React Chat App With Video Calling and Personal AI Bot using Chat GPT API

Overview

This document serves as a comprehensive guide to understanding and utilizing the functionalities of React-based chat application. The application integrates video call features and incorporates an AI bot powered by ChatGPT, enhancing user interactions.

Table of Contents

1. Introduction

This React application combines real-time video calling capabilities with an AI-powered chatbot to provide users with a seamless and interactive experience. Users can engage in video calls while simultaneously interacting with the AI bot, powered by ChatGPT, enabling intelligent responses and assistance during conversations.

2. Installation and Setup

Prerequisites

Ensure you have the following installed:

- Node.js
- npm or yarn

Steps to Install

Clone the repository from https://github.com/vishoov/chat-app

Navigate to the project directory.

Run npm install or yarn install to install dependencies separately from client and server folders

Configure environment variables for the application, including API keys for the video call service and ChatGPT.

3. Features

Video Call

The application offers real-time video call functionalities using socket.io

Users can:

- Initiate and join video call rooms.
- Access chat functionality alongside the video call.

AI Bot Integration

The application integrates ChatGPT's AI bot to provide intelligent conversational support. Key features include:

- Seamless integration of the gpt-3-turbo using ChatGPT API
- Natural language understanding for user queries and responses.
- Customizable bot responses and interactions based on user input.

4. Usage

- Video Call Usage
- Sign in or create an account.
- Navigate to the video call section.
- Initiate or join a video call.
- Utilize the provided controls for audio, video, and screen sharing.
- AI Bot Interaction
- Access the chat interface during a video call or separately.
- Engage with the AI bot by typing messages.
- Receive intelligent responses based on the conversation context.

5. Architecture Overview

- React.js for the user interface.
- Integration with socket.io for real-time video calling.
- API requests to ChatGPT for AI bot interactions.

6. Dependencies

"@testing-library/jest-dom": "^5.16.2",

"@testing-library/react": "^12.1.2",

"@testing-library/user-event": "^13.5.0",

"axios": "^0.25.0",

"buffer": "^6.0.3",

"emoji-picker-react": "^3.5.1",

"react": "^17.0.2",

"react-dom": "^17.0.2",

"react-icons": "^4.3.1",

"react-router-dom": "^6.2.1",

"react-scripts": "5.0.0",

"react-toastify": "^8.1.1",

"socket.io-client": "^4.4.1",

"styled-components": "^5.3.3",

"uuid": "^8.3.2",

"web-vitals": "^2.1.4"

7. Future Improvement Scope

1. Enhanced AI Bot Capabilities

Contextual Understanding: Implement context retention in the AI bot to maintain conversational context across multiple interactions, providing a more seamless user experience.

Multilingual Support: Extend the AI bot's language capabilities to support a broader range of languages for a more inclusive user base.

2. Advanced Video Call Features

Customizable Layouts: Introduce customizable video call layout options, enabling users to arrange video feeds according to their preferences.

Recording and Playback: Implement functionality for recording video calls and playback options for users to review past conversations.

3. User Experience Enhancements

Accessibility Features: Improve accessibility by integrating screen reader support and ensuring compliance with accessibility standards.

User Feedback Mechanism: Implement a feedback system to gather user input, allowing us to continuously improve the application based on user suggestions.

4. Performance Optimization

Reduced Latency: Optimize network calls and video streaming to reduce latency during video calls, ensuring a smoother real-time experience.

5. Security Enhancements

End-to-End Encryption: Implement end-to-end encryption for video calls to enhance user privacy and security.

Bot Interaction Security: Ensure secure data transmission between the user and the AI bot, especially for sensitive information exchange.

6. Platform Expansion

Mobile Application Support: Develop mobile applications (iOS and Android) to extend the reach of the application to mobile users.

Browser Compatibility: Ensure compatibility with a wider range of browsers for a more inclusive user experience.

