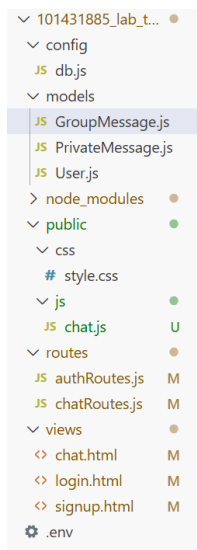


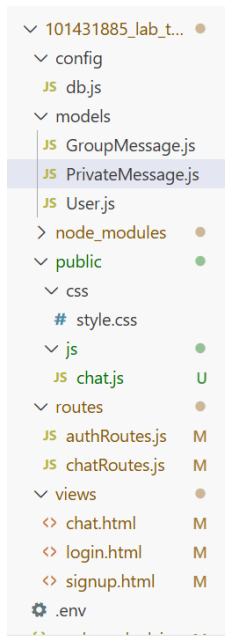
```
1  const mongoose = require("mongoose");
2  require("dotenv").config();
3
4  const connectDB = async () => {
5      try {
6          await mongoose.connect(process.env.MONGO_URI, {
7              useNewUrlParser: true,
8              useUnifiedTopology: true,
9          });
10         console.log("✅ MongoDB Connected...");
11     } catch (error) {
12         console.error("❌ MongoDB Connection Failed:", error);
13         process.exit(1);
14     }
15 };
16
17 module.exports = connectDB;
18
```

/



```
1  const mongoose = require('mongoose');
2
3  const groupMessageSchema = new mongoose.Schema({
4      from_user: {
5          type: String,
6          required: true
7      },
8      room: {
9          type: String,
10         required: true
11     },
12     message: {
13         type: String,
14         required: true
15     },
16     date_sent: {
17         type: Date,
18         default: Date.now
19     }
20 });
21
22 const GroupMessage = mongoose.model('GroupMessage', groupMessageSchema);
23 module.exports = GroupMessage;
24
```

/



```
1  const mongoose = require('mongoose');
2
3
4  const privateMessageSchema = new mongoose.Schema({
5    from_user: {
6      type: String,
7      required: true
8    },
9    to_user: {
10     type: String,
11     required: true
12   },
13   message: {
14     type: String,
15     required: true
16   },
17   date_sent: {
18     type: Date,
19     default: Date.now
20   }
21 });
22
23 const PrivateMessage = mongoose.model('PrivateMessage', privateMessageSchema);
24 module.exports = PrivateMessage;
```

/



```
1  const mongoose = require('mongoose');
2
3
4  const userSchema = new mongoose.Schema({
5    username: {
6      type: String,
7      required: true,
8      unique: true, // Ensure the username is unique
9    },
10   firstname: String,
11   lastname: String,
12   password: {
13     type: String,
14     required: true
15   },
16   creation: {
17     type: Date,
18     default: Date.now
19   }
20 });
21
22 const User = mongoose.model('User', userSchema);
23 module.exports = User;
```

/

- models
  - JS User.js
- node\_modules
- public
  - css
    - # style.css
  - js
    - JS chat.js U
- routes
  - JS authRoutes.js M
  - JS chatRoutes.js M
- views
  - chat.html M
  - login.html M
  - signup.html M
- .env
- package-lock.js... M
- package.json M
- server.js M
- config
- OUTLINE
- TIMELINE
- CS-SCRIPT - ACTIVE

```

2  body {
3      font-family: Arial, sans-serif;
4      background-color: #f5f5f5;
5      color: #333;
6  }
7
8  #chat-window {
9      background-color: #fff;
10     padding: 10px;
11     border-radius: 5px;
12     max-width: 800px;
13     margin: 0 auto;
14 }
15
16 #messages {
17     max-height: 400px;
18     overflow-y: scroll;
19 }
20
21 .message {
22     margin: 5px 0;
23 }
24
25 #typing-indicator {
26     font-style: italic;
27     color: gray;
28 }
29

