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

1. Introduction:

This report provides an overview of the task, explaining the rationale behind integrating Auth0 with Microsoft Identity Platform for web application authentication. The introduction cogently articulates the imperative behind establishing a secure and scalable authentication solution. It underscores the significance of crafting a compelling proof of concept (POC) to vividly showcase the seamless implementation of advanced login and registration features. This report stands as a beacon, illuminating the path toward a robust and sophisticated authentication system.

2. Research on Auth0 and Microsoft Identity:

What is Microsoft Identity

Microsoft Identity, also known as the Microsoft Identity Platform, is a dedicated cloud identity service. It facilitates the development of applications that enable users and customers to sign in using their Microsoft identities or social accounts. Additionally, it authorizes access to APIs, both proprietary and those belonging to Microsoft, such as Microsoft Graph.

What is Auth0

Auth0 is a comprehensive identity management platform that simplifies the process of implementing authentication and authorization in applications. It provides developers with the tools to add robust user authentication features to their applications quickly. Auth0 supports various identity providers, social logins, and enterprise connections, offering flexibility in user authentication methods.

Microsoft Identity vs Azure: Are Both the Same?

Microsoft Identity and Azure are both services provided by Microsoft, but they serve different purposes and are not the same. Microsoft Identity is one of the many services provided under the Azure umbrella. It's a specific service that handles identity management, allowing users to sign into applications using their Microsoft identities or social accounts.

Moreover, Azure encompasses a diverse array of essential services beyond **Microsoft identity management,** these include AI + Machine Learning, Databases, Data Analytics, and numerous others services.

Features

The main features of Microsoft Identity are:

1. **Azure Active Directory (Azure AD):**

   Microsoft Identity stands as a linchpin in the realm of identity and access management, particularly due to its profound integration with Azure Active Directory (Azure AD). Beyond offering a comprehensive authentication and authorization service, this synergy extends across a spectrum of Microsoft services and third-party applications. This holistic integration ensures not only a secure access gateway but also a unified identity solution that harmonizes user experiences across diverse platforms.

2. **Single Sign-On (SSO)**

   Microsoft Identity incorporates Single Sign-On (SSO) capabilities, enabling users to log in once and access multiple applications without the need to re-enter credentials. This enhances user convenience and reduces the burden of remembering multiple sets of login information.

3. **Multi-factor Authentication (MFA)**

   Emphasizing a proactive stance on security, Microsoft Identity adopts Multi-factor Authentication (MFA) to fortify user accounts against unauthorized access and potential security threats. Going beyond traditional password-centric security, MFA incorporates additional layers of authentication, elevating the defense mechanisms. This layered approach ensures a resilient security posture, safeguarding sensitive data and user identities with a multi-faceted security shield.

4. **Conditional Access**

   Microsoft Identity introduces Conditional Access, a feature that enables organizations to define access policies based on specific conditions. This allows for granular control over user access, considering factors such as device health, location, and user roles. With Conditional Access, organizations can ensure that the right people have the right access to the right resources under the right conditions, thereby enhancing security and compliance.

The main features of Auth0

1. **Universal Login**

Auth0 offers a Universal Login feature, providing a seamless and customizable login experience for users. This centralized login system ensures consistency across various applications, enhancing user familiarity and usability.

2. **Social Logins**

Auth0 supports integration with a wide array of social identity providers, allowing users to log in using their existing social media credentials.

3. **Multi-factor Authentication (MFA)**

To bolster security, Auth0 implements Multi-factor Authentication (MFA). This additional layer of protection requires users to go through multiple verification steps, such as entering a code sent to their mobile device, ensuring a higher level of authentication and safeguarding against unauthorized access.

4. **Authorization**

Auth0 provides robust authorization features, allowing for fine-grained access control and the implementation of sophisticated authorization policies. This ensures that users only have access to the resources and features they are authorized to use, enhancing overall security.

Use cases for Microsoft Identity:

- Provides a comprehensive identity management solution for applications and services.
- Includes single sign-on (SSO), allowing users to authenticate once and gain access to multiple applications.
- Offers conditional access, enabling organizations to enforce policies that control access to applications based on a variety of conditions, such as user location and device state.

Use cases for Auth0:

- Provides a universal identity platform for secure authentication in applications.
- Supports social logins, allowing users to authenticate using their existing social media accounts.
- Supports multi-factor authentication, adding an extra layer of security by requiring users to provide two or more pieces of evidence to authenticate their identity.

The Microsoft Identity Pros

1. **Azure Integration:** Seamlessly integrates with other Azure services.
2. **Single Sign-On (SSO):** Enhances user experience with single sign-on capabilities.
3. **Conditional Access:** Allows granular control over access policies.

Microsoft Identity Cons

1. **Complexity:** The extensive feature set may be overwhelming for simpler applications.
2. **Azure-Centric**: Primarily designed for Azure-centric environments.

Auth0 Pros

1. **Flexible Integration:** Easily integrates with various identity providers.
2. **Universal Login**: Provides a customizable and secure universal login experience.
3. **Comprehensive Documentation**: Well-documented with extensive resources.

