

Task Title:

Development of Login and Registration Feature using Auth0 and Microsoft Identity

Objective:

To create a proof of concept (POC) demonstrating the implementation of a login and registration feature in a web application, utilising Auth0 for authentication and Microsoft Identity for login.

Project Overview:

The objective of this project was to develop a proof of concept (POC) demonstrating the implementation of a robust login and registration feature in a web application. The project aimed to leverage Auth0 for authentication and Microsoft Identity for login, providing users with a seamless and secure authentication experience. The decision to integrate Auth0 and Microsoft Identity was driven by the need for a scalable and reliable authentication solution that adheres to industry standards.

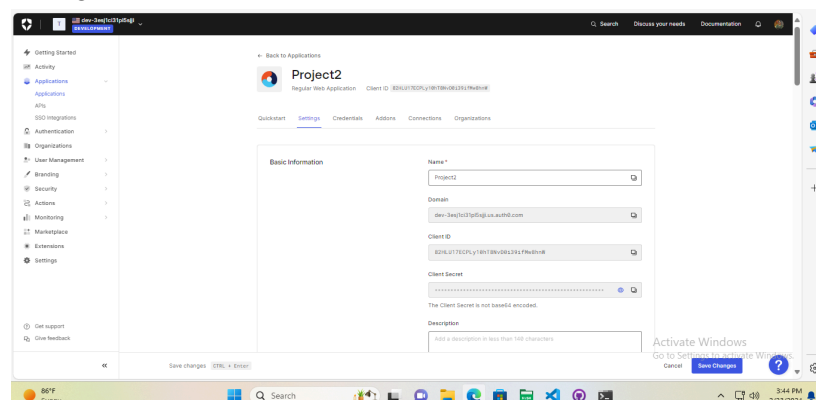
Architecture Overview:

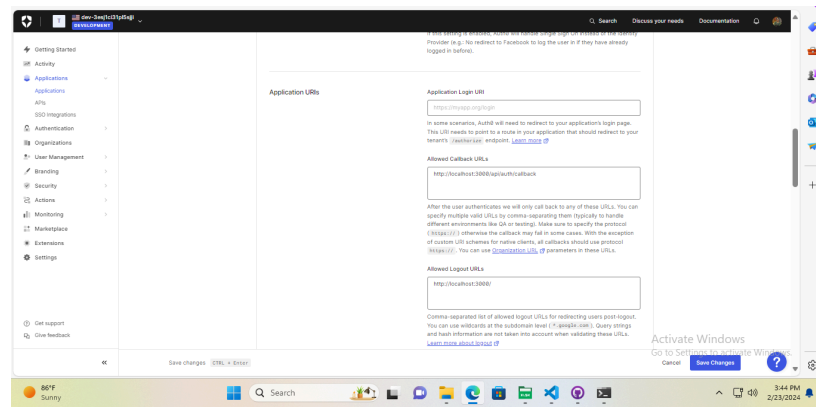
The architecture of the web application follows a client-server model, with Next.js serving as the frontend framework and an appropriate backend for handling authentication processes. Auth0 and Microsoft Identity are integrated into the application to manage user authentication and registration. The flow of authentication involves redirecting users to the respective login pages of Auth0 and Microsoft Identity, where they authenticate using their credentials. Upon successful authentication, users are redirected back to the application, logged in, and authenticated.

Setup Process and Integration of Auth0 with Microsoft Identity:

