

# USER AUTHENTICATION

## Workflow for user authentication using POST and GET APIs in a web application

### 1. Signing Up

- ① The user visits the sign-up page of the application.
- ② Sign-up page displays a form with input fields for **username**, **email** & **password** and a 'SIGN UP' button.
- ③ When the user clicks "SIGN UP" button,  
⇒ a 'POST' request is sent to the server-side API with the **username**, **email** & **password**.
- ④ "The server-side API creates a new-user account"
- ⑤ If the account creation is 'Successful',  
⇒ the server responds with a '**200 OK**' HTTP status code.
- ⑥ If the account creation is not successful,  
⇒ the server responds with an error

message and an appropriate HTTP status code (e.g; **400 Bad Request**)

- ⑦ The client-side code then redirects the user to the login page of the application.

## II. Logging In

- ① The user visits the login page of the application.
- ② It displays a form with input fields for email and password, as well as a Login button

Diagram of a Login form:

Login

Enter email

Enter password

LOGIN

- ③ When the user clicks the 'Login' button,  
⇒ a 'POST' request is sent to the server-side



API with user's email & password.

④ The server-side API validates the user's credentials.

⑤ If the credentials are valid,

⇒ The server generates 'an authentication token' and responds with this authentication token & '200 OK' HTTP request status code.

⑥ If the credentials are not valid,

⇒ the server responds with an error message and an appropriate HTTP status code (e.g; 401 unauthorised).

⑦ The client-side code receives the authentication token and stores it in local storage or a cookie for future requests.

⑧ The client-side code then redirects the user to the homepage of the application.

⑨ If the user is not authenticated, the homepage redirects the user to the login page with appropriate message.

### III Log out

(1). When the user logs out,

⇒ the client-side code removes the authentication token from local storage or a cookie.

(2). The client-side code then redirects the user to the login page.