

Report: Development of Login and Registration Feature using Auth0 and Microsoft Identity

1. Introduction:

The purpose of this report is to document the development process and outcomes of implementing a login and registration feature in a web application using Auth0 for authentication and Microsoft Identity for login. This project aimed to create a proof of concept (POC) demonstrating the integration of these technologies to provide a secure and seamless authentication experience for users.

2. Background:

Auth0 is a leading identity and access management platform that provides secure authentication and authorization solutions for web and mobile applications. Microsoft Identity, on the other hand, is Microsoft's identity platform that allows users to sign in to applications using their Microsoft accounts. Integrating Auth0 with Microsoft Identity offers a robust authentication solution for web applications.

3. Overview of Auth0:

Auth0 provides a comprehensive set of features for identity management, including authentication, authorization, and user management. It supports various authentication protocols such as OAuth 2.0, OpenID Connect, and SAML, making it versatile and adaptable to different use cases. Auth0 also offers customization and extensibility options, allowing developers to tailor authentication flows to their specific requirements.

4. Overview of Microsoft Identity:

Microsoft Identity is Microsoft's cloud-based identity platform that enables developers to add authentication and authorization to their applications using Microsoft accounts, Azure Active Directory (Azure AD), or other identity providers. It supports industry-standard protocols like OAuth 2.0 and OpenID Connect, making it interoperable with a wide range of applications and services. Microsoft Identity offers seamless integration with Azure services, providing a unified identity and access management solution for Microsoft ecosystem.

5.Features of Microsoft Identity:

Features

The key attributes of Microsoft Identity comprise:

→Azure Active Directory (Azure AD):

Microsoft Identity serves as a cornerstone in the domain of identity and access management, notably due to its deep integration with Azure Active Directory (Azure AD). Beyond providing a comprehensive authentication and authorization framework, this synergy extends throughout various Microsoft services and third-party applications. This comprehensive integration ensures not only a secure access gateway but also a cohesive identity solution that harmonizes user experiences across diverse platforms.

→Seamless Sign-On (SSO)

Microsoft Identity integrates Seamless Sign-On (SSO) capabilities, allowing users to authenticate once and access multiple applications without the need to re-enter credentials. This streamlines user experience and reduces the burden of managing multiple sets of login information.

→Multi-layered Authentication (MLA)

Emphasizing a proactive approach to security, Microsoft Identity employs Multi-layered Authentication (MLA) to strengthen user accounts against unauthorized access and potential security threats. Going beyond traditional password-centric security, MLA incorporates additional layers of authentication, enhancing the defensive measures. This layered approach ensures a resilient security posture, safeguarding sensitive data and user identities with a multi-faceted security shield.

→Context-aware Access

Microsoft Identity introduces Context-aware Access, a feature empowering organization to define access policies based on specific contextual cues. This enables granular control over user access, considering factors such as device posture, geographical location, and user roles. With Context-aware Access, organizations can ensure that the right individuals have the appropriate access to resources under the right circumstances, thereby enhancing security and compliance.

5.1 Pros of Microsoft Identity:

→Seamless Integration with Microsoft Ecosystem:

Microsoft Identity is deeply integrated with various Microsoft products and services, such as Azure Active Directory (Azure AD), Microsoft 365, Azure services, and more. This seamless integration simplifies identity management for organizations already using Microsoft technologies, providing a cohesive experience across the ecosystem.

→Comprehensive Authentication and Authorization:

Microsoft Identity provides robust authentication and authorization capabilities, including Single Sign-On (SSO), Multi-factor Authentication (MFA), Conditional Access, and Identity Protection. This ensures secure access to resources while offering flexibility and granular control over user permissions.

→Scalability and Reliability:

Being part of the Microsoft cloud infrastructure, Microsoft Identity benefits from the scalability and reliability of Azure services. It can effortlessly handle large user bases and fluctuating workloads, ensuring high availability and performance for identity services.

→Unified Identity Management:

Microsoft Identity offers a unified platform for managing identities across cloud and on-premises environments. With Azure AD Connect, organizations can synchronize identities from Active Directory to Azure AD, enabling centralized identity management and seamless access to cloud resources.

→Development Environment:

Microsoft Identity provides robust developer tools, SDKs, and APIs for integrating identity services into custom applications and workflows. Developers can leverage Microsoft Identity Platform to build secure and scalable identity solutions while adhering to industry standards and best practices.