Auth0 Cons

1. **Cost:** Depending on usage, costs may escalate.
2. **Learning Curve:** Advanced features may require time to master.

This section provides an overview of Auth0 and Microsoft Identity, their features, differences, and the distinct advantages and disadvantages associated with each solution.

3. Identified Key Uses:

For Auth0

1. Universal Identity Platform: Auth0 provides a unified platform for identity management across applications.
2. Extensive Social Identity Providers: Supports integration with a wide array of social identity providers.
3. Rules Engine: Allows developers to customize and extend identity management logic.

For Microsoft Identity

1. Azure AD B2C: Enables businesses to customize and control how customers sign up, sign in, and manage their profiles.
2. Azure AD Connect: Facilitates integration between on-premises directories and Azure AD.
3. Conditional Access Policies: Allows organizations to enforce specific access conditions based on various parameters.

4.Setup and Configuration

4.1 Configure Auth0:

4.1.1 Create an Auth0 Account

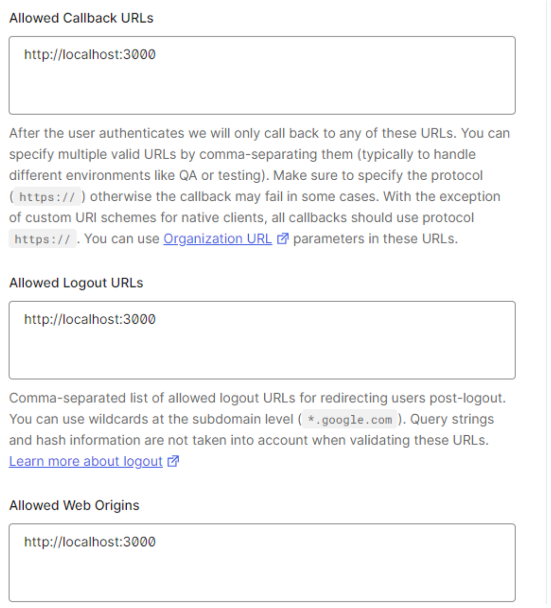
- Visit the Auth0 website and sign up for a new account **(https://auth0.com)**
- Once logged in, navigate to the Auth0 Dashboard.

4.1.2 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.

4.1.3 Configure Callback URLs

- Set the callback URL to http://localhost:3000 in the Auth0 Dashboard.
- Configure logout URLs to http://localhost:3000.
- Define Allowed Web Origins as http://localhost:3000.
- Create a new react application using a command
  - **npx create-react-app auth0**
  - **cd auth0**



Allowed Callback URLs

http://localhost:3000

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 (`https://`) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol `https://` . You can use Organization URL parameters in these URLs.

Allowed Logout URLs

http://localhost:3000

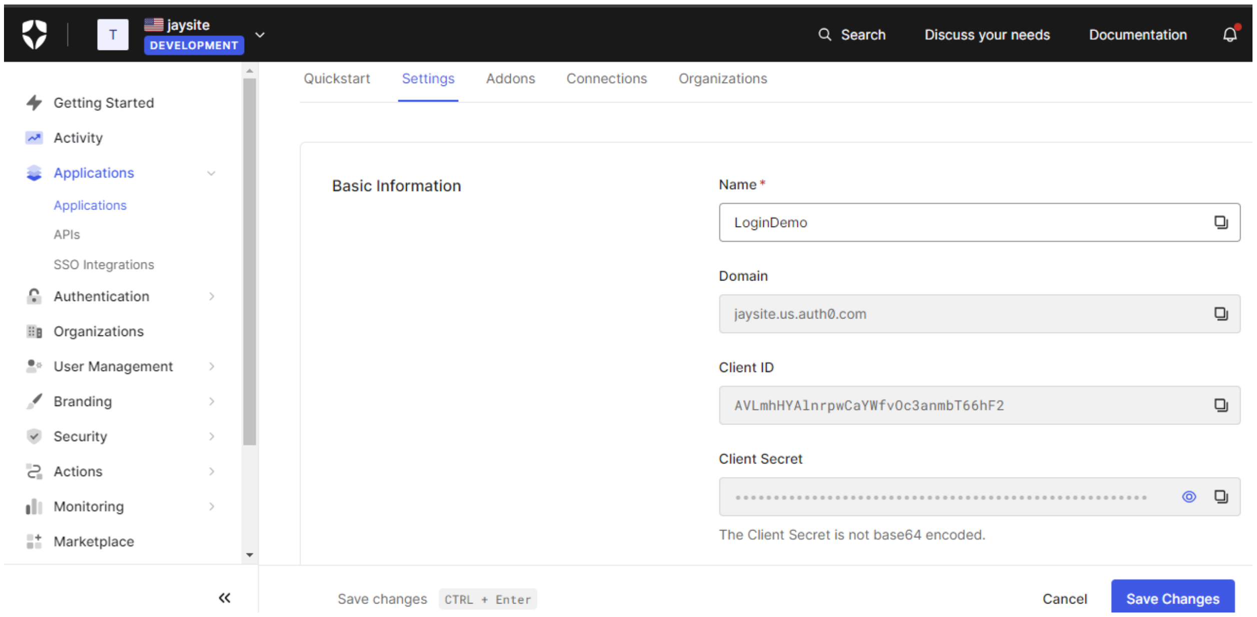
Comma-separated list of allowed logout URLs for redirecting users post-logout. You can use wildcards at the subdomain level (`*.google.com`). Query strings and hash information are not taken into account when validating these URLs. Learn more about logout

Allowed Web Origins

http://localhost:3000

4.1.4 Configure Auth0Provider Component:

- Set properties in the Auth0Provider component:
- Domain: Auth0 tenant domain.
- ClientID: Client ID from Auth0 Dashboard.
- authorizationParams.redirect_uri: http://localhost:3000.



