Assignment 2: Software Security Web Application

1 - Objective

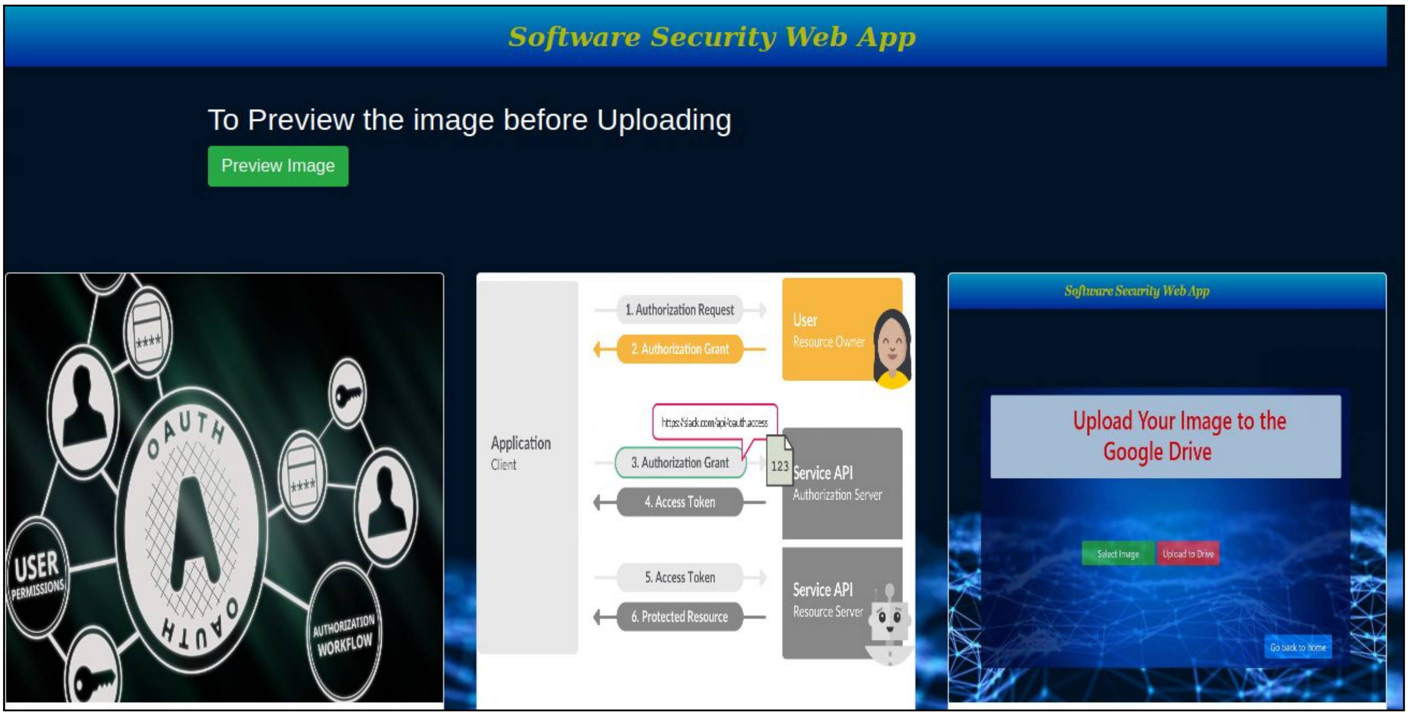
To develop a JavaScript based online application for GDrive image upload. A link is available for the app in terms of a sign-in option, which redirects to Google where an access token for OAuth 2.0 is obtained once a user successfully completes the authentication. Then the application would provide a browse button where the user could select the image file to upload and it would be automatically uploaded to the Google Drive of the user. For uploading the file, the application would use the OAuth access token obtained and call the **Google APIs**. [1]

2 - Technologies

Framework: **OAuth 2.0 framework**
Public OAuth Server: **Google**
Client-side Programming Language: **JavaScript**
Deployment: **localhost**
Web Server Solution: **XAMPP**
O/S Platform: **Windows**
Web Browser: **Internet Explorer, Chrome**

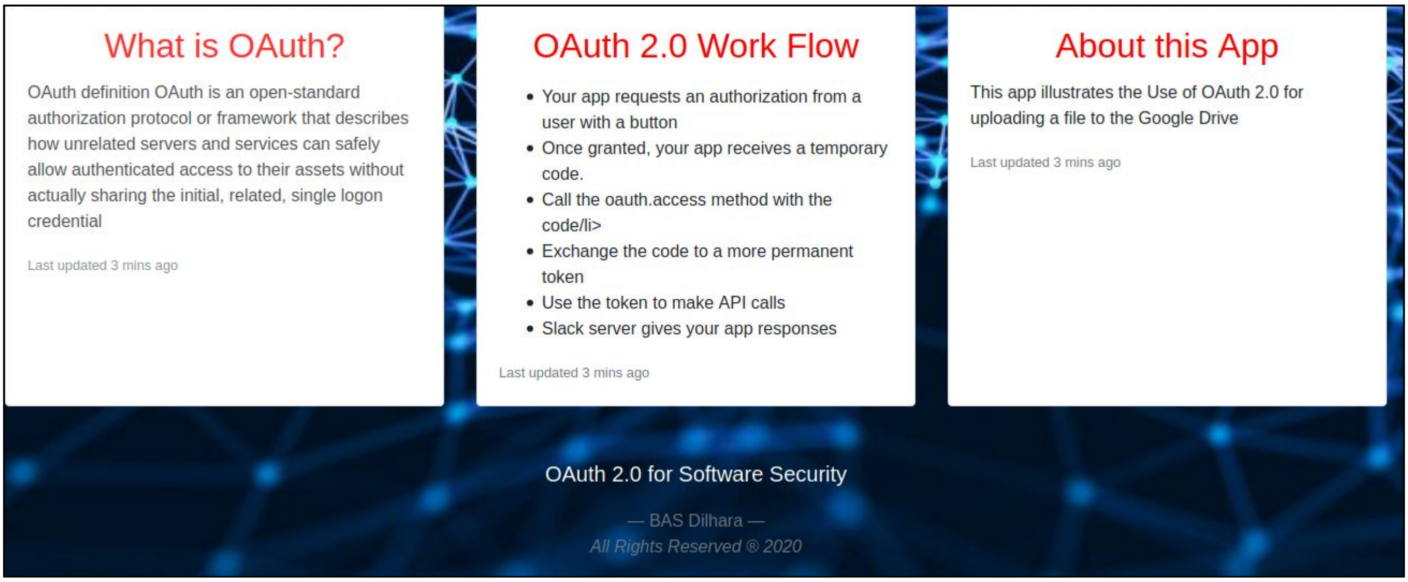
3 - Phase 1

Upon startup, the user is presented with the application context **index.html** that explains practical usage of OAuth 2.0. Informative sections are available namely, What is OAuth?, OAuth 2.0 Workflow and About this App along-with a descriptive animated GIF as well. A preview button has the functionality to provide the user with a thumbnail of the particular image/s that can be chosen for upload.



