Test Assignment: (MERN, AWS, ThreeJS)

Objective:
Create a full-stack web application that allows users to visualize and manipulate 3D objects using Three.js, with user authentication and data persistence. The backend should be built using Node.js, Express, and MongoDB (MERN stack) hosted on AWS, with the frontend rendering the 3D scene and handling user interactions.

Project Requirements:

1. 3D Visualization and Manipulation (Three.js):

- Develop a 3D object viewer using **Three.js**.
- Allow users to **upload a 3D object file** (e.g., `.obj` or `.glb`) and visualize it within the application.
- Implement basic object manipulation features (rotation, zoom, pan).
- Users should be able to save **camera positions** or any other interaction states, which should persist across sessions.

2. User Authentication:

- Implement user registration, login, and logout functionality using **JWT (JSON Web Token)** for authentication.
- Restrict 3D manipulation functionality to authenticated users only.
- Store user data and interaction states in **MongoDB** (deployed on AWS).

3. Data Persistence (AWS + MongoDB):

- The application must store 3D object information, user interaction states (e.g., camera angles), and user authentication data in **MongoDB**.
- Host the MongoDB instance and the application backend on **AWS** (use services like EC2, S3, or RDS as necessary).
- Provide **auto-scaling** and **load balancing** configurations in the backend for optimal performance.

4. Frontend (React):

- The frontend should be developed using **React**.
- Ensure a clean, user-friendly UI for uploading 3D objects, interacting with them, and managing user sessions.
- Use **React hooks** and follow best practices for state management (such as using **Context API** or **Redux** for global state).

5. Backend (Node.js + Express):

- Develop the backend using **Node.js** and **Express.js**.
- Expose RESTful APIs for user authentication, object upload, and saving interaction states.
- Ensure the backend code is modular, well-structured, and optimized for performance.

6. Deployment and Hosting:

- Host the application on **AWS** using **EC2** for the server, **S3** for any static file hosting, and **MongoDB** for database.
- Ensure **high availability** and **scalability** of the application using AWS services.
- Use **CloudFront** or other AWS services to optimize the delivery of content.

Submission Deliverables:

1. **Application URL**:
   The application should be hosted and accessible via a public URL. Provide the link where the project is hosted and running live on AWS.
2. **Technical Documentation**:
   A detailed technical document explaining the following:
   - The architecture of the application (including backend, frontend, and deployment structure).
   - A diagram showing how various services (React, Express, MongoDB, AWS) interact with each other.
   - Highlights of optimization techniques used in the project (e.g., caching, lazy loading, code splitting).
   - How security measures (e.g., JWT, HTTPS, password encryption) are implemented.
   - Explanation of the cloud setup on AWS, including auto-scaling, load balancing, and data persistence strategies.
3. **Video Walkthrough**:
   A short video (5-10 minutes) explaining the project features and showing the working application. The video should demonstrate:
   - User registration, login, and logout.
   - Uploading and manipulating 3D objects.
   - Saving interaction states and persistence across sessions.
   - AWS deployment overview.

Evaluation Criteria:

- **Code Quality & Optimization**: Clean, modular, well-documented code following industry best practices. Proper optimization in both frontend and backend.
- **Architecture & Scalability**: The solution should be architected for scalability, making good use of AWS services.
- **Security**: Proper implementation of security measures such as JWT, HTTPS, and secure password storage.
- **Creativity and UI/UX**: Intuitive and user-friendly interface for the 3D visualization tool.
- **Documentation & Video**: Clear and concise documentation with an easy-to-understand video walkthrough of the application.

Note:
Make sure the application is production-ready, highly performant, and optimized for a scalable deployment. You will be judged on your ability to lead development and deliver robust solutions that scale effectively while maintaining code quality and security standards.

Submission Deadline: **Within 5 days of task assignment.**

NO CODE SUBMISSION PLEASE
