**Feature Development Plan Template**

**Feature Name**

*Auth 0 integration*

# Requirement Analysis

**User Story Reference:** *[Link or reference to the User Story] (optinal)*

**Summary:** *Enable authentication and authorization using Auth0 for enhanced security and user management..*

# Impact Analysis (Optional)

**Affected APIs:** *Auth API, User Management API.*

**Analysis:** *The integration will impact user authentication and authorization processes. Dependencies include a seamless connection to the Auth0 service. Potential conflicts may arise if existing authentication methods are not properly deprecated.*

# Database Schema Changes (Optional)

**Current Schema Overview:** *No direct changes*

* **Proposed Changes:** 
  + ***api/auth/login*** *(POST) - User login*
  + ***api/auth/logout*** *(POST) - User logout*
  + ***api/auth/user-profile*** *(GET) - Fetch user profile*

# API Endpoints Design

**Endpoint Summary:** *List the new or modified endpoints required for this feature.*

**Design Details:** *For each endpoint, provide the following:*

* *Endpoint URL :*
* *api/auth/login (POST) - User login*
* *api/auth/logout (POST) - User logout*
* *api/auth/user-profile (GET) - Fetch user profile*
* *Request Parameters : None*
* *Request Body Schema:*

*{*

*"username": "string",*

*"password": "string"*

*}*

* *Response Body Schema*
* *Error Codes and Messages : 401 Unauthorized, 400 Bad Request*

# Pseudo Code for Key Functionalities

**Function Overview:** *Briefly describe the key functions to be implemented.*

**Pseudo Code:** *For each key function, provide pseudo code to illustrate the core logic and flow.*

Pseudo code is a way to describe the functionality of a program or algorithm in a format that’s more readable than actual code, but not as detailed as plain language. It helps in planning and understanding the logic before diving into the actual coding.

Let’s say the feature in question involves adding a new functionality to a user profile system, where the backend needs to handle a request to update a user’s email address. Here’s an example of how the pseudo code for this functionality might look:

**Pseudo Code for Update Email Function Functionality:** Auth0

**Pseudo Code:**

FUNCTION AuthenticateUser(username, password)

IF username and password are valid THEN

token = Auth0Service.generateToken(username, password)

user = Auth0Service.getUserProfile(token)

RETURN { "token": token, "user": user }

ELSE

RETURN "Invalid credentials"

END IF

END FUNCTION

This pseudo code outlines the basic logic for updating a user’s email. It includes input validation, checking if the user exists, and handling the database transaction, which are common steps in backend operations. Note that pseudo code is not meant to be executed, and the syntax is not language-specific. It serves as a blueprint for the actual code that will be written.

# Deployment and Configuration Changes (Optional)

**New App Settings:** *Auth0 credentials (Client ID, Client Secret).*

**Deployment Changes:** *Integrate Auth0 settings during deployment.*

* *Testing and Verification Conduct unit tests for authentication and user profile retrieval.*
* *Perform integration tests for the overall authentication flow.*
* *Manual testing for login, logout, and user profile functionality*