imageUrl;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
}
```

```

    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getImageUrl() {
        return imageUrl;
    }

    public void setImageUrl(String imageUrl) {
        this.imageUrl = imageUrl;
    }
}

}

```

## **@Controllers**

### 1) OAuthHomeController

```
package com.GroupAssignment.OAuth2.Controller;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
@Controller
```

```
public class HomeController {
```

```
    //create rest controller for url mapping
```



```

//redirect to home page
@GetMapping(value = "/")
public String HomeView() {
    return "view/index";
}

//logout button controller mapping. when
@GetMapping(value="/logout")
public String Logout() {
    return "view/index";
}
}

```

## 2) OAuthFacebookController

```

package com.GroupAssignment.OAuth2.Controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.social.facebook.api.User;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.view.RedirectView;

import com.GroupAssignment.OAuth2.Model.LoginUserInfo;
import com.GroupAssignment.OAuth2.Service.OAuthFacebookService;

```

@Controller

```
public class OAuthFacebookController {
```

```
    // define AuthFacebookService private property
```

```
    @Autowired private OAuthFacebookService oAuthFacebookService;
```

```
    //request from server
```

```
    @GetMapping(value = "/facebookloginOAuth")
```

```
    public RedirectView OAuthFacebookLogin() {
```

```
        RedirectView redirectView = new RedirectView();
```

```
        String url = oAuthFacebookService.OAuthFacebookLogin();
```

```
        System.out.println(url);
```

```
        redirectView.setUrl(url);
```

```
        return redirectView;
```

```
    }
```

```
    //request to token
```

```
    @GetMapping("/facebookApp")
```

```
    public String facebookApp(@RequestParam("code") String code) {
```

```
        String accessToken = oAuthFacebookService.getFacebookOAuthAccessToken(code);
```

```
        return "redirect:/facebookProfileData/"+accessToken;
```

```
    }
```

```
    //get token and return user info value to UserProfile redirect page
```

```
    @GetMapping(value = "/facebookProfileData/{accessToken:.+}")
```

```
    public String facebookProfileData(@PathVariable String accessToken, Model model) {
```

```
        User user = oAuthFacebookService.getFacebookUserProfile(accessToken);
```

```
        LoginUserInfo loginUserInfo = new LoginUserInfo(user.getFirstName(), user.getLastName(),
user.getEmail());
```

```
        // print user , token and user info
```

```
        System.out.println("user print here = "+user);
```

```
        System.out.println("Token print here = "+accessToken);
```

```
        System.out.println("Login user info print here = "+loginUserInfo);
```

```
        model.addAttribute("user", loginUserInfo);
```

```
        return "view/UserProfile";
```

```
    }
```

```
}
```

### 3) OAuthGoogleController

```
package com.GroupAssignment.OAuth2.Controller;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.social.google.api.plus.Person;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import org.springframework.web.servlet.view.RedirectView;
```

```

import com.GroupAssignment.OAuth2.Model.LoginUserInfo;

import com.GroupAssignment.OAuth2.Service.OAuthGoogleService;


@Controller

public class OAuthGoogleController {


    // define AuthGoogleService private property
    @Autowired private OAuthGoogleService oAuthGoogleService;


    //request from server
    @GetMapping(value = "/googleloginOAuth")

    public RedirectView OAuthGoogleLogin() {


        RedirectView redirectView = new RedirectView();

        String url = oAuthGoogleService.OAuthGoogleLogin();

        System.out.println(url);

        redirectView.setUrl(url);

        return redirectView;

    }


    //request to token
    @GetMapping("/google")

    public String google(@RequestParam("code") String code) {

        String accessToken = oAuthGoogleService.getGoogleOAuthAccessToken(code);

        return "redirect:/googleProfileData/"+accessToken;

    }


    //get token and return user info value to UserProfile redirect page
    @GetMapping(value = "/googleProfileData/{accessToken:.+}")

    public String googleProfileData(@PathVariable String accessToken, Model model) {

```

```

        Person person = oAuthGoogleService.getGoogleUserProfile(accessToken);

        LoginUserInfo loginUserInfo = new LoginUserInfo(person.getGivenName(),
        person.getFamilyName(), person.getImageUrl());

        // print user , token and user info

        System.out.println("user print here = "+person);

        System.out.println("Token print here = "+accessToken);

        System.out.println("Login user info print here = "+loginUserInfo);

        model.addAttribute("person", loginUserInfo);

        return "view/UserProfile";

    }

}

```

## **@Services**

### 1) OAuthFacebookServices

```

package com.GroupAssignment.OAuth2.Service;

import org.springframework.social.facebook.api.User;