6)Features of AuthO:

Features

The key attributes of AuthO comprise:

→Universal Authentication

Auth0 provides a customizable Universal Authentication experience, enabling users to access applications using diverse identity providers, including social media platforms (e.g., Google, Facebook), enterprise authentication systems (e.g., Active Directory, LDAP), or traditional username/password combinations.

→Unified Sign-On (SSO):

Auth0 facilitates Unified Sign-On across multiple applications, permitting users to authenticate once and seamlessly access different services without repetitive logins. This streamlines user interaction and minimizes authentication hurdles.

→Multi-layered Security (MLS):

Auth0 offers Multi-layered Security, enhancing user account protection through additional verification steps beyond passwords. This includes options such as SMS verification, email verification, authenticator apps, or hardware tokens, fortifying authentication processes.

→Tailored Authentication Workflows:

Auth0 provides flexible customization for Authentication Workflows, empowering developers to adapt registration, login, and password reset processes to meet specific application needs. This ensures consistency with branding and user experience requirements.

→Identity Federation:

Auth0 supports Identity Federation, enabling smooth integration with various identity providers and protocols like OAuth, OpenID Connect, SAML, among others. This simplifies third-party identity provider integration into applications, streamlining authentication mechanisms.

6.1)Pros of AuthO:

→Extensive Integration Options:

Auth0 supports a wide range of identity protocols and standards, including OAuth, OpenID Connect, SAML, LDAP, and more. This enables seamless integration with various identity providers, applications, and platforms, offering flexibility in authentication and authorization mechanisms.

→Customizable and Brandable Authentication Experience:

Auth0 provides a highly customizable authentication experience, allowing developers to tailor login screens, signup flows, and password reset processes to match the branding and user experience requirements of their applications. This ensures a consistent and branded authentication experience for end-users.

→Single Sign-On (SSO) Across Multiple Applications:

Auth0 facilitates Single Sign-On (SSO) across multiple applications, enabling users to authenticate once and access different services without the need for repeated logins. This enhances user convenience, improves productivity, and reduces authentication friction.

→Developer-Friendly Tools and APIs:

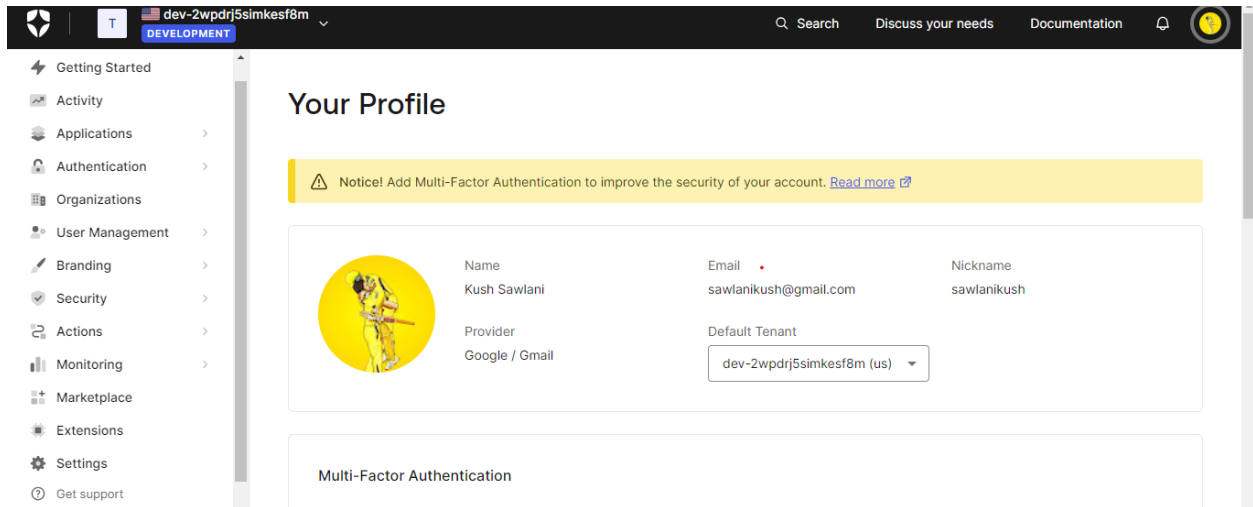
Auth0 offers a comprehensive set of developer tools, SDKs, and APIs, making it easy to integrate authentication and authorization features into applications. Developers can leverage pre-built components, libraries, and documentation to expedite development and ensure security best practices.