7. Machine Learning Integration

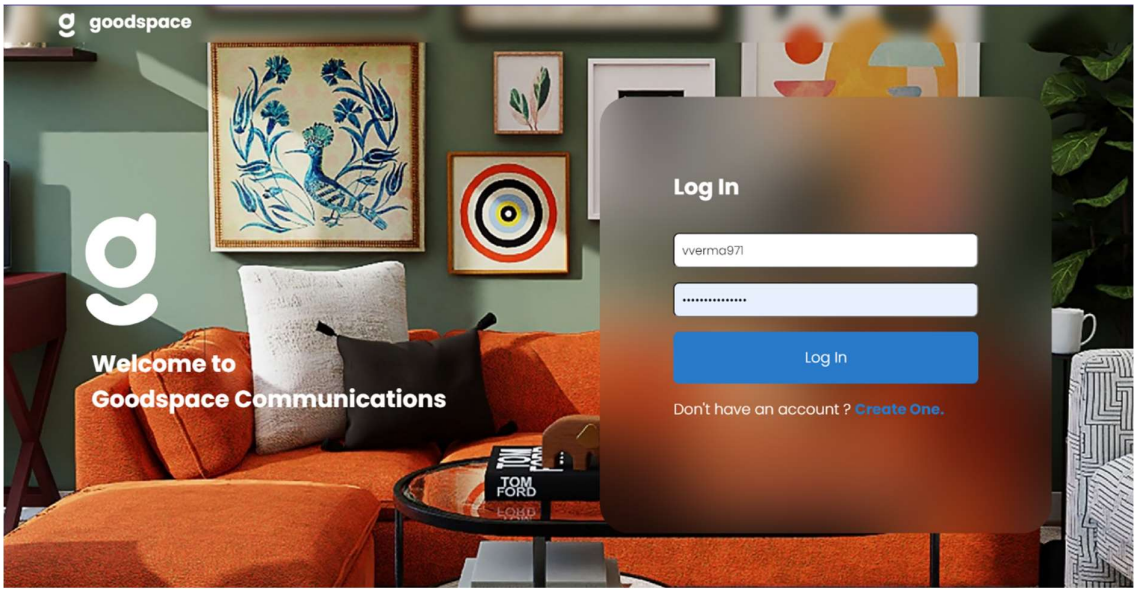
AI Personalization: Utilize machine learning to personalize the AI bot's responses based on user behavior and preferences.

Content Filtering: Implement content filtering mechanisms within the AI bot to ensure appropriate and safe interactions.

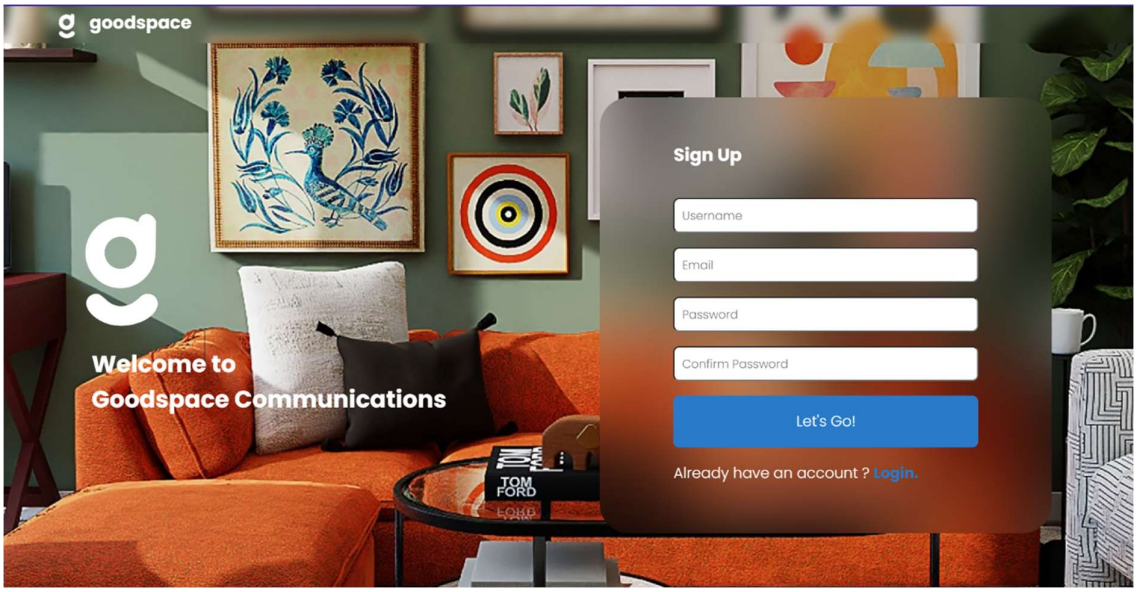
These improvements aim to enhance various aspects of the application, providing a more robust, user-friendly, and feature-rich experience. They will be prioritized based on user feedback, technological feasibility, and strategic goals.

