**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 02 -10-2025 |
| Team ID | **SWUID20250216599** |
| Project Name | Smart Meet |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

The **Solution Architecture** of the Video Conferencing App defines the overall structure of the system, showing how different components interact to provide real-time communication, secure data handling, and a smooth user experience. The architecture is based on a **client-server model** integrated with real-time media streaming technologies

**1. Client Layer (Frontend)**

* Built using **React.js** for a dynamic and responsive user interface.
* Provides functionalities like **user authentication, meeting creation/joining, chat, screen sharing, and media controls**.

**2. Application Layer (Backend)**

* Developed using **Node.js** and **Express.js** to manage business logic and API requests.
* Handles **user authentication**, **meeting management**, **real-time signaling**, and **chat handling**.

**3. Media Server Layer**

* Handles the **real-time transmission of audio and video streams** using **WebRTC** protocols (STUN/TURN servers).
* Supports recording, screen sharing, and multi-participant conferencing.

**4. Database Layer**

* Utilizes **MongoDB** to store data such as user credentials, meeting details, chat logs, and recording links.
* Provides scalability and quick data retrieval for live sessions.
* Ensures data integrity through proper indexing and access control.

**5. Cloud & Storage Layer**

* Cloud services (like AWS, Firebase, or Azure) store **recorded meetings and shared files**.
* Supports backup and retrieval for future access.
* CDN (Content Delivery Network) ensures fast media delivery worldwide.

**6. Security Layer**

* Implements **JWT (JSON Web Token)** for secure authentication.
* Uses **end-to-end encryption** for audio/video streams and **HTTPS** for API calls.
* Ensures privacy by restricting unauthorized access to meetings.