```

/

- OPEN EDITORS 1 unsaved
- LABTE...
  - 101431885\_lab\_t...
    - models
      - JS User.js
    - node\_modules
    - public
      - css
        - # style.css
      - js
        - JS chat.js U
    - routes
      - JS authRoutes.js M
      - JS chatRoutes.js M
    - views
      - chat.html M
      - login.html M
      - signup.html M
    - .env
    - package-lock.js... M
    - package.json M
    - server.js M
    - config
    - OUTLINE
    - TIMELINE
    - CS-SCRIPT - ACTIVE

```

101431885_lab_test1_chat_app > public > js > JS chat.js > ...
1  // public/js/chat.js
2  const socket = io(); // Connect to WebSocket
3
4  const token = localStorage.getItem("token");
5  if (!token) {
6      window.location.href = "/login";
7  }
8
9  const sendBtn = document.getElementById("sendMessageBtn");
10 const roomSelect = document.getElementById("roomSelect");
11 const messagesDiv = document.getElementById("messages");
12 const messageInput = document.getElementById("messageInput");
13 const typingIndicator = document.getElementById("typingIndicator");
14
15 let typingTimer;
16
17 // Send message
18 sendBtn.addEventListener("click", async () => {
19     const message = messageInput.value.trim();
20     const room = roomSelect.value;
21     const from_user = localStorage.getItem("username");
22
23     if (!message) return;
24
25     await fetch("/chat/sendMessage", {
26         method: "POST",
27         headers: { "Content-Type": "application/json" },
28         body: JSON.stringify({ from_user, room, message }),
29     });
30

```

101431885_lab_t...	
models	
JS User.js	
node_modules	
public	
css	
# style.css	
js	
JS chat.js	U
routes	
JS authRoutes.js	M
JS chatRoutes.js	M
views	
<> chat.html	M
<> login.html	M
<> signup.html	M
.env	
{ } package-lock.js...	M
{ } package.json	M
JS server.js	M
config	
OUTLINE	
TIMELINE	
CS-SCRIPT - ACTIVE	

```

30
31     messageInput.value = "";
32     socket.emit("stopTyping", { room });
33 });
34
35 // Listen for new messages in real-time (room-filtered)
36 socket.on("newMessage", (data) => {
37     const currentRoom = roomSelect.value;
38     if (data.room !== currentRoom) return;
39
40     messagesDiv.innerHTML += `<p><strong>${data.from_user}</strong> ${data.message}</p>`;
41     messagesDiv.scrollTop = messagesDiv.scrollHeight;
42 });
43
44 // Load messages when the room changes
45 async function loadMessages(room) {
46     const response = await fetch(`/chat/messages/${room}`);
47     const messages = await response.json();
48
49     messagesDiv.innerHTML = messages
50         .map((msg) => `<p><strong>${msg.from_user}</strong> ${msg.message}</p>`)
51         .join("");
52
53     messagesDiv.scrollTop = messagesDiv.scrollHeight;
54
55     // Join the room using Socket.io
56     socket.emit("joinRoom", room);
57 }

```

101431885_lab_t...	
models	
JS User.js	
node_modules	
public	
css	
# style.css	
js	
JS chat.js	U
routes	
JS authRoutes.js	M
JS chatRoutes.js	M
views	
<> chat.html	M
<> login.html	M
<> signup.html	M
.env	
package-lock.js...	M
package.json	M
server.js	M
config	
OUTLINE	
TIMELINE	

```

59 // Change room
60 roomSelect.addEventListener("change", () => {
61     const room = roomSelect.value;
62     loadMessages(room);
63 });
64 // Load messages when page loads
65 window.onload = () => {
66     const room = roomSelect.value;
67     loadMessages(room);
68 };
69 // Typing indicator (required)
70 messageInput.addEventListener("input", () => {
71     const room = roomSelect.value;
72     const username = localStorage.getItem("username");
73
74     socket.emit("typing", { room, username });
75
76     clearTimeout(typingTimer);
77     typingTimer = setTimeout(() => {
78         socket.emit("stopTyping", { room });
79     }, 700);
80 });
81 socket.on("typing", ({ username }) => {
82     if (typingIndicator) typingIndicator.textContent = `${username} is typing...`;
83 });
84 socket.on("stopTyping", () => {
85     if (typingIndicator) typingIndicator.textContent = "";
86 });

```

/

OPEN EDITORS 2 unsaved

LABTE... [Icons]

101431885\_lab\_t...

