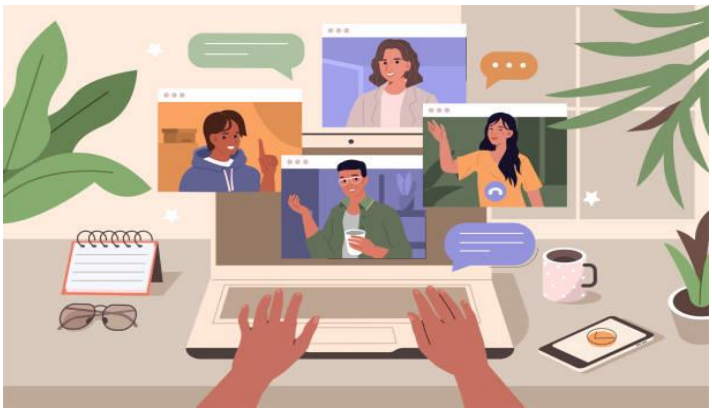
FULL STACK DEVELOPMENT WITH MERN

Project Title :

ConvoConnect : Simplifying Video Conferencing



Submitted By :

Velithota Yamini – 218X1A04G8

Mahankali Harish – 218X1A04E7

Vadde Poojitha – 218X1A04G5

Yalagala Lakshmi Tanuja – 218X1A04H4

Vemula Kiran Kumar – 218X1A04I5

List of contents :

1.Project overview

ConvoConnect: Simplifying Video Conferencing

Introducing our revolutionary video conference app! Designed to redefine remote collaboration, our webapp offers an intuitive and innovative platform for seamless communication.

Experience a new level of connectivity with our group video conference feature. Whether you're working with colleagues, connecting with friends, or hosting virtual events, our webapp ensures a smooth and immersive experience that brings everyone together.

Break down barriers and enhance collaboration with real-time screen sharing. Whether you're delivering dynamic presentations, collaborating on projects, or providing hands-on demonstrations, our screen sharing feature allows for seamless interaction and understanding.

Never miss a moment with our convenient meet recording functionality. Capture important discussions, presentations, or brainstorming sessions for later playback and sharing with absent participants. Your valuable insights are preserved, ensuring nothing gets lost in translation.

Stay engaged and connected with in-meet chat. Instantly communicate, share links, or ask questions alongside the live audio and video streams, making collaboration efficient and effective.

Effortlessly plan and organize your video conferences with our meet scheduling feature. Set up meetings, invite participants, and send automated reminders, ensuring everyone is prepared and punctual for productive sessions.

Your privacy and security are paramount. Our app employs robust encryption to safeguard your data, ensuring all communications remain confidential and protected from unauthorized access.

Step into a new era of remote collaboration and communication. Discover the unmatched convenience of seamless video conferencing, innovative screen sharing, meet recording, in-meet chat, and meet scheduling features – all within our game-changing video conference app. Connect, collaborate, and achieve more together!

Purpose :

The purpose of video conferencing apps is to allow people to communicate in real time, regardless of their location, using video and audio streaming:

- **Connect people**

Video conferencing apps allow people to connect with each other, even if they're in different locations.

- **Enhance collaboration**

Video conferencing apps can help with collaboration by allowing people to see each other's nonverbal communication, which can lead to richer context and stronger relationships.

- **Save money**

Video conferencing can save money on travel by allowing people to communicate without having to travel long distances.

- **Boost team spirit**

Video conferencing can help boost team spirit by making it easier for people to collaborate on projects and discussions.

- **Improve accountability**

Video conferencing can help improve accountability among employees by making it easier for colleagues to see each other.

Video conferencing apps can be used for a variety of purposes, including:

- **Team meetings**
- **Webinars**
- **Product demos**
- **Job interviews**
- **Virtual lectures**
- **Catching up with friends**
- **Hosting business meetings**

To use a video conferencing app, you'll need a device with a webcam, microphone, and speaker, as well as internet access. You'll also need to install or download the video conferencing app.

Features :

Some features of video conferencing apps include:

Screen sharing

A prominent feature that allows users to collaborate on a project by sharing their screens

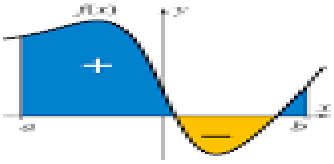
Chat

Allows participants to exchange messages, create personal conversations, and ask questions

Security

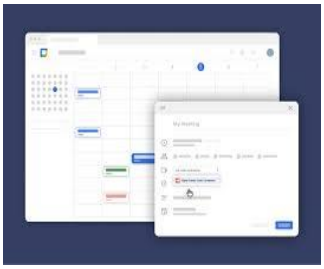
Most cloud video conferencing solutions use end-to-end encryption to protect users' data

Integrations



Allows users to connect with others around the world without being limited by time zones or geographical location

