

## Project Design Phase

### Solution Architecture

Date	15 February 2025
Team ID	Team -145206
Project Name	Sociopedia
Maximum Marks	4 Marks

### Solution Architecture:

This report outlines the technical architecture of a full-stack social media web application built using the MERN stack. The platform allows users to register, authenticate, create posts, comment, like, and interact with each other. The architecture ensures scalability, modularity, and maintainability.

### 2. Project Objectives

- Build a responsive social media web application
- Enable secure user authentication and authorization
- Implement CRUD operations for posts and comments
- Store and serve multimedia files (profile pics, post images)
- Deploy the full application with scalability in mind

### 3. High-Level Architecture

The system consists of:

- Frontend (React.js): UI built with reusable components.
- Backend (Node.js & Express): RESTful APIs for business logic and data handling.
- Database (MongoDB): NoSQL storage for users, posts, comments.
- Authentication (JWT, Passport.js): Secure session management.
- File Storage (Cloudinary or AWS S3): For image/media hosting.
- Deployment (Heroku or AWS): Application hosting.

### 4. Component Breakdown

#### 4.1 Frontend (React.js)

- Uses React Router for navigation
- Axios for API communication
- Redux (optional) for state management
- Responsive design with CSS/Styled Components

#### 4.2 Backend (Node.js + Express)

- RESTful API structure
- Express middleware for routing, error handling, auth checks
- JSON parsing and CORS handling

### **4.3 Database (MongoDB)**

- NoSQL schema using Mongoose
- Key collections:
  - Users (username, email, password, bio, avatar)
  - Posts (text, image, user, timestamp)
  - Comments (post, user, content, timestamp)
  - Chats

### **4.4 Authentication**

- JWT tokens stored in localStorage
- Bcrypt for password hashing
- Role-based access (admin/user)

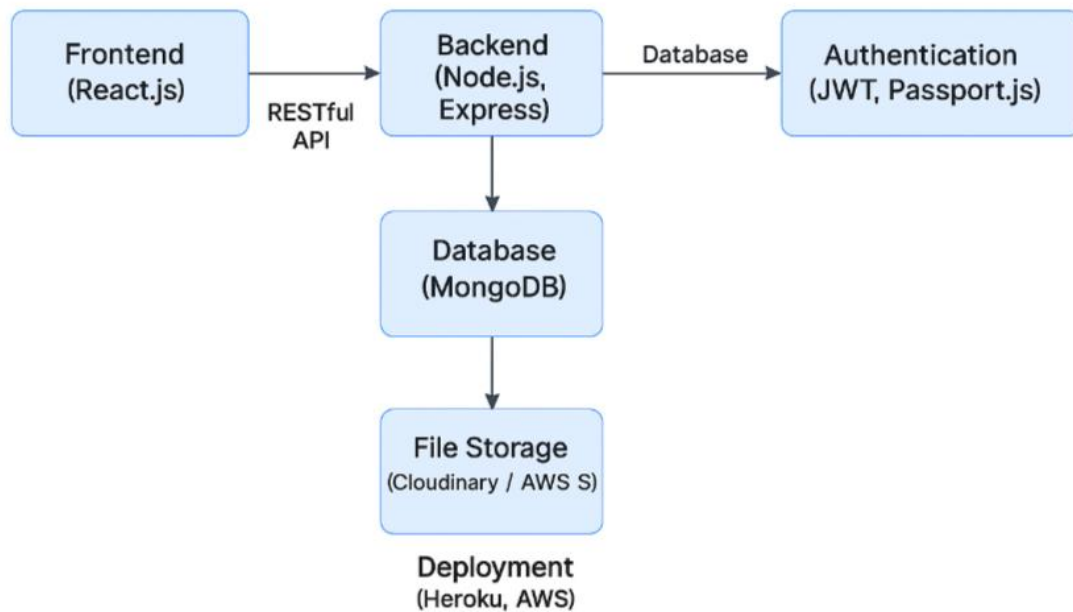
### **4.5 File Storage**

- Integration with Cloudinary for handling uploads
- Multer middleware to handle image uploads

## **5. Data Flow**

1. User interacts with React UI
2. React sends API requests via Axios to Express backend
3. Express handles business logic and interacts with MongoDB
4. Auth tokens are used for protected routes
5. Media files are sent to Cloudinary, and URLs are stored in the DB

# Social Media MERN Solution Architecture



## 6. Development Workflow

- Git for version control
- GitHub for repo management and collaboration
- ESLint + Prettier for code linting
- Postman for testing APIs

---

## 7. Conclusion

This solution architecture provides a solid foundation for building and scaling a full-featured social media application. The separation of concerns, modular structure, and use of cloud-based services ensure flexibility and scalability for future enhancements.

**Reference:** <https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/>