// interface for all facebook services

public interface OAuthFacebookService {

    String OAuthFacebookLogin();
}

```

```
String getFacebookOAuthAccessToken(String code);

User getFacebookUserProfile(String accessToken);

}
```

## 2) OAuthGoogleServices

```
package com.GroupAssignment.OAuth2.Service;

import org.springframework.social.google.api.plus.Person;

//interface for all facebook services

public interface OAuthGoogleService {

    String OAuthGoogleLogin();

    String getGoogleOAuthAccessToken(String code);

    Person getGoogleUserProfile(String accessToken);

}
```

## @ServiceImpl

### 3) OAuthFacebookServiceImpl

```
package com.GroupAssignment.OAuth2.ServiceImpl;
```

```
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.social.facebook.api.Facebook;  
import org.springframework.social.facebook.api.User;  
import org.springframework.social.facebook.api.impl.FacebookTemplate;  
import org.springframework.social.facebook.connect.FacebookConnectionFactory;  
import org.springframework.social.oauth2.OAuth2Parameters;  
import org.springframework.stereotype.Service;
```

```
import com.GroupAssignment.OAuth2.Service.OAuthFacebookService;
```

```
@Service
```

```
public class OAuthFacebookServiceImpl implements OAuthFacebookService{
```

```
    // get facebook id and secret value from application.properties
```

```
    @Value("${spring.social.facebook.app-id}")
```

```
    private String facebookId;
```

```
    @Value("${spring.social.facebook.app-secret}")
```

```
    private String facebookSecret;
```

```
    // facebook connection create
```

```
    private FacebookConnectionFactory CreateOAuthFacebookConnection() {
```

```
        return new FacebookConnectionFactory(facebookId, facebookSecret);
```

```
    }
```

```

// create auth server and assign scope and redirect page
@Override
public String OAuthFacebookLogin() {
    OAuth2Parameters parameters = new OAuth2Parameters();
    parameters.setRedirectUri("http://localhost:3000/facebookApp");
    parameters.setScope("public_profile,email,user_photos");
    return
CreateOAuthFacebookConnection().getOAuthOperations().buildAuthenticateUrl(parameters);
}

// return access token
@Override
public String getFacebookOAuthAccessToken(String code) {
    return CreateOAuthFacebookConnection().getOAuthOperations().exchangeForAccess(code,
"http://localhost:3000/facebookApp", null).getAccessToken();
}

// return authorized user information from facebook server
@Override
public User getFacebookUserProfile(String accessToken) {
    Facebook facebook = new FacebookTemplate(accessToken);
    String[] values = {"id", "first_name", "last_name", "cover", "email"};
    return facebook.fetchObject("me", User.class, values);
}

}

```

#### 4) OAuthGoogleServiceImpl



```

package com.GroupAssignment.OAuth2.ServiceImpl;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.social.google.api.Google;
import org.springframework.social.google.api.impl.GoogleTemplate;
import org.springframework.social.google.api.plus.Person;
import org.springframework.social.google.connect.GoogleConnectionFactory;
import org.springframework.social.oauth2.OAuth2Parameters;
import org.springframework.stereotype.Service;

import com.GroupAssignment.OAuth2.Service.OAuthGoogleService;

@Service
public class OAuthGoogleServiceImpl implements OAuthGoogleService{

    // get google id and secret value from application.properties
    @Value("${spring.social.google.app-id}")
    private String googleId;
    @Value("${spring.social.google.app-secret}")
    private String googleSecret;

    // google connection create
    private GoogleConnectionFactory CreateOAuthGoogleConnection() {
        return new GoogleConnectionFactory(googleId, googleSecret);
    }

    // create auth server and assign scope and redirect page
    @Override
    public String OAuthGoogleLogin() {
        OAuth2Parameters parameters = new OAuth2Parameters();
        parameters.setRedirectUri("http://localhost:3000/google");
    }
}

```

```

        parameters.setScope("profile");

        return CreateOAuthGoogleConnection().getOAuthOperations().buildAuthenticateUrl(parameters);
    }

    // return access token

    @Override

    public String getGoogleOAuthAccessToken(String code) {

        return CreateOAuthGoogleConnection().getOAuthOperations().exchangeForAccess(code,
"http://localhost:3000/google", null).getAccessToken();

    }

    // return authorized user information from facebook server

    @Override

    public Person getGoogleUserProfile(String accessToken) {

        Google google = new GoogleTemplate(accessToken);

        Person person = google.plusOperations().getGoogleProfile();

        return person;

    }

}

```