

Task Accomplishment: Integrating Google Sign-In with OTP Generation Using PHP

Objective: The goal is to create a PHP web application that allows user to sign in using their Google account and subsequently generates and displays one-time password (OTP) after successful authentication.

Tools and Technology used:

- PHP: Server-side scripting language used for backend logic and session management.
- Google API Client Library for PHP: Used for handling Google OAuth2 authentication.
- Composer: Dependency manager for PHP used to install and manage libraries, including the Google API Client.
- Text Editor/IDE: Used for writing and editing PHP scripts.
- Web Server: Local or remote server setup (e.g., Apache, Nginx) for hosting and testing PHP scripts.

Setting Up Google OAuth Credentials:

Created OAuth credentials on the Google Developers Console.

Configured authorized redirect URIs for local testing and production environments.

Downloaded the credentials JSON file (`credentials.json`) containing OAuth client ID and client secret.

Installing Google API Client Library:

Installed the Google API Client Library for PHP using Composer: `composer require google/apiclient:^2.0`

Creating PHP Web Application Components:

Config.php

Contains configuration settings such as `client_id`, `client_secret`, `redirect_uri`, and `scopes` required for Google OAuth.

ErrorHandle.php

Handles error messages and redirects users to the appropriate error page (`ErrorHandler.php`) with specific error messages based on session errors (`$_SESSION['error']`).

Index.php

Displays a link to initiate Google OAuth authentication (`$authUrl`) using Google's PHP client library.

login.php

- Handles the callback after a user authorizes the application via Google OAuth.
- Fetches the access token using the authorization code.

- Retrieves user information (email and name) from Google.
- Generates a 6-digit OTP and stores it in the session with a 30-second expiration.
- Displays user information and the OTP.
- Redirects to `ErrorHandler.php` in case of any exceptions during Google OAuth or OTP generation.

Logout.php

- Clears all session variables and destroys the session.
- Redirects the user back to the login page (`index.php`).

otp.php:

independently generates an OTP (though somewhat redundant with `login.php`).

Retrieves the OTP from the session if available and valid.

Displays the OTP and handles expiration logic.

Provides a logout link to end the session.