DESIGN AND IMPLEMENTATION OF SIMPLE AUTHENTICATION APPLICATION USING AWS

OVERVIEW:

In this application, we create a sample authentication flow for the user to register, confirm their registration, login into an existing account and reset their password if forgotten. Initially, the login page will be displayed on default if the user opens the application. The user can directly login into the application with their credentials if they have an account. If they don't have an account, there is an option to open the registration page which allow the user to register with the application followed by a confirmation page for the registration. If the user forgot their password, it can also be reset. In the backend, we are using Amazon Web Services namely Lambda, Cognito and API Gateway and boto3 package functions for creating new users, accessing them, and many more.

COMPONENTS USED:

Lambda Functions	Dynamic UI	Manual Functions	Data elements
API Gateway	Boto3	On Click Listener	Override Methods
Cognito	Intents	AsyncTask	UI Elements

UI:

The UI of this application mainly consists of three main flows namely Login, Registration and Reset Password. The application opens with the Login UI, and we can navigate between these three different flows back and forth. We try to make use of the dynamic UI elements in the Reset Password flow instead of creating multiple activities. The UI is a simple click-and-display and based on the success/failure of that flow, a toast message will be displayed indicating the same.

APPLICATION FLOW:

In simple terms, we will be using intents to navigate between the activities created for each flow. For each flow, we are using AWS services to create user, authenticate user and reset the user's password. We can divide the application into different flows:

1. In the first screen, the application displays the login UI, and the user is presented with the username and password boxes to enter the values into it. If the user details match and the login is successful, a toast message will be displayed on the screen stating the same and the MainActivity screen is displayed. In the backend, we will be using a lambda function to send the username and password of the user using "admin_initiate_auth" which is a CognitoIdentityProvider API that can allow us to perform operations in Cognito. If the

credentials entered by the user are incorrect, then a toast message will be displayed stating the same.

2. If the user doesn't have an account, we provide a clickable TextView in the LoginUI which can redirect the user to the RegisterActivity screen, the user can then create an account with details like name, username, email, and password. For registration, we will use the "sign_up" boto3 function. A toast message will be displayed if the user's registration is successful, and a confirmation registration screen will appear where the user must enter the code sent to his email. Once the code is entered, the user should click confirm button. If the confirmation is done successfully, a toast message will be displayed and the Continue to Login option will appear.

3. If the user forgets his password, the user can select the Forgot Password link in the LoginUI. A username prompt will appear and if the user clicks the reset button, a code will be sent to their email and a new screen will appear where the user should enter his new password and the code sent to mail. After these details are entered, the user should click the set password button. If the details are correct and everything works well in the backend, the user will get a screen stating, "Your password is confirmed!". For the forgot password, we are using two lambda functions namely "admin_reset_user_password" and "confirm_forgot_password".

4. At all times in the Reset Password flow, the user has the option to go back. Once the user's new password is confirmed, the user can select the back arrow in the top screen to go back to the Login UI.

5. Five APIs are created for registration, confirmation of registration, login, reset password and confirm forgot password in the API Gateway respectively.There are five APIs we are using in the API Gateway which are linked to the lambda function. These API resources are all created under one REST API called the user. Each API resource will have its own method depending on the type of request it will perform.

    1. Registration: This API resource will use the POST method, which we will use to register a new user.
    2. Confirm Registration will be used to confirm the registration of a new user and POST method will be used to request the code.
    3. Login: Any existing user can login using this resource. Since we need confirmation, we will be using the POST method.
    4. Reset Password: The user can reset their password using this API resource. A code will be sent for which we will use a POST request method.
    5. Confirm Reset Password: The user can enter a new password using this API resource. With the code sent using previous resource, username, and new password we can set a new password for the user. POST method request will be used for this API as well.

6. We will be maintaining a user pool in Cognito with all the user credentials. All the details of the users are stored namely username, full name, email, and password. We will provide each lambda function that accesses AWS Cognito with PowerAccess policy

which will grant all the necessary permissions to the lambda functions to modify, access and change any details in the user pool.

APPLICATION LIFECYCLE