Course Name: EECS 2311 Z
Team Number: 8
Students:
Abdullah Khatib 219742436

Date: March 24, 2025

| User Story | Main Developers | Assigned Tester |
|---|---|---|
| Creating a group | Haaris / Kashif | Farzin |
| Viewing course Schedules | Marko | Lukasz |
| Chat | Faaris | Abdullah |
| Create an account | Lukasz/Abdullah | Marko/Haaris |

# Manual Test Cases:

**1. Sending a Valid Message:**
- Open the chat page.
- Type a valid message (e.g., “Hello, team!”) in the input field.
- Click the “Send” button.
- **Expected Outcome:** The message appears immediately in the chat window with a timestamp.

**2. Empty Message Submission:**
- Open the chat page.
- Attempt to send a message with only spaces or no content.
- **Expected Outcome:** The system should block the submission and display an error message like “Cannot send an empty message.”

**3. Input Field Clearing:**
- Open the chat page.
- Type a valid message and click “Send.”
- **Expected Outcome:** The input field is cleared immediately after sending the message.

**4. Auto-Scrolling Behavior:**
- Open the chat page and send multiple messages to create an overflow.
- **Expected Outcome:** The chat window auto-scrolls to display the latest message without user intervention.

**5. Long Message Handling:**
- Open the chat page.
- Type an extremely long message that exceeds typical length limits.
- **Expected Outcome:** The application either truncates the message appropriately or displays an error indicating the message exceeds allowed limits.

**Problem Report 1:**

- **Problem report number:** PR-201
- **Reported by:** Abdullah
- **Date reported:** March 24, 2025
- **Program (or component) name:** Chat Feature – Message Persistence
- **Release number:** v2.0
- **Configuration(s):** Local browser environment (no server/database)
- **Report type:** Functionality bug
- **Can reproduce:** Yes
- **Severity:** High
- **Priority:** High
- **Problem summary:** Chat messages disappear after a page refresh
- **Problem description and how to reproduce it:**
    1. Open the chat page.
    2. Type a message and click “Send.”
    3. Refresh the page.
    4. Observe that the sent message is no longer displayed.
- **Suggested fix:** Implement local storage or a client-side caching mechanism to retain messages temporarily.
- **Status:** Open
- **Resolution:** Pending
- **Resolved by:** [To be determined]

**Problem Report 2:**

- **Problem report number:** PR-202
- **Reported by:** Abdullah
- **Date reported:** March 24, 2025
- **Program (or component) name:** Chat Feature – UI Scrolling
- **Release number:** v2.0
- **Configuration(s):** All supported browsers, local environment
- **Report type:** UI/UX bug
- **Can reproduce:** Yes
- **Severity:** Medium
- **Priority:** Medium
- **Problem summary:** Chat window does not auto-scroll to the latest message
- **Problem description and how to reproduce it:**
  1. Open the chat page and send multiple messages.
  2. Observe that the chat window does not automatically scroll to show the most recent message.
- **Suggested fix:** Add a scroll-to-bottom function whenever a new message is rendered.
- **Status:** Open
- **Resolution:** Pending
- **Resolved by:** [To be determined]

---

**Problem Report 3:**

- **Problem report number:** PR-204
- **Reported by:** Abdullah
- **Date reported:** March 24, 2025
- **Program (or component) name:** Chat Feature – Input Field Behavior
- **Release number:** v2.0
- **Configuration(s):** Local browser environment
- **Report type:** Functionality bug
- **Can reproduce:** Yes
- **Severity:** Low
- **Priority:** Medium
- **Problem summary:** Input field does not clear after sending a message
- **Problem description and how to reproduce it:**
    1. Open the chat page.
    2. Type a message and click “Send.”
    3. Notice that the input field retains the sent text instead of clearing.
- **Suggested fix:** Clear the input field upon successful rendering of the new message.
- **Status:** Open
- **Resolution:** Pending
- **Resolved by:** [To be determined]