Scheduling



Allows users to schedule meetings with others

Video and audio quality

Provides high-quality audio and video to ensure smooth interaction

Video call recording

Allows users to record and transcribe meetings for future reference

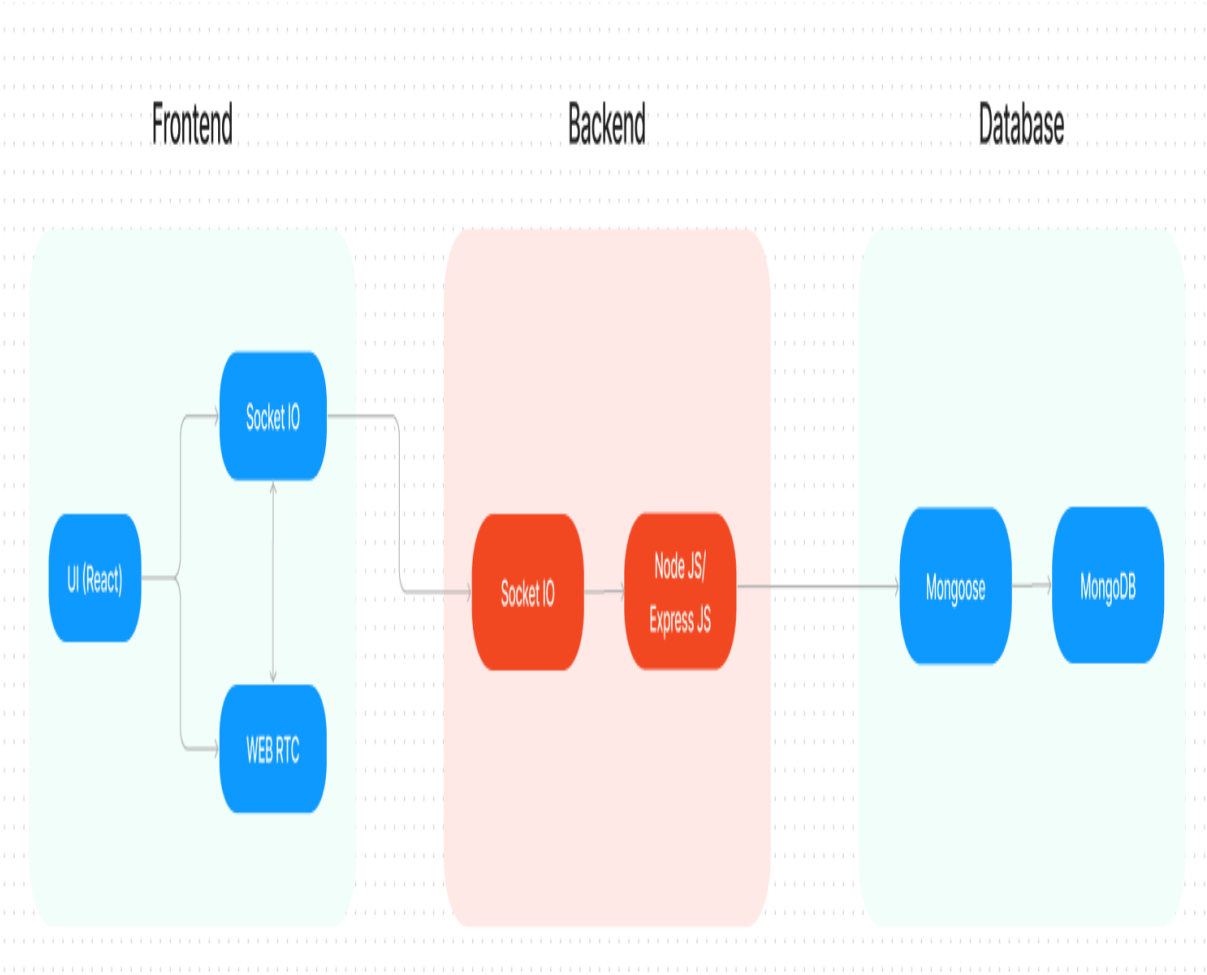
File sharing

Allows participants to share documents with others for viewing and editing

Other features of video conferencing apps include:

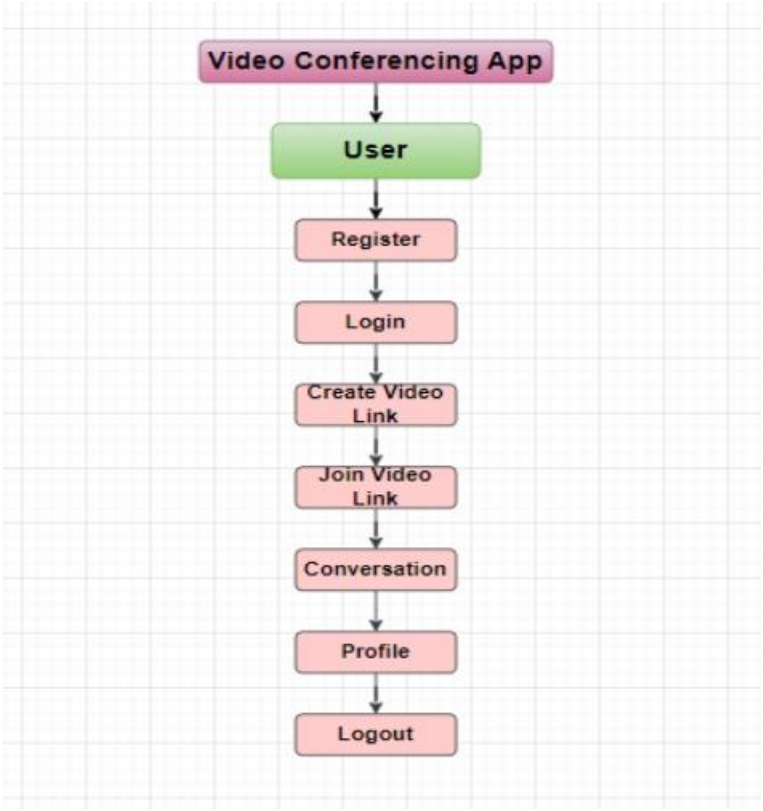
- **User management**
- **Mobile support**
- **Secure server**
- **Firewalls**
- **Authentication**
- **Data encryption**
- **Active speaker view**
- **Switching between speakers**
- **Live streaming**

2.Architecture



The technical architecture of our video conference app follows a client-server model, where the frontend serves as the client and the backend acts as the server. The frontend encompasses not only the user interface and presentation but also incorporates the socket.io-client and WebRTC API.

Work Flow of Proposed system:



3.Setup Instructions

Prerequisites :

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, React.js, Socket.io, Agora RTC, and Agora RTM:

- **Node.js and npm:**

  Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server-side. It provides a scalable and efficient platform for building network applications.

  Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

- **Download:** https://nodejs.org/en/download/
- **Installation instructions:** https://nodejs.org/en/download/package-manager/

- **Express.js:**

  Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture.

  Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.

  Installation: Open your command prompt or terminal and run the following command:

  ```
  npm install express
  ```

- **MongoDB:**

  MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

  Set up a MongoDB database to store your application's data.

- **Download: https://www.mongodb.com/try/download/community**
- **Installation instructions: https://docs.mongodb.com/manual/installation/**

.React.js:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

Follow the installation guide: https://reactjs.org/docs/create-a-new-react-app.html

- **Socket.io:**

  Socket.io is a real-time bidirectional communication library that enables seamless communication between the server and clients. It allows for real-time data exchange, event-based messaging, and facilitates the development of real-time applications such as chat, collaboration, and gaming platforms.

Install Socket.io, a real-time bidirectional communication library for web applications.

Installation :

- **Open your command prompt or terminal of server and run the following command:**

  npm install socket.io

- **Open your command prompt or terminal of client and run the following command:**

  npm install socket.io-client

- **Agora RTC:**

  Agora RTC (Real-Time Communication): Agora RTC provides a platform for real-time audio and video communication. It offers a range of features like video conferencing, live streaming, and interactive broadcasting, enabling developers to create immersive and interactive communication experiences.

- **Sign up for an Agora developer account to access their RTC platform.**
- **Integration guide and documentation: https://docs.agora.io/en/**

- **Agora RTM:**

  Agora RTM enables real-time messaging and data synchronization between users. It provides reliable and low-latency messaging capabilities, allowing developers to build chat applications, collaborative tools, and real-time notification systems.

