Step 2: Express Server with WebSocket

ChatGPT React Frontend and Node.js Server

Start React & Web

The next step is to build Express server.

1. Start the React app (port 3000)

```
npm install
npm start
```

2. Start the Server app (port 5000)

```
npm start
node index.js
```

Changing default port number

1. Change port number in `socketConnection/socketConn.js`.

```
import { io } from "socket.io-client";

export const connectWithSocketServer = () => {
  socket = io("http://localhost:5001"); // <-- Change this
  ...
};
```

2. Set different port before running `node index.js`.

```
PORT=5001 node index.js # to use other port
```

Server

index.js

We process users' request using WebScoket; however, we also use Express, because it gives developers full control over how requests are processed and responses are sent.

- HTTP requests and responses
- Routing (e.g., /users, /login)
- Middleware for authentication, logging, validation, etc.

We use WebSocket and Express server.

```
const express = require("express");
const http = require("http");
const cors = require("cors");
const socketServer = require("./src/socketServer");

const app = express();

const server = http.createServer(app);
socketServer.registerSocketServer(server);

app.use(cors());

app.get("/", (req, res) => {
  res.send("Hello server is working");
});
```

We use port 5000 for this Express server.

```
const PORT = process.env.PORT || 5000;

server.listen(PORT, () => {
  console.log(`App started listening at port ${PORT}`);
});
```

src/socketServer.js

```
const { Server } = require("socket.io");

const registerSocketServer = (server) => {
  const io = new Server(server, {
    cors: {
      origin: "*",
      methods: ["GET", "POST"],
    },
  });

  io.on("connection", (socket) => {
    console.log(`user connected ${socket.id}`);
  });
};

module.exports = { registerSocketServer };
```

Client

App.js

```
import { connectWithSocketServer } from "./socketConnection/socketConn";

function App() {
  useEffect(() => {
    connectWithSocketServer();
  }, []);

  return (
    <div className="App">
      <Dashboard />
    </div>
  );
}
```

- useEffect runs after the component first renders
- The empty array [] tells React: “Run this effect only one time, on component mount.”

socketConnection/socketConn.js

This function connects to server at port 5000.

- When the server emits "connect" API, the `socket.on("connect")` is invoked.

```
import { io } from "socket.io-client";

let socket;

export const connectWithSocketServer = () => {
  socket = io("http://localhost:5000"); // Change port if necessary

  socket.on("connect", () => {
    console.log("connected with socket.io server");
    console.log(socket.id);
  });
};
```