Software Security Web App

To Preview the image before Uploading



What is OAuth?

OAuth definition OAuth is an open-standard authorization protocol or framework that describes how unrelated servers and services can safely allow authenticated access to their assets without actually sharing the initial, related, single logon credential

Last updated 3 mins ago

OAuth 2.0 Work Flow

- Your app requests an authorization from a user with a button
- Once granted, your app receives a temporary code.
- Call the oauth.access method with the code/li>
- Exchange the code to a more permanent token
- Use the token to make API calls
- Slack server gives your app responses

Last updated 3 mins ago

About this App

This app illustrates the Use of OAuth 2.0 for uploading a file to the Google Drive

Last updated 3 mins ago

OAuth 2.0 for Software Security

— BAS Dilhara —

All Rights Reserved ® 2020

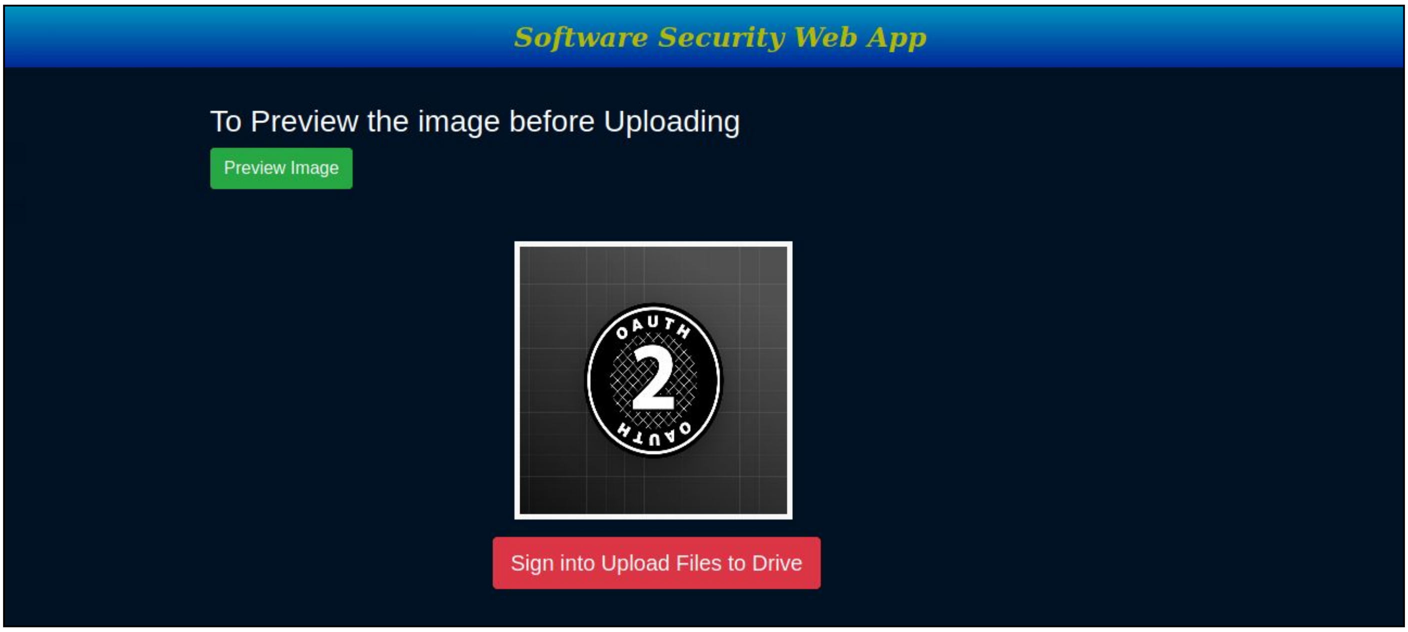
Figure 1: Application Interface

Code Review

```
<form id="upload-file" method="GET" enctype="multipart/form-data">
<h3>To Preview the image before Uploading</h4>
<label for="imageUpload" class="btn btn-success" id="slct">
Preview Image
</label>
<input type="file" name="file" id="imageUpload" accept=".png,.jpg,.jpeg"><br>
<br>
</form>
```

4 - Phase 2

On clicking preview, the traditional Windows File Explorer is loaded where any **compatible image file** can be located. Once the file is taken onto the application, **a snapshot view** is available with an additional button to **sign-in to the Google Drive** of the user if he/she wishes to follow-through with the upload process.



Software Security Web App

To Preview the image before Uploading

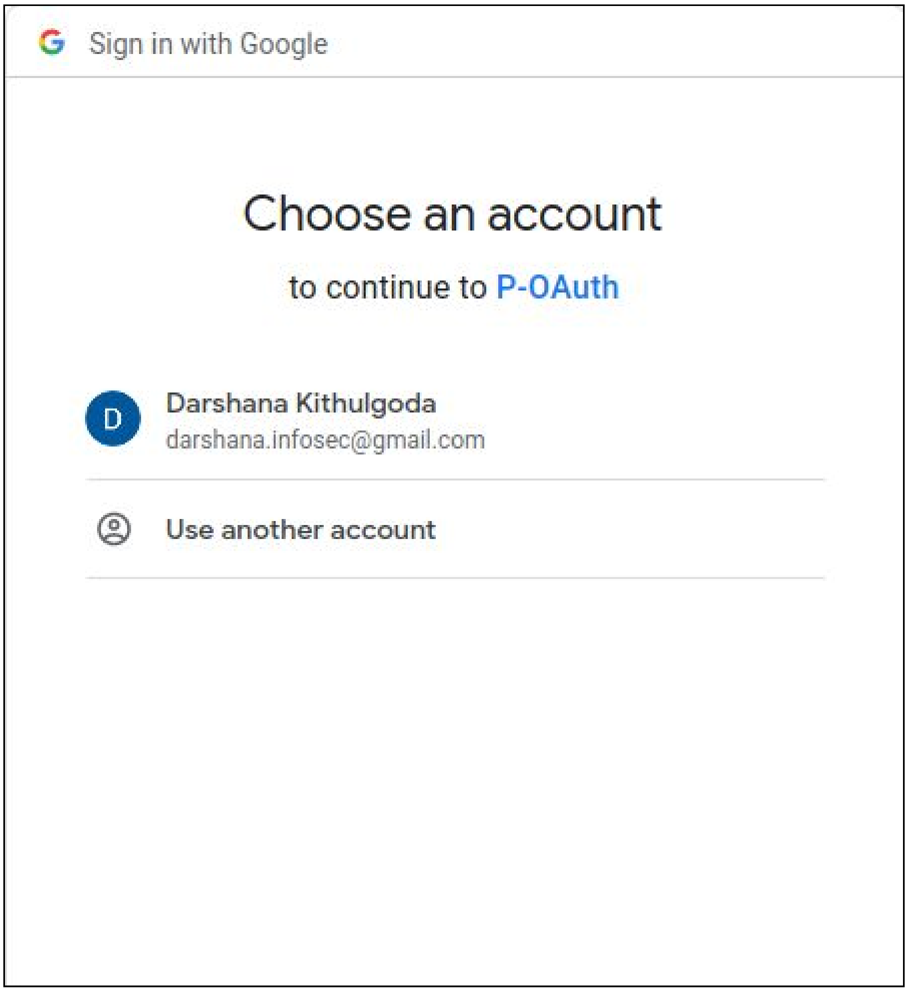
Figure 2: Image Preview

Code Review

```
<div class="image-section" style="display:none;">
<div class="img-preview">
<div id="imagePreview">
</div></div>
<div>
<button id="login" class="btn btn-danger btn-lg ">Sign into Upload Files to
Drive</button>
</div>
</div>
```