- **Sign up for an Agora developer account to access their RTM platform.**
- **Integration guide and documentation: https://docs.agora.io/en/Real-time-Messaging/index.html**

- **HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.**

- **Database Connectivity: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations. To Connect the Database with Node JS go through the below provided link:**

  • https://www.section.io/engineering-education/nodejs-mongoosejs-mongodb/

- **Front-end Framework: Utilize Angular to build the user-facing part of the application, including products listings, booking forms, and user interfaces for the admin dashboard.**

- **Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.**

  • Git: Download and installation instructions can be found at: https://git-scm.com/downloads

- **Development Environment: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.**

  • Visual Studio Code: Download from https://code.visualstudio.com/download

  • Sublime Text: Download from https://www.sublimetext.com/download

  • WebStorm: Download from https://www.jetbrains.com/webstorm/download

  To run the existing Video Conference App project downloaded from Google drive:

  Follow below steps:

  .Download the code:

  • Download the code from the below link:

  https://drive.google.com/drive/folders/1c1SBTyz8HLUh6g_TfPdNrojvMI3KIK0g?usp=sharing

- **Install Dependencies:**

  • Navigate into the cloned repository directory:

  ```
  cd smart-meet
  ```

  • Install the required dependencies by running the following commands:

  ```
  cd client
  npm install
  cd ../server
  npm install
  ```

- **Start the Development Server:**

• **To start the development server, execute the following command:**

```
npm start
```

• **The video conference app will be accessible at http://localhost:3000**

- **Access the App:**

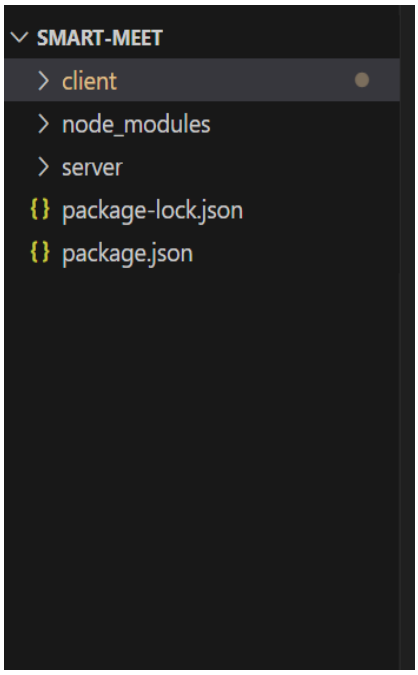
• **Open your web browser and navigate to http://localhost:3000.**

• **You should see the video conference app's homepage, indicating that the installation and setup were successful.**

You have successfully installed and set up the e-commerce app on your local machine. You can now proceed with further customization, development, and testing as needed.

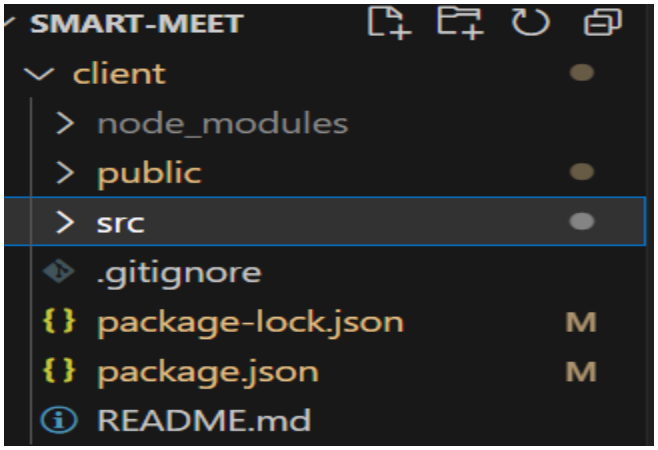
4.Folder Structure

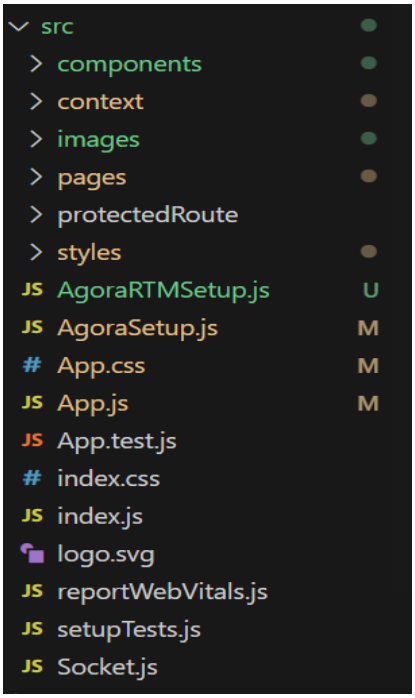
- **Inside the smart-meet (video conference app) directory, we have the following folders**

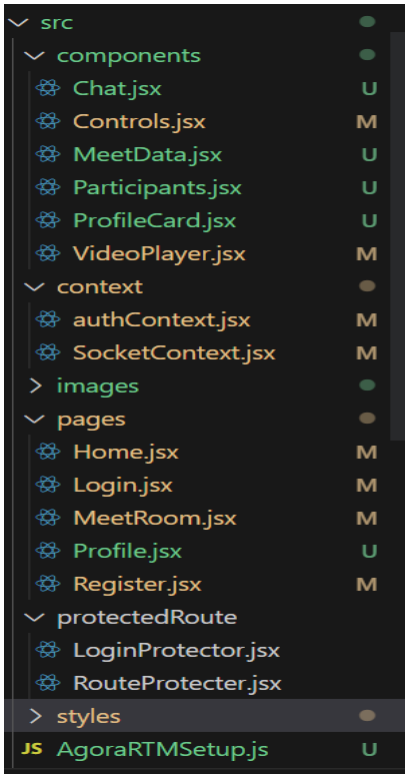


Client directory :

The below directory structure represents the directories and files in the client folder (front end) where, react js is used along with Api's such as socket.io and agora.

