

Project 3: Implementing an Online Meeting Software

Due Dec. 9th

Hand in: Summarize your implementation into a report. Also snapshot the interfaces/executions, post the snapshots into your report. Electronic submission of entire project (source, compiled, executable files, etc.) on Canvas. Please zip up your whole project directory and submit the zip file.

Overview/Preparation

In this project, we want to implement/complete an online meeting software. WebRTC (<https://webrtc.org/>) is a powerful and popular real-time communication platform. By using WebRTC, it is easy to have video conference, audio conservation, and text chat.

In this project, we will NOT build the program from scratch. Instead, we will build the program from an existing one.

1. In this project, we will use Visual Studio Code (<https://code.visualstudio.com/>) as the development environment. If you do not have it yet, please install Visual Studio Code on your computer.
2. After installation, click “Extensions” icon on the left bar, search extensions for “javascript”, “html”, and “json”, choose one package for each and install them.
3. On your computer, create a working directory for this project. After creating the folder, open Visual Studio Code, click “Files”, then “Open Folder”, choose the working directory for this project.
4. In Visual Studio Code, click “Terminal”, then “New Terminal” to get a terminal inside Visual Studio Code.

Project Requirements

Watch the video at <https://www.youtube.com/watch?v=DvlyzDZDEq4&t=524s> . This project uses WebRTC to implement online video meeting. Follow this video, build the project on your computer. Make sure this project works on your computer.

When you watch the video and build the project, there are a few things to keep in mind:

1. Under your working directory, build “public”, “views” folders.
2. You can copy the source code from <https://github.com/sitepoint-editors/simplewebrtc-messenger> to your working directory. Make sure the different script files will be saved to different directories on your computer.
3. One line in “package.json” file needs to be modified. Watch the video, modify the line.

4. Before typing any command to install any package, run the following command inside terminal (the one in Visual Studio Code under working directory)

npm install --no-optional

5. To save your time, I list the commands that you need to type in the terminal

npm init -y

npm i express ejs socket.io

npm i uuid

npm i --save-dev nodemon

npm i -g peer

6. One command you need pay attention is listed below. If terminal returns error message after running the first command, then execute the second command first, then run the first command again.

peerjs --port 3001

Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

7. Once your program is ready to run, you can start the server by running the following command in a SEPARATE terminal inside Visual Studio Code.

npm run devStart

Once the sample code works, modify it to add text chatting function. In other words, your implementation should have not only video conference but also text chat. The chat box can be displayed under video boxes. Your implementation can use any package to implement this function, but preferably WebRTC, which can transfer any data, including text.