



**COLLEGE CODE :- 9509**

**COLLEGE NAME :- Holy Cross Engineering College**

**DEPARTMENT :- CSE**

**STUDENT NM-ID:-**

**40302445BB0FD49BB18403F6BA781322**

**DATE :- 22/09/2025**

**Completed the project named as**

**Phase\_\_TECHNOLOGY PROJECT NAME :**

**CHAT UI APPLICATION**

**SUBMITTED BY,**

**NAME :- U.Marimuthu**

**MOBILE NO :-9790270667**

## Chat Application UI - MVP Implementation

I'll create a responsive chat application UI with core features, local state management, and a clean interface. This implementation will focus on the frontend with simulated backend functionality.

html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>ChatApp - MVP</title>
```

```
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
```

```
  <style>
```

```
    * {  
      margin: 0;  
      padding: 0;  
      box-sizing: border-box;  
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
    }
```

```
    body {  
      background: linear-gradient(135deg, #6e8efb, #a777e3);  
      height: 100vh;  
      display: flex;  
      justify-content: center;  
      align-items: center;  
      color: #333;  
    }
```

```
    .chat-container {  
      width: 90%;  
      max-width: 1000px;  
      height: 90vh;  
      background: white;  
      border-radius: 12px;  
      box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);  
      display: flex;  
      overflow: hidden;  
    }
```

```
    .sidebar {  
      width: 30%;
```

```
background: #f5f7fb;
border-right: 1px solid #e1e4e8;
display: flex;
flex-direction: column;
}
```

```
.header {
padding: 20px;
background: #6e8efb;
color: white;
display: flex;
justify-content: space-between;
align-items: center;
}
```

```
.user-info {
display: flex;
align-items: center;
gap: 10px;
}
```

```
.avatar {
width: 40px;
height: 40px;
border-radius: 50%;
background: #a777e3;
display: flex;
align-items: center;
justify-content: center;
color: white;
font-weight: bold;
}
```

```
.search-container {
padding: 15px;
border-bottom: 1px solid #e1e4e8;
}
```

```
.search-box {
display: flex;
align-items: center;
background: white;
border-radius: 20px;
padding: 8px 15px;
```

```
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
}
```

```
.search-box input {  
    flex: 1;  
    border: none;  
    outline: none;  
    padding: 5px;  
    font-size: 14px;  
}
```

```
.conversation-list {  
    flex: 1;  
    overflow-y: auto;  
    padding: 10px;  
}
```

```
.conversation {  
    display: flex;  
    align-items: center;  
    padding: 12px;  
    border-radius: 8px;  
    margin-bottom: 8px;  
    cursor: pointer;  
    transition: background 0.2s;  
}
```

```
.conversation:hover {  
    background: #e6ebf5;  
}
```

```
.conversation.active {  
    background: #e0e7ff;  
}
```

```
.conversation-avatar {  
    width: 50px;  
    height: 50px;  
    border-radius: 50%;  
    background: #a777e3;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    color: white;
```

```
    font-weight: bold;
    margin-right: 15px;
}

.conversation-info {
    flex: 1;
}

.conversation-name {
    font-weight: 600;
    margin-bottom: 4px;
}

.conversation-preview {
    font-size: 13px;
    color: #666;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    max-width: 150px;
}

.conversation-time {
    font-size: 12px;
    color: #999;
}

.chat-area {
    flex: 1;
    display: flex;
    flex-direction: column;
}

.chat-header {
    padding: 15px 20px;
    border-bottom: 1px solid #e1e4e8;
    display: flex;
    align-items: center;
}

.chat-messages {
    flex: 1;
    padding: 20px;
    overflow-y: auto;
```

```
background: #f0f4fb;
display: flex;
flex-direction: column;
gap: 15px;
}
```

```
.message {
  max-width: 70%;
  padding: 12px 16px;
  border-radius: 18px;
  margin-bottom: 5px;
  position: relative;
  animation: fadeIn 0.3s ease;
}
```

```
@keyframes fadeIn {
  from { opacity: 0; transform: translateY(10px); }
  to { opacity: 1; transform: translateY(0); }
}
```

```
.message.sent {
  align-self: flex-end;
  background: #6e8efb;
  color: white;
  border-bottom-right-radius: 5px;
}
```

```
.message.received {
  align-self: flex-start;
  background: white;
  color: #333;
  border-bottom-left-radius: 5px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.05);
}
```