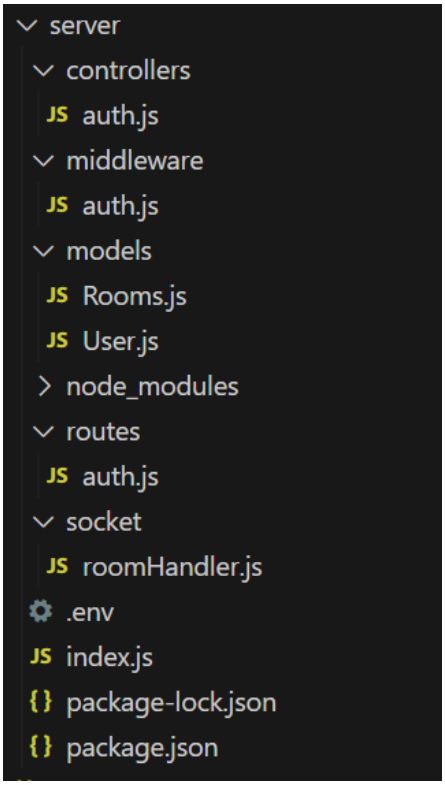






Server directory :

The below directory structure represents the directories and files in the server folder (back end) where, node js, express js and mongodb are used along with socket.io Api.


server
controllers
auth.js
middleware
auth.js
models
Rooms.js
User.js
node_modules
routes
auth.js
socket
roomHandler.js
.env
index.js
package-lock.json
package.json

5.Running the Application

provide commands to start the frontend and backend servers locally.

Frontend:

npm start in the client directory.

Backend:

npm start in the server directory.

6.API Documentation

Backend Development :

Setup express server

1. Create index.js file in the server (backend folder).
2. Create a .env file and define port number to access it globally.
3. Configure the server by adding cors, body-parser.

Create socket.io connection

1. Import socket.io in the index.js file.
2. Create server using the socket.io.
3. Start the server.

Configure MongoDB

1. Import mongoose.
2. Add Database URL to the .env file.
3. Connect the database to the server.
4. Create a 'models' folder in the server to store all the DB models.

Add authentication

1. Create the "User" model for the MongoDB.
2. Create auth controller file to control the authentication actions.
3. Import "bcrypt" – used to hash(encode) the password to make it secure.
4. Import "JWT" to create authentication tokens.

5. **Define registration & login activities.**
6. **Configure frontend & backend for authentication and store authenticated data in Context API in frontend.**

Create rooms

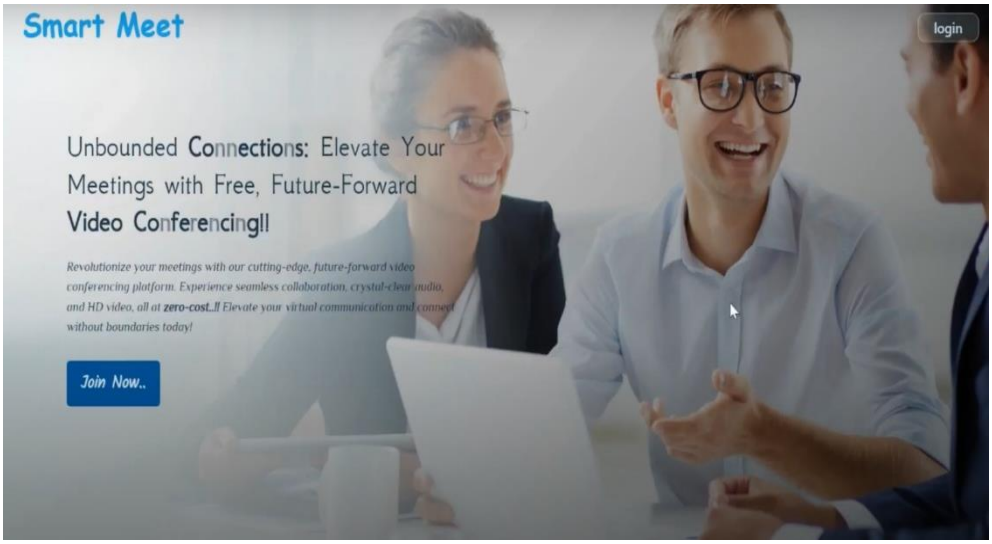
- **Create “Rooms” model to store data of rooms (meet rooms) in Database.**
- **Configure socket.io-client to make client-server communication simpler.**
- **Send message to server if client requested to create a room.**
- **Create a new room and add user to it.**
- **Allow other users to join the room.**
- **Manage user leaving the room.**
- **Use the calendar input, if the user wants to schedule meet for later.**

7.Authentication

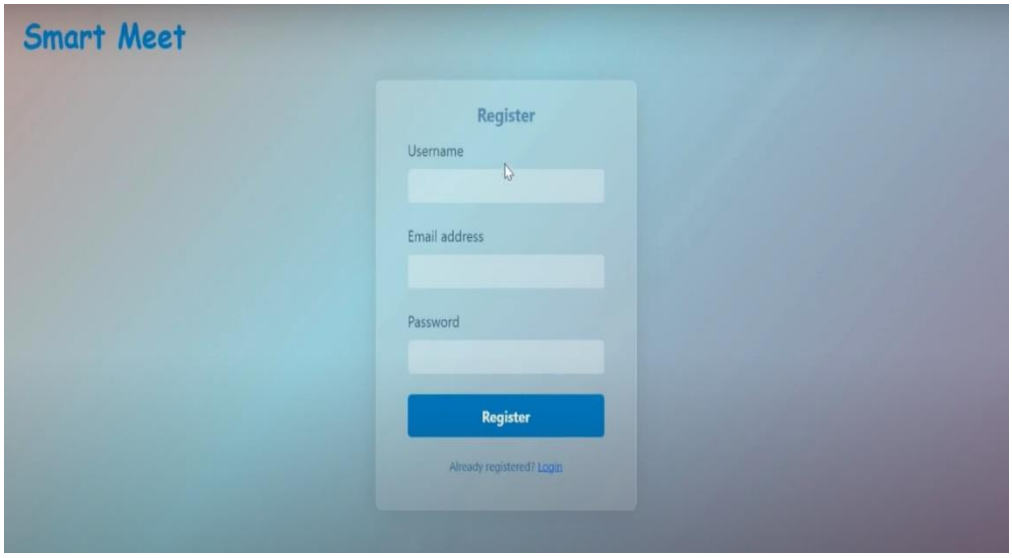
1. **Create the “User” model for the MongoDB.**
2. **Create auth controller file to control the authentication actions.**
3. **Import “bcrypt” – used to hash(encode) the password to make it secure.**
4. **Import “JWT” to create authentication tokens.**
5. **Define registration & login activities.**
6. **Configure frontend & backend for authentication and store authenticated data in Context API in frontend.**

