Angular or React.js & Node.js User Connection System: Practical Description

Overview: The practical involves creating a user connection system with both frontend and backend components. The primary functionalities include user signup, login, profile completion, a home page with a list of registered users, a follow/unfollow mechanism, user details page, and a personal profile page displaying follower and following details.

Frontend:

1. Signup & Login:
   * Users must sign up and log in to access the system.
2. Profile Completion Wizard:
   * After login, users are directed to a wizard to complete their profile.
   * The wizard includes fields for display name, address, profile picture, and other relevant details.
3. Profile Completion Requirement:
   * Users must complete the profile wizard to access other pages.
   * Incomplete profiles restrict access to further functionalities.
4. Redirect to Home Page:
   * Upon completing the profile wizard, users are redirected to the home page.
5. Home Page:
   * Displays a list of registered users on the site.
   * Provides follow/unfollow buttons next to each user.
6. User Details Page:
   * Clicking on a user's profile leads to a details page.
   * Follow/unfollow buttons are available on this page as well.
   * Follower and following counts are displayed.
7. Own Profile Detail Page:
   * Users have a dedicated page for their own profile details.
   * Displays follower and following counts.
   * Lists all followers and those being followed.

Backend:

1. API Creation:
   * Develop APIs to handle user signup and login.
   * Implement endpoints for profile completion, home page user list retrieval, follow/unfollow actions, and user details retrieval.
2. Data Storage:
   * Store user data securely, including profile details, follower/following relationships, and login credentials.
3. Authentication:
   * Implement secure authentication mechanisms to protect user data.
4. Profile Completion Validation:
   * Ensure that profile completion is validated before allowing access to other functionalities.
5. Follow/Unfollow Logic:
   * Implement logic to handle follow and unfollow actions between users.
6. User Details Retrieval:
   * Develop APIs to retrieve user details for the home page and individual user details pages.
7. Personal Profile Information:
   * Create APIs to fetch follower and following counts, as well as lists of followers and those being followed for a user's own profile.

This practical assesses the candidate's ability to integrate frontend and backend components, handle user authentication, and implement various user-related functionalities.