

# MERN Stack Blog with Authentication - Project Guide

This document outlines the step-by-step guide and project map for building a blog application using the MERN stack (MongoDB, Express.js, React.js, Node.js) with user authentication. This guide avoids coding and instead provides a clear architectural and implementation overview.

## 1. Project Setup

- Create the main project folder (e.g., mern-blog).
- Create two subfolders: 'client' (for React frontend) and 'server' (for Express backend).

## 2. Backend Setup (Node + Express + MongoDB)

- Initialize a Node.js project using npm.
- Install necessary dependencies like Express, Mongoose, Bcrypt, JWT, CORS, dotenv.
- Connect to MongoDB using Mongoose.
- Setup environment variables for database URI and JWT secret.
- Create an Express server and enable CORS and JSON parsing.

## 3. User Authentication (Backend)

- Design a User model with fields like username, email, and password.
- Implement a registration route that hashes the password and saves user data.
- Implement a login route that verifies credentials and returns a JWT token.
- Create middleware to verify JWT token for protected routes.

## 4. Blog Post Management (Backend)

- Design a Blog model with fields like title, content, author, userId, createdAt.
- Implement routes to create, read, update, and delete blog posts.
- Protect blog creation, update, and delete routes using JWT middleware.

## 5. Frontend Setup (React + Vite + Bootstrap)

- Initialize a React project using Vite.

- Install dependencies like Axios, React Router DOM, and Bootstrap.
- Setup routing structure using React Router.

## **6. Pages & Components (Frontend)**

- Pages: Home, Login, Register, Create Blog, Edit Blog, My Blogs.
- Components: Navbar, BlogCard, BlogForm (reusable).

## **7. Authentication Handling (Frontend)**

- Allow user registration and login via backend API.
- Store JWT token in localStorage after login.
- Send token in Authorization header for protected API requests.
- Implement logout functionality.

## **8. CRUD Blog Features (Frontend)**

- Fetch and display all blog posts publicly.
- Allow logged-in users to create, update, and delete their blog posts.
- Display only users own blogs in the 'My Blogs' section.

## **9. Route Protection (Frontend)**

- Create a PrivateRoute component that checks if the user is authenticated.
- Redirect users to login if they try to access protected routes without a token.
- Modify the UI to show/hide options based on login status.

## **10. Polish & Final Touches**

- Add loading indicators and error handling.
- Add toast alerts or notifications.
- Improve responsive UI using Bootstrap.
- Optionally add categories, comments, or likes.