**Scope Document**

**1. Project Overview**

This project is a MERN (MongoDB, Express, React, Node.js) stack application, structured with two main folders:

* **Client**: The frontend of the application, built using React.
* **Server**: The backend of the application, built using Express and Node.js, and connected to MongoDB for database management.

The primary objective of this project is to provide a robust, full-stack web application that facilitates seamless interaction between the client and server, ensuring high performance and scalability.

**2. In-Scope**

The scope of this project includes the following:

**Functional Features:**

1. User Authentication and Authorization:
   * Login and registration functionality.
   * Secure password storage using bcrypt.
2. CRUD Operations:
   * Create, Read, Update, and Delete functionality for core entities.
3. Database Integration:
   * MongoDB integration for storing and retrieving data.
   * Data validation and schema enforcement using Mongoose.
4. Frontend Features:
   * A responsive and intuitive user interface designed with React.
   * State management (using React hooks)
5. API Development:
   * RESTful APIs to handle client-server communication.
   * Proper error handling and response codes.

**Non-Functional Requirements:**

1. Performance:
   * The application should handle concurrent users efficiently.
2. Scalability:
   * Design to accommodate future enhancements or increased user load.
3. Security:
   * Implementation of HTTPS.
   * Authentication using JWT
4. Deployment:
   * Deployment of the server on a platform like Render.
   * Deployment of the client on platforms like Render or Vercel.

**3. Out of Scope**

The following items are beyond the scope of this project:

1. Third-party integrations or plugins not explicitly mentioned in the requirements.
2. Advanced analytics or reporting systems.
3. Extensive custom UI/UX designs beyond the standard React implementation.
4. Real-time features such as live chatting or notifications.

**4. Objectives**

1. Deliver a fully functional MERN stack web application with the specified features.
2. Ensure secure and efficient communication between the client and server.
3. Provide clear documentation for installation, usage, and future enhancements.
4. Ensure deployment readiness and provide an easily accessible application for end-users.

**5. Stakeholders**

1. **Project Manager**: Oversees the project timeline and deliverables.
2. **Developers**: Responsible for implementing the client and server components.
3. **End Users**: Individuals or entities who will use the application upon completion.

**6. Dependencies**

1. **Technical Tools**:
   * Node.js and npm for backend development.
   * MongoDB for database management.
   * React for frontend development.
2. **Development Environment**:
   * A code editor like VS Code.
   * Postman for API testing.
3. **Hosting Services**:
   * Render (or equivalent) for backend deployment.
   * Render or Vercel for frontend deployment.
4. **Browsers**:
   * Modern browsers like Chrome, Firefox, or Edge for testing.

**7. Assumptions**

1. All required third-party libraries or dependencies will be installed during the setup process.
2. The client and server will communicate over a secure API.
3. MongoDB is configured and accessible during development and deployment.
4. The development team has access to hosting services for deployment.

**8. Limitations**

1. The application is not optimized for legacy browsers.
2. Advanced scaling and high-availability configurations are not included in the initial scope.
3. Limited to CRUD operations and basic user management for this phase.

**9. Future Enhancements**

1. Integration of additional features such as real-time notifications or live chat.
2. Migration to a microservices architecture for scalability.
3. Addition of advanced analytics and reporting tools.
4. UI/UX redesign based on user feedback.