

SMARTBRIDGE
Final Project Documentation

Video Conferencing App
ConvoConnect

Submitted by:

Patel Shubh Bhargavkumar (22BCE10218)

Devansh Jadon (22BCE10134)

Pratham Amit Singh (22BCE10959)

VIT BHOPAL UNIVERSITY
April 2025

1. INTRODUCTION

- **Project Title:** Convoconnect: Simplifying Video Conferencing
- **Team Members:** 1. Patel Shubh Bhargavkumar
2. Devansh Jadon
3. Pratham Amit Singh

2. PROJECT OVERVIEW

- **Purpose:** Convoconnect is a modern web-based video conferencing platform designed to streamline virtual communication. With the rise in remote work, online education, and virtual meetings, users require tools that are fast, reliable, and easy to use. Convoconnect eliminates common barriers such as sign-up processes, software installations, and complex interfaces, offering an instant and secure connection.

- **Features:**

- One-click room creation and sharing
- Real-time video and audio using WebRTC
- Secure room links with optional passcode protection
- In-room group chat functionality
- Screen sharing capabilities
- Lightweight and responsive UI for all devices
- Peer-to-peer connection for high-quality calls

3. ARCHITECTURE

- **Frontend:** The frontend is developed using React.js with functional components and React Router for navigation. Tailwind CSS is used for responsive and consistent styling. The UI is designed to be minimal and accessible, ensuring ease of use for all age groups.
- **Backend:** The backend is powered by Node.js and Express.js, handling API requests, managing room metadata, and enabling real-time communication through Socket.IO. It supports secure room creation, connection signaling, and message relays.
- **Database:** MongoDB, a NoSQL database, is used to persist room data and optional chat logs. Mongoose ODM simplifies schema creation and query logic for managing the database effectively.

4.SETUP INSTRUCTIONS

- **Prerequisites:** Node.js, MongoDB
- **Installation:**
 1. Clone the repository using `git clone`
 2. Navigate to the client and server directories respectively
 3. Run `npm install` to install dependencies
 4. Set environment variables in a `.env` file for backend configuration
 5. Use `npm start` in both directories to run the application

5. FOLDER STRUCTURE

- **Client:** Organized into components, context providers, and pages. Components include the video player, chat box, control buttons, and layout utilities.
- **Server:** Structured with routes, controllers, models, and real-time communication logic for maintaining WebSocket connections.

6. RUNNING THE APPLICATION

- **Frontend:** Run ``npm start`` inside the client directory
- **Backend:** Run ``npm start`` inside the server directory

Ensure MongoDB is running locally or set the appropriate remote database URI in ``.env``.

7. API DOCUMENTATION

- **POST /api/room** - Creates a new video room and returns a unique room ID and join URL
- **GET /api/room/:id** - Fetch room details including creation time and status
- **POST /api/message** - (Optional) Store messages exchanged in chat

8. AUTHENTICATION

Currently, Convoconnect supports lightweight authentication in the form of room passcodes. Users can set a password when creating a room which must be entered by others to join. For enhanced security, JWT-based user authentication is planned for future versions.