7) Setup and Configuration:

→ Configure Auth0:

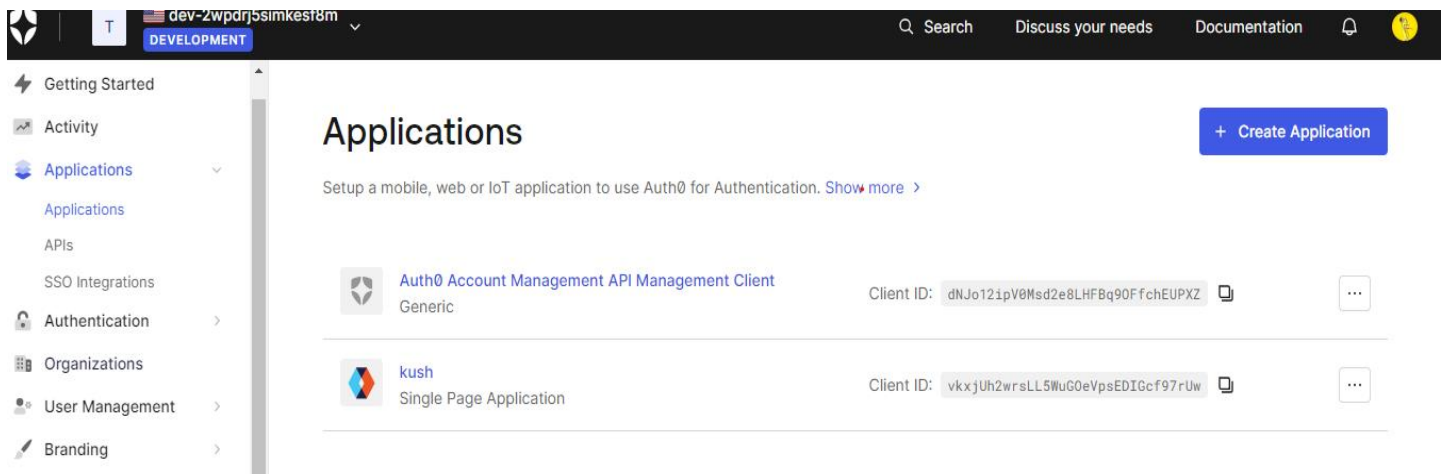
1. Create an Auth0 Account

- Visit the Auth0 website and sign up for a new account (<https://auth0.com>)
- Once logged in, navigate to the Auth0 Dashboard.



2. Create Auth0 Application

- Sign up for a free Auth0 account or log in.
- Use the Auth0 Dashboard to create a new Auth0 application or select an existing one.
- Note the alphanumeric client ID for your application.



3. Configure Callback URLs


- Set the callback URL to `http://localhost:5173/` in the Auth0 Dashboard.
- Created Through React Vite so Local Host: to `http://localhost:5173/`.
- Configure logout URLs to `http://localhost:5173/`.
- Define Allowed Web Origins as `http://localhost:5173/`
- Create a new react application using a command

Application URIs	Application Login URI
	<input type="text" value="https://myapp.org/login"/>
	<p>In some scenarios, Auth0 will need to redirect to your application's login page. This URI needs to point to a route in your application that should redirect to your tenant's <code>/authorize</code> endpoint. Learn more</p>
	Allowed Callback URLs
	<input type="text" value="http://localhost:5173/"/>
	<p>After the user authenticates we will only call back to any of these URLs. You can specify multiple valid URLs by comma-separating them (typically to handle different environments like QA or testing). Make sure to specify the protocol (<code>https://</code>) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol <code>https://</code> . You can use Organization URL parameters in these URLs.</p>
	Allowed Logout URLs
	<input type="text" value="http://localhost:5173/"/>
	<p>Comma-separated list of allowed logout URLs for redirecting users post-logout. You can use wildcards at the subdomain level (<code>*.google.com</code>). Query strings and hash information are not taken into account when validating these URLs. Learn more about logout</p>

4. Configure Auth0Provider Component:

- Set properties in the Auth0Provider component:
- Domain: Auth0 tenant domain.
- ClientID: Client ID from Auth0 Dashboard.
- Authorization Params .redirect_url : http://localhost:5173/.

← Back to Applications

**kush**
Single Page Application Client ID vkxjUh2wrsLL5WuG0eVpsEDIGcf97rUw

Quickstart **Settings** Addons Connections Organizations

Basic Information

Name *

Domain

Client ID

Client Secret

The Client Secret is not base64 encoded.

→Add Login and Logout in Web Code:

- Create a file App.jsx and use the login With Redirect () method to enable login.
- Update main.jsx to include the new login button.
- Verify successful redirection to Auth0 Universal Login page and user login/signup.

