

CODING:

Front-End:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>T-800 Fitness Coach</title>

  <style>

    :root {

      --primary: #2c3e50;

      --secondary: #3498db;

      --accent: #e74c3c;

      --light: #ecf0f1;

      --dark: #2c3e50;

      --success: #2ecc71;

    }

    * {

      box-sizing: border-box;

      margin: 0;

      padding: 0;

      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

    }

    body {

      background-color: #f5f5f5;

      color: var(--dark);

      line-height: 1.6;
```

```
}
```

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

```
header {  
  background-color: var(--primary);  
  color: white;  
  padding: 20px 0;  
  text-align: center;  
  border-radius: 0 0 10px 10px;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
}
```

```
h1 {  
  font-size: 2.5rem;  
  margin-bottom: 10px;  
}
```

```
.tagline {  
  font-size: 1.2rem;  
  opacity: 0.9;  
}
```

```
.chat-container {  
  display: flex;
```

```
flex-direction: column;

height: 70vh;

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

margin: 30px 0;

overflow: hidden;

}
```

```
.chat-header {

background-color: var(--secondary);

color: white;

padding: 15px;

font-weight: bold;

display: flex;

align-items: center;

}
```

```
.chat-header img {

width: 40px;

height: 40px;

margin-right: 10px;

border-radius: 50%;

}
```

```
.chat-messages {

flex: 1;

padding: 20px;

overflow-y: auto;
```

```
    background-color: #f9f9f9;
}
```

```
.message {
    margin-bottom: 15px;
    display: flex;
    flex-direction: column;
}
```

```
.message-user {
    align-items: flex-end;
}
```

```
.message-bot {
    align-items: flex-start;
}
```

```
.message-content {
    max-width: 70%;
    padding: 12px 15px;
    border-radius: 18px;
    margin-top: 5px;
    position: relative;
}
```

```
.user-message {
    background-color: var(--secondary);
    color: white;
    border-bottom-right-radius: 5px;
```

```
}
```

```
.bot-message {  
  background-color: var(--light);  
  color: var(--dark);  
  border-bottom-left-radius: 5px;  
  white-space: pre-wrap;  
  line-height: 1.6;  
}
```

```
.bot-message b {  
  color: var(--primary);  
  font-weight: bold;  
}
```

```
.bot-message ul {  
  padding-left: 20px;  
  margin: 10px 0;  
  list-style-type: disc;  
}
```

```
.bot-message li {  
  margin-bottom: 5px;  
}
```

```
.bot-message p {  
  margin-bottom: 10px;  
}
```

```
.message-sender {  
  font-size: 0.8rem;  
  font-weight: bold;  
  margin-bottom: 3px;  
}
```

```
.input-area {  
  display: flex;  
  padding: 15px;  
  background-color: white;  
  border-top: 1px solid #eee;  
}
```

```
#user-input {  
  flex: 1;  
  padding: 12px 15px;  
  border: 1px solid #ddd;  
  border-radius: 25px;  
  outline: none;  
  font-size: 1rem;  
}
```

```
#send-btn {  
  background-color: var(--secondary);  
  color: white;  
  border: none;  
  border-radius: 25px;  
  padding: 0 20px;  
  margin-left: 10px;
```

```
    cursor: pointer;
    font-weight: bold;
    transition: background-color 0.3s;
}
```

```
#send-btn:hover {
    background-color: #2980b9;
}
```

```
.user-info-form {
    background-color: white;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    margin: 30px 0;
}
```

```
.user-info-form h2 {
    margin-bottom: 20px;
    color: var(--primary);
}
```

```
.form-group {
    margin-bottom: 15px;
}
```

```
.form-group label {
    display: block;
    margin-bottom: 5px;
```

```
    font-weight: bold;
}
```

```
.form-group input, .form-group select {
    width: 100%;
    padding: 10px;
    border: 1px solid #ddd;
    border-radius: 5px;
    font-size: 1rem;
}
```

```
#submit-info {
    background-color: var(--success);
    color: white;
    border: none;
    padding: 12px 20px;
    border-radius: 5px;
    cursor: pointer;
    font-weight: bold;
    font-size: 1rem;
    transition: background-color 0.3s;
}
```

```
#submit-info:hover {
    background-color: #27ae60;
}
```

```
.typing-indicator {
    display: none;
```



```
padding: 10px 15px;
background-color: var(--light);
border-radius: 18px;
margin-bottom: 15px;
width: fit-content;
}
```