4 - Phase 3

It leads to the usual **Google login prompt** when the sign-in button is clicked where an account can be chosen or created on-point. This is where **OAuth functionality** has been enabled, which will be discussed in detail in following sections. **Selecting a Google account** to access the Drive here creates a warning by default which says '**This App is not verified**'. This is because the developer is not identified by Google as a trusted entity. A walk-around for this issue is to select '**Go to (Unsafe)**' in advanced options. **Access permissions to Google resources** should then be provided to the developer's app. To ensure fault-less operation of the application, it is best to choose '**See, edit, create and delete all of your Google Drive files.**' Once allowed, the user is taken back to the uploading function as mentioned before (**Redirect URL**) for further processing.



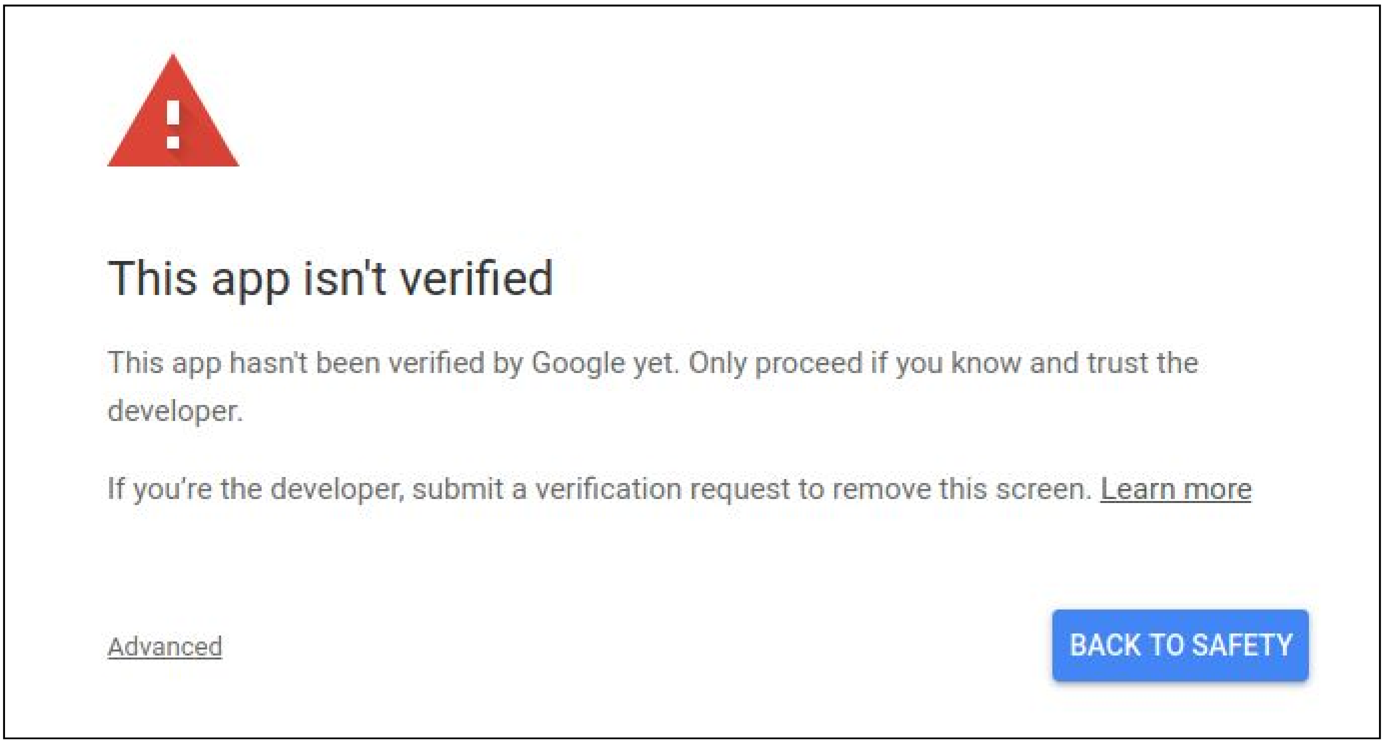
Sign in with Google

Choose an account

to continue to P-OAuth

Darshana Kithulgoda
darshana.infosec@gmail.com

Figure 3: Google login prompt

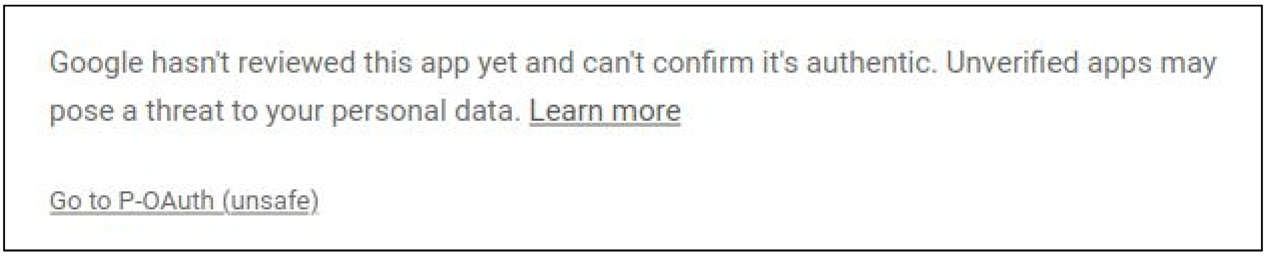


This app isn't verified

This app hasn't been verified by Google yet. Only proceed if you know and trust the developer.

