# **Architectural Design and Process Flow for User Management System.**

**Requirements**

**Endpoints:**

* + User should be able to login using email and password
  + Incase of inactive user set is\_active=1
  + User should be able to sign up using email and password
  + User should be able to get their details (profile information)
  + User should be able to update their information
  + User should be able to deactivate their profile(set is\_active=0)
  + Other user should be able to get other user profile using user id
  + User should be able to get other user profile details

**Process Flow:**

**User Signup:**

* User sends a POST request to /signup with email, password details.
* FastAPI validates the request data.
* If valid, FastAPI hashes the password and saves user details to the database.
* User account is created with default values for is\_active and is\_verified.

**User Login:**

* User sends a POST request to /login with email and password.
* FastAPI validates the credentials.
* If valid, FastAPI generates a JWT token and returns it to the user for authentication.

**User Profile Management:**

* User sends GET request to /user to view their profile with info like firstname, lastname , address.
* User sends PUT request to /user to update their profile information.
* User sends DELETE request to /user to deactivate their profile.

**View Other User's Profile:**

* User sends GET request to /users/{id} with the ID of the user whose profile they want to view.

**View All Users:**

* User sends GET request to /users to get a list of all user profiles.

**Feature Enhancements:**

**Password Reset:**

* Add a /reset-password endpoint for users to reset their passwords.

**Email Verification:**

* Implement email verification during signup.
* Add a /verify-email endpoint for email verification.

**EndPoints to be protected**

* /rest-password
* /verify-email
* /users
* /users/{id}

**Architecture flow:**