---

**Problem Report 4:**

- **Problem report number:** PR-205

- **Reported by:** Abdullah
- **Date reported:** March 24, 2025
- **Program (or component) name:** Chat Feature – Message Length Handling
- **Release number:** v2.0
- **Configuration(s):** Local browser testing environment
- **Report type:** Functionality bug
- **Can reproduce:** Yes
- **Severity:** Medium
- **Priority:** Medium
- **Problem summary:** No error handling for excessively long messages
- **Problem description and how to reproduce it:**
    1. Open the chat page and type an extremely long message.
    2. Click “Send.”
    3. Observe that the application does not restrict the message length and may lead to UI issues.
- **Suggested fix:** Implement a character limit and provide appropriate feedback if exceeded.
- **Status:** Open
- **Resolution:** Pending
- **Resolved by:** [To be determined]

---

**Problem Report 5:**

- **Problem report number:** PR-206

- **Reported by:** Abdullah
- **Date reported:** March 24, 2025
- **Program (or component) name:** Chat Feature – Timestamp Formatting
- **Release number:** v2.0
- **Configuration(s):** Local browser environment
- **Report type:** Functionality/UI bug
- **Can reproduce:** Yes
- **Severity:** Low
- **Priority:** Low
- **Problem summary:** Inconsistent or missing timestamps on messages
- **Problem description and how to reproduce it:**
    1. Open the chat page and send several messages.
    2. Observe the timestamps; some messages may show incorrect formatting or be missing.
- **Suggested fix:** Standardize timestamp generation using a consistent format and ensure every message is timestamped.
- **Status:** Open
- **Resolution:** Pending
- **Resolved by:** [To be determined]

**Chat Feature (GroupChatSidebar)**

**Overview:**
For the code review, I examined the GroupChatSidebar component with the following objectives in mind:

- **Maintainability:** Ensuring the code is modular, readable, and easy to update.
- **Performance:** Verifying that state updates and UI rendering are efficient.
- **Functionality:** Checking that the component behaves as expected.
- **User Experience:** Confirming proper UI feedback and error handling.
- **Code Smells:** Reviewing common smells such as duplication, overly complex logic, and poor naming conventions.

# Code Review:

1. **State Update Pattern (Potential Bug/Code Smell):**
    - **Issue:** The messages state is updated using `setMessages([...messages, newMessage])`, which can lead to potential race conditions when multiple updates occur quickly.
    - **Code Smell:** Using a non-functional update pattern may cause bugs in rapidly updating UIs.

**Suggestion:** Use a functional update:

```
setMessages((prevMessages) => [...prevMessages, newMessage]);
```

   -
   - **GitHub Bug Report Reference:** (See PR-203 in bug reports)

2. **Lack of Input Feedback (User Experience):**
    - **Issue:** Currently, there is no visual feedback if the user tries to send an empty message or when a message is being processed.
    - **Code Smell:** Missing affordance for error handling can lead to a poor user experience.
    - **Suggestion:** Disable the “Send” button or show an error message when the input is empty.
    - **GitHub Bug Report Reference:** (See PR-204 in bug reports)

3. **Debouncing / Multiple Message Submission (Functionality Bug):**

    - **Issue:** Rapid pressing of the “Send” button could lead to multiple submissions if the debounce is not implemented.

    - **Code Smell:** The potential for duplicate logic and race conditions if event handling isn’t optimized.

    - **Suggestion:** Implement debouncing or temporarily disable the “Send” button after submission.

    - **GitHub Bug Report Reference:** (See PR-203 in bug reports)

4. **Component Modularity (Design Smell):**

    - **Issue:** The component currently contains the complete chat logic (header, message list, input field) in a single file.

    - **Code Smell:** “Large Class” or “Long Method” smell — the component might become difficult to maintain as features are added.

    - **Suggestion:** Consider breaking the component into smaller sub-components (e.g., ChatHeader, MessageList, MessageInput).

    - **GitHub Bug Report Reference:** (This is a design suggestion to enhance maintainability)

5. **Accessibility and ARIA Attributes (UX/Design Smell):**

    - **Issue:** There are no ARIA labels or accessibility hints on interactive elements like buttons and input fields.

    - **Code Smell:** “Inadequate Feedback” — Lack of accessibility can limit usability for users with disabilities.

    - **Suggestion:** Add ARIA attributes (e.g., `aria-label`) to improve accessibility.

    - **GitHub Bug Report Reference:** (Design improvement recommendation)