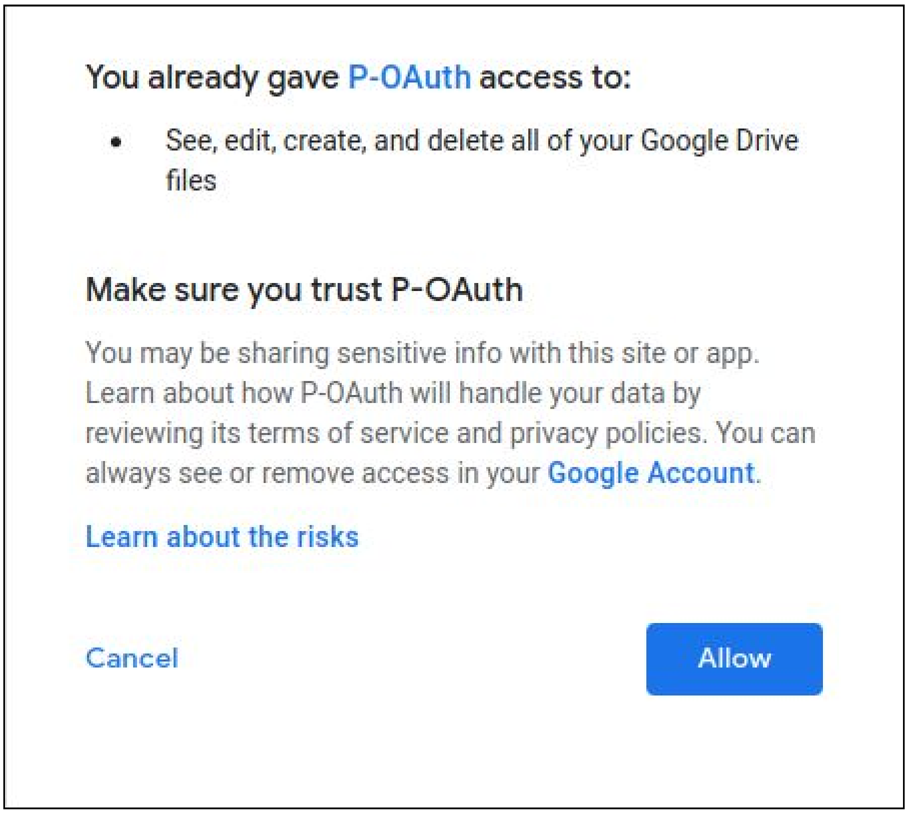
If you're the developer, submit a verification request to remove this screen. Learn more

Figure 4: Untrusted developer warning



Google hasn't reviewed this app yet and can't confirm it's authentic. Unverified apps may pose a threat to your personal data. Learn more

Figure 5: Go to (Unsafe)



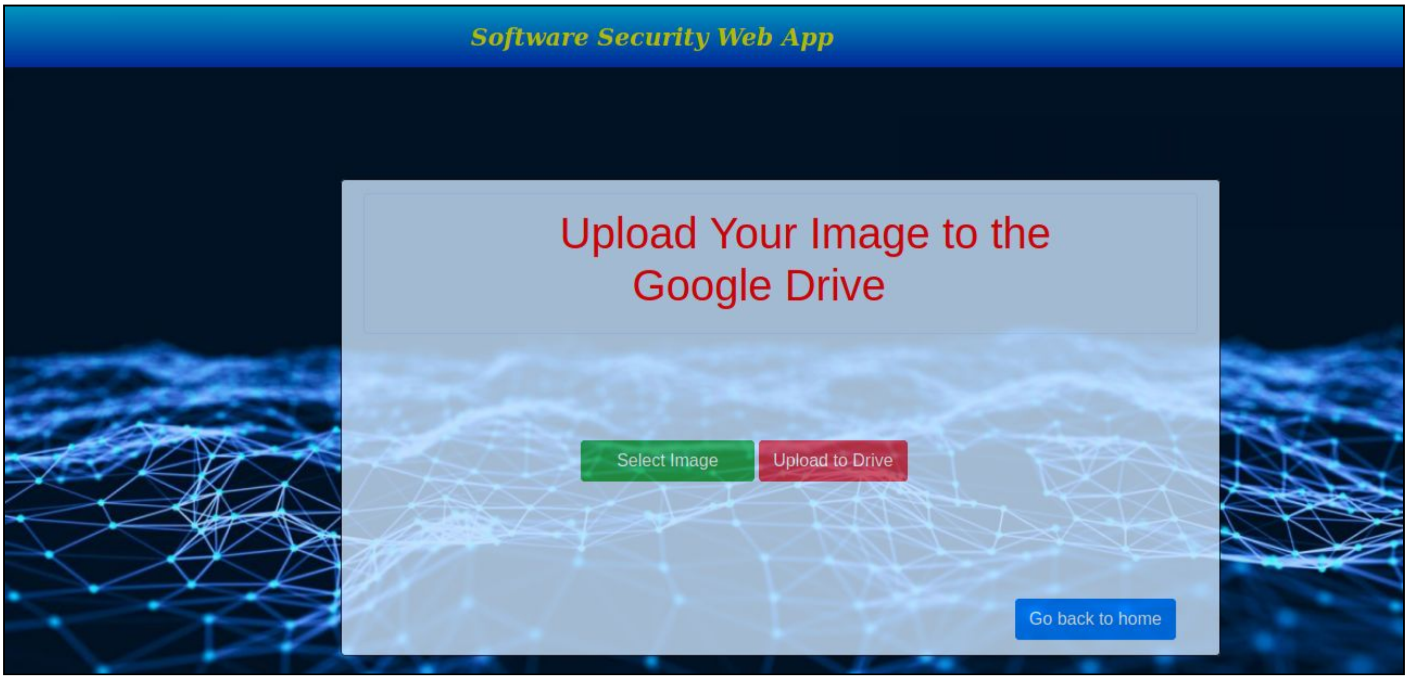
You already gave P-OAuth access to:

- See, edit, create, and delete all of your Google Drive files

Make sure you trust P-OAuth

You may be sharing sensitive info with this site or app. Learn about how P-OAuth will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Figure 6: Access permissions to Google resources


Software Security Web App

Upload Your Image to the Google Drive

Figure 6: Redirect URL

Code Review

mainpage.js is invoked when signing in. Given below are key components of it.

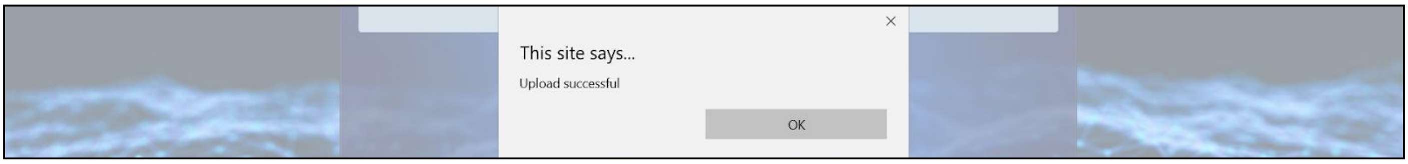
```
function googleLogin(){
let clientId =
"914067404437-qdq1e5jstp8bv5nbr4o98phftl3bgq7l.apps.googleusercontent.com";
let redirect_uri = "http://localhost/webapp/upload.html";
let scope = "https://www.googleapis.com/auth/drive";
let url = "";

$("#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;
}}
```

5 - Phase 4

Message flow is diverted into the **upload** app resource. The required image can be directly uploaded to the drive at this point and an alert notification mentioning that '**the upload was successful**' is given.


