**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 1st, April 2025 |
| Team ID | SWTID1742493942 |
| Project Name | Connectify - Social Media App |
| Maximum Marks | 4 Marks |

**Solution Architecture**

Below is a high-level overview of the architecture of the system:

**1. Client Layer (Frontend)**

* **React.js** based SPA (Single Page Application)
* **Axios** used for API communication
* **Tailwind CSS** for responsive and consistent styling

**Features:**

* User interface for creating posts, viewing feeds, uploading stories
* Interactive story display and reminder management
* Responsive design for both desktop and mobile views

**2. Server Layer (Backend)**

* **Node.js + Express.js** server for handling logic and routing
* Manages API endpoints for posts, stories, reminders, and user data
* Middleware used for request parsing and security
* Implements **JWT** for secure user session management

**3. Business Logic Layer**

* Handles user operations such as:
  + Post/story creation
  + Reminder scheduling
  + Feed sorting based on timestamps and user interactions
* Ensures authorization checks for content editing/deletion

**4. Database Layer**

* **MongoDB (Atlas)** used for storing:
  + User profiles and authentication data
  + Posts, stories, and reminders
* **Mongoose** for schema definition and query abstraction
* Indexed fields (e.g., timestamps, userId) for optimized feed loading
* Data relationships maintained through ObjectId references

**5. Storage & Media Handling**

* Uses **Cloudinary** or **Firebase Storage** for:
  + Uploading and serving images/media in posts and stories
  + Media compression and optimization for performance

**6. Authentication & Security**

* **JWT** tokens for maintaining authenticated sessions
* **Bcrypt** for hashing user passwords
* Input validation to prevent XSS/SQL injection attacks
* CORS and HTTPS for secure cross-origin communication