9. USER INTERFACE

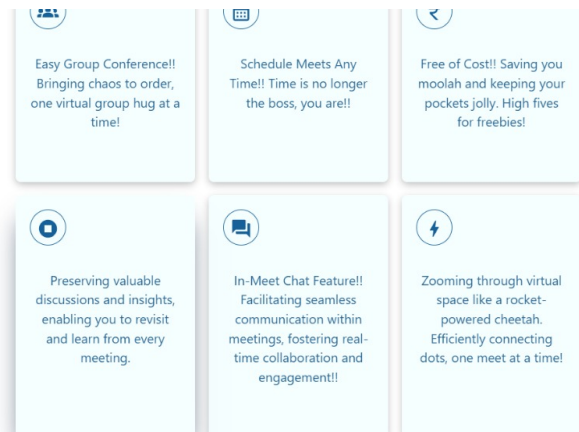
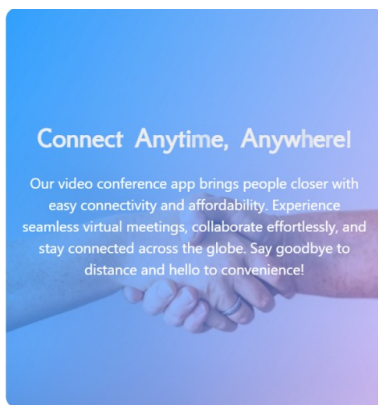
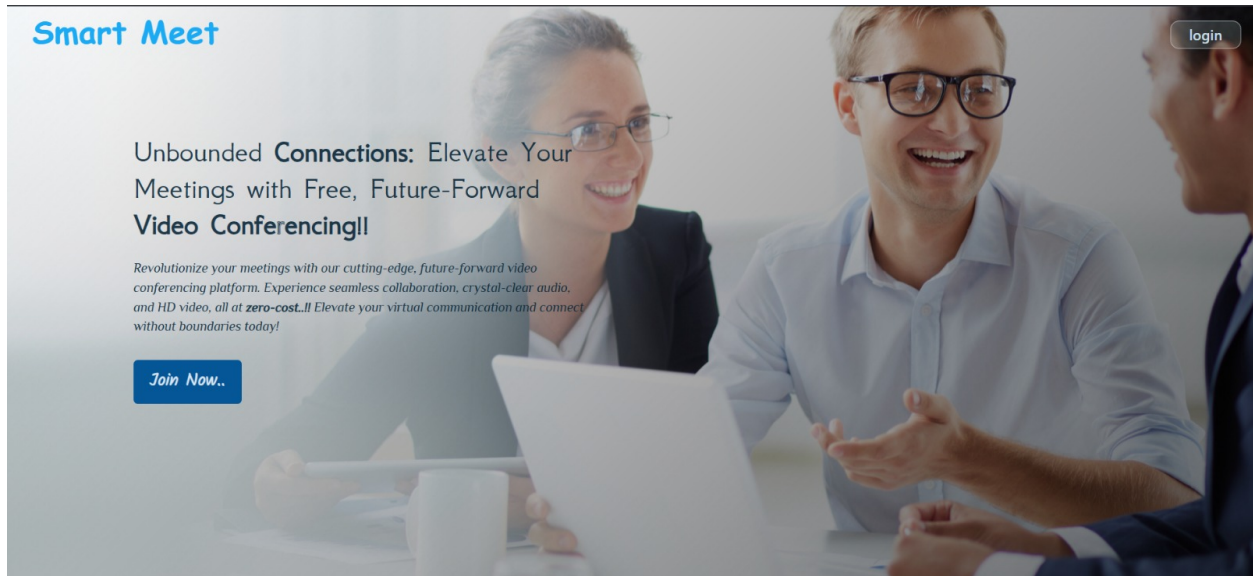
The interface is kept clean and focused with essential controls such as video toggle, microphone control, screen sharing, and end-call options. The room UI adjusts seamlessly across device sizes, ensuring a smooth experience on mobile, tablet, and desktop devices.

