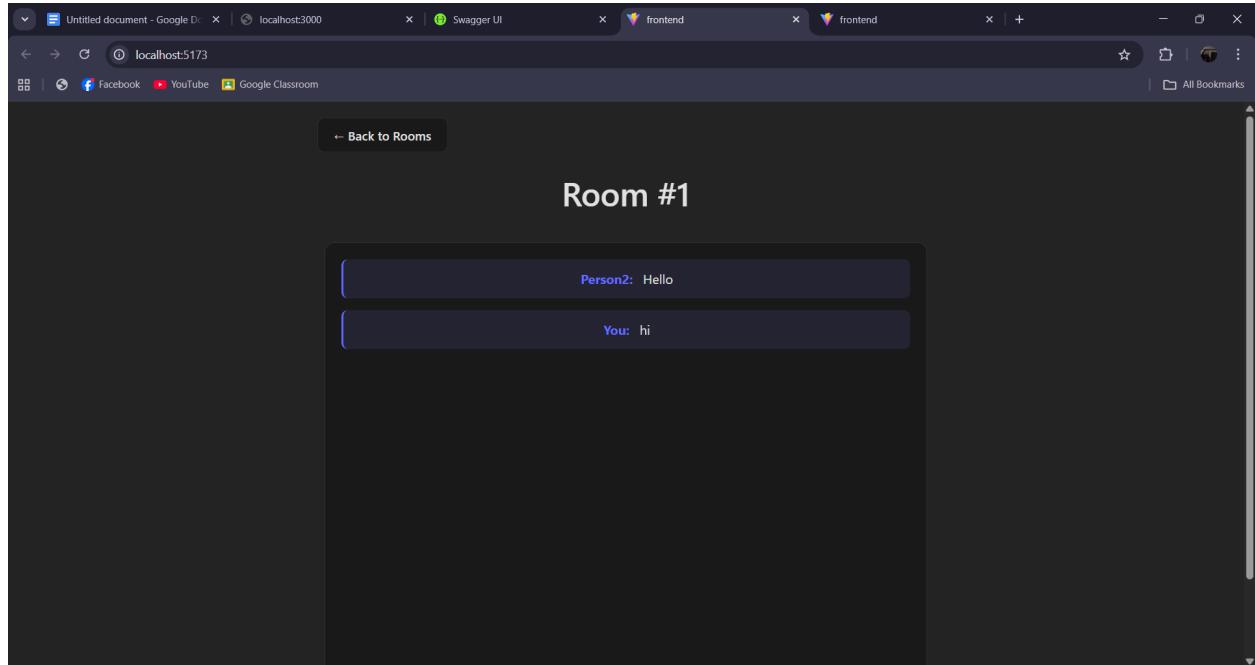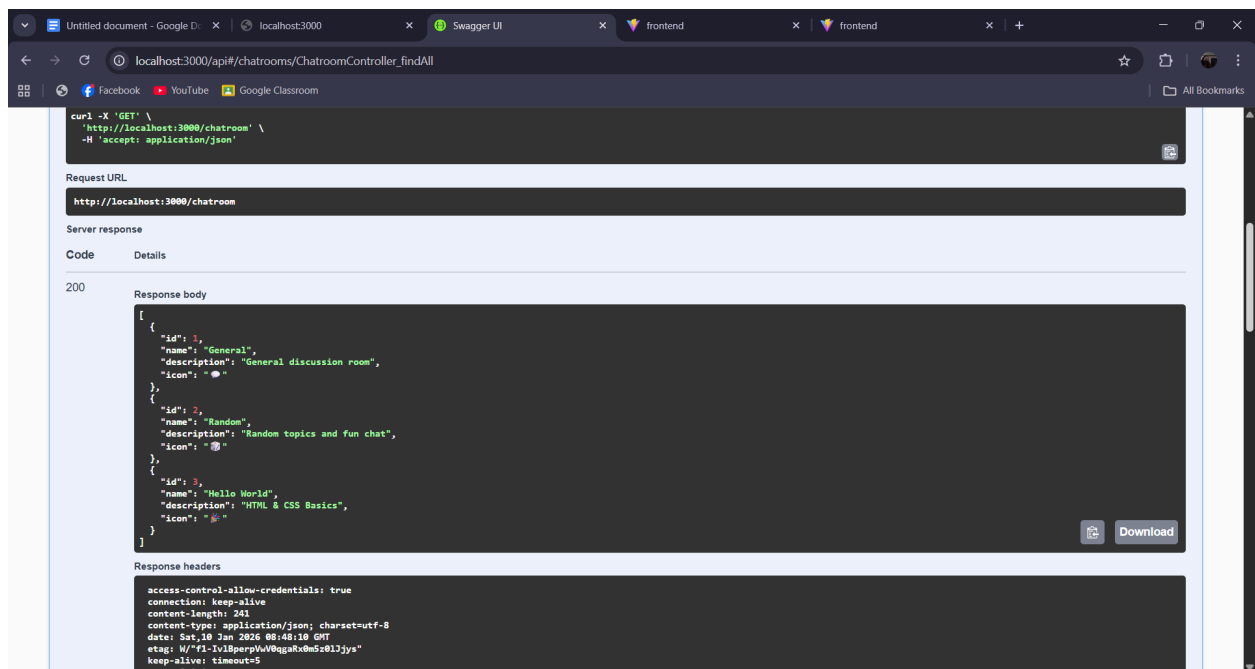**Activity 8: Chatroom REST API + UI** is a web-based real-time chat application where users can join different chatrooms and send messages instantly. The system uses a REST API to manage chatrooms and WebSocket technology for real-time message delivery. The frontend is built with ReactJS, while the backend is powered by Node.js, NestJS, and TypeScript.

