Project Design Phase-I
Proposed Solution Template

Date	22 June 2024
Team ID	PNT2022TMID1720113552
Project Name	Project - Video Conferencing Application
Maximum Marks	3 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
	Problem Statement (Problem to be solved)	Effective distant communication is critical, but current video conferencing systems are costly, difficult to integrate, and lack vital collaborative capabilities. This necessitates the development of a platform that is accessible, feature-rich, and easy to use.
	Idea / Solution description	We propose a scalable video conferencing web application built on the MERN stack (MongoDB, Express.js, React, and Node.js). It will include high-quality video and audio communication, screen sharing, chat, and collaboration features like as whiteboarding and document sharing, all available from any device with an internet connection.
	Novelty / Uniqueness	Our solution distinguishes out because: Real-time Collaboration: Integrated tools for white-boarding and document editing. Scalability: Uses the MERN stack to deliver efficient, scalable performance. Customisation: Open source, easily customisable, and compatible with various services. Security features end-to-end encryption and strong authentication.
	Social Impact / Customer Satisfaction	Our tool improves productivity and connectivity by enabling remote collaboration for businesses, schools, and social groups. We provide a positive user experience and strong customer support with an easy interface and great

	Business Model (Revenue Model)	We use a freemium approach with basic functions for free and paid memberships for additional features such as increased participant limitations and greater security. In-app purchases, enterprise solutions, and collaborations all contribute to increased income.
	Scalability of the Solution	Our MERN stack application is designed for high scalability. MongoDB handles large data efficiently, Node.js supports numerous simultaneous connections, and React ensures a responsive user interface. The architecture allows for seamless horizontal scaling, enabling the addition of more servers to manage increased loads without compromising performance.