

## **Assignment: Building a Posts and Comments System with React and JSON Server**

### **Setup:**

- Create a new React application using create-react-app.
- Set up a JSON Server to serve as a mock backend for handling user data, posts, and comments.

### **User Registration and Login:**

- Implement a user registration form where users can enter their profile\_url, name, email, phone number, password, and confirm password.
- Validate user input to ensure all fields are filled correctly.
- Store user registration data in a JSONServer database.
- Implement a user login form where users can enter their email or phone and password.
- Validate user login credentials against the JSONServer database.
- Upon successful login, maintain a logged-in user session using local storage.

### **User Profile Management:**

- Create a user profile page for each registered user.
- Display the user's profile\_url, name, email, phone number, and registration date.
- Provide an option for users to edit their profile information.
- Upon profile update, save the changes to the JSONServer database.
- Display a list of all posts created by the user on their profile page.

**Post Creation and Management:**

- Create a component for posting new content.
- Allow users to create new posts with image\_url, title and body content.
- Associate each post with the currently logged-in user.
- Store post data in the JSONServer database.
- Display a list of all posts on the main page.
- Implement a mechanism to delete posts by the logged-in user who created.

**Comment Creation and Management:**

- Allow users to comment on existing posts.
- Associate each comment with the corresponding post and the logged-in user.
- Store comment data in the JSONServer database.
- Display comments below each post.
- Implement a mechanism for users to delete their own comments.

**Routing:**

- Use React Router to handle navigation between different pages (e.g., home, login, registration, user profile).

**Authentication Guard:**

- Implement a higher-order component or a custom hook to guard routes that require authentication. Unauthorized users should be redirected to the login page.

**Error Handling:**

- Implement proper error handling for user registration, login, and API requests.

**Styling:**

- Apply styles to make the application visually appealing and user-friendly.

**Additional Challenges (Optional):**

- Implement token for more secure authentication.
- Add pagination for the posts or comments to improve performance.

**Submission:**

- Submit a ZIP file containing the source code of your React application.
- Include a README file explaining how to run your application, any additional information about your project, and how you used array methods and the `fetch()` method.