**User Management Tool API Explaination Documentation**

**1. User Sign Up API :-**

* This is a sign up API endpoint.
* It takes in the user's username, email, password, and confirm password as input and validates them.
* If any of the required fields are missing, the API returns an error message with a 400 status code.
* It also checks if the email is valid, the password is at least 6 characters long, and the password and confirm password fields match.
* If any of these conditions are not met, the API returns an error message with a 400 status code.
* If the username is not already taken, a new user is created and saved to the database, and the API returns a success message with a 201 status code and the user's ID.

*endpoint :-*

*method:* POST

/sign\_up :

Request Body :

username - string - Required\*

email - string - Required\*

password - string - Required\*

confirm\_password – string - Required\*

Response Body:

{

“success”: “User Created Successfully”,

“user\_id”: “1”

}

**2. User Login API :-**

* This is a user login API endpoint.
* It takes in the user's username and password as input and validates them.
* If either of the fields is missing, the API returns an error message with a 400 status code.
* It then checks if the user is authenticated by calling the authenticate() function with the username and password.
* If the user is not authenticated, the API returns an error message with a 401 status code.
* If the user is authenticated, the API logs in the user by calling the login() function with the request and user objects.
* It then returns a success message with a 200 status code and the user's ID.

*endpoint :-*

*method:* POST

/login :

Request Body :

username - string - Required\*

password - string - Required\*

Response Body:

{

“success”: “User Logged in Successfully”,

“user\_id”: “1”

}

**3. User Profile Create API :-**

* This is a user profile API endpoint.
* It takes in the user's ID, profile photo, location, and designation as input and validates the user ID.
* If the user ID is missing, the API returns an error message with a 400 status code.
* If the user record does not exist, the API returns an error message with an appropriate error message.
* The API then checks the size of the profile photo and compresses it if it is larger than 1MB.
* If the photo is compressed successfully, the compressed photo is used to create the profile.
* If the photo cannot be compressed, the API returns an error message with a 400 status code.
* If the photo is larger than 1MB, it is not create the profile.
* Finally, the API creates the user profile and returns a success message with a 200 status code.

*endpoint :-*

*method:* POST

/create\_user\_profile :

Request Body :

user\_id - integer - Required\*

profile\_photo - file - Not Required

location - string - Not Required

designation - string - Not Required

Response Body:

{

“success”: “Profile Create Successfully”

}