8.User Interface

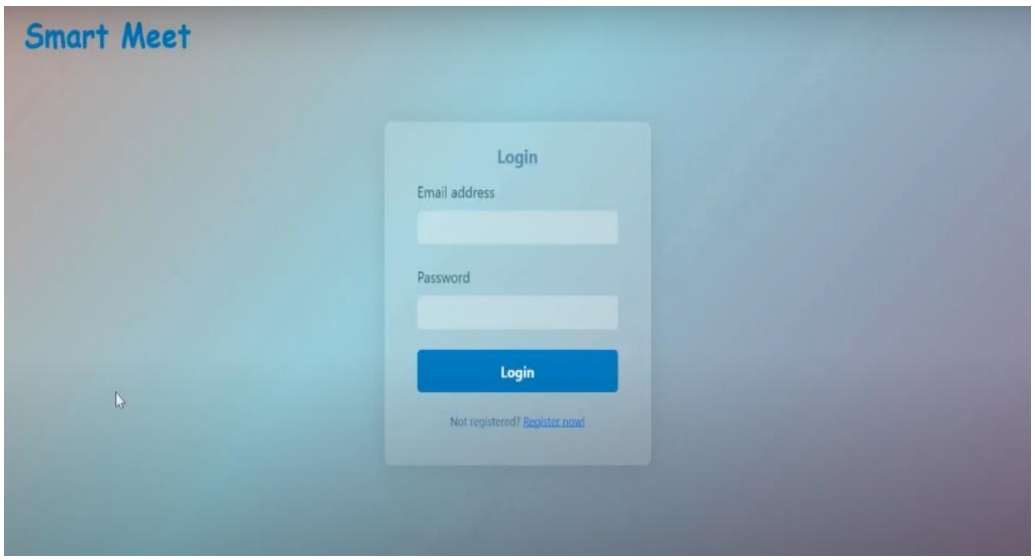
Landing page :



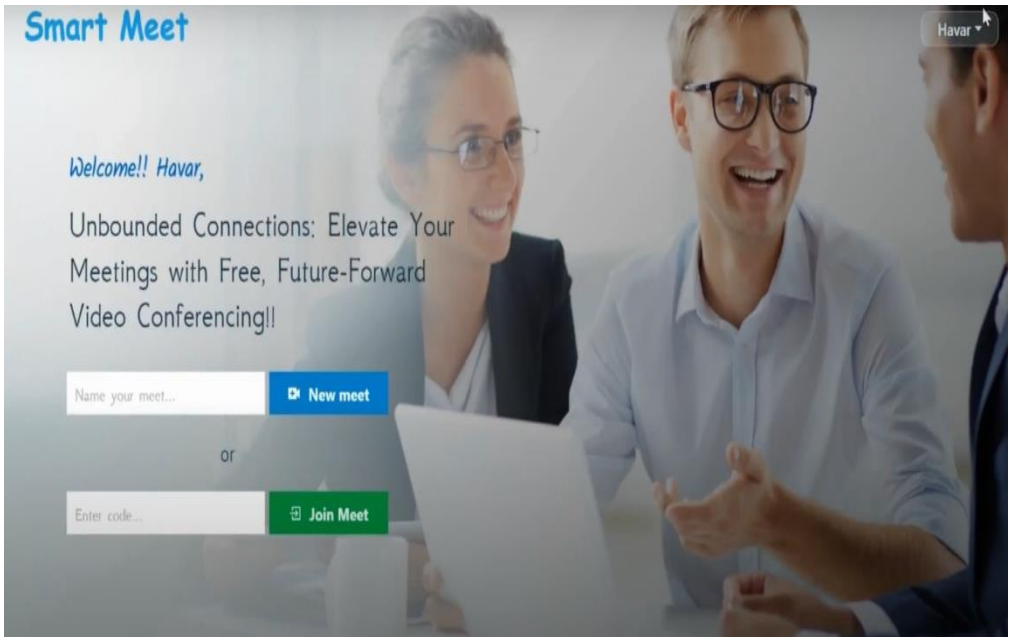
Registration page :



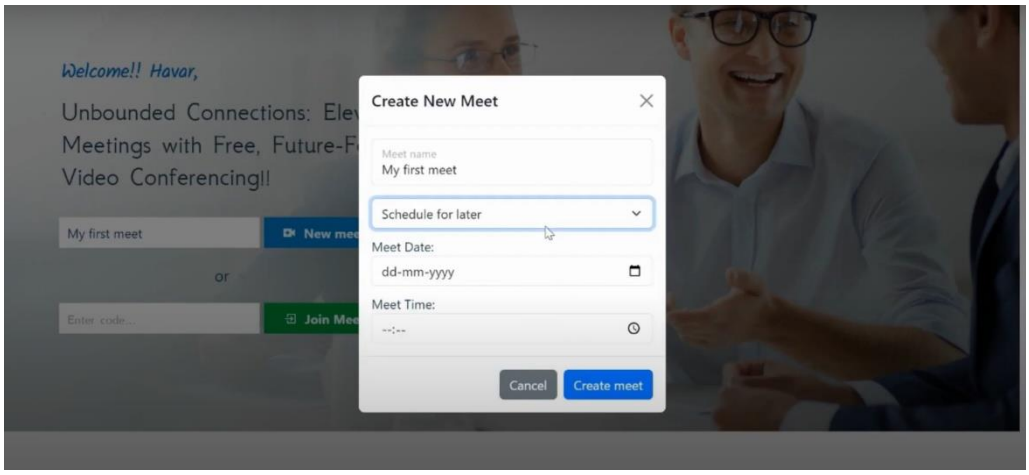
Login page :



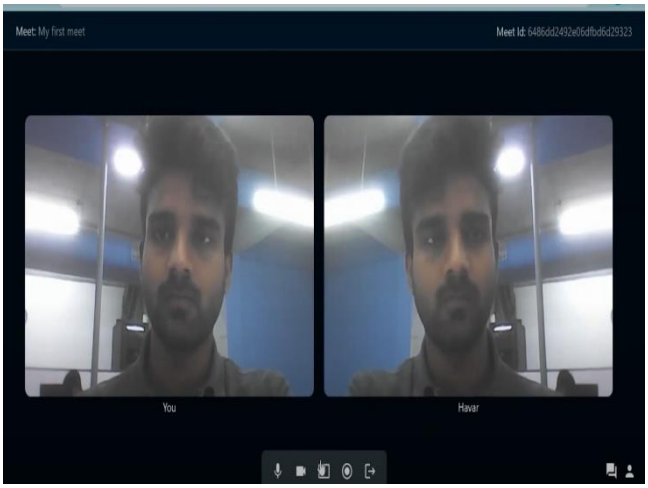
Home page :