10. TESTING

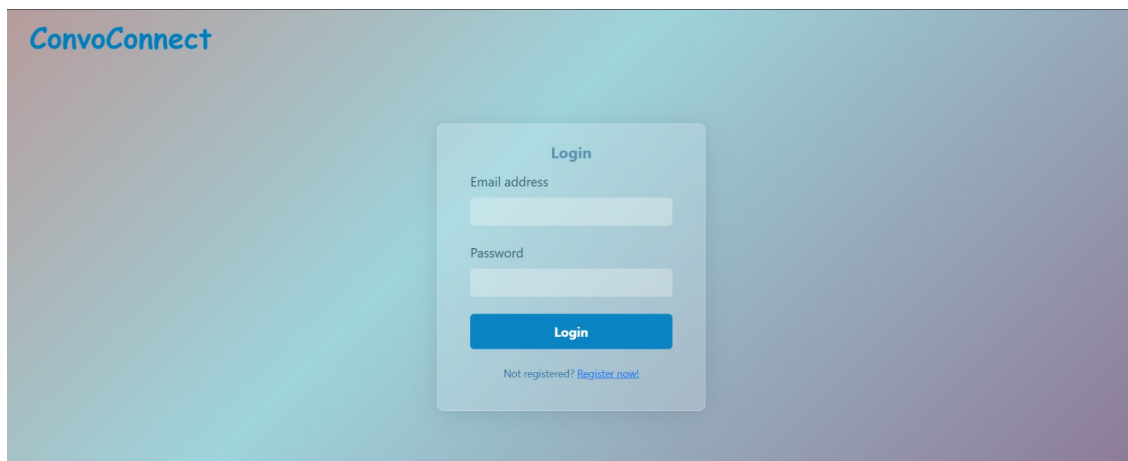
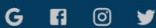
- Manual testing was conducted on Chrome, Firefox, and Edge to ensure compatibility.
- Responsiveness tested on mobile and tablet simulators.
- Functional components tested using Jest and React Testing Library (where applicable).
- Socket.IO communication tested with multiple peer connections.

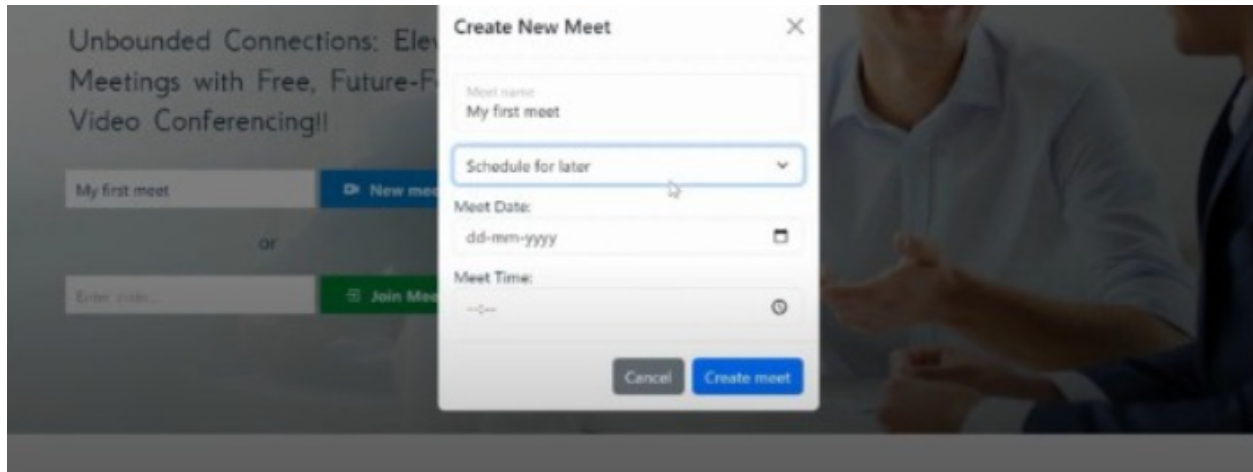
11. SCREENSHOTS OR DEMO

• Screenshots:



Contact us @:





Live Demo:

<https://www.loom.com/share/45e4503863fe4241815ef20ac62e72de?sid=36e7625d-9846-4de0-9a30-52d2f85a84b7>

12. KNOWN ISSUES

- Screen sharing may not be fully supported in Safari.
- Mobile devices may restrict microphone access in some conditions.
- No session recording implemented yet.

13. FUTURE ENHANCEMENTS

- Integration of meeting recording and cloud storage
- AI-based live transcription and automatic meeting summaries
- Breakout rooms for group collaboration
- User authentication and profile management
- Polls and file sharing inside the room