MERN Blogging Application:

---

1. Environment Setup:

- MongoDB: Set up a local or cloud-based MongoDB instance.
- Node & Express: Initialize a new Node project, and set up an Express server.
- React: Initialize a new React app using Create React App or another preferred method.

---

2. Backend Development (Node, Express, MongoDB):

- **Database Models**:

    - User: With fields such as `username`, `email`, `hashedPassword`, `role`, and `createdBlogs`.
    - Blog: With fields such as `title`, `content`, `author`, `comments`, and `status`.
    - Comment: With fields like `text`, `author`, and `blogID`.

- **API Routes**:

    - User: Signup, Login, Logout, and GetProfile.
    - Blog: Create, Update, Delete, Get (all and by ID).
    - Comment: Create, Update, and Delete.

- **Authentication & Authorization**:

    - Implement JWT for authentication.
    - Middleware for role-based access controls.

---

3. Frontend Development (React):

- **Core Components**:

    - Navbar: With links based on user's authentication and role.
    - Home: Displaying a list of blogs.
    - BlogView: For reading a full blog and its comments.

- **User Pages**:

    - Signup & Login.
    - Profile Page: Displaying user information and blogs authored.

- **Admin Features**:

    - User Management: View all users, upgrade/downgrade roles, delete users.
    - Blog Management: View all blogs, edit, or delete any blog.

- **Author Features**:

    - Blog Creation: A page or modal to create new blogs.
    - Blog Management: Edit or delete their own blogs.

- **Moderator Features**:

  - Comment Moderation: Approve, edit, or delete any comment.
  - Blog Moderation: Approve or reject blogs before they're publicly visible.

- **Common User Features**:

  - Comment: Add comments to blogs.
  - Blog Viewing: View and search for all approved blogs.

- **State Management**:

  - Use Context API, Redux, or another state management tool to handle user authentication, blog data, etc.

- **Routing**:

  - Implement client-side routing with protected routes based on user roles and authentication.

---

4. Testing & Deployment:

- **Backend**:

  - Write unit tests for API routes using a tool like Mocha, Chai, or Jest.
  - Deploy backend to a platform like Heroku, AWS, or any preferred hosting.

- **Frontend**:

  - Write unit tests for critical components.
  - Deploy frontend to a platform like Vercel, Netlify, or any preferred hosting.

---

5. Additional Enhancements:

- Responsive design.
- Implement blog categories or tags.
- Add a rich text editor for blog creation.
- Implement blog analytics (views, likes, etc.).