```
.typing-dots {
  display: flex;
}
```

```
.typing-dot {
  width: 8px;
  height: 8px;
  background-color: #999;
  border-radius: 50%;
  margin: 0 2px;
  animation: typingAnimation 1.4s infinite ease-in-out;
}
```

```
.typing-dot:nth-child(1) {
  animation-delay: 0s;
}
```

```
.typing-dot:nth-child(2) {
  animation-delay: 0.2s;
}
```

```
.typing-dot:nth-child(3) {
```

```
    animation-delay: 0.4s;
}
```

```
@keyframes typingAnimation {
    0%, 60%, 100% {
        transform: translateY(0);
    }
    30% {
        transform: translateY(-5px);
    }
}
```

```
@media (max-width: 768px) {
    .container {
        padding: 10px;
    }
}
```

```
.message-content {
    max-width: 85%;
}
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<header>
```

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

```
<h1>T-800 Fitness Coach</h1>
```

```
<p class="tagline">Your AI-powered fitness and nutrition assistant</p>
```

```
</div>
```

```
</header>
```

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

```
  <div id="user-info-section" class="user-info-form">
```

```
    <h2>Let's Get Started!</h2>
```

```
    <p>Please provide some information to personalize your fitness journey:</p>
```

```
    <form id="user-info-form">
```

```
      <div class="form-group">
```

```
        <label for="age">Age</label>
```

```
        <input type="number" id="age" required min="10" max="100">
```

```
      </div>
```

```
      <div class="form-group">
```

```
        <label for="weight">Weight (kg)</label>
```

```
        <input type="number" id="weight" required min="30" max="200" step="0.1">
```

```
      </div>
```

```
      <div class="form-group">
```

```
        <label for="goal">Fitness Goal</label>
```

```
        <select id="goal" required>
```

```
          <option value="">Select your goal</option>
```

```
          <option value="muscle gain">Muscle Gain</option>
```

```
          <option value="weight loss">Weight Loss</option>
```

```
          <option value="maintenance">Maintenance</option>
```

```
          <option value="endurance">Endurance</option>
```

```
          <option value="strength">Strength</option>
```

```
        </select>
```

```
      </div>
```

```
      <div class="form-group">
```

```
        <label for="fitness-level">Fitness Level</label>
```

```
        <select id="fitness-level" required>
```

```
        <option value="">Select your level</option>
        <option value="beginner">Beginner</option>
        <option value="intermediate">Intermediate</option>
        <option value="advanced">Advanced</option>
    </select>
</div>

<div class="form-group">
    <label for="region">Region</label>
    <input type="text" id="region" required placeholder="e.g., South India, North
India">
</div>

<div class="form-group">
    <label for="diet">Diet Preference</label>
    <select id="diet" required>
        <option value="">Select your diet</option>
        <option value="non-veg">Non-vegetarian</option>
        <option value="veg">Vegetarian</option>
        <option value="vegan">Vegan</option>
    </select>
</div>

<button type="submit" id="submit-info">Start Chat</button>
</form>
</div>

<div id="chat-section" class="chat-container" style="display: none;">
    <div class="chat-header">
        
        <span>T-800 Fitness Coach</span>
    </div>
```

```
<div class="chat-messages" id="chat-messages">
  <!-- Messages will appear here -->
</div>

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

<div class="input-area">
  <input type="text" id="user-input" placeholder="Type your message here...">
  <button id="send-btn">Send</button>
</div>
</div>
</div>

<script>
document.addEventListener('DOMContentLoaded', function() {
  const userInfoForm = document.getElementById('user-info-form');
  const chatSection = document.getElementById('chat-section');
  const userInfoSection = document.getElementById('user-info-section');
  const chatMessages = document.getElementById('chat-messages');
  const userInput = document.getElementById('user-input');
  const sendBtn = document.getElementById('send-btn');
  const typingIndicator = document.getElementById('typing-indicator');

  let userInfo = {};
  let messageHistory = [];
```

```
// Handle user info submission

userInfoForm.addEventListener('submit', function(e) {
  e.preventDefault();

  userInfo = {
    age: document.getElementById('age').value,
    weight: document.getElementById('weight').value,
    goal: document.getElementById('goal').value,
    fitness_level: document.getElementById('fitness-level').value,
    region: document.getElementById('region').value,
    diet: document.getElementById('diet').value
  };

  // Hide user info form and show chat
  userInfoSection.style.display = 'none';
  chatSection.style.display = 'flex';

  // Add welcome message
  addMessage('bot', "Hello! I'm T-800, your AI fitness coach. How can I help you
today?");

  // Initialize message history with system prompt
  const systemPrompt = `You are T-800, a friendly and knowledgeable fitness and
nutrition coach.