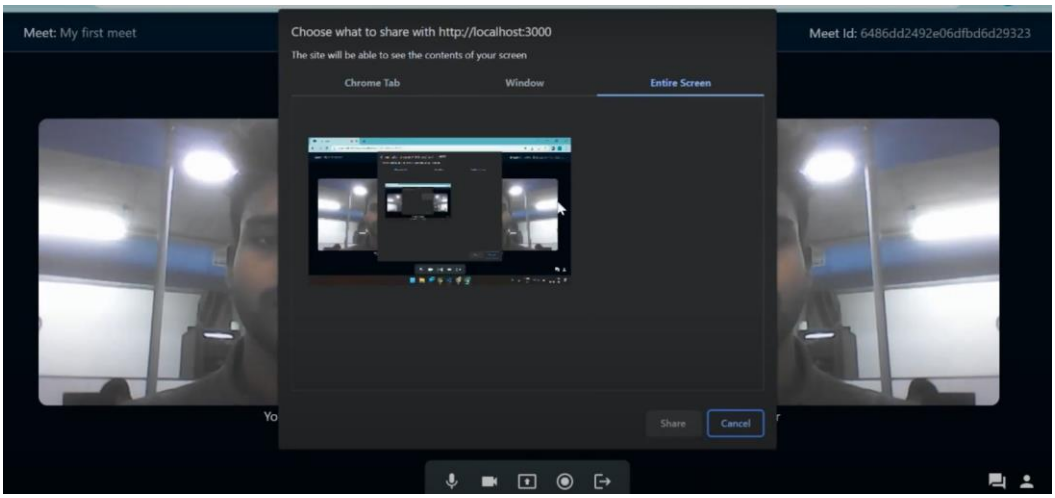
Creating new meet :



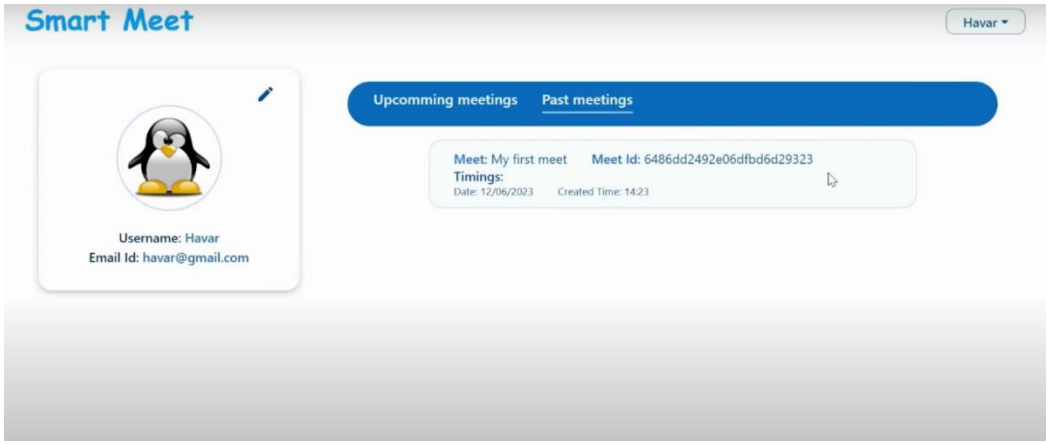
Meet page :



Screen sharing :



Profile page :



9.Testing

Test objectives

- **All field entries must work properly.**
- **Pages must be activated from the identified link.**
- **The entry screen, messages and rsesponses must not be delayed.**

Features to be tested

- **Verify that the entries are of the correct format**
- **No duplicate entries should be allowed**
- **All links should take the user to the correct page.**

Test cases :

User Module:

Test Case ID	Test Scenario	Precondition	Test Steps	Expected Result
TC01	Register a new user	User is on the registration page	1. Enter valid user details (e.g., username, email, password). 2. Click on "Register" button.	User is successfully registered, and a confirmation message appears.
TC02	Register with existing email	User is on the registration page	1. Enter details with an existing email.	Registration fails with an error message stating that

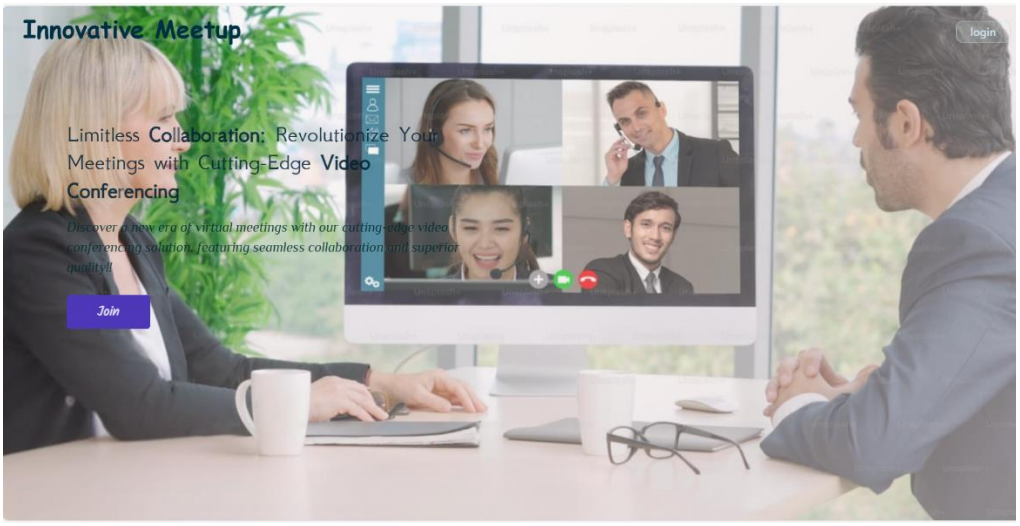
			2. Click on "Register" button.	the email is already in use.
TC03	Login with valid credentials	User is on the login page	1. Enter correct email and password. 2. Click on "Login" button.	User is successfully logged in and redirected to the dashboard.
TC04	Login with invalid credentials	User is on the login page	1. Enter incorrect email or password. 2. Click on "Login" button.	Login fails with an error message indicating incorrect email or password.
TC05	Create a video conference link	User is logged in	1. Navigate to the "Create Conference" section. 2. Click on "Generate Link" button.	A unique conference link is generated and displayed to the user.
TC06	Join a video conference	User has a valid conference link	1. Click on the shared conference link. 2. Wait for the	User successfully joins the video conference, and audio/video

			conference to load.	streams are established.
TC07	Join a conference with an invalid link	User has an invalid conference link	1. Click on an invalid or expired conference link. 2. Wait for the conference to load.	User receives an error message indicating that the link is invalid or expired.
TC08	View user profile	User is logged in	1. Navigate to the "Profile" section.	User profile details (e.g., name, email, etc.) are displayed.
TC09	Update profile information	User is logged in and on Profile page	1. Click on "Edit Profile". 2. Modify profile details. 3. Click on "Save" button.	Profile information is updated, and a success message is shown.
TC10	Logout from the application	User is logged in	1. Click on the "Logout" button.	User is logged out, and the session is terminated, redirecting to the login page.