App.jsx:

```
import React, { useState } from 'react';
import { useAuth0 } from "@auth0/auth0-react";
import './App.css';

function App() {
  const { loginWithRedirect, logout, user, isAuthenticated, isLoading } =
useAuth0();
  const [isLoggedIn, setIsLoggedIn] = useState(false); // State to track user's
login status
  const [userName, setUserName] = useState('John Doe'); // State to store user's
name

  const handleLogin = () => {
    // Simulate login functionality
    loginWithRedirect()
  };

  const handleLogout = () => {
    // Simulate logout functionality
    logout({ logoutParams: { returnTo: window.location.origin } })
  };

  return (
    <div className="App">
      <nav className="navbar">

        <div className="container">
          <a href="#" className="logo">My Webpage</a>
          <div className="navbar-menu">
            <ul className="menu-items">
              <li><a href="#">Home</a></li>
              <li><a href="#">About</a></li>
              <li><a href="#">Services</a></li>
              <li><a href="#">Contact</a></li>
            </ul>
          </div>
          <div className="user-info">
            {isAuthenticated ? (

              // Display user info if logged in
```

```

        <div className="user-info-logged-in">
          {console.log(user)}
          <span className="user-name">Hi, {user.name}</span>
          <img src={user.picture} alt="Profile" className="profile-icon" />
          <button className="btn" onClick={handleLogout}>Logout</button>
        </div>
      ) : (
        // Display login button if not logged in
        <button className="btn" onClick={handleLogin}>Login</button>
      )}
    </div>
  </div>
  </nav>
  { /* <ImageCarousel className="carousel-container"/> */ }
</div>
);
}

export default App;

```

main.jsx:

```

import React from 'react';
import { createRoot } from 'react-dom/client';
import { Auth0Provider } from '@auth0/auth0-react';
import App from './App';

const root = createRoot(document.getElementById('root'));

root.render(
  <Auth0Provider
    domain="dev-2wpdrj5simkesf8m.us.auth0.com"
    clientId="vkxjUh2wrsLL5WuG0eVpsEDIGcf97rUw"
    authorizationParams={{
      redirect_uri: window.location.origin
    }}
  >
    <App />
  </Auth0Provider>,
);

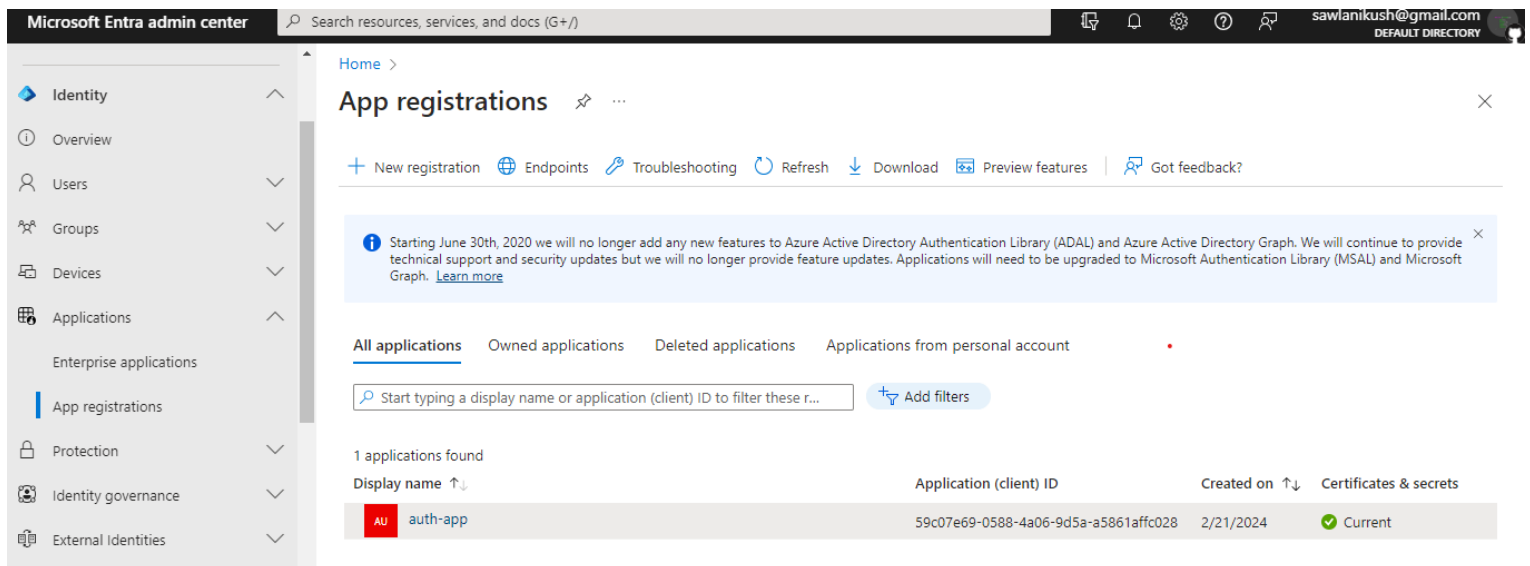
```