This site says...

Upload successful

Figure 7: Success Notification

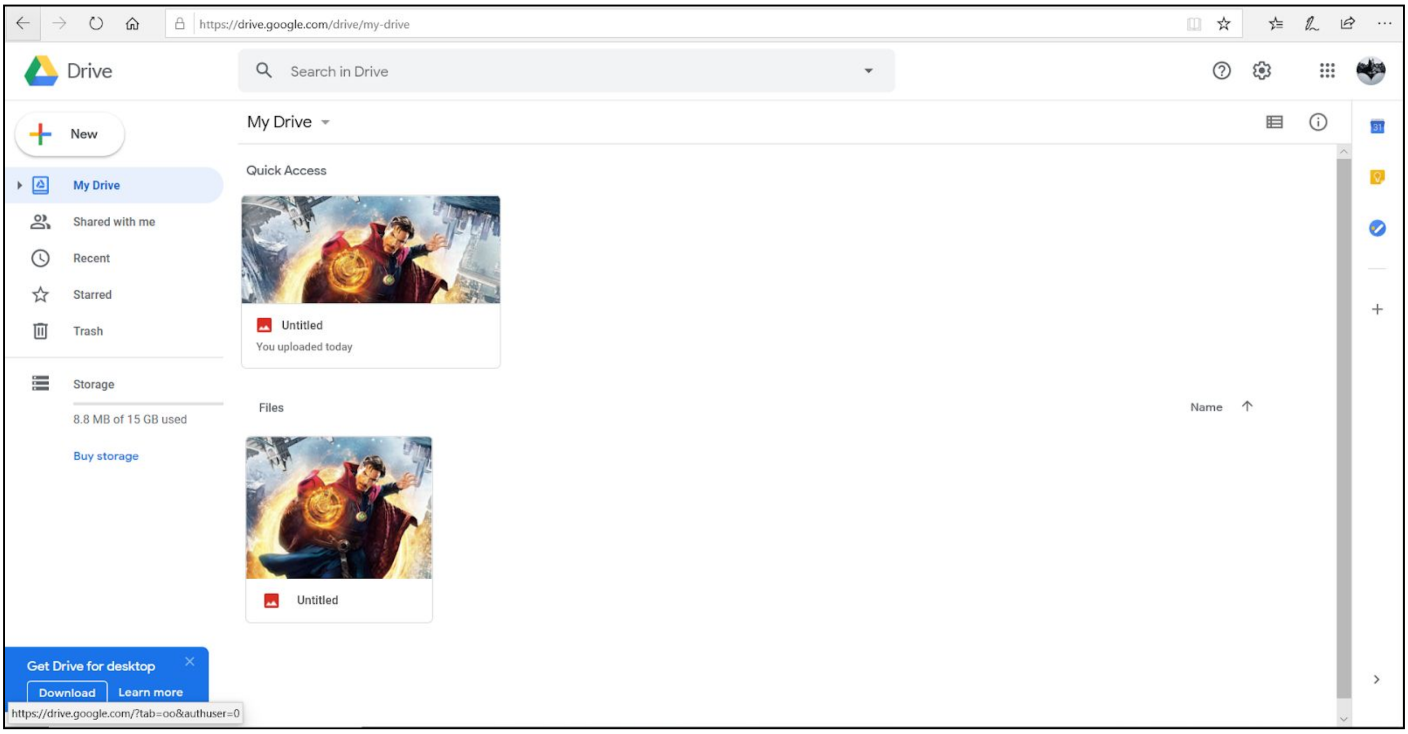


Figure 8: Image successfully uploaded to the My Drive section

Code Review

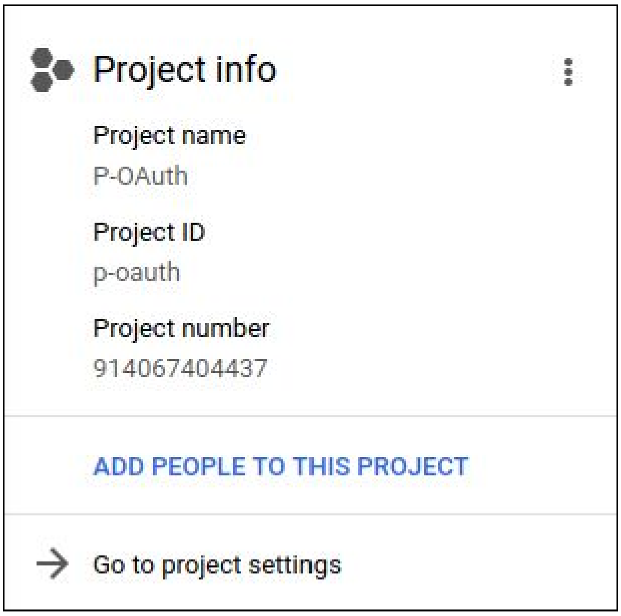
upload.js is invoked when transferring the image file to Google. The obtained access token is used for the whole process. Message flow is shown below.

```
$.ajax({
type: "POST",
beforeSend: function (request){
request.setRequestHeader("Authorization", "Bearer" + " " +
localStorage.getItem("accessToken"));
},
url: googleDriveApiUrl,
data:{
uploadType: "media"
},
success: function (data){
console.log(data);
alert("Upload successful");
},
error: function (error){
console.log(error);
},
async: true,
data: formData,
cache: false,
contentType: false,
processData: false,
timeout: 60000
});

$("#upload").on("click", function (e){
let file = $("#files")[0].files[0];
let upload = new Upload(file);
upload.doUpload();
});
```

6 - Obtaining developer credentials from Google APIs & Services

The **Client ID and Client Secret credentials** are mandatory for OAuth implementation with the Drive API. In https://console.developers.google.com, a web application project should be created first.



