Team Members

[Venkatesh Bonageri](https://uncc.instructure.com/groups/17907/users/53827)

Sai Nishanth Dilly

**Introduction:**

The assignment was to create a API for the user authentication functionality with all the error handling. So to implement this, we choose to implement with php and mysql, the client side to be as an Android Application.

**MYSQL:**

The backend (MYSQL) has been deployed in 000 Web hosting service. It contains a users table which maintains the user details which contains email, password, first name, last name, age and weight.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Collation** | **Attributes** | **Null** | **Default** | **Comments** | **Extra** |
| 1 | **fname** |  | utf8\_unicode\_ci |  | Yes | *NULL* |  |  |
| 2 | **lname** |  | utf8\_unicode\_ci |  | Yes | *NULL* |  |  |
| 3 | **email** |  | utf8\_unicode\_ci |  | No | *None* |  |  |
| 4 | **age** |  |  |  | Yes | *NULL* |  |  |
| 5 | **weight** |  |  |  | Yes | *NULL* |  |  |
| 6 | **password** |  | utf8\_unicode\_ci |  | Yes | *NULL* |  |  |

**PHP:**

The PHP has been deployed to a free hosting website 000webhost.com. It contains 2 different files

The request types used are GET and POST and the response would be in the json format.

* **signup.php:** It contains two parameters namely email, name and password which is used for the creation of the user with the parameters given by queries the amazon aws database. Different scenarios handled on signup action
  + Check for amazon aws database connectivity, if it is not reachable then there is a response called *database not reached*
  + On signup success we would get a response saying *registration successful* and with a status code 200. The json response is show below

{

"status": 200,

"status\_message": "registration sucesfull",

"id": 1

}

* + On using same name, then the response is the *user exists* with the status code as 200

{

"status": 200,

"status\_message": "user already registered",

"id": 0

}

* + On *invalid request* the status code would be 404.

{

"status": 404,

"status\_message": "Invalid request",

"id": null

}

* **login.php:** It contains the php code which queries to the database with the name and the password as the parameters. Different scenarios handled on the login action
  + On login success we would get a row of data and an alert would be displayed saying *login successful* with a status code as 200

{

"status": 200,

"status\_message": "login sucesfull",

"id": 1,

"data": {

"0": "pawan@gmail.com",

"1": "Pawan",

"2": "pawan",

"email": "pawan@gmail.com",

"name": "Pawan",

"password": "pawan"

}

}

* + On login failed we would get an alert displaying *invalid credentials* and the response status is 200

{

"status": 200,

"status\_message": "invalid credentials",

"id": 0,

"data": null

}

* + On *invalid request* the status code would be 404.

{

"status": 404,

"status\_message": "Invalid request",

"id": null

}

**Android Mobile Application:**

This is a basic user authentication application which uses the API which has been created to serve the user authentication functionalities. Considering the requests and response paradigm we have created different views to handle the functionality.

* **MainActivity:** In this activity we are hitting the API

[**https://advancemad.000webhostapp.com/index.php**](https://advancemad.000webhostapp.com/index.php)with email, password and name taken from the mobile application. It requests the API for login generates the token jwt and sends to mobile app and this service stores this jwt token ii=n shared preference.

* **SignUp:** In this Activity we are hitting the API <https://advancemad.000webhostapp.com/signup.php>with email, password, first name last name and other details as the credentials taken from the mobile input. If the user exists, the appropriate response would be delivered in JSON as a response from the API.
* **ProfileDetails:** It contains the user specific data with the specific email, if the token is already in shared preference and expired this activity deletes it from shared preference and redirects to login and if token is not expired we can update user details the api which hits in this activity are [**https://advancemad.000webhostapp.com/updateDetails.php**](https://advancemad.000webhostapp.com/updateDetails.php) **to update user details.** [**https://advancemad.000webhostapp.com/index.php**](https://advancemad.000webhostapp.com/index.php) **to display user details**