Chatrooms

```
    icon :
}
```

Responses

Code	Description	Links
201	Chatroom created	No links

Media type

```
application/json
```

Controls Accept header.

Example Value | Schema

```
{
  "id": 1,
  "name": "General Chat",
  "description": "General discussion room",
  "icon": "💬"
}
```

GET	/chatroom/{id}	Get a chatroom by ID	⌄
PUT	/chatroom/{id}	Update a chatroom	⌄
DELETE	/chatroom/{id}	Delete a chatroom	⌄

messages Message management endpoints

Execute	Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:3000/chatroom/1' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:3000/chatroom/1
```

Server response

Code	Details
200	Response body

```
{
  "id": 1,
  "name": "General",
  "description": "General discussion room",
  "icon": "💬"
}
```

Download

Response headers

```
access-control-allow-credentials: true
connection: keep-alive
content-length: 79
content-type: application/json; charset=utf-8
date: Sat,10 Jan 2026 08:49:38 GMT
etag: W/"4f-6b6Kms/ZrKK6/IrGrAzRrQgI9qs"
keep-alive: timeout=5
vary: Origin
x-powered-by: Express
```

Untitled document - Google D... | localhost:3000 | Swagger UI | frontend | frontend | +
localhost:3000/api#/chatrooms/ChatroomController_update
Facebook | YouTube | Google Classroom | All Bookmarks

```
'http://localhost:3000/chatroom/1' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "name": "Updated Chat",
  "description": "Updated description",
  "icon": "🎉"
}'
```

Request URL

```
http://localhost:3000/chatroom/1
```

Server response

Code	Details

200

Response body

```
{
  "id": 1,
  "name": "Updated Chat",
  "description": "Updated description",
  "icon": "🎉"
}
```

[Download]

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:3000
connection: keep-alive
content-length: 80
content-type: application/json; charset=utf-8
date: Sat,10 Jan 2026 08:50:19 GMT
etag: W/"50-wI41++4Nm7cxx9Y7+VEpNBOL4qA"
keep-alive: timeout=5
vary: Origin
x-powered-by: Express
```

Responses

Code	Description	Links
200	Chatroom updated	No links

Untitled document - Google D... | localhost:3000 | Swagger UI | frontend | frontend | +
localhost:3000/api#/chatrooms/ChatroomController_remove
Facebook | YouTube | Google Classroom | All Bookmarks

Execute	Clear

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:3000/chatroom/3' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:3000/chatroom/3
```

Server response

Code	Details

200

Response body

```
{
  "success": true
}
```

[Download]

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:3000
connection: keep-alive
content-length: 16
content-type: application/json; charset=utf-8
date: Sat,10 Jan 2026 08:50:51 GMT
etag: W/"10-oV4hJxRVSENxc/wX8+mA4/Pe4tA"
keep-alive: timeout=5
vary: Origin
x-powered-by: Express
```

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:3000/message?chatroomId=1' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:3000/message?chatroomId=1
```

Server response

Code	Details
200	**Response body**

```json
[
  {
    "id": 1,
    "chatroomId": 1,
    "username": "Person1",
    "message": "hi"
  },
  {
    "id": 2,
    "chatroomId": 1,
    "username": "Person2",
    "message": "hello"
  },
  {
    "id": 3,
    "chatroomId": 1,
    "username": "Person1",
    "message": "hi"
  },
  {
    "id": 4,
    "chatroomId": 1,
    "username": "Person3",
    "message": "uyy"
  },
  {
    "id": 5,
```

Response headers

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:3000/message' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "chatroomId": 1,
  "username": "John Doe",
  "message": "Hello, world!"
}'
```

Request URL

```
http://localhost:3000/message
```

Server response

Code	Details
201	**Response body**

```json
{
  "id": 17,
  "chatroomId": 1,
  "username": "John Doe",
  "message": "Hello, world!"
}
```