Project info

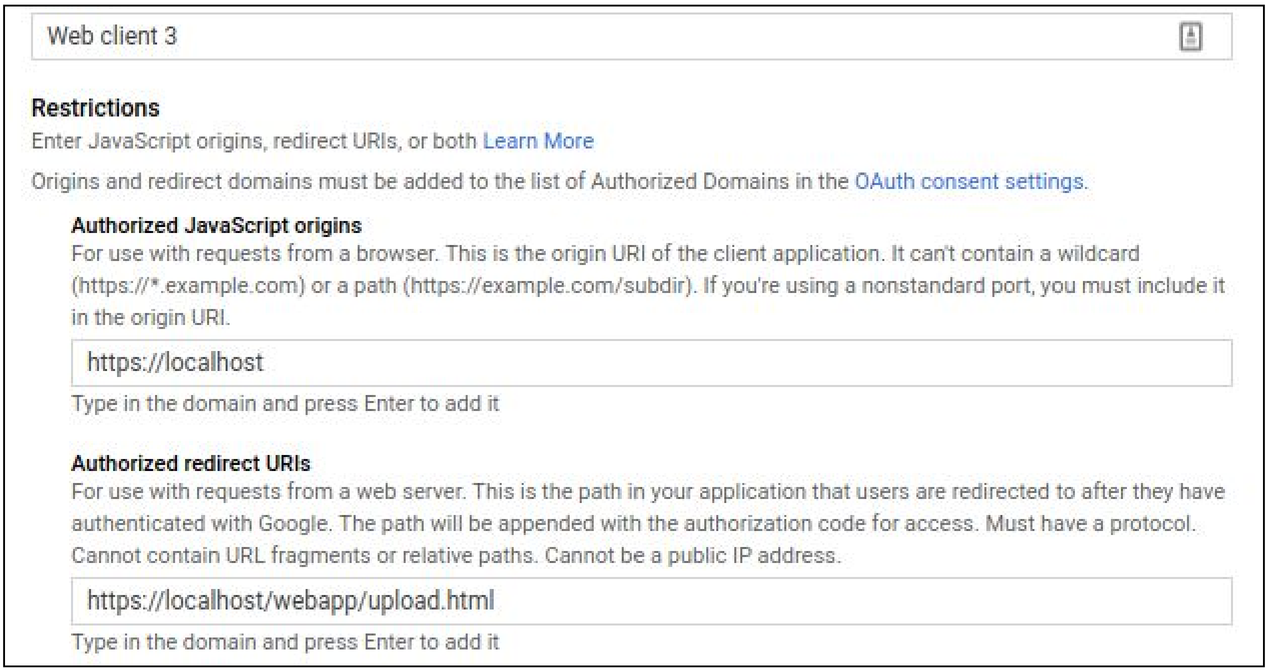
Project name
P-OAuth

Project ID
p-oauth

Project number
914067404437

Figure 9: Web App Project on Google Developer Console

Credentials can be created under the APIs and Services tab. Under ‘Create OAuth client ID’, the application type has to be chosen. ‘Authorized JavaScript origins’ is used to indicate the Origin URI along-with a port if specified. ‘Authorized redirect URIs’ must contain the application path that users are redirected to after they have passed Google authentication.



Web client 3

Restrictions
Enter JavaScript origins, redirect URIs, or both Learn More
Origins and redirect domains must be added to the list of Authorized Domains in the OAuth consent settings.

Authorized JavaScript origins
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

https://localhost

Type in the domain and press Enter to add it

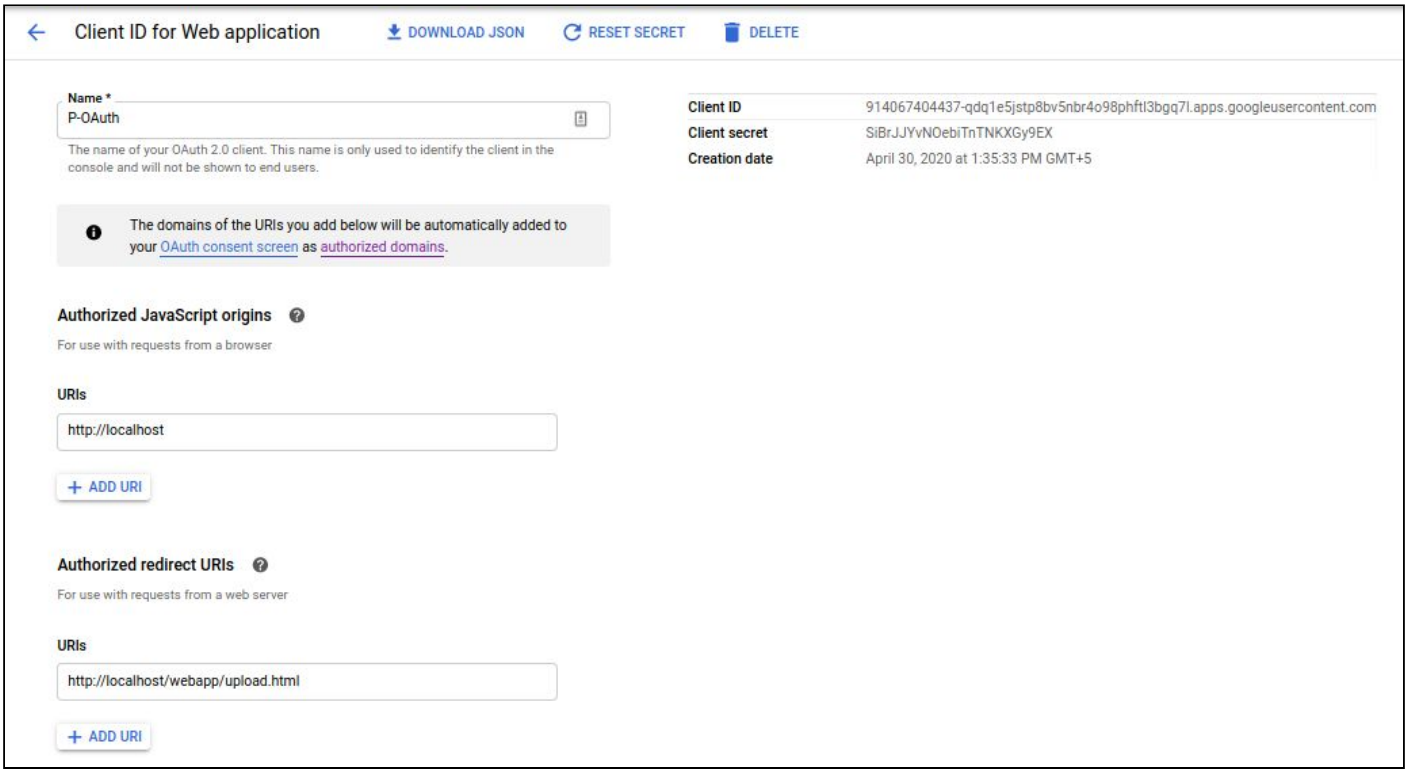
Authorized redirect URIs
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

https://localhost/webapp/upload.html

Type in the domain and press Enter to add it

Figure 10: Creating OAuth Credentials

OAuth 2.0 client name is used to identify the client within the console only. Multiple **URI**s can be added to the **OAuth consent screen** as **authorized domains**. JavaScript Origin URIs are for requests from a browser. In previous explanations as well, the redirect URIs are for requests from a web server.



Name *
P-OAuth
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.