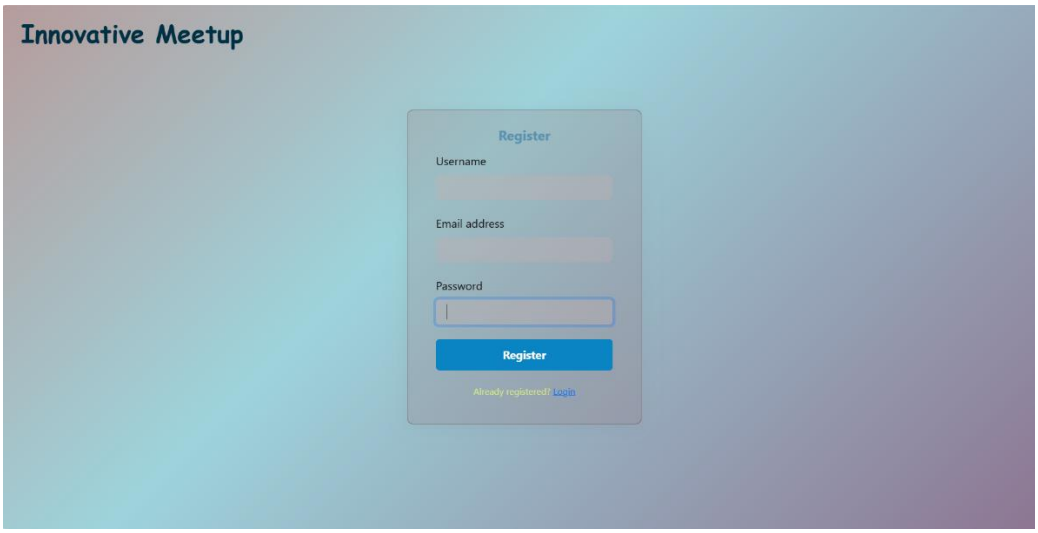
10.Screenshots or Demo

Finally, after finishing coding the projects we run the whole project to test it's working process and look for bugs. Now, let's have a final look at the working of our video conference application.

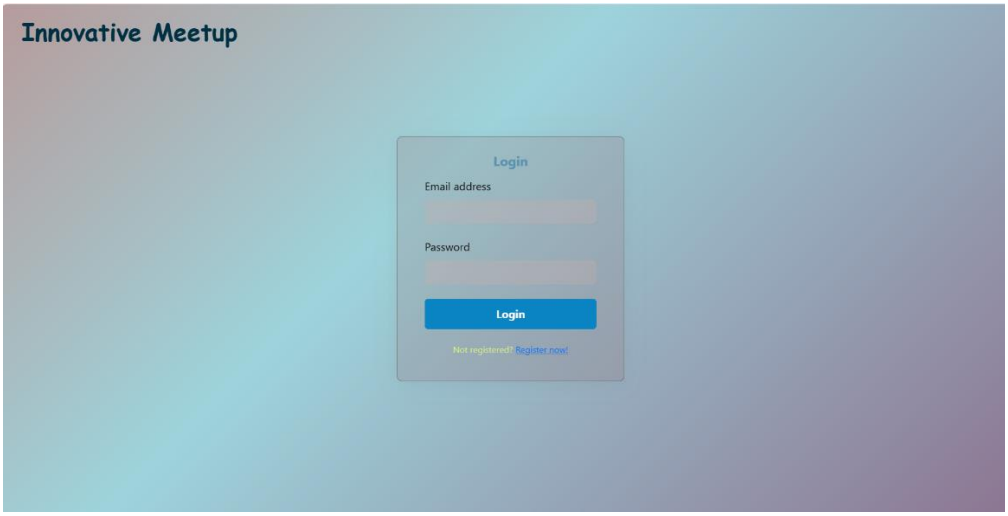
Landing page :



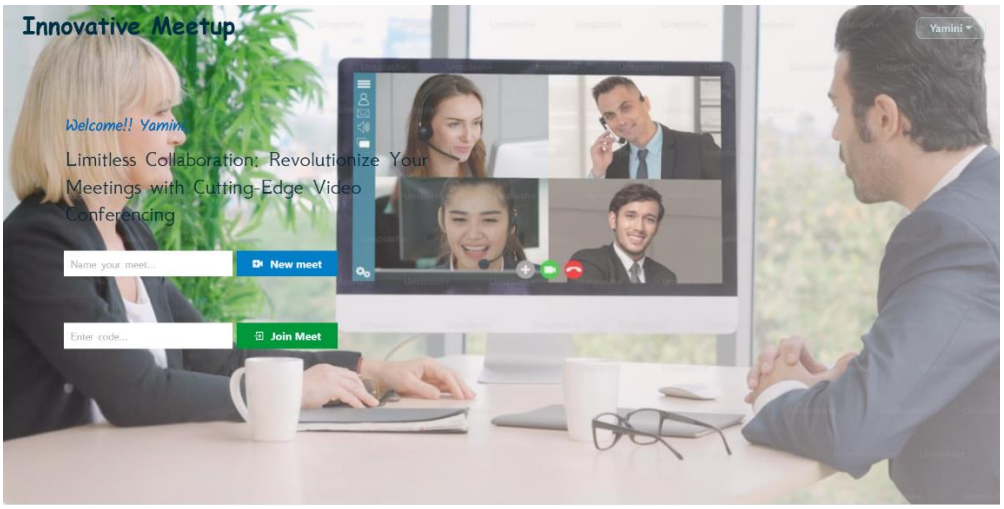
Registration :