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:3000
connection: keep-alive
content-length: 72
content-type: application/json; charset=utf-8
date: Sat,10 Jan 2026 08:52:03 GMT
etag: W/"48-yaQ8wxdtDRcgvjUqPbtysi0j7a8"
keep-alive: timeout=5
vary: Origin
x-powered-by: Express
```

Execute   Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:3000/message/1' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:3000/message/1
```

Server response

Code	Details
200	Response body

```
{
  "id": 1,
  "chatroomId": 1,
  "username": "Person1",
  "message": "hi"
}
```

Download

Response headers

```
access-control-allow-credentials: true
connection: keep-alive
content-length: 59
content-type: application/json; charset=utf-8
date: Sat,10 Jan 2026 08:52:28 GMT
etag: W/"3b-Jne+NollWlv59T4jqYyi5I7j1VM"
keep-alive: timeout=5
vary: Origin
x-powered-by: Express
```

Responses

Execute   Clear

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:3000/message/1' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "message": "Updated message text"
}'
```

Request URL

```
http://localhost:3000/message/1
```

Server response

Code	Details
200	Response body

```
{
  "id": 1,
  "chatroomId": 1,
  "username": "Person1",
  "message": "Updated message text"
}
```

Download

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:3000
connection: keep-alive
content-length: 77
content-type: application/json; charset=utf-8
```
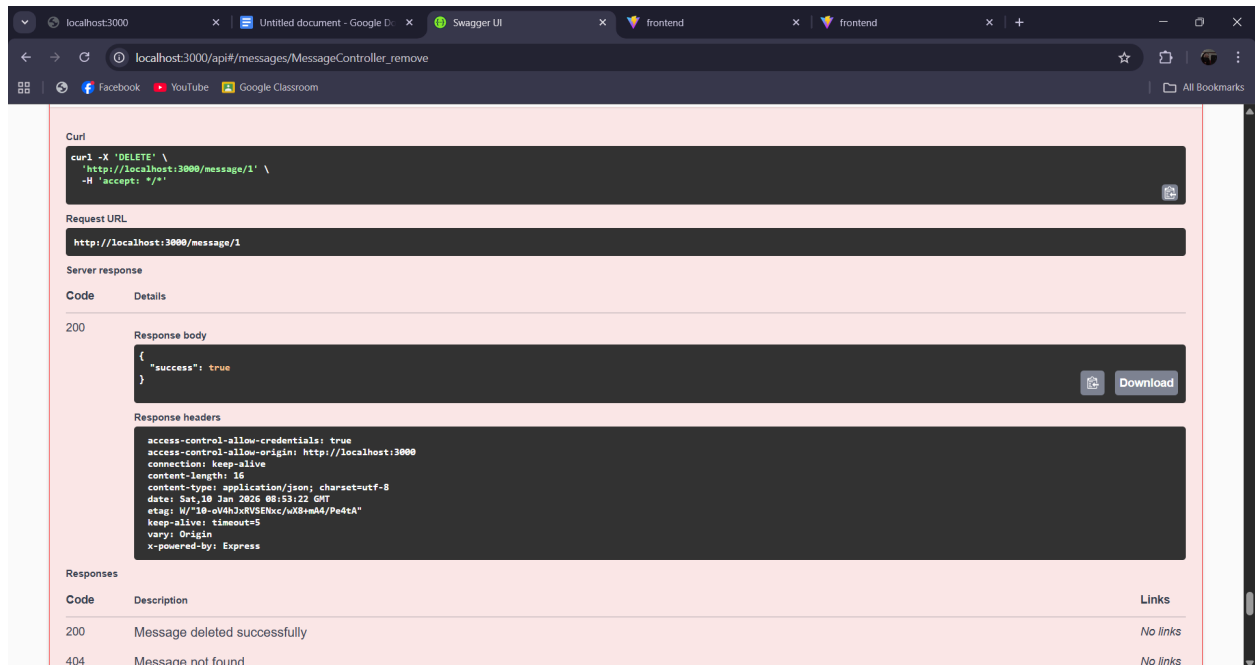
Instructions: How to Run the System

Start the Backend Server

1. Open a terminal

Go to the backend folder:

```
cd Activity-8/backend
```

Install dependencies:

```
npm install
```

2.

Run the backend:

```
npm run start:dev
```

3.

Make sure you see:

```
Application is running on http://localhost:3000
```

4.

Start the Frontend (UI)

1. Open another terminal

Go to the frontend folder:

cd Activity-8/frontend

2.

Install dependencies:

npm install

3.

Run the frontend:

npm run dev

4.

Open the browser and go to:

http://localhost:5173

How to Use the Chatroom

1. Select or add a chatroom from the list

2. Open the same chatroom in another browser tab

3. Send a message

4. Messages will appear in real-time without refreshing

API Testing (Optional)

Use Postman to test the chatroom API:

- GET /chatrooms – view all chatrooms

- POST /chatrooms – add a new chatroom

Expected Output

- Chatrooms can be added via UI or API

- Messages are delivered in real-time

- No page refresh is needed