```
.message-time {
  font-size: 11px;
  margin-top: 5px;
  opacity: 0.8;
  text-align: right;
}
```

```
.chat-input {
  padding: 15px 20px;
```

```
border-top: 1px solid #e1e4e8;  
display: flex;  
align-items: center;  
gap: 10px;  
}
```

```
.message-input {  
  flex: 1;  
  padding: 12px 15px;  
  border: 1px solid #e1e4e8;  
  border-radius: 24px;  
  outline: none;  
  font-size: 15px;  
}
```

```
.send-button {  
  width: 45px;  
  height: 45px;  
  border-radius: 50%;  
  background: #6e8efb;  
  color: white;  
  border: none;  
  cursor: pointer;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  transition: background 0.2s;  
}
```

```
.send-button:hover {  
  background: #5a7ce2;  
}
```

```
.typing-indicator {  
  font-size: 13px;  
  color: #666;  
  font-style: italic;  
  padding: 0 20px;  
  height: 20px;  
}
```

```
.status {  
  width: 10px;  
  height: 10px;
```

```
border-radius: 50%;  
margin-left: 10px;  
}
```

```
.status.online {  
  background: #4caf50;  
}
```

```
.status.offline {  
  background: #ccc;  
}
```

```
@media (max-width: 768px) {  
  .sidebar {  
    width: 80px;  
  }
```

```
.conversation-info, .conversation-time {  
  display: none;  
}
```

```
.conversation {  
  justify-content: center;  
}
```

```
.conversation-avatar {  
  margin-right: 0;  
}
```

```
.search-container {  
  display: none;  
}
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="chat-container">
```

```
<!-- Sidebar with conversations -->
```

```
<div class="sidebar">
```

```
<div class="header">
```

```
<div class="user-info">
```

```
<div class="avatar">JD</div>
```

```
<span>John Doe</span>
```

```
</div>
```



```
    <i class="fas fa-ellipsis-v"></i>
</div>
```

```
<div class="search-container">
  <div class="search-box">
    <i class="fas fa-search"></i>
    <input type="text" placeholder="Search conversations...">
  </div>
</div>
```

```
<div class="conversation-list">
  <div class="conversation active">
    <div class="conversation-avatar">AS</div>
    <div class="conversation-info">
      <div class="conversation-name">Alice Smith <span class="status online"></span></div>
      <div class="conversation-preview">Hey, how are you doing?</div>
    </div>
    <div class="conversation-time">10:45 AM</div>
  </div>
```

```
<div class="conversation">
  <div class="conversation-avatar">BJ</div>
  <div class="conversation-info">
    <div class="conversation-name">Bob Johnson <span class="status offline"></span></div>
    <div class="conversation-preview">I'll send you the files tomorrow</div>
  </div>
  <div class="conversation-time">Yesterday</div>
</div>
```

```
<div class="conversation">
  <div class="conversation-avatar">EW</div>
  <div class="conversation-info">
    <div class="conversation-name">Emma Wilson <span class="status online"></span></div>
    <div class="conversation-preview">Meeting at 3 PM confirmed</div>
  </div>
  <div class="conversation-time">Wednesday</div>
</div>
```

```
<div class="conversation">
  <div class="conversation-avatar">MR</div>
  <div class="conversation-info">
    <div class="conversation-name">Mike Ross <span class="status online"></span></div>
    <div class="conversation-preview">Thanks for your help!</div>
  </div>
```

```
    <div class="conversation-time">Monday</div>
  </div>
</div>
</div>
```

```
<!-- Main chat area -->
```

```
<div class="chat-area">
  <div class="chat-header">
    <div class="user-info">
      <div class="avatar">AS</div>
      <div>
        <div class="conversation-name">Alice Smith <span class="status online"></span></div>
        <div class="conversation-preview">Online</div>
      </div>
    </div>
  </div>
</div>
```

```
<div class="chat-messages" id="chat-messages">
  <div class="message received">
    <div class="message-text">Hey there! How's your project coming along?</div>
    <div class="message-time">10:30 AM</div>
  </div>

  <div class="message sent">
    <div class="message-text">It's going well! Just finished the UI for the chat application.</div>
    <div class="message-time">10:32 AM</div>
  </div>
```

```
<div class="message received">
  <div class="message-text">That's great! Can you show me a screenshot?</div>
  <div class="message-time">10:33 AM</div>
</div>
```

