```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AI FAQ Chatbot</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <style>
    @keyframes float {
      0% { transform: translateY(0px); }
      50% { transform: translateY(-10px); }
      100% { transform: translateY(0px); }
    }

    .bot-icon {
      animation: float 3s ease-in-out infinite;
    }

    .message {
      opacity: 0;
      transform: translateY(20px);
      transition: all 0.3s ease-out;
    }

    .message.visible {
      opacity: 1;
      transform: translateY(0);
    }

    .typing-indicator span {
      display: inline-block;
      width: 8px;
      height: 8px;
      border-radius: 50%;
      background-color: #4f46e5;
      margin: 0 2px;
      opacity: 0.4;
    }

    .typing-indicator span:nth-child(1) {
      animation: bounce 1s infinite;
    }

    .typing-indicator span:nth-child(2) {
            animation: bounce 1s infinite 0.2s;
        }

        .typing-indicator span:nth-child(3) {
            animation: bounce 1s infinite 0.4s;
        }

        @keyframes bounce {
            0%, 100% { transform: translateY(0); opacity: 0.4; }
            50% { transform: translateY(-5px); opacity: 1; }
        }

        .feedback-button {
            transition: all 0.2s ease;
        }

        .feedback-button:hover {
            transform: scale(1.1);
        }
    </style>
</head>
<body class="bg-gray-100 min-h-screen flex items-center justify-center p-4">
    <div class="w-full max-w-md">
        <div class="bg-white rounded-2xl shadow-xl overflow-hidden">
            <!-- Chat header -->
            <div class="bg-indigo-600 text-white p-4 flex items-center">
                <div class="bot-icon bg-white rounded-full p-2 mr-3">
                    <svg xmlns="http://www.w3.org/2000/svg" class="h-8 w-8 text-indigo-600"
fill="none" viewBox="0 0 24 24" stroke="currentColor">
                        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9
3v2m6-2v2M9 19v2m6-2v2M5 9H3m2 6H3m18-6h-2m2 6h-2M7 19h10a2 2 0 002-2V7a2 2 0
00-2-2H7a2 2 0 00-2 2v10a2 2 0 002 2zM9 9h6v6H9V9z" />
                    </svg>
                </div>
                <div>
                    <h1 class="text-xl font-bold">AI FAQ Assistant</h1>
                    <p class="text-indigo-200 text-sm">Ask me anything about our services</p>
                </div>
            </div>

            <!-- Chat messages container -->
            <div id="chat-container" class="h-96 overflow-y-auto p-4 space-y-3">
                <!-- Initial bot message -->
                <div class="message visible">
                    <div class="flex items-start">
                        <div class="bg-indigo-100 rounded-lg p-3 max-w-xs">
                            <p class="text-gray-800">Hello! I'm your AI assistant. Ask me anything about
our services, pricing, or features. I'm here to help!</p>
                        </div>
                    </div>
                </div>
            </div>

            <!-- User input area -->
            <div class="border-t border-gray-200 p-4 bg-gray-50">
                <div class="flex space-x-2">
                    <input
                        id="user-input"
                        type="text"
                        placeholder="Type your question here..."
                        class="flex-1 border border-gray-300 rounded-full px-4 py-2 focus:outline-none
focus:ring-2 focus:ring-indigo-500"
                        autocomplete="off"
                    >
                    <button
                        id="send-button"
                        class="bg-indigo-600 text-white rounded-full p-2 hover:bg-indigo-700
focus:outline-none focus:ring-2 focus:ring-indigo-500 transition"
                    >
                        <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none"
viewBox="0 0 24 24" stroke="currentColor">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12
19l9 2-9-18-9 18 9-2zm0 0v-8" />
                        </svg>
                    </button>
                </div>
                <div class="mt-2 text-xs text-gray-500 text-center">
                    Try asking: "What are your business hours?" or "How can I contact support?"
                </div>
            </div>
        </div>
    </div>

    <script>
        // FAQ knowledge base with keywords and responses
        const faqDatabase = [
            {
                keywords: ['hello', 'hi', 'hey'],
                response: "Hello there! How can I assist you today?",
                followUp: []
            },
            {
                keywords: ['hours', 'open', 'close', 'business hours', 'time'],
                response: "Our business hours are Monday to Friday from 9:00 AM to 6:00 PM. We're
closed on weekends and public holidays.",
                followUp: [
                    "Would you like to schedule an appointment outside these hours?",
                    "Do you need directions to our location?"
                ]
            },
            {
                keywords: ['contact', 'email', 'phone', 'number', 'address'],
                response: "You can contact us via:\n- Phone: (123) 456-7890\n- Email:
support@example.com\n- Address: 123 Business Ave, City, Country\n\nOur support team is
available 24/7 for urgent matters.",
                followUp: [
                    "Would you like me to connect you with a specific department?",
                    "Should I send these contact details to your email?"
