

# Practical Exam - MERN Stack Developer

## ***Project Requirements:***

Develop a **User Management System** with the following features:

### **1. User Registration**

- Create a **registration form** with the following fields:
  - Name (required, min 3 characters)
  - Email (required, valid format)
  - Password (required, min 8 characters, one uppercase, one number)
  - Confirm Password (must match password)
  - Role (Static roles: **Admin, User, Moderator**)
- Validate all fields on the frontend before submission.
- Store user data securely in MongoDB (password must be hashed).
- Show error messages when validation fails.
- Display success message after successful registration.

### **2. User Login (Role-Based Access Control)**

- Implement **role-based login** with JWT authentication.
- Redirect users based on their role:
  - **Admin** → Can view all users.
  - **User** → Can only see their own profile.
  - **Moderator** → Can search and view users but cannot modify.
- Protect routes based on roles.
- Show error messages for incorrect credentials.

### **3. User Listing with Search and Pagination**

- Fetch users from the database and display them in a paginated table.
- Implement a **search bar** where users can type an **ID**, and the search should work only when the complete ID is entered (not for partial search).
- Pagination must be implemented with **next, previous, and page numbers**.

### **4. Implement Redux for State Management**

- Use Redux to manage authentication and user listing.
- Store logged-in user details in Redux.

- Store user list data in Redux.
- Use Redux actions to fetch users, search users, and handle pagination.

## Technology Stack:

- **Frontend:** React, Redux Toolkit, React Router, TailwindCSS (optional)
- **Backend:** Node.js, Express.js, MongoDB, JWT for authentication
- **Database:** MongoDB (Mongoose ORM)