Project Title: ChatBlink – Real-Time Messaging & Video Calling

Group Members:

- Member 1: Rayyan-ur-Rehman 23k-0634
- Member 2: Muhammad Abdullah Khan 23k-0607
- Member 3: Abdul Ahad Munaf 23k-0590

Proposed Project Description

ChatBlink is a web-based application designed to randomly connect two users for real-time text messaging and video chatting. Inspired by platforms like Omegle and OrzChat, it enables users to instantly interact with strangers through chat and video without prior registration or friend lists.

How it Works:

1. Users visit the ChatBlink website and click “Open Chat.”

2. The system randomly pairs them with another online user.

3. Users can exchange text messages instantly via the chat interface built with Bootstrap and Node.js/Express backend.

4. Users can also initiate a video call, using WebRTC for peer-to-peer video streaming.

5. Users can end the chat or video call at any time and start a new session with a different random user.

Functional Features:

- ➔ Random User Matching: Automatically pairs two online users.

- ➔ Instant Messaging: Real-time text communication using Socket.io.

- ➔ Video Calling: Peer-to-peer video using WebRTC.

- ➔ Chat History (Temporary): Users can see messages exchanged during a session only.

Plan of Work (Next 5 Weeks)

Week 1:

- Finalize UI design using Bootstrap.
- Set up Node.js & Express server structure.
- WebRTC & Socket.io dependencies.

Week 2:

- Implement random user matching.
- Integrate basic text chat functionality.

Week 3:

- Implement WebRTC-based video calling.
- Test video call connections
- Enhance UI responsiveness for video chat interface.

Week 4:

- Add user controls (end chat, start new chat).
- Debugging

Week 5:

- Final testing
- Polish UI and UX for smoother user experience.

Team Contributions:

Team Member	Responsibilities
Member 1	Implement chat functionality, Socket.io integration, text messaging UI, chat moderation, typing indicators.
Member 2	Implement video calling using WebRTC, video UI design, mute/unmute, connection status
Member 3	Implement random user matching algorithm, connection management, assist in debugging, support chat functionality integration, help with video UI design,assist with chat moderation, assist with typing indicators, monitor connection status, and coordinate integration of chat and video features across the system.

References

MDN Web Docs – WebRTC API (https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API)

Socket.io Documentation – https://socket.io/docs/

Bootstrap Documentation – https://getbootstrap.com/docs/