**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 10 April 2025 |
| Team ID | Team ID : SWTID1743172790 |
| Project Name | Social Media Web App |
| Maximum Marks | 4 marks |

**Solution Architecture:**

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

* Find the best tech solution to solve existing business problems.
* Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
* Define features, development phases, and solution requirements.
* Provide specifications according to which the solution is defined, managed, and delivered.

**Solution Architecture of Project:**

1. **Client:**

* User: Represents the end-user who interacts with the Social Media Web app.
* Social Media Web App: The front-end application accessed by the user through their browser.

1. **Browser:** The web browser used by the client to access the web application.
2. **WebServer:**

* Node.js: The JavaScript runtime environment used for building the server-side of the application.
* Express.js: A web application framework for Node.js, used to handle HTTP requests and route them to the appropriate handlers.

1. **AppServer**:

* React.js: The JavaScript library used for building the user interface of the web app. React components are rendered in the browser, providing a dynamic and responsive user experience.

1. **Database**:

* MongoDB: The NoSQL database used to store user information, post and history data, and other relevant data for the social media web app.

1. **ExternalServices**:

* It includes API endpoints for features like authentication, post management, and stories. No third-party authentication (e.g., OAuth) or external APIs (like cloud storage or email services) are explicitly listed in the repository**.**

**Interaction Flow:**

1. **Client Interaction:**

* The user interacts with the social media web app through their browser.

1. **Request Flow:**

* The browser sends an HTTP request to the WebServer (Node.js with Express.js) to access the web app.

1. **User Actions:**

* The user interacts with the React components in the app
* These interactions result in API calls from the React app to the WebServer.

1. **Data Operations:**

* The WebServer handles these API calls and performs CRUD (Create, Read, Update, Delete) operations on the MongoDB database.

1. **External Services Interaction:**

* It includes API endpoints for features like authentication, post management, and stories
* The fetched data is saved in the MongoDB database for future use and quick access.

1. **Response Flow:**

* The WebServer processes the data and returns the API responses to the React app.
* The React app updates the user interface based on the responses, providing the user with real-time data and feedback.

**Solution Architecture Diagram:**

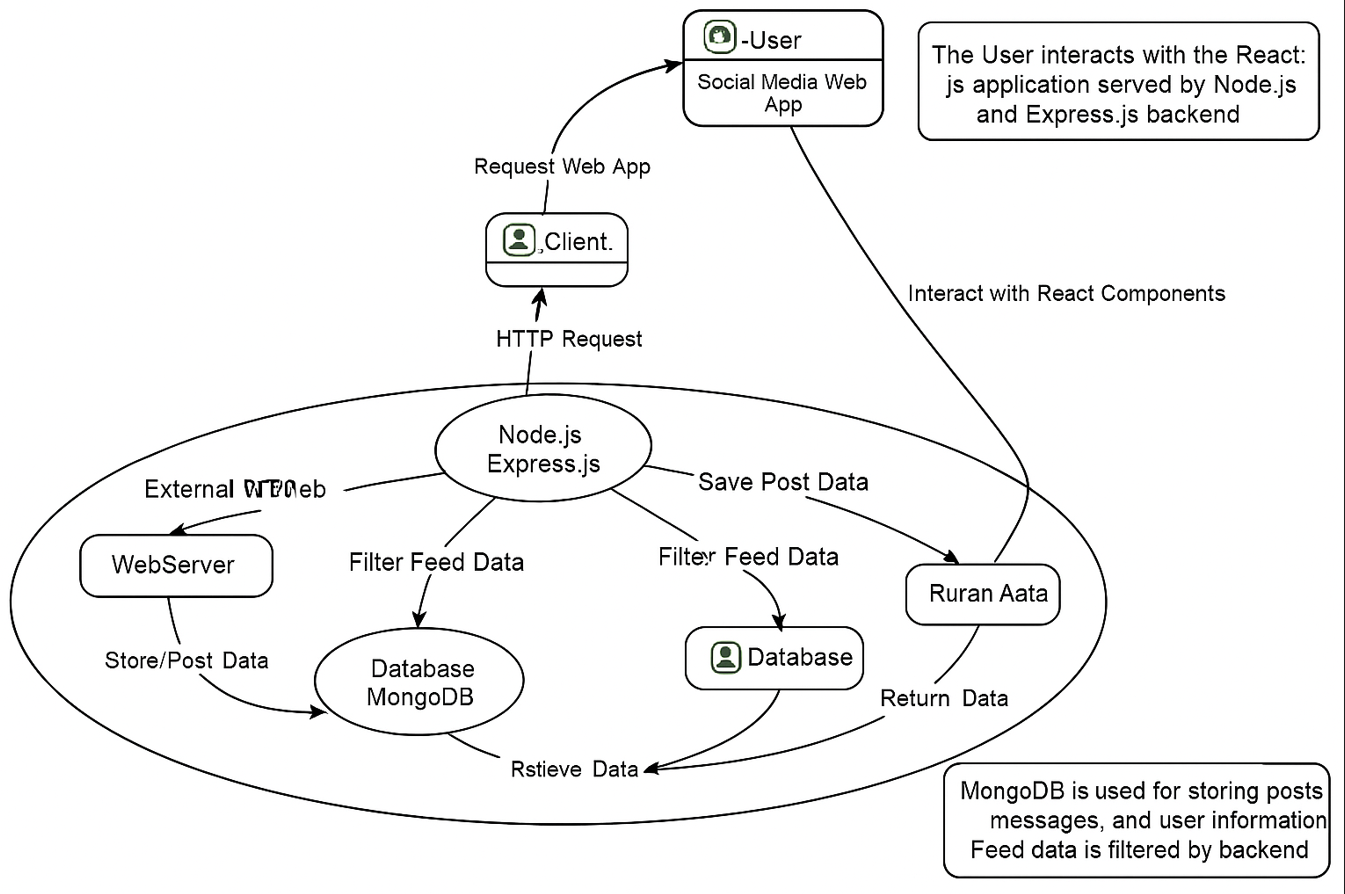
****

Figure 1: Architecture and data flow of the social media web application.