- models
  - JS User.js
- node\_modules
- public
  - css
    - # style.css
  - js
    - JS chat.js U
- routes
  - JS authRoutes.js M
  - JS chatRoutes.js M
- views
  - <> chat.html M
  - <> login.html M
  - <> signup.html M
- .env
- { } package-lock.js... M
- { } package.json M
- JS server.js M
- config

OUTLINE

TIMELINE

CS-SCRIPT - ACTIVE

```

101431885_lab_test1_chat_app / routes / authRoutes.js / ...
You, 1 hour ago | 1 author (You)
1 // routes/authRoutes.js
2 const express = require("express");
3 const bcrypt = require("bcryptjs");
4 const jwt = require("jsonwebtoken");
5 const User = require("../models/User");
6
7 const router = express.Router();
8
9 /*
10 =====
11 Signup Route
12 POST /auth/signup
13 =====
14 */
15 router.post("/signup", async (req, res) => {
16   const { username, firstname, lastname, password } = req.body;
17
18   try {
19     // Check if user exists
20     const existingUser = await User.findOne({ username });
21     if (existingUser) {
22       return res.status(400).json({ error: "Username already taken" });
23     }
24
25     // Hash password
26     const hashedPassword = await bcrypt.hash(password, 10);
27
28     // Create user
29     const newUser = new User({
30       username,
31       firstname,
32       lastname,
33       password: hashedPassword,
34     });
35     await newUser.save();
36     return res.status(201).json(newUser);
37   } catch (error) {
38     return res.status(500).json({ error: "Server error" });
39   }
40 });

```

/

OPEN EDITORS 2 unsaved

LABTE... [Icons]

101431885\_lab\_t...

- models
  - JS User.js
- node\_modules
- public
  - css
    - # style.css
  - js
    - JS chat.js U
- routes
  - JS authRoutes.js M
  - JS chatRoutes.js M
- views
  - <> chat.html M
  - <> login.html M
  - <> signup.html M
- .env
- { } package-lock.js... M
- { } package.json M
- JS server.js M
- config

OUTLINE

TIMELINE

CS-SCRIPT - ACTIVE

```

101431885_lab_test1_chat_app > routes > JS chatRoutes.js > ...
You, 1 hour ago | 1 author (You)
1 // routes/chatRoutes.js
2 const express = require("express");
3 const GroupMessage = require("../models/GroupMessage");
4
5 const router = express.Router();
6
7 module.exports = (io) => {
8   // Send message (stores in MongoDB)
9   router.post("/sendMessage", async (req, res) => {
10     const { from_user, room, message } = req.body;
11
12     // Basic validation
13     if (!from_user || !room || !message) {
14       return res.status(400).json({ error: "Missing required fields" });
15     }
16
17     try {
18       const newMsg = new GroupMessage({ from_user, room, message });
19       await newMsg.save();
20
21       // Realtime emit to the room
22       io.to(room).emit("newMessage", {
23         from_user,
24         message,
25         room,
26         date_sent: newMsg.date_sent,
27       });
28     } catch (error) {
29       return res.status(500).json({ error: "Server error" });
30     }
31   });
32 };

```

/

OPEN EDITORS2 unsaved

LABTEST1101431885\_lab\_test1\_chat\_app

modelsUser.jsnode\_modulespubliccssstyle.cssjschat.jsroutesauthRoutes.jsMchatRoutes.jsMviewschat.htmlMlogin.htmlMsignup.htmlM.envpackage-lock.js...Mpackage.jsonMserver.jsMconfigOUTLINETIMELINECS-SCRIPT - ACTIVE