7.1) Setup and Configure Microsoft Identity:

Microsoft Identity Platform, integrated with Azure Active Directory, provides a secure and scalable identity and access management solution. Here is a step-by-step guide on setting up Microsoft Identity for our project.

→ Azure Portal Setup

- Navigate to the Azure portal and sign in with your Azure account.
- In the Azure portal, select "Azure Active Directory" from the left-hand navigation.
- Choose "App registrations" and create a new application registration for your web application.
- Note down the Application (client) ID and Directory (tenant) ID.



The screenshot displays the Microsoft Entra admin center interface. The left-hand navigation pane shows the 'Identity' section expanded, with 'App registrations' selected. The main content area is titled 'App registrations' and includes a search bar and a list of applications. A notification banner at the top states: 'Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)'. Below this, the 'All applications' tab is active, showing a table with one application: 'auth-app'.

Display name	Application (client) ID	Created on	Certificates & secrets
auth-app	59c07e69-0588-4a06-9d5a-a5861affc028	2/21/2024	Current

→ Configure Authentication in Auth0

- In the application registration settings, go to the "Authentication" tab.
- Add the appropriate redirect URIs for your application.
- Configure the "Implicit grant" and "ID tokens" settings.
- Save the changes.

Home > App registrations >

auth-app

Search

Overview Quickstart Integration assistant Manage Branding & properties Authentication Certificates & secrets Token configuration

Delete Endpoints Preview features

Essentials

Display name
[auth-app](#)

Application (client) ID
59c07e69-0588-4a06-9d5a-a5861affc028

Object ID
3fe47c0e-f4ec-4e02-b95d-11dbfaa646df

Directory (tenant) ID
9e2aab1b-6490-4f3a-b895-a705c7648f7b

Supported account types
[My organization only](#)

Client credentials
[0 certificate, 2 secret](#)

Redirect URIs
[1 web, 0 spa, 0 public client](#)

Application ID URI
[Add an Application ID URI](#)

Managed application in local directory
[auth-app](#)

→ Set Permissions

- In the application registration settings, go to the "API permissions" tab.
- Add the necessary permissions required for your application, such as user. Read.
- Grant admin consent for the added permissions.

→ Obtain Client Secret

- In the application registration settings, go to the "Certificates & secrets" tab.
- Generate a new client secret and note down the value.

auth-app | Certificates & secrets

Search

Overview Quickstart Integration assistant Manage Branding & properties Authentication Certificates & secrets Token configuration API permissions Expose an API App roles Owners

Got feedback?

secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) **Client secrets (2)** Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

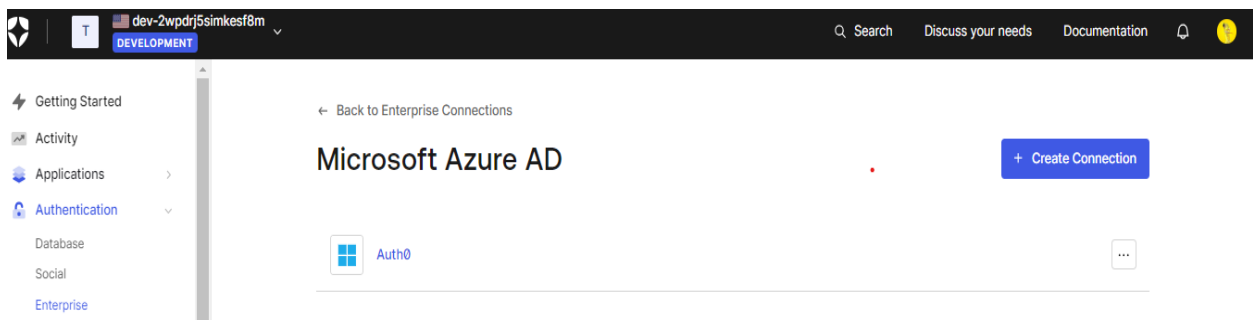
+ New client secret

Description	Expires	Value	Secret ID
auth0	8/19/2024	Gcm*****	243f5620-0cf6-45bd-baca-...
auth0-app	8/19/2024	dVw*****	cd012a7f-6c0a-4c53-b96c-...