```
<div class="message sent">
  <div class="message-text">Sure, I'll send it in a bit. Just adding some final touches.</div>
  <div class="message-time">10:34 AM</div>
</div>
```

```
<div class="message received">
  <div class="message-text">Awesome! Looking forward to seeing it. By the way, are we still
meeting tomorrow?</div>
  <div class="message-time">10:35 AM</div>
</div>
```

```

<div class="message sent">
  <div class="message-text">Yes, 10 AM as planned. I've sent the calendar invite.</div>
  <div class="message-time">10:36 AM</div>
</div>

<div class="message received">
  <div class="message-text">Perfect. I'll see you then!</div>
  <div class="message-time">10:37 AM</div>
</div>
</div>

<div class="typing-indicator" id="typing-indicator"></div>

<div class="chat-input">
  <input type="text" class="message-input" placeholder="Type a message..." id="message-input">
  <button class="send-button" id="send-button">
    <i class="fas fa-paper-plane"></i>
  </button>
</div>
</div>
</div>

<script>
document.addEventListener('DOMContentLoaded', function() {
  const chatMessages = document.getElementById('chat-messages');
  const messageInput = document.getElementById('message-input');
  const sendButton = document.getElementById('send-button');
  const typingIndicator = document.getElementById('typing-indicator');

  // Sample responses for demo purposes
  const responses = [
    "That's interesting!",
    "I see what you mean.",
    "Tell me more about that.",
    "I agree with you.",
    "That makes sense.",
    "What do you think about this weather?",
    "I've been thinking about that too.",
    "Let's discuss this tomorrow."
  ];

  // Function to add a message to the chat
  function addMessage(text, isSent = true) {
    const messageDiv = document.createElement('div');

```

```

messageDiv.classList.add('message');
messageDiv.classList.add(isSent ? 'sent' : 'received');

const messageText = document.createElement('div');
messageText.classList.add('message-text');
messageText.textContent = text;

const messageTime = document.createElement('div');
messageTime.classList.add('message-time');

// Get current time
const now = new Date();
let hours = now.getHours();
let minutes = now.getMinutes();
const ampm = hours >= 12 ? 'PM' : 'AM';
hours = hours % 12;
hours = hours ? hours : 12; // the hour '0' should be '12'
minutes = minutes < 10 ? '0'+minutes : minutes;
messageTime.textContent = `${hours}:${minutes} ${ampm}`;

messageDiv.appendChild(messageText);
messageDiv.appendChild(messageTime);

chatMessages.appendChild(messageDiv);

// Scroll to bottom
chatMessages.scrollTop = chatMessages.scrollHeight;
}

// Function to simulate receiving a message
function simulateResponse() {
  typingIndicator.textContent = "Alice is typing...";

  // Random delay between 1-3 seconds
  const delay = 1000 + Math.random() * 2000;

  setTimeout(() => {
    typingIndicator.textContent = "";
    const randomResponse = responses[Math.floor(Math.random() * responses.length)];
    addMessage(randomResponse, false);
  }, delay);
}

// Event listener for send button

```

```

sendButton.addEventListener('click', function() {
  const message = messageInput.value.trim();
  if (message) {
    addMessage(message);
    messageInput.value = "";

    // Simulate response after a short delay
    setTimeout(simulateResponse, 500);
  }
});

// Event listener for Enter key
messageInput.addEventListener('keypress', function(e) {
  if (e.key === 'Enter') {
    sendButton.click();
  }
});

// Demo: Simulate receiving a message after 2 seconds on load
setTimeout(() => {
  addMessage("Hey, glad to see you online! How's everything going?", false);
}, 2000);
});
</script>
</body>
</html>

```

## Features Implemented

1. **Project Setup** - Complete HTML5 structure with responsive design
2. **Core Features:**
  - Conversation list with online/offline status
  - Real-time message sending and receiving
  - Message history with timestamps
  - Typing indicators
  - Responsive design for mobile and desktop
3. **Data Storage:**
  - Local state management using JavaScript
  - Simulated conversation data
4. **UI/UX Elements:**
  - Clean, modern chat interface
  - Message bubbles with different styles for sent/received
  - Animated message appearance
  - User avatars and status indicators
5. **Testing:**

- The interface is fully functional for demonstration
- You can send messages and receive simulated responses