                ]
            },
            {
                keywords: ['pricing', 'cost', 'price', 'fee', 'how much'],
                response: "Our pricing depends on the service you need:\n\n- Basic Plan:
$9.99/month\n- Pro Plan: $19.99/month\n- Enterprise Plan: Custom pricing\n\nAll plans come
with a 30-day money-back guarantee.",
                followUp: [
                    "Would you like a detailed breakdown of features for each plan?",
                    "Can I help you choose the right plan for your needs?"
                ]
            },
            {
                keywords: ['features', 'what can you do', 'services', 'offer'],
                response: "We offer a comprehensive suite of services including:\n\n- AI-powered
analytics\n- 24/7 customer support\n- Secure cloud storage\n- Custom integrations\n- Regular
feature updates\n\nIs there a specific feature you'd like to know more about?",
                followUp: []
            },
            {
                keywords: ['support', 'help', 'issue', 'problem'],
                response: "I'm sorry to hear you're experiencing issues. Our support team can help
with:\n\n- Technical problems\n- Account issues\n- Billing questions\n- Feature
requests\n\nWould you like me to create a support ticket for you?",
                followUp: [
                    "Can you describe the issue in more detail?",
                    "Is this affecting your ability to use our service?"
                ]
            },
            {
                keywords: ['thank', 'thanks', 'appreciate'],
                response: "You're very welcome! Is there anything else I can help you with today?",
                followUp: []
            }
        ];

        // DOM elements
        const chatContainer = document.getElementById('chat-container');
        const userInput = document.getElementById('user-input');
        const sendButton = document.getElementById('send-button');

        // Add a message to the chat
        function addMessage(text, isUser = false) {
            const messageDiv = document.createElement('div');
            messageDiv.className = 'message';

            if (isUser) {
                messageDiv.innerHTML = `
                    <div class="flex items-start justify-end">
                        <div class="bg-indigo-600 text-white rounded-lg p-3 max-w-xs">
                            <p>${text}</p>
                        </div>
                    </div>
                `;
            } else {
                messageDiv.innerHTML = `
                    <div class="flex items-start">
                        <div class="bg-indigo-100 rounded-lg p-3 max-w-xs">
                            <p class="text-gray-800">${text}</p>
                        </div>
                    </div>
                `;
            }

            chatContainer.appendChild(messageDiv);

            // Scroll to bottom and make message visible
            setTimeout(() => {
                chatContainer.scrollTop = chatContainer.scrollHeight;
                messageDiv.classList.add('visible');
            }, 10);
        }

        // Show typing indicator
        function showTypingIndicator() {
            const typingDiv = document.createElement('div');
            typingDiv.className = 'flex items-start typing-indicator';
            typingDiv.innerHTML = `
                <div class="bg-indigo-100 rounded-lg p-3">
                    <div class="flex space-x-1">
                        <span></span>
                        <span></span>
                        <span></span>
                    </div>
                </div>
            `;
            chatContainer.appendChild(typingDiv);
            chatContainer.scrollTop = chatContainer.scrollHeight;
            return typingDiv;
        }

        // Hide typing indicator
        function hideTypingIndicator(typingDiv) {
            if (typingDiv && typingDiv.parentNode) {
                typingDiv.parentNode.removeChild(typingDiv);
            }
        }

        // Process user input and generate response
        function processInput() {
            const inputText = userInput.value.trim();
            if (!inputText) return;

            // Add user message to chat
            addMessage(inputText, true);
            userInput.value = '';

            // Show typing indicator
            const typingIndicator = showTypingIndicator();

            // Simulate AI processing delay
            setTimeout(() => {
```

```
                hideTypingIndicator(typingIndicator);

                // Find the best matching FAQ response
                const lowerInput = inputText.toLowerCase();
                let bestMatch = null;
                let maxScore = 0;

                for (const faq of faqDatabase) {
                    let score = 0;
                    for (const keyword of faq.keywords) {
                        if (lowerInput.includes(keyword)) {
                            score++;
                        }
                    }

                    if (score > maxScore) {
                        maxScore = score;
                        bestMatch = faq;
                    }
                }

                // Generate response
                if (bestMatch && maxScore > 0) {
                    // Add main response
                    addMessage(bestMatch.response);

                    // Add follow-up questions if available
                    if (bestMatch.followUp.length > 0) {
                        setTimeout(() => {
                            addMessage("Follow-up:\n" + bestMatch.followUp.join("\n\n"));
                        }, 800);
                    }
                } else {
                    // No match found
                    addMessage("I'm not sure I understand. Could you rephrase your question or try
asking about our services, pricing, or contact information?");
                    addMessage("I'll make sure to learn from this interaction to better assist you in the
future!");
                }
            }, 1500 + Math.random() * 1000); // Random delay between 1.5-2.5 seconds
        }

        // Event listeners
        sendButton.addEventListener('click', processInput);
        userInput.addEventListener('keypress', (e) => {
            if (e.key === 'Enter') {
                processInput();
            }
        });
    </script>
</body>
</html>
```