101431885\_lab\_test1\_chat\_app > views > chat.html > ...  
You, 1 hour ago | 1 author (You)  
1<!DOCTYPE html>  
2<html lang="en">  
3<head>  
4<meta charset="UTF-8">  
5<meta name="viewport" content="width=device-width, initial-scale=1.0">  
6<title>Chat - Chat App</title>  
7<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">  
8<link rel="stylesheet" href="/css/style.css">  
9</head>  
10<body>  
11<div class="container">  
12<h2 class="mt-5">Welcome to the Chat</h2>  
13<br>  
14<button id="logoutBtn" class="btn btn-warning">Logout</button>  
15<button id="leaveRoomBtn" class="btn btn-danger">Leave Room</button>  
16<br>  
17<div class="mt-4">  
18<label for="roomSelect">Select a Room:</label>  
19<select id="roomSelect" class="form-select">  
20<option value="devops">DevOps</option>  
21<option value="cloud">Cloud Computing</option>  
22<option value="covid19">COVID-19</option>  
23<option value="sports">Sports</option>  
24<option value="nodejs">Node.js</option>  
25</select>  
26</div>  
27<br>  
28<div id="messages" class="mt-3"></div>

/

OPEN EDITORS2 unsaved

LABTEST1101431885\_lab\_test1\_chat\_app

modelsUser.jsnode\_modulespubliccssstyle.cssjschat.jsroutesauthRoutes.jsMchatRoutes.jsMviewschat.htmlMlogin.htmlMsignup.htmlM.envpackage-lock.js...Mpackage.jsonMserver.jsMconfigOUTLINETIMELINECS-SCRIPT - ACTIVE

101431885\_lab\_test1\_chat\_app > views > login.html > ...  
You, 1 hour ago | 1 author (You)  
1<!DOCTYPE html>  
2<html lang="en">  
3<head>  
4<meta charset="UTF-8">  
5<title>Login</title>  
6<link rel="stylesheet" href="/css/style.css">  
7</head>  
8<body>  
9<br>  
10<div id="chat-window">  
11<h2>Login</h2>  
12<br>  
13<form id="loginForm">  
14<input type="text" id="username" placeholder="Username" required />  
15<br>  
16<input type="password" id="password" placeholder="Password" required />  
17<br>  
18<button type="submit">Login</button>  
19</form>  
20<br>  
21<p>No account? <a href="/signup">Signup here</a></p>  
22</div>  
23<br>  
24<script src="/js/login.js"></script>  
25<br>  
26</body>  
27</html>

/

OPEN EDITORS 2 unsaved

LABTEST1

- 101431885\_lab\_test1\_chat\_app
  - models
    - User.js
  - node\_modules
  - public
    - css
      - style.css
    - js
      - chat.js
  - routes
    - authRoutes.js
    - chatRoutes.js
  - views
    - chat.html
    - login.html
    - signup.html
  - .env
  - package-lock.js...
  - package.json
  - server.js
  - config

OUTLINE

TIMELINE

101431885\_lab\_test1\_chat\_app > views > <> signup.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6   <title>Signup</title>
7   <link rel="stylesheet" href="/css/style.css" />
8 </head>
9 <body>
10
11   <div id="chat-window">
12     <h2>Signup</h2>
13
14     <form id="signupForm">
15       <input type="text" id="username" placeholder="Username" required />
16       <br><br>
17
18       <input type="text" id="firstname" placeholder="First name" required />
19       <br><br>
20
21       <input type="text" id="lastname" placeholder="Last name" required />
22       <br><br>
23
24       <input type="password" id="password" placeholder="Password" required />
25       <br><br>
26
27       <button type="submit">Create Account</button>
28     </form>
```

/

OPEN EDITORS 2 unsaved

LABTEST1

- 101431885\_lab\_test1\_chat\_app
  - models
    - User.js
  - node\_modules
  - public
    - css
      - style.css
    - js
      - chat.js
  - routes
    - authRoutes.js
    - chatRoutes.js
  - views
    - chat.html
    - login.html
    - signup.html
  - .env
  - package-lock.js...
  - package.json
  - server.js
  - config

OUTLINE

TIMELINE

101431885\_lab\_test1\_chat\_app > {} package-lock.json > {} packages > {} ""

