# **Assignment**

### CRUD Using Angular/ReactJs, NodeJS & MongoDB

### **This Assignment is divided into two parts:**

Part 1 Backend: API Creation with NodeJs & MongoDB.

<u>Part 2</u> Frontend: Integrating API using Angular/ReactJs.

# Part 1 - NodeJs with MongoDB.

We want you to create a REST API using NodeJS & MongoDB that has the following features. You can use ExpressJs and Mongoose for defining the routes and creating database.

- 1. The API routes should start with /api/v1/
- 2. Following resource will be required to complete CRUD operation.
  - POST /api/v1/users To create user
  - GET /api/v1/users To get list of all the created users
  - PUT /api/v1/users/:id To Update the user
  - DELETE /api/v1/users/:id To Delete the user
  - POST /api/v1/session To generate session id for login
  - DELETE /api/v1/session To Delete the session id for logout
- 3. Implement API Authentication so that API routes are protected with session id/token. You can use JWT for the same.
- \*\* Read Part 2 Angular/ReactJs below to understand what type of data you will be storing.

Following fields will be saved for each user.

- Name: String (min 3 to 20)
- Email Address: String (should be unique)
- **Phone Number:** Number (10 digits)
- **Password:** String (min 8 and max 20) The password will be stored in as hashed do not store it as plain text. You can use **bcrypt.**

**<u>Dummy JSON Format:</u>** You can define your JSON format as shown below while sending the user details in the POST request.

```
{
  "name": "John Doe",
  "email": "john@mail.com",
  "phone_number": 9865242510,
  "created_by": "<user id who is creating this user>" // [default value is null]
  "password": "John@123"
}
```

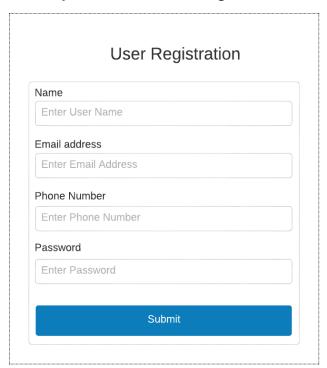
# Part 2 - Angular/ReactJs

We want you to create an Angular or ReactJs application that has the following areas. You may use bootstrap to design your UI.

# **User Registration:**

You have to create page which will have following fields

- **1. Name:** String (min 3 to 20)
- 2. Email Address: String (Unique email for every user)
- 3. Phone Number: Number (10 digits)
- **4. Password:** String (min 8 and max 20)
- **5. Submit Button:** On Clicking on submit button user will be registered and will be able to login back into to the system.
- \*\* Store password as hashed string i.e. do not store password as plain text.



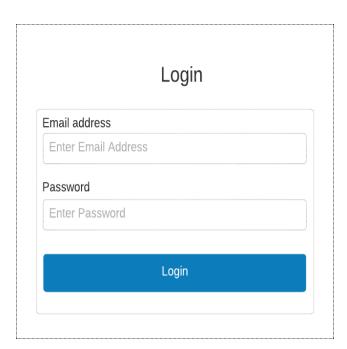
# **User Login:**

You have to create page which will have following fields

- 1. Email Address: String (Password entered while registration)
- **2. Password:** String (Password entered while registration)
- **3. Login Button:** On Clicking on login button if email address and password exists in the database then a session id will be generated and sent back to UI and user will be redirected to **Home Page.**

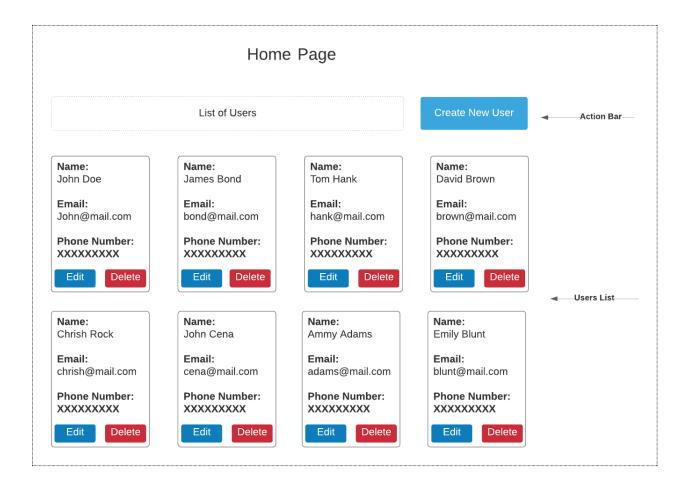
#### Note:

- \* For authentication you can implement **JWT** in the API (**Part 1 NodeJs with MongoDB above**) or you are free to implement other mechanism.
- \* After successful login user should be redirected to home page and the session should be maintained even if user refreshes the home Page the session should not get deleted.
- \* Without login (session) user can view Login and Registration page only and other pages are restricted to logged in users.



### **Home Page**

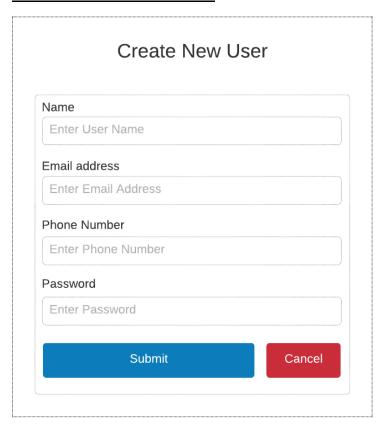
The home should have an **Action bar** and a **List of Users**.



### Action bar

- There should a button labeled as Create New User.
  - a. On Clicking on Create New User a new page will open (See Form Below) which will have a form with following fields Name, Email Address, Phone Number, Password and Submit Button.
  - **b.** After filling the form once we click the **Submit Button** a new user should be created and we should be redirected back to home page where newly created user should be available in the list.

#### Create New User Form:



#### List of Users –

- Create a card view to show the users details where each card represent a user.
- On each card show the **Name**, **Email**, and **Phone Number** of each user.
- At the bottom of the each card there should be 2 button one for edit and second for delete.
  - Delete Button: On clicking on delete button a confirmation popup should come which will have 2 buttons labeled as Yes & No, Clicking on Yes will delete the user from Database as well as from UI (users list) and on clicking on No confirmation box will be closed.
  - Edit Button: On clicking on edit button a new page should open which will have a form (same as Create New User form above but without password field) with fields (Name, Email and Phone Number) prefilled and Update button.
    - On clicking on update button user details will be updated and we will be redirected back to Home page where we can see the updated user details.
- Users created by Logged in user will be displayed in the list. i.e. if a user has created 5 users
   then after login that user will be seeing those 5 users only and not the users created by others.