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

URIs
http://localhost

Authorized redirect URIs
For use with requests from a web server

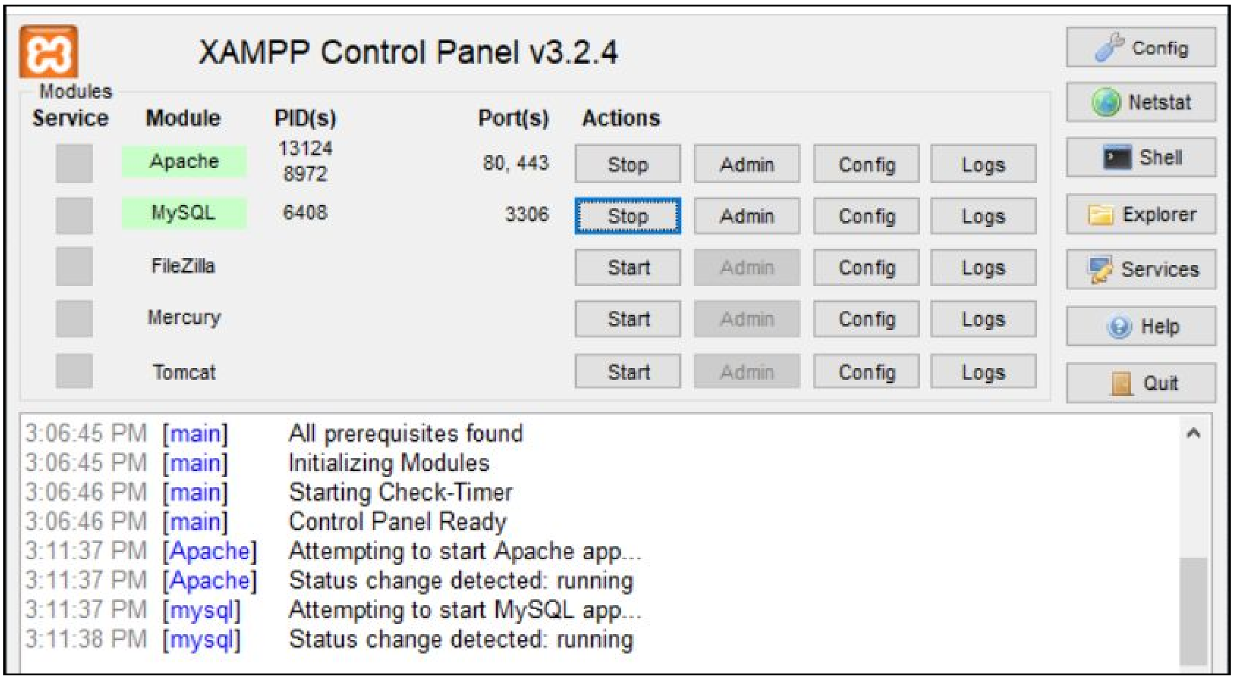
URIs
http://localhost/webapp/upload.html

Client ID	914067404437-qdq1e5jstp8bv5nbr4o98phftl3bgq7l.apps.googleusercontent.com
Client secret	SiBrJJYvNOebiTnTNKXGy9EX
Creation date	April 30, 2020 at 1:35:33 PM GMT+5

Figure 11: OAuth 2.0 Credentials

7 - Deployment

The project 'Software Security Web Application' can simply be loaded onto localhost with **XAMPP** Server after initializing **Apache** and **MySQL** services on available ports. All artifacts must be placed in the htdocs folder of XAMPP (**images, index.html, mainpage.css, mainpage.js, upload.css, upload.html, upload.js**).



Service	Module	PID(s)	Port(s)	Actions			
	Apache	13124 8972	80, 443	Stop	Admin	Config	Logs
	MySQL	6408	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
3:06:45 PM  [main]      All prerequisites found
3:06:45 PM  [main]      Initializing Modules
3:06:46 PM  [main]      Starting Check-Timer
3:06:46 PM  [main]      Control Panel Ready
3:11:37 PM  [Apache]    Attempting to start Apache app...
3:11:37 PM  [Apache]    Status change detected: running
3:11:37 PM  [mysql]     Attempting to start MySQL app...
3:11:38 PM  [mysql]     Status change detected: running
```

Figure 12: Launching the app with XAMPP

APPENDIX

index.html

```
<!DOCTYPE html>
<html>
<head>
	<meta charset="UTF-8">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<meta http-equiv="X-UA-Compatible" content="ie=edge">
	<title>Software Security Project</title>
	<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
	<script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
	<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
	<script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
	<link rel="stylesheet" type="text/css" href="mainpage.css">
	<script src="mainpage.js"></script>
</head>
<body>
	<nav class="navbar navbar-light" style="background-color: #e3f5fd;">
	<div class="container">
		<a class="navbar-brand" href="#" id="asd">Software Security Web App</a>
	</div>
	</nav>
	<div class="container">
	<div id="content" style="margin-top:2em">
	<div>
	<form id="upload-file" method="GET" enctype="multipart/form-data">
	<h3>To Preview the image before Uploading</h4>
	<label for="imageUpload" class="btn btn-success" id="slct">
		Preview Image
	</label>
	<input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg"><br><br>
	</form>
	<div class="image-section" style="display:none;">
	<div class="img-preview">
		<div id="imagePreview">
		</div>
	</div>
	<div>
	<button id="login" class="btn btn-danger btn-lg ">Sign into Upload Files to
Drive</button>
	</div>
	</div>
	<div class="loader" style="display:none;"></div>
	<h3 id="result">
	<span> </span>
	</h3>
	</div>
<br><br>
</div>
</div>
<div class="card-deck">
<div class="card" >
```

```
<img src="images/img1.jpg" class="card-img-top" alt="..." height="400px" width="10px">
<div class="card-body" id="sd">
<h5 class="card-title">What is OAuth?</h5>
<p class="card-text">OAuth definition OAuth is an open-standard authorization protocol or 
framework that describes how unrelated servers and services can safely allow 
authenticated access to their assets without actually sharing the initial, related, 
single logon credential</p>
<p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>
</div>
</div>
<div class="card" >
<img src="images/img2.gif" class="card-img-top" alt="..." height="400px" width="10px">
<div class="card-body">
      <h5 class="card-title">OAuth 2.0 Work Flow </h5>
      <p class="card-text"><ul><li>Your app requests an authorization from a user with a 
button</li><li>Once granted, your app receives a temporary code.</li>
      <li>Call the oauth.access method with the code/li><li>Exchange the code to a more 
permanent token</li><li>Use the token to make API calls</li><li>Slack server gives your 
app responses</li></ul></p>
      <p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>
      </div>
  </div>
<div class="card" >
      <img src="images/img4.jpg" class="card-img-top" alt="..." height="400px" 
width="10px">
      <div class="card-body" >
      <h5 class="card-title">About this App</h5>
      <p class="card-text">This app illustrates the Use of OAuth 2.0 for uploading a file 
to the Google Drive</p>
      <p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>
      </div>
  </div>
</div>
</div>
<br><br>
<blockquote class="blockquote text-center">
  <p style="color:white;">OAuth 2.0 for Software Security</p>
  <footer class="blockquote-footer">BAS Dilhara &mdash;<br><cite title="Source Title">
All Rights Reserved &reg; 2020   </cite></footer>
</blockquote>
</body>
</html>
```

mainpage.js

