TASK 2

We should have translator option on comment section for user comments along with like and dislike options. user can comment on any language as well as we should have an option to translate the comment in desired language. we should show user exact city name as well along with comments. we should not allow comments which has special characters and remove the comments automatically if it's get 2 dislike from others.

Step 1: Set Up Your Project in VS Code

Open VS Code and create a new folder for your project.

Inside the folder, create the following files:

index.html

Step 2: Write the HTML Structure

Create and open the index.html file, then add the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Comment Section with Translation</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 20px; }
    .comment-box { margin-bottom: 15px; padding: 10px; border: 1px solid #ccc; }
    .comment-header { font-weight: bold; margin-bottom: 5px; }
    .actions { margin-top: 10px; }
    button { margin-right: 5px; }
  </style>
```

```
</head>
<body>

  <h2>Comment Section</h2>
  <textarea id="commentInput" placeholder="Write your comment..." rows="3"
cols="50"></textarea><br>
  <button onclick="postComment()">Post Comment</button>

  <div id="commentsSection"></div>

  <script>
    let commentId = 0;

    // Dummy city name (mocked)
    function getUserCity() {
      return "München, Germany"; // includes special characters
    }

    function postComment() {
      const commentText = document.getElementById("commentInput").value;
      if (!commentText.trim()) return;

      const city = getUserCity();
      const id = `comment-${commentId++}`;

      const commentHTML = `
        <div class="comment-box" id="${id}">
          <div class="comment-header">User from ${city}</div>
          <div class="comment-text">${commentText}</div>
          <div class="actions">
            <button onclick="likeComment('${id}')">👍 Like</button>
            <button onclick="dislikeComment('${id}')">👎 Dislike (<span id="${id}-
dislikes">0</span>)</button>
            <button onclick="translateComment('${id}')">🌐 Translate</button>
          </div>
        </div>
      `;

document.getElementById("commentsSection").insertAdjacentHTML("beforeend",
commentHTML);
      document.getElementById("commentInput").value = '';
    }

    function likeComment(id) {
      alert("Liked comment: " + id);
    }

    function dislikeComment(id) {
      const countSpan = document.getElementById(`${id}-dislikes`);
      let count = parseInt(countSpan.textContent) + 1;
      countSpan.textContent = count;

      if (count >= 2) {
        document.getElementById(id).remove();
      }
    }

    function translateComment(id) {
      const commentEl = document.querySelector(`#${id} .comment-text`);
      const originalText = commentEl.textContent;


      // Mock translation
      commentEl.innerHTML = `🔁 Translated: "${originalText}" (en)`;
    }
  </script>
</body>
</html>
```

Step 3: Test the Chat Interface

Open the index.html file in a browser (double-click it or use Live Server in VS Code).

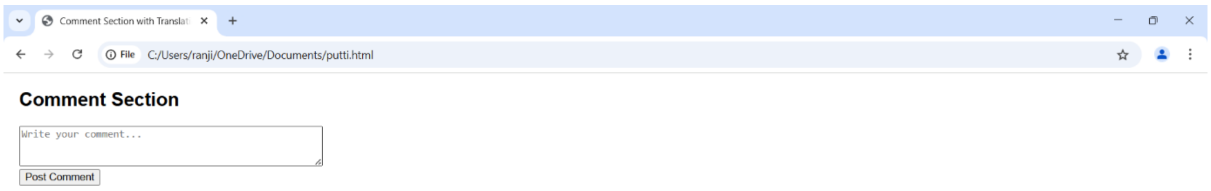
Type a Commends and click "**POST COMMEND**."

The comment should appear in the comment section.

Step 4: Add Real-Time Communication (Optional)

To make this a real-time comment, you need WebSockets using Socket.io and a Node.js backend. Would you like to continue with that?

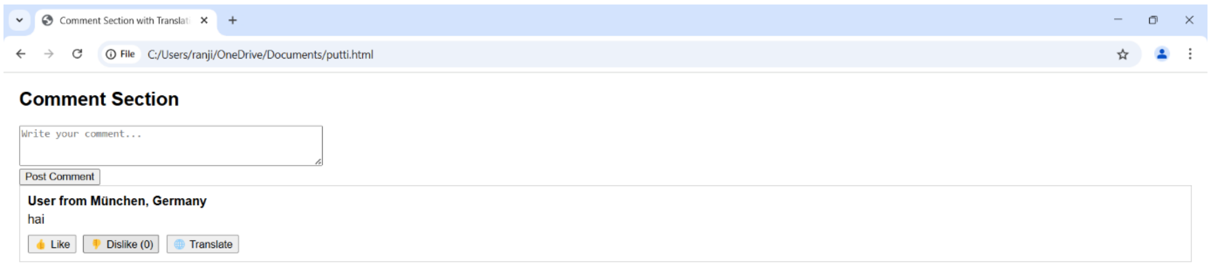
Step 6: Add Real-Time Communication (Optional) To make this a real-time chat, you need WebSockets using Socket.io and a Node.js backend. Would you like to continue with that?


Comment Section with Translat
File C:/Users/ranji/OneDrive/Documents/putti.html

Comment Section

Write your comment...

Post Comment


Comment Section with Translat
File C:/Users/ranji/OneDrive/Documents/putti.html

Comment Section

Write your comment...

Post Comment

User from München, Germany

hai

Like | Dislike (0) | Translate