7.2) Auth0 Enterprise Integration

→Auth0 Enterprise Account:

- Ensure you have access to an Auth0 Enterprise account. If not, contact your Auth0 administrator to set up an Auth0 Enterprise account.
- Log in to Auth0 Dashboard, go to "Authentication," and select "Enterprise Connections"
- Choose "Microsoft Azure AD."
- Click on "Create a New Application" on the left-hand side.
- Enter application details - Name, Domain Name, Client ID, and Secret Key.



→ Obtain Auth0 Enterprise Credentials:

- In the Auth0 Enterprise Dashboard, obtain the necessary credentials:
- Auth0 Enterprise Domain
- Enterprise Client ID
- Enterprise Client Secret

→ Update Auth0Provider Component:

- Update Auth0Provider Configuration:
- Modify the Auth0Provider component in your application (typically in index.js or an entry file) to include Auth0 Enterprise credentials:

→ Explore Auth0 Enterprise Dashboard:

- Log in to the Auth0 Enterprise Dashboard.
- Explore and configure enterprise-specific features such as:
- Identity Providers
- Custom Domains
- Multi-factor Authentication (MFA)



General

Connection name *

Auth0

This is a logical identifier of the connection. This name cannot be changed.

Microsoft Azure AD Domain *

sawlanikushgmail.onmicrosoft.com

Client ID *

59c07e69-0588-4a06-9d5a-a5861affc028

[How to obtain a Client ID? ↗](#)

Client Secret *

.....

For security purposes, we don't show your existing Client Secret.

Use common endpoint

☐

Use "https://login.windows.net/common" instead of default endpoint (https://login.windows.net/your_domain)). This is typically enabled if you're using this for a Multi-tenant application in Azure AD.

Activate Windows
Go to Settings to activate Windows

→After Integration Test the Enterprise Connection:

Creation of Enterprise Account:

The enterprise administrator creates an enterprise account on the identity management platform, such as Auth0.

Initiating Connection:

The enterprise user attempts to connect to a service, such as GitHub, using the enterprise account credentials.

Clicking on "Try":

Upon initiating the connection process, the user clicks on the "Try" button or initiates the login process for the desired service (e.g., GitHub) from within the enterprise account interface.

Redirect to Microsoft Sign-In:

The user is redirected to the Microsoft sign-in page, where they are prompted to enter their enterprise account credentials (e.g., username and password) to authenticate themselves.

Choosing GitHub in Sign-In Options:

After successfully signing in to the Microsoft account, the user is presented with sign-in options. In this case, the user selects GitHub as the desired sign-in option among the available choices.

Entering GitHub Credentials:

The user then enters their GitHub credentials (e.g., username and password) into the sign-in form provided by GitHub.

Connection Verification:

After providing the GitHub credentials, the identity management platform (e.g., Auth0) processes the authentication request and attempts to establish a connection between the enterprise account and the GitHub service.

Confirmation of Successful Connection:

If the connection process is successful and the provided credentials are valid, the user receives a confirmation message indicating that the connection was successful. They may also be redirected back to the original application or service they were trying to access, now authenticated with their enterprise account.