```
// Image Preview
$(document).ready(function () {
      googleLogin();
$('.image-section').hide();
      let imagePreview = $('#imagePreview');
      let result = $('#result');
$('.loader').hide();
      result.hide();
function readURL(input) {
      if (input.files && input.files[0]) {
            let reader = new FileReader();
            reader.onload = function (e) {
            imagePreview.css('background-image', 'url(' + e.target.result + ')');
            imagePreview.hide();
            imagePreview.fadeIn(650);
            };
            reader.readAsDataURL(input.files[0]);
      }
      }
      $("#imageUpload").change(function () {
      $('.image-section').show();
      $('#btn-predict').show();
      result.text('');
      result.hide();
      readURL(this);
      });
});
function googleLogin() {
      let clientId =
"914067404437-qdq1e5jstp8bv5nbr4o98phftl3bgq7l.apps.googleusercontent.com";
      let redirect_uri = "http://localhost/webapp/upload.html";
      let scope = "https://www.googleapis.com/auth/drive";
      let 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;
      }
}
```

upload.html

```
<!DOCTYPE html>
<html>
<head>
      <meta charset="utf-8" />
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <title>Software Security project</title>
      <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
      <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
      <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
      <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
      <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>
</head>
<body>
 <nav class="navbar navbar-light" style="background-color: #e3f5fd;">
      <div class="container">
            <a class="navbar-brand" href="#" id="asd">Software Security Web App</a>
      </div>
      </nav>
   <div id="outer" class="alert alert-primary">
       <div id="middle" class="alert alert-primary">
      <h1>Upload Your Image to the <br>&nbsp; &nbsp;&nbsp;&nbsp;&nbsp;Google Drive</h1>
       </div>
      <br><nr>
      <div id="inner">
      <label for="files" class="btn btn-success" id="slct">
            Select Image
      </label>
      <input class="btn btn-success" id="files" type="file" name="files[]" multiple/>
      <button id="upload" class="btn btn-danger" >Upload to Drive</button>
   </div>
<div id='home'>
<button  class="btn btn-primary" onclick=”window.location.href='/webapp/index.html'">Go
back to home</button>
      </div>
   </div>
   </div>
</body>
```

upload.js

```
$(document).ready(function () {
      const urlParameters = new URLSearchParams(window.location.search);
      const code = urlParameters.get('code');
      const RedirectUri = "http://localhost/webapp/upload.html";
      const ClientSecret = "SiBrJJYvNOebiTnTNKXGy9EX";
      const SCOPE = "https://www.googleapis.com/auth/drive";
      let access_token = "";
      let ClientId =
"914067404437-qdq1e5jstp8bv5nbr4o98phftl3bgq7l.apps.googleusercontent.com";
      let googleAuthApiUrl = "https://www.googleapis.com/oauth2/v4/token";
      let googleDriveApiUrl = "https://www.googleapis.com/upload/drive/v2/files";
      $.ajax({
      type: 'POST',
      url: googleAuthApiUrl,
      data: {
            code: code
            , redirect_uri: RedirectUri,
            client_secret: ClientSecret,
            client_id: ClientId,
            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, "/SoftwareSecurityApp/" +
"upload.html");
      }
      });
      let 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 () {
      let that = this;
      let formData = new FormData();
      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: googleDriveApiUrl,
            data: {
            uploadType: "media"
            },
            success: function (data) {
            console.log(data);
            alert("Upload successful");
            },
            error: function (error) {
            console.log(error);
            },
            async: true,
            data: formData,
            cache: false,
            contentType: false,
            processData: false,
            timeout: 60000
      });
      };
      $("#upload").on("click", function (e) {
      let file = $("#files")[0].files[0];
      let upload = new Upload(file);
      upload.doUpload();
      });
});
```

mainpage.css

```
body {
      background-image:url('images/back.jpg');
      background-size: cover;
}
```

```
H3 {
      margin-left:100px;
      color: white;
}

#slct {
      margin-left:100px;
      color: white;
}

a {
      margin-left:380px;
}

#asd {
      color: yellow ;
      font: italic bold 25px/30px Georgia, serif;
}

Label {
      margin-left:-160px;
}

.image-section>.img-preview {
      margin-left:380px;
}

Button {
      margin-left:420px;
}

Nav {
      background: linear-gradient(to bottom, #00ccff 0%, #0033cc 100%);
      opacity:0.7;
}

#login {
      margin-left: 360px;
}

H5 {
      color:red;
      font-style: bold;
      font-size: 2em;
      text-align: center;
}

.img-preview {
      width: 256px;
      height: 256px;
      position: relative;
      border: 5px solid #F8F8F8;
      box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
      margin-top: 1em;
      margin-bottom: 1em;
}
```

```
.img-preview>div {
	width: 100%;
	height: 100%;
	background-size: 256px 256px;
	background-repeat: no-repeat;
	background-position: center;
}

input[type="file"] {
	display: none;
}

.upload-label {
	display: inline-block;
	padding: 12px 30px;
	background: #39D2B4;
	color: #fff;
	font-size: 1em;
	transition: all .4s;
	cursor: pointer;
}

.upload-label:hover {
	background: #34495E;
	color: #39D2B4;
}

.loader {
	border: 8px solid #f3f3f3; /* Light grey */
	border-top: 8px solid #3498db; /* Blue */
	border-radius: 50%;
	width: 50px;
	height: 50px;
	animation: spin 1s linear infinite;
}

@keyframes spin {
	0% { transform: rotate(0deg); }
	100% { transform: rotate(360deg); }
}

#carouselExampleControls {
	width:400px;
	height:40px;
}

#sd {
	opacity: 0.8;
}
```

upload.css

```
body {
	background-image: url("images/back.jpg");
	background-size: cover;
}
```

```
#outer {
      border: 1PX solid black;
      width: 810px;
      height: 440px;
      margin-left: 24%;
      margin-top: 8%;
      background-image: url("images/ai.JFIF");
      background-size: cover;
      opacity:0.8;
}

input[type="file"] {
      display: none;
}

#slct {
      margin-top: 8px;
      width:160px;
      margin-left: 200px;
}

H1 {
      margin-left: 160px;
      vertical-align: middle;
      color: red;
}

nav {
      background: linear-gradient(to bottom, #00ccff 0%, #0033cc 100%);
      opacity:0.7;
}

#asd {
      color: yellow ;
      margin-left: 340px;
      font: italic bold 22px/30px Georgia, serif;
}

Label {
      margin-left: 100px;
      width: 300px;
}

#inner {
      margin-top: 50px;
}

#home {
      margin-left: 600px; margin-top: 100px; }
```

REFERENCES

[1] - Google Developers. 2020. Google Drive API Video Library | Google Developers. [online] Available at: <https://developers.google.com/drive/api/v3/videos> [Accessed 11 May 2020].