

COMP 3133 – Lab Test – 1 (06%)

Due Date: 11th Feb 2026 08:00 PM

Create a Chat Application

Objective: In this lab test, you will create a real-time chat application that includes features such as user authentication, room-based messaging, and persistence using MongoDB.

Technologies:

- Use Socket.io, Express, and Mongoose for the backend.
- Use HTML5, CSS, Bootstrap, fetch, and jQuery for the frontend design.

Features to Implement:

1. GitHub Repository:

- Create a GitHub repository named **studentID_lab_test1_chat_app** and push your code there.

2. Connect to the Chat Server:

- Set up a server that users can connect to in order to use the chat application.

3. Signup Page: ([view/signup.html](#))

- Implement a **Signup page** where new users can create an account using a **unique username**.
- Store user details in MongoDB.

4. Login Page: ([view/login.html](#))

- Create a **Login page** to authenticate existing users.
- Use **localStorage** to keep the user session after successful login.

5. Join Room:

- After logging in, allow users to join any room from a list of predefined rooms (e.g., **devops**, **cloud computing**, **covid19**, **sports**, **nodeJS**, etc.).
- Users can only chat within the room they joined.

6. Leave Room:

- Users should have the ability to leave the room they are currently in.

7. Room-based Chat:

- Users should only be able to send and receive messages within the room they are currently in.
- Implement real-time communication using **Socket.io**.

8. Message Storage:

- Store all chat messages and user details in **MongoDB** for future retrieval.
- The messages can be either **group messages** (within a room) or **private messages** (between two users).

9. Typing Indicator:

- Implement a feature that shows "**User is typing...**" whenever a user starts typing a message.

10. Logout Functionality:

- Allow users to log out of the application.
-

MongoDB Schemas

- **User Schema:** Store user details when they sign up.

```
{
    "_id": "j35nn35hjdksqjkdsqs",
    "username": "pritamworld",
    "firstname": "pritesh",
    "lastname": "Patel",
    "password": "secret",
    "createon": "01-28-2022 18:30 PM"
}
```

- **Group Message Schema:** Store group chat messages.

```
{
    "_id": "847het8nieigouy4v",
    "from_user": "pritamworld",
    "room": "covid19",
    "message": "What about covid19 vaccine?",
    "date_sent": "01-28-2021 18:30 PM"
}
```

- **Private Message Schema:** Store private chat messages between users.

```
{
    "_id": "847het8nieigouy4v",
    "from_user": "pritamworld",
    "to_user": "moxdroid",
    "message": "What about covid19 vaccine?",
    "date_sent": "01-28-2021 18:30 PM"
}
```

Evaluation Criteria:

1. GitHub Repository:

- Maintain your GitHub repository and commit your code regularly.
No file upload to GitHub is be accepted. (10 points)

2. Working Signup Page:

- Ensure that the signup page is functional, records user data in MongoDB, and integrates with the Express API. **(10 points)**

3. Working Login/Logout:

- Implement a login/logout page with session management (use **localStorage** for storing the session). **(10 points)**

4. MongoDB Validation:

- Ensure you have proper validation on your Mongoose schemas (e.g., ensure unique usernames). **(10 points)**

5. Room Join/Leave:

- Implement functionality to allow users to join and leave rooms. **(10 points)**

6. Typing Indicator

- Message typing indicator for 1-to-1 chat **(10 points)**

7. Chat Functionality with MongoDB Storage:

- Users should be able to chat within the room and the messages should be saved in MongoDB. This is the most critical part of the project. **(40 points)**
-

Submission Guidelines:

1. Submission Guidelines:

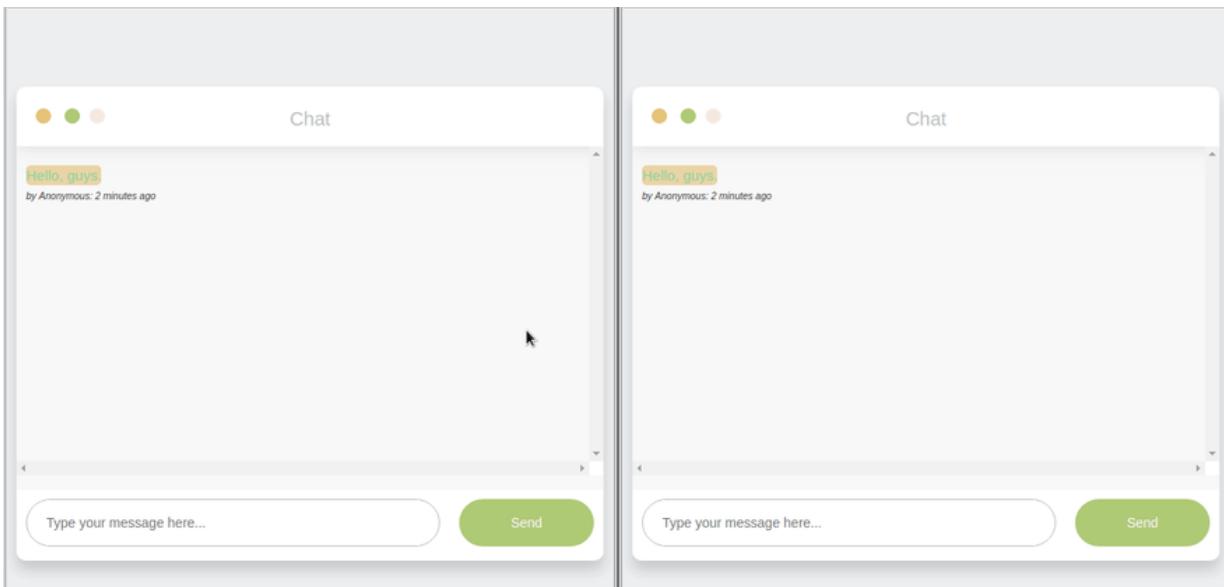
- **Upload** a ZIP file containing your **source code** to the D2L by the deadline.
- **Provide** the **GitHub link** to your repository when submitting.
- **Include evidence** of your application's functionality (screenshots or a short video) and upload it to D2L.
- **No email submissions** will be accepted. Emails asking for exceptions will not be responded to.

2. Important:

- Any attempt to cheat will result in a grade of **ZERO** for the lab test.
- Cheating includes submitting code written by someone else or copying from online sources.

Good Luck!

Sample Chat screen are only for reference



Chat app

Username
pritamworld

Room
News

Join Chat Room

Chat app

Room name: News
Members:
Pritamworld
Moxdroid

Chat App Bot 2:56 PM
Welcome to chat app :)

Chat App Bot 2:56 PM
pritamworld has joined the chat

Chat App Bot 2:56 PM
pritamworld has left the chat

Chat App Bot 2:57 PM
moxdroid has joined the chat

Moxdroid 2:57 PM
hi

Pritamworld 2:57 PM
hello

Type your message here... Send

Chat app

Room name: News
Members:
Pritamworld
Moxdroid

Chat App Bot 2:57 PM
Welcome to chat app :)

Moxdroid 2:57 PM
hi

Pritamworld 2:57 PM
hello

Type your message here... Send