Basic Information

Name *
LoginDemo

Domain
jaysite.us.auth0.com

Client ID
AVLmhHYAlnrpwCaYWfvOc3anmbT66hF2

Client Secret
••

The Client Secret is not base64 encoded.

<u>4.1.5. Add Login and logout to Your Application:</u>

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

- Index.js

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

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

root.render(
<Auth0Provider
    domain="jaysite.us.auth0.com"
    clientId="AVLmhHYAlnrpwCaYWfvOc3anmbT66hF2"
    authorizationParams={{
      redirect_uri: window.location.origin
    }}
  >
    <App />
  </Auth0Provider>,
);
```

- **App.js**

```
 function App() {
  const { user, isAuthenticated, isLoading, loginWithRedirect, logout } =
useAuth0();


  return (
    <div className="App">
      <Navbar />
      <div className="homepage">
        <h1>Welcome to Auth0 Login Site</h1>
        <h5>This one is Auth0 Login & Logout Testing Purpose Page</h5>

        {user ? (
          <>
            <button onClick={() => logout({ returnTo: window.location.origin })}
style={{ marginTop: "20px" ,marginBottom:"20px" }}>
              Log Out
            </button>
          </>
        ) : (

          <button onClick={() => loginWithRedirect()} style={{ marginTop:
"20px",marginBottom:"20px" }}>Login</button>
        )}
        {isLoading ? (
          <div>Loading ...</div>
        ) : (
          isAuthenticated && (
            <div>
              <img src={user.picture} alt={user.name} />
              <h2>{user.name}</h2>
              <p>{user.email}</p>
            </div>
          )
        )}
```

4.1.6. Show User Profile Information:

- Utilize the user property provided by Auth0 React SDK to display user information.
- Use is Authenticated and is Loading properties to handle user authentication status.

4.2. Set up Microsoft Identity Account

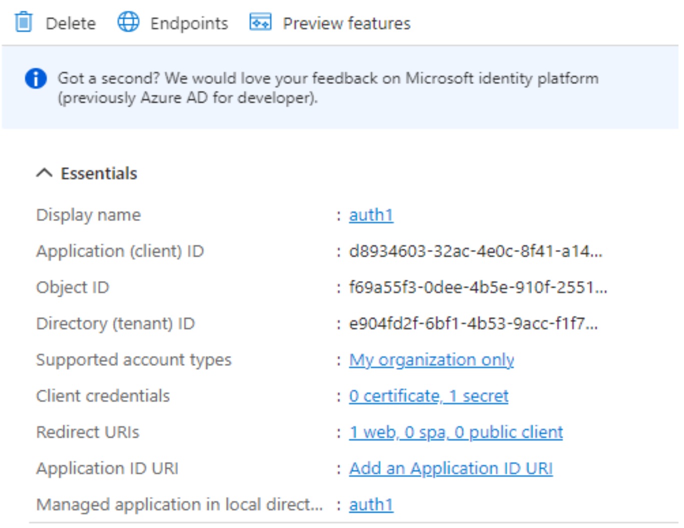
Microsoft Identity Platform, integrated with Azure Active Directory, provides a secure and scalable identity and access management solution. Here is a step-by-step guide on setting up Microsoft Identity for our project.

4.2.1 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.

4.2.2 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.



Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer).

Essentials

Display name	: auth1
Application (client) ID	: d8934603-32ac-4e0c-8f41-a14...
Object ID	: f69a55f3-0dee-4b5e-910f-2551...
Directory (tenant) ID	: e904fd2f-6bf1-4b53-9acc-f1f7...
Supported account types	: My organization only
Client credentials	: 0 certificate, 1 secret
Redirect URIs	: 1 web, 0 spa, 0 public client
Application ID URI	: Add an Application ID URI
Managed application in local direct...	: auth1

4.2.3 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.

4.2.4 Obtain Client Secret

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

4.3 Auth0 Enterprise Integration

4.3.1 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.

4.3.2 Obtain Auth0 Enterprise Credentials:

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

4.3.3 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:

4.3.4 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)

4.3.5 Test the Integration:

- Run your React application.
- Verify that the Auth0Provider is initializing correctly.
- Ensure there are no errors related to Auth0 during application runtime.

5. Conclusion:

The successful completion of this project will result in a functional proof of concept demonstrating the integration of Auth0 and Microsoft Identity for secure and efficient user authentication in a web application. By following the outlined tasks, the team will ensure the development of a robust and user-friendly login and registration feature.

6. References

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