```
1 {
2   "name": "101431885_lab_test1_chat_app",
3   "version": "1.0.0",
4   "lockfileVersion": 3,
5   "requires": true,
6   "packages": {
7     "": {
8       "name": "101431885_lab_test1_chat_app",
9       "version": "1.0.0",
10      "dependencies": {
11        "bcryptjs": "^2.4.3",
12        "cors": "^2.8.5",
13        "dotenv": "^16.4.7",
14        "express": "^4.21.2",
15        "jsonwebtoken": "^9.0.2",
16        "mongodb": "^6.21.0",
17        "mongoose": "^8.10.0",
18        "socket.io": "^4.8.1"
19      },
20      "devDependencies": {
21        "nodemon": "^3.0.0"
22      }
23    },
24    "node_modules/@mongodb-js/saslprep": {
25      "version": "1.4.6",
26      "resolved": "https://registry.npmjs.org/@mongodb-js/saslprep/-/saslprep-1.4.6.tgz",
27      "integrity": "sha512-1+UWj14pD8DnZtO7WZVNvJ4S7iUajJ6I49pTatYF00I6ZuLk960uF+hV9Vh6Xr8FkI7F9LXUgYUg==",
28      "license": "MIT"
```

/

OPEN EDITORS2 unsaved

LABTE...

101431885\_lab\_t...

models

JS User.js

node\_modules

public

css

# style.css

js

JS chat.jsU

routes

JS authRoutes.jsM

JS chatRoutes.jsM

views

chat.htmlM

login.htmlM

signup.htmlM

.env

package-lock.js...M

package.jsonM

server.jsM

config

OUTLINE

TIMELINE

101431885\_lab\_test1\_chat\_app > {} package.json > {} dependencies  
You, 54 minutes ago | 1 author (You)  
1 {  
2   "name": "101431885\_lab\_test1\_chat\_app",  
3   "version": "1.0.0",  
4   "main": "server.js",  
5   "scripts": {  
6     "start": "node server.js",  
7     "dev": "nodemon server.js"  
8   },  
9   "dependencies": {  
10     "bcryptjs": "^2.4.3",  
11     "cors": "^2.8.5",  
12     "dotenv": "^16.4.7",  
13     "express": "^4.21.2",  
14     "jsonwebtoken": "^9.0.2",  
15     "mongodb": "^6.21.0",  
16     "mongoose": "^8.10.0",  
17     "socket.io": "^4.8.1"  
18   },  
19   "devDependencies": {  
20     "nodemon": "^3.0.0"  
21   }  
22 }  
23

/

OPEN EDITORS2 unsaved

LABTE...

101431885\_lab\_t...

models

JS User.js

node\_modules

public

css

# style.css

js

JS chat.jsU

routes

JS authRoutes.jsM

JS chatRoutes.jsM

views

chat.htmlM

login.htmlM

signup.htmlM

.env

package-lock.js...M

package.jsonM

server.jsM

config

OUTLINE

TIMELINE

CS-SCRIPT - ACTIVE

101431885\_lab\_test1\_chat\_app > JS server.js > ...  
You, 1 hour ago | 1 author (You)  
1   const path = require("path");  
2   const express = require("express");  
3   const dotenv = require("dotenv");  
4   const http = require("http");  
5   const cors = require("cors");  
6   const { Server } = require("socket.io");  
7  
8   const connectDB = require("./config/db");  
9   const authRoutes = require("./routes/authRoutes");  
10   const chatRoutes = require("./routes/chatRoutes");  
11  
12   dotenv.config();  
13  
14   const app = express();  
15   const server = http.createServer(app);  
16   const io = new Server(server, { cors: { origin: "\*" } });  
17  
18   // DB  
19   connectDB();  
20  
21   // Middleware  
22   app.use(cors());  
23   app.use(express.json());  
24   app.use(express.urlencoded({ extended: true }));  
25  
26   // Serve static files (JS/CSS)  
27   app.use(express.static(path.join(\_\_dirname, "public")));  
28  
29   ...  
30