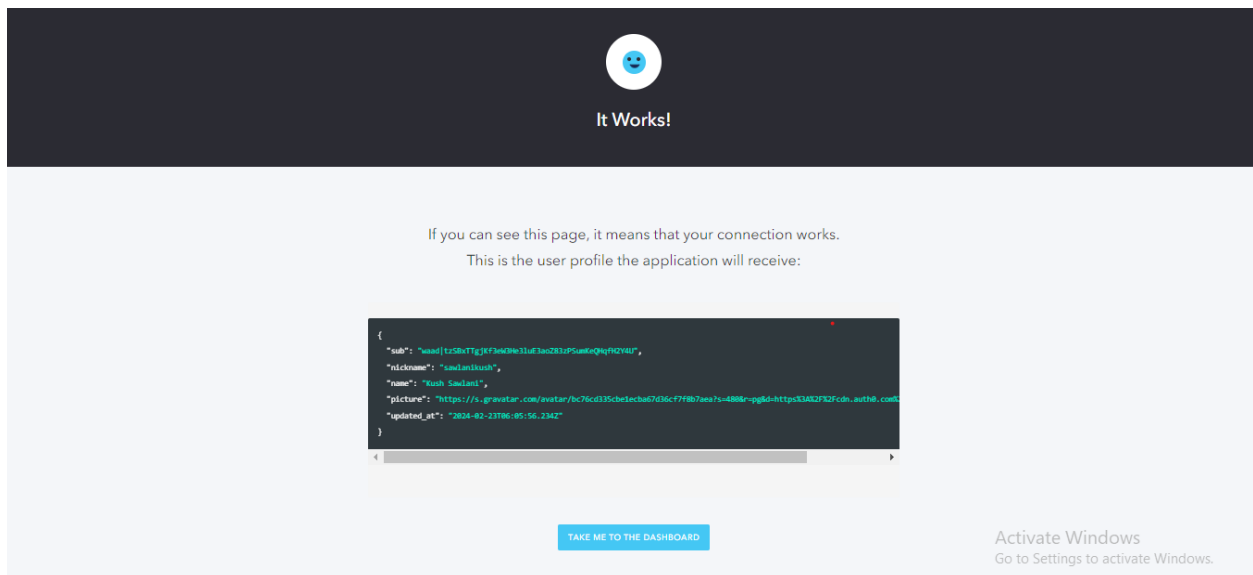
Testing the Connection:

The user may proceed to test the connection by accessing the desired resources or functionalities within the connected service (e.g., GitHub). They can verify whether they can perform the intended actions with their enterprise account.

Confirmation of Functionality:

If the connection worked as expected, the user receives confirmation that they can successfully access and utilize the connected service (e.g., GitHub) using their enterprise account credentials. They may also receive further instructions or guidance on using the service within the enterprise environment.

These steps outline the typical process of connecting to a service (such as GitHub) via Microsoft sign-in within an enterprise environment using an identity management platform like Auth0.

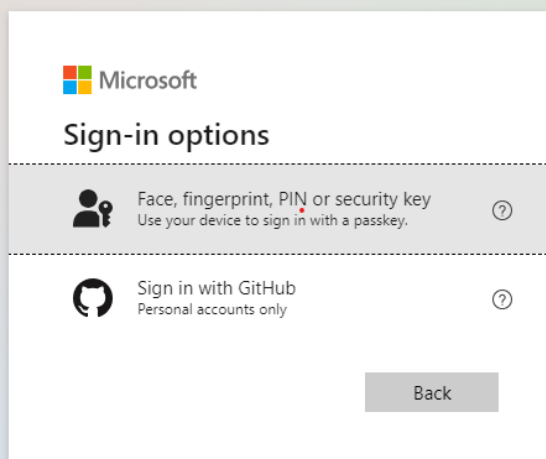
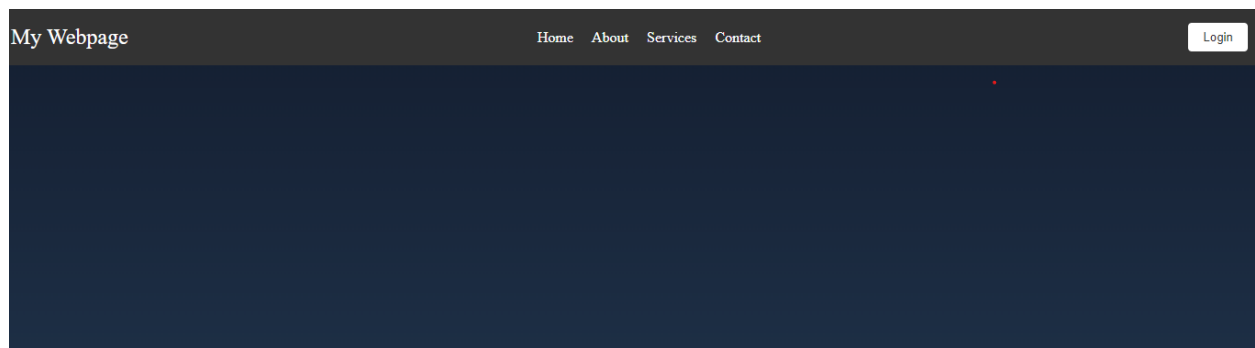


8)User Guide:

1. Logging in with Microsoft Account:

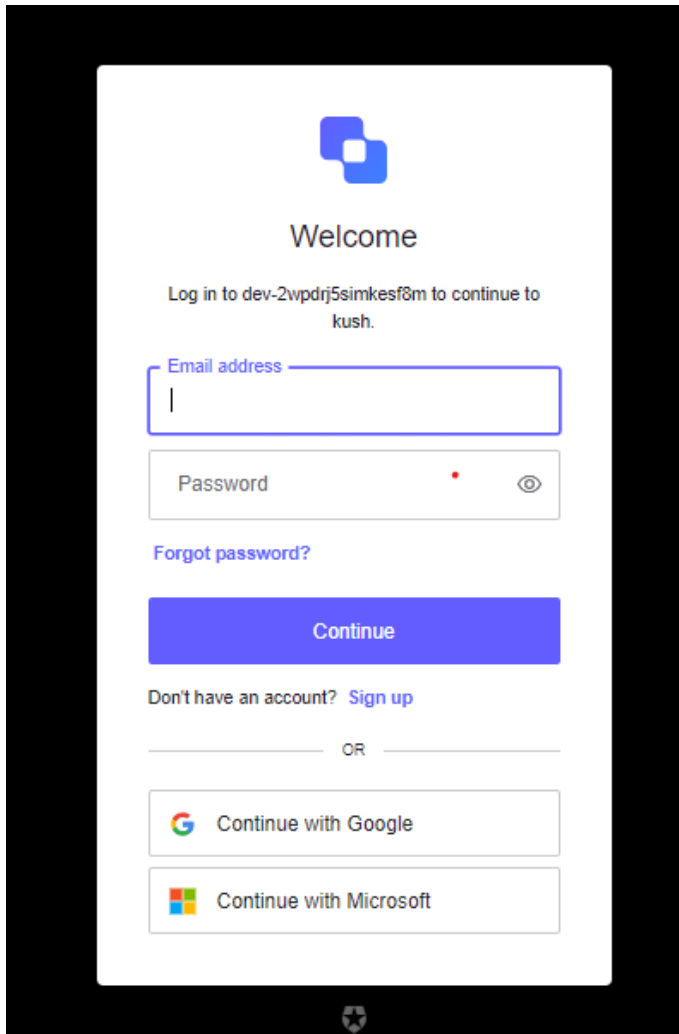
- To log in to the web application using your Microsoft account, follow these steps:
- Open the web application in your preferred web browser.
- Navigate to the login page.
- Look for the "Login with Microsoft" button on the login page.
- Click on the "Login with Microsoft" button. This action redirects you to the Microsoft login page.
- On the Microsoft login page, enter your Microsoft account credentials (email address and password).
- After successfully authenticating with Microsoft, you will be automatically redirected back to the web application.
- Always Choose Sign Option As GitHub Only.

You are now logged in to the web application using your Microsoft account.



1. Log In Process:

- Always have GitHub as a sign in option because it is a Microsoft Enterprise so it will not take personal mail id.
- After selecting GitHub Enter the Valid GitHub Username and Password and then login with all the credentials.



A login form for a web application. At the top is a blue logo consisting of two overlapping squares. Below the logo is the word "Welcome" in a bold, sans-serif font. Underneath "Welcome" is a line of text: "Log in to dev-2vpdrj5simkesf8m to continue to kush." Below this text are two input fields: "Email address" and "Password". The "Email address" field has a blue border and a cursor. The "Password" field has a red dot and an eye icon. Below the "Password" field is a link "Forgot password?". Below the links is a blue button labeled "Continue". Below the button is a link "Don't have an account? Sign up". Below the link is a horizontal line with "OR" in the center. Below the line are two buttons: "Continue with Google" and "Continue with Microsoft". At the bottom of the form is a small, dark, circular icon.

Welcome

Log in to dev-2vpdrj5simkesf8m to continue to kush.

Email address

Password

[Forgot password?](#)

Continue

Don't have an account? [Sign up](#)

OR

[Continue with Google](#)

[Continue with Microsoft](#)

2. Logging Out:

- To log out of the web application, follow these steps:
- Ensure that you are logged in to the web application.
- Look for the logout option, which is typically located in the user profile section or navigation menu.
- Click on the logout option. This action securely terminates your user session.
- After logging out, you may be redirected to the login page or a confirmation page, depending on the application's configuration.
- By following these steps, you can effectively navigate the login feature integrated with Auth0 and Microsoft Identity in the web application.

9)Conclusion:

The integration of Auth0 and Microsoft Identity for implementing the login and registration feature was successfully completed, providing users with a secure and seamless authentication experience. This project demonstrates the effectiveness of leveraging these technologies to enhance the authentication capabilities of web applications.

10)References:

- <https://auth0.com/docs/>
- <https://learn.microsoft.com/en-us/entra/identity-platform/>