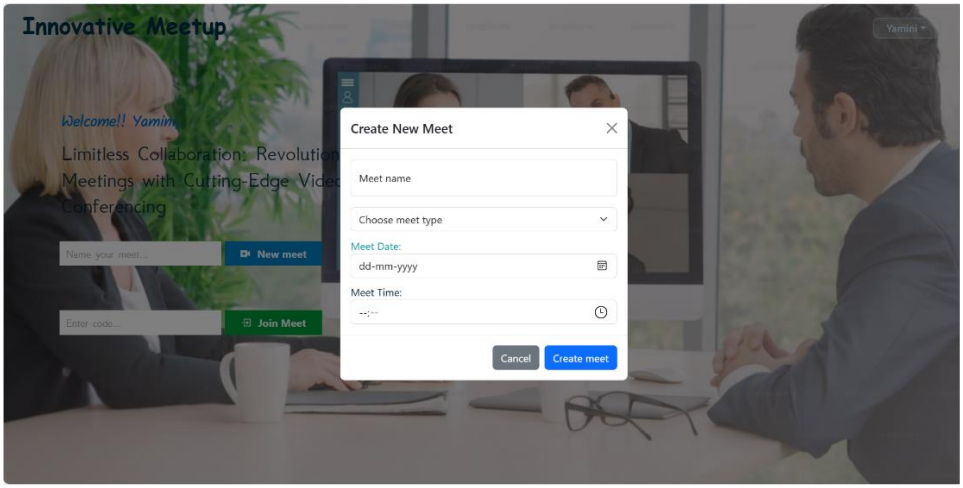
Login page :



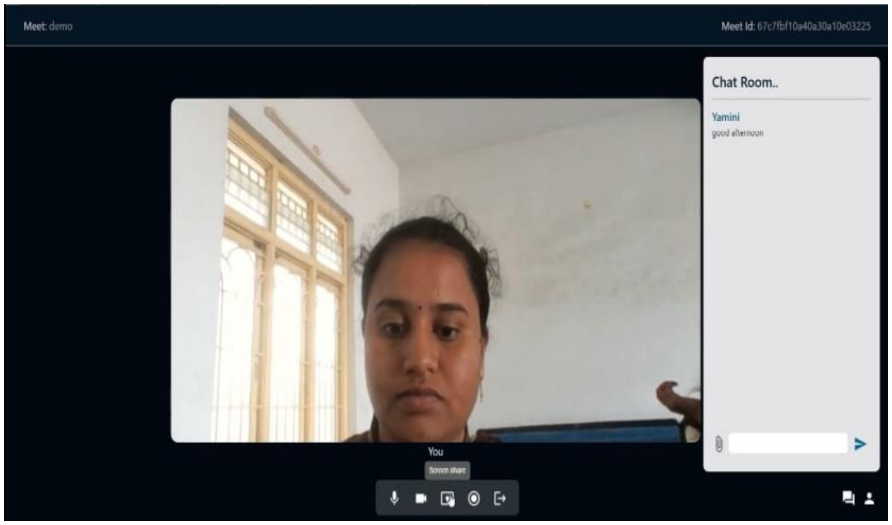
Home page :



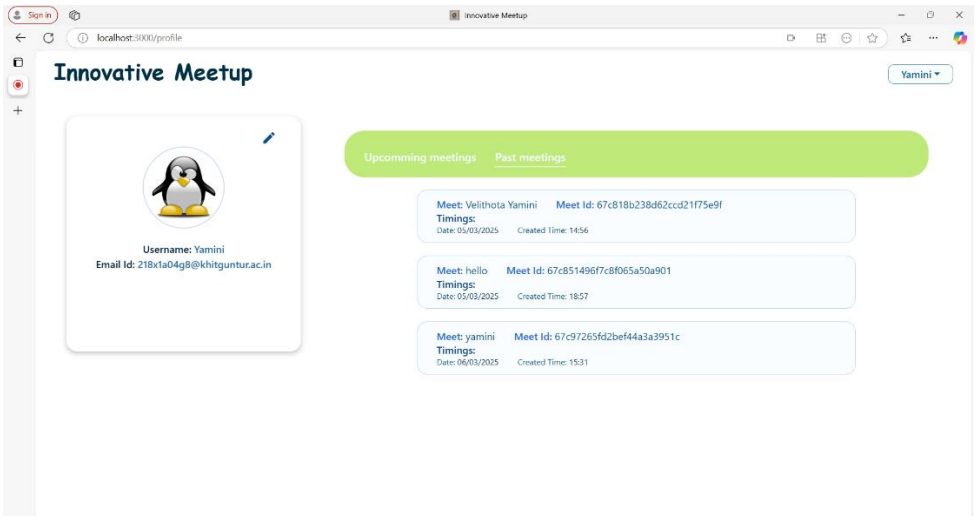
Creating new meet :



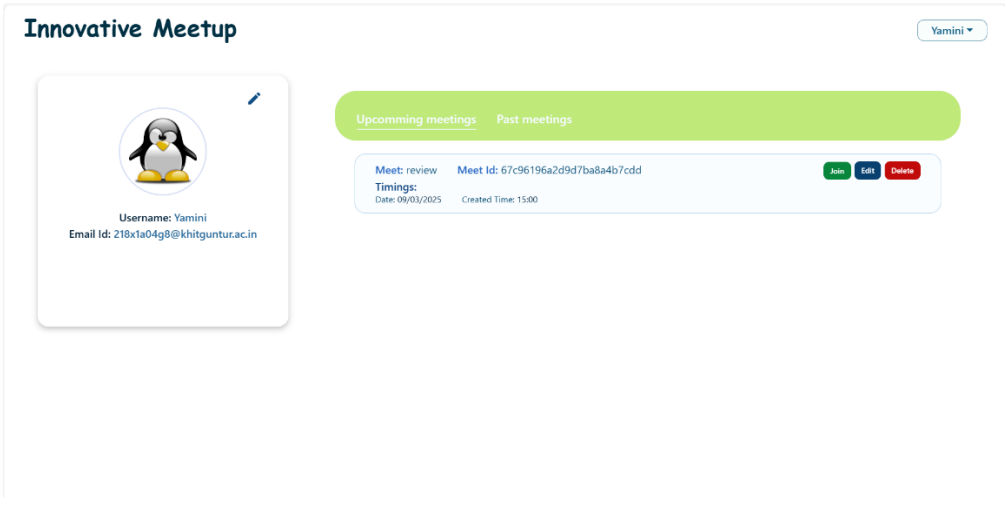
Meeting :



Past Meetings :



Upcomming Meetings :



12.Future Enhancement

As the demand for seamless communication continues to grow, several enhancements can be made to elevate the Video Conferencing App experience. First, integrating advanced features such as screen sharing and collaborative document editing would enhance real-time collaboration during conferences. Additionally, implementing end-to-end encryption would significantly improve security and privacy for users. Introducing an AI-powered background noise cancellation feature can further enhance audio quality, making conversations clearer and more professional. To cater to a wider audience, the app could be expanded to support multilingual interfaces and real-time translation services. Furthermore, incorporating scheduling capabilities for recurring meetings, along with automated reminders, would facilitate better time management for users. Finally, leveraging analytics to provide insights on user engagement and meeting effectiveness could help users optimize their conferencing experience.