**Login and Sign Up**

1. Take Full Name, Email, Password and Role from the current user.
2. On the submission initialize this new user’s object in the users collection with following properties:

{

FullName, username, Email, Password, Role

}

1. When signing up, the user would give his/her username and email which should be unique. When the submission action will be performed, post the sign up data to the backend server for the uniqueness check. Bring all the email addresses and usernames of the users into an array and with find method, check whether the incoming email address and username already exists or not. If not then, send the status code 200 and save the user in the database, if exists, send status code 500 with a message “user email already exists” or “username is not available” depending upon what already exists. On the front end, status code 200 proceeds the app to the second screen, status code 500, displays an error message.
2. When Logging in, the user would post his/her username and password to the backend server, the server would bring all the user objects {username, password, role} from the db into an array and check whether the username exists or not and password matches or not, if true, send the status code 200 and the role value to the front end and the app will proceed to the next screen (here tokens should be sent and the session would start), if not then send the status code 500 which will prompt the incorrect username password error. If the role value is student, set the redux state of the app { currentUser: username }, create a GET request to backend server – localhost:3000/student/username (here authorization is required, to ensure that this link is only accessible if the current token is authorized to visit it). At backend, the incoming parameters (student and username) would go into the jobs collection in DB and iterate over it to find out the jobs posted by this username, the resultant list would be sent as the response which will then update the state of the front end.

**Admin Login and Sign Up**

* Admins can’t sign up. If someone has to be an admin, their id will be created directly on the database by the company.
* Admins collection will be created in the database which would hold the adminId and passwords.
* Admin has to go to the link localhost:3000/admin where they’ll input their adminId and password and post it to the backend. Typical, login mechanism will follow, allowing admin to log in.

When signing up, the user would put in their credentials and the API request will be initiated directly from the front end.