1. Created an Auth0 Account and Configured a New Application:

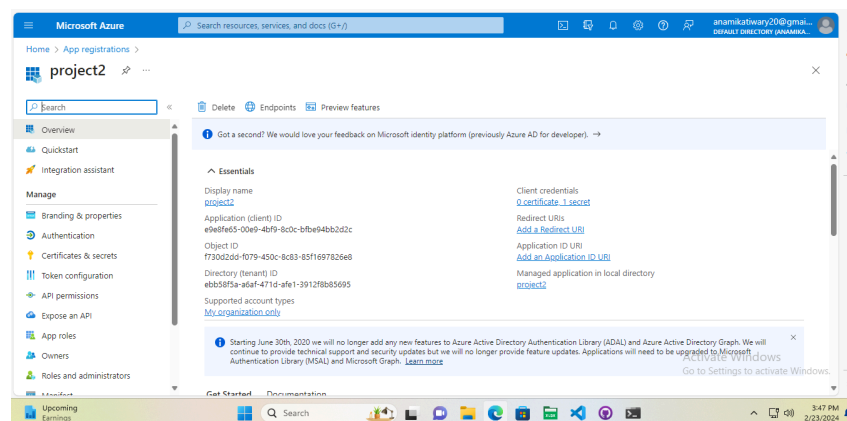
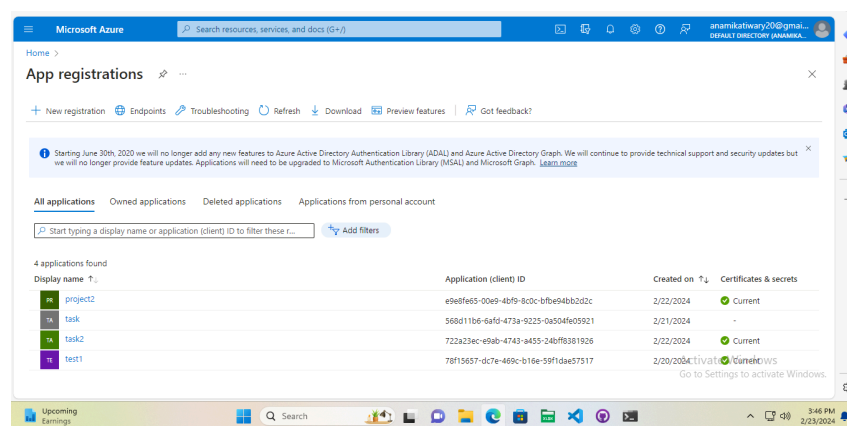
- Navigated to the Auth0 website and signed up for an account.
- Logged in and accessed the Applications section to create a new application.
- Selected the application type and provided a name for the application.
- Configured the settings as per the requirements, specifying appropriate callback URLs and allowed web origins.

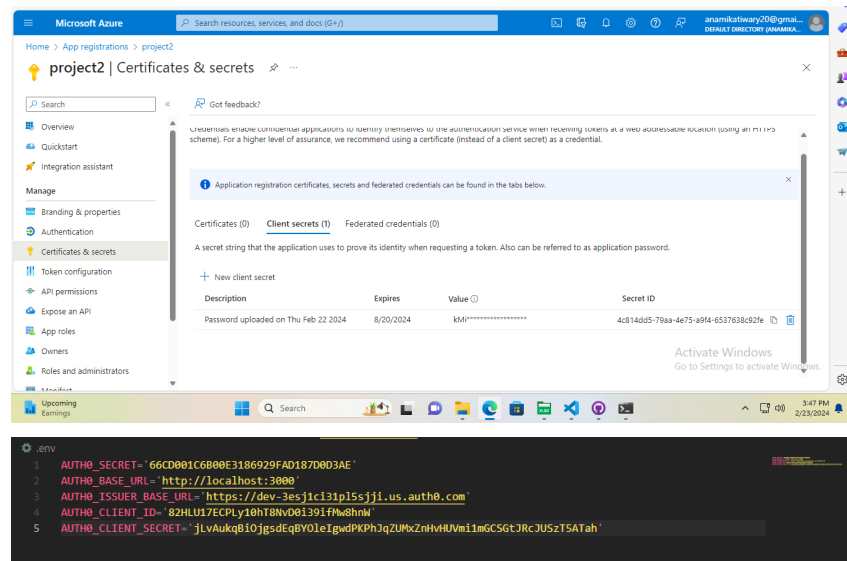




2. Set Up Microsoft Identity in the Azure Portal and Configured it as an Identity Provider in Auth0:

- Logged in to the Azure portal and navigated to the Azure Active Directory section.
- Created a new Azure Active Directory tenant or used an existing one.
- Registered the application with Azure Active Directory and noted down the Application (client) ID and Directory (tenant) ID.
- Configured the necessary permissions and authentication settings for the application.
- In Auth0, navigated to the Connections section and selected "Microsoft Azure AD" as an identity provider.
- Entered the Application (client) ID, Directory (tenant) ID, and any additional configuration settings required.





Code Snippet

User Guide for Login and Registration Feature

Logging In:

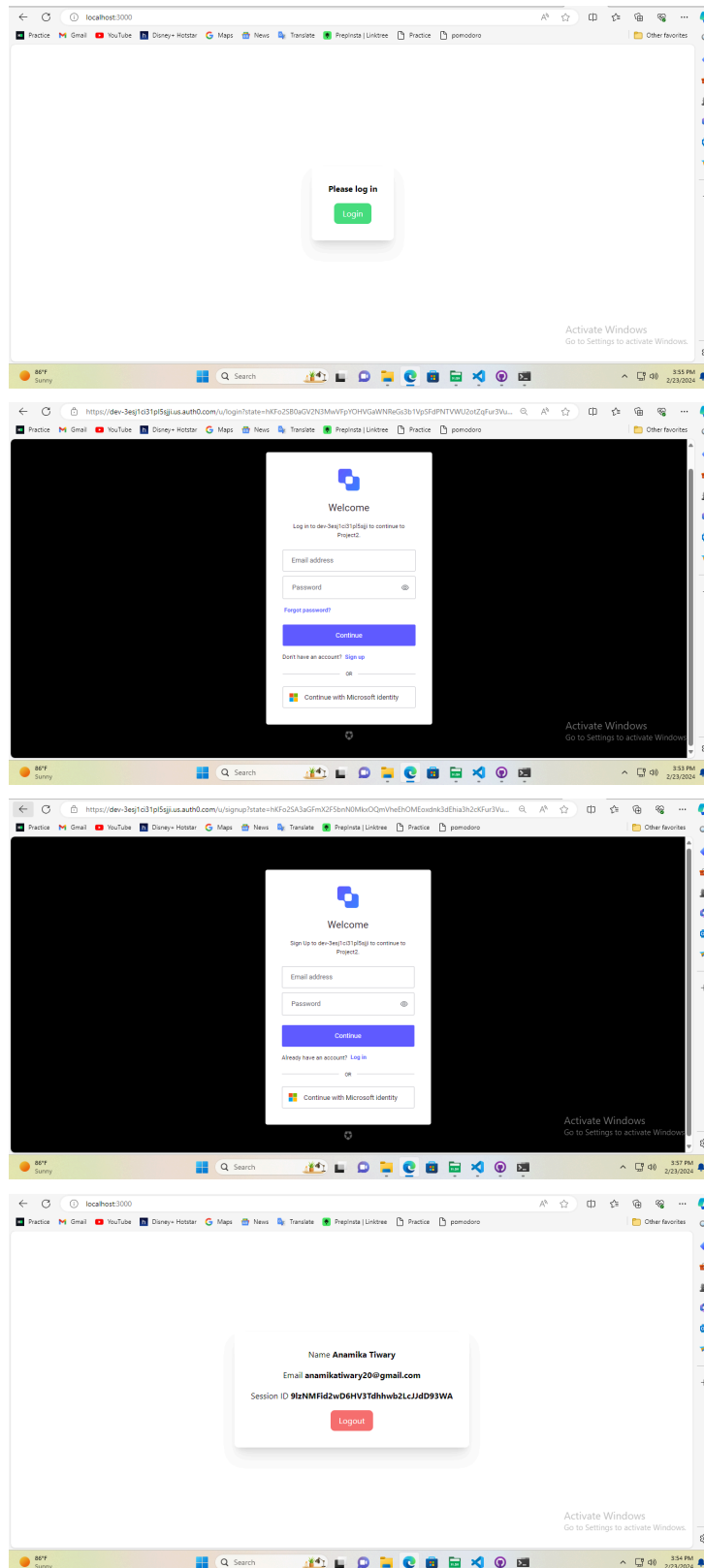
1. Accessed the login page of the Next.js web application.
2. Clicked on the "Login with Microsoft Identity" button.
3. Was redirected to the Microsoft login page. Entered Microsoft otp and authenticated.
4. Upon successful authentication, was redirected back to the Next.js web application, logged in, and authenticated.

Registration:

1. Accessed the registration page of the Next.js web application.
2. Clicked on the "Register with Microsoft" button.
3. Was redirected to the Microsoft login page. Entered Microsoft credentials and authenticated.
4. Upon successful authentication, a new account was created using Microsoft credentials.
5. Was automatically logged in to the Next.js web application after registration.

Logging Out:

1. Clicked on the "Logout" button located in the user profile section.
2. Session was securely terminated, and was redirected to the home page.



Conclusion:

In conclusion, the integration of Auth0 and Microsoft Identity into the Next.js web application has resulted in a reliable and secure authentication solution. By leveraging industry-leading authentication providers, users can enjoy a seamless and secure login and registration

experience. The project has demonstrated the importance of robust authentication mechanisms in web applications and the value added by integrating Auth0 and Microsoft Identity.

Challenges & Solutions:

Challenge: Configuring Auth0 and Microsoft Identity settings, including callback URLs, allowed web origins, and client IDs, proved to be complex and time-consuming.

Solution: Took a systematic approach by carefully following the documentation provided by Auth0 and Microsoft Identity. Utilised step-by-step guides and consulted online forums for clarification when encountering difficulties. Ensured that all configurations were accurately set up in both Auth0 and Azure portals to establish a seamless integration.

References:

- Auth0 Documentation: [\[Link\]](#)
- Microsoft Identity Platform Documentation: [\[Link\]](#)
- Next.js Documentation: [\[Link\]](#)