Here are some screens in the app that user navigates through:

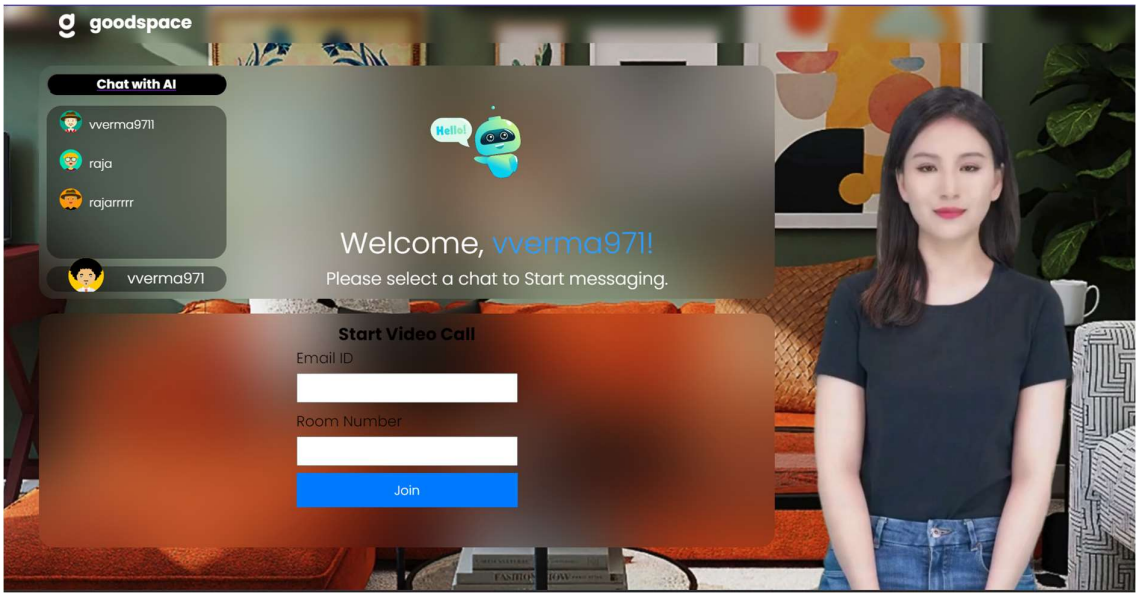
1.Login



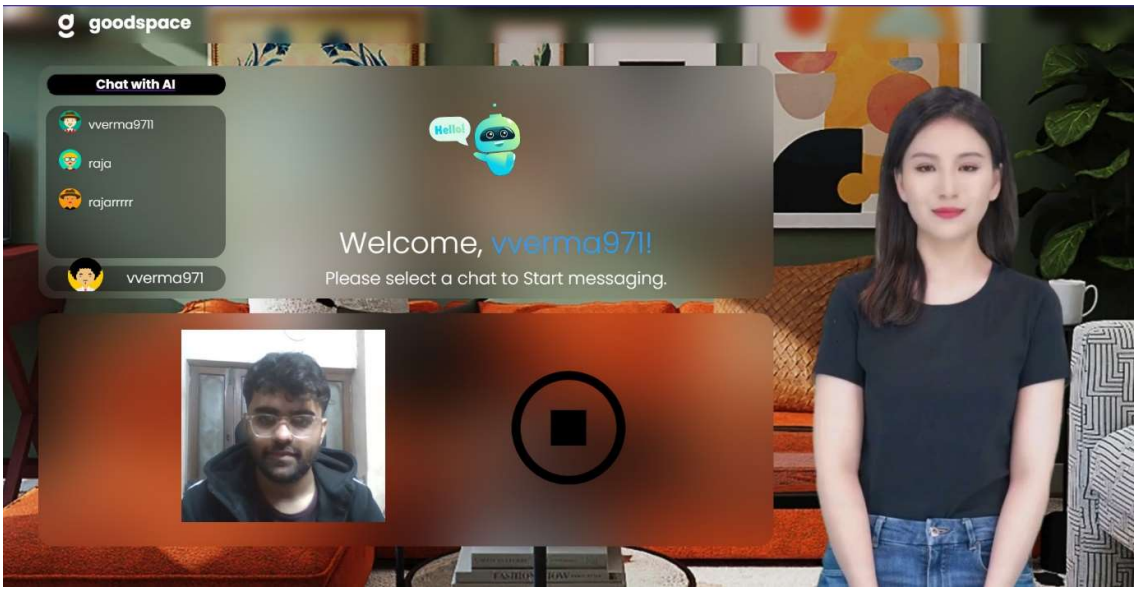
2.Sign Up



3. Home Screen



4.Video Calling



5.Chat Bot Conversation