You help users with:
- Personalized workout plans (based on their age, goal, fitness level)
- Custom diet plans (based on calories, region, food preference)
- Answering fitness, health, and nutrition-related questions
- Having friendly, motivational conversation`
```

Response Formatting Guidelines:

1. Use clear section headings
2. Use bullet points for lists
3. Separate paragraphs with blank lines
4. Use **asterisks** for emphasis
5. Structure complex information logically

User Info: \${JSON.stringify(userInfo)}`;

```
messageHistory = [  
  { role: "system", content: systemPrompt }  
];  
});
```

```
// Handle sending messages  
sendBtn.addEventListener('click', sendMessage);  
userInput.addEventListener('keypress', function(e) {  
  if (e.key === 'Enter') {  
    sendMessage();  
  }  
});
```

```
function sendMessage() {  
  const message = userInput.value.trim();  
  if (message === "") return;  
  
  addMessage('user', message);  
  messageHistory.push({ role: "user", content: message });  
}
```

```
userInput.value = "";

// Show typing indicator
typingIndicator.style.display = 'block';
chatMessages.scrollTop = chatMessages.scrollHeight;

// Call the backend API
fetch('/chat', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    messages: messageHistory
  })
})
.then(response => response.json())
.then(data => {
  // Hide typing indicator
  typingIndicator.style.display = 'none';

  if (data.error) {
    addMessage('bot', "Sorry, I encountered an error. Please try again.");
    console.error(data.error);
  } else {
    const aiMessage = data.response;
    addMessage('bot', aiMessage);
    messageHistory.push({ role: "assistant", content: aiMessage });
  }
})
```



```

    })
    .catch(error => {
        typingIndicator.style.display = 'none';
        addMessage('bot', "Sorry, I'm having trouble connecting. Please try again later.");
        console.error('Error:', error);
    });
}

```

```

function addMessage(sender, text) {
    const messageDiv = document.createElement('div');
    messageDiv.className = `message message-${sender}`;

    const senderDiv = document.createElement('div');
    senderDiv.className = 'message-sender';
    senderDiv.textContent = sender === 'user' ? 'You' : 'T-800';

    const contentDiv = document.createElement('div');
    contentDiv.className = `message-content ${sender}-message`;

    // Format the text differently for bot vs user
    if (sender === 'bot') {
        contentDiv.innerHTML = formatBotResponse(text);
    } else {
        contentDiv.textContent = text;
    }

    messageDiv.appendChild(senderDiv);
    messageDiv.appendChild(contentDiv);
    chatMessages.appendChild(messageDiv);
}

```

```

    // Scroll to bottom

    chatMessages.scrollTop = chatMessages.scrollHeight;
}

function formatBotResponse(text) {
    // Convert markdown-like formatting to HTML

    let formatted = text

    // Preserve line breaks

    .replace(/\n/g, '<br>')

    // Convert *bold* to <b>bold</b>

    .replace(/\*([^\*]+)\*/g, '<b>$1</b>')

    // Convert - bullet points to proper list items

    .replace(/^- (.+?)(<br>|$)/gm, '<li>$1</li>')

    // Add extra spacing after headings

    .replace(/(<br><b>.+?<\b><br>)/g, '$1<br>')

    // Convert numbered lists

    .replace(/^(\\d+)\. (.+?)(<br>|$)/gm, '<li>$1. $2</li>');

    // Wrap list items in unordered list tags

    if (formatted.includes('<li>')) {
        formatted = formatted.replace(/(<li>.+?<\li>)+/g, '<ul>$&</ul>');
    }

    return formatted;
}

});
</script>
</body>

```

</html>

Back-End:

```
from flask import Flask, request, jsonify, send_from_directory
```

```
import requests
```

```
import os
```

```
app = Flask(__name__, static_folder='static')
```

```
# Configuration
```

```
TOGETHER_API_KEY = "api-key" # 🔑 Replace with your actual key
```

```
MODEL_NAME = "mistralai/Mistral-7B-Instruct-v0.2"
```

```
# Serve the frontend
```

```
@app.route('/')
```

```
def serve_frontend():
```

```
    return send_from_directory('.', 'index.html')
```

```
# Chat API endpoint
```

```
@app.route('/chat', methods=['POST'])
```

```
def chat():
```

```
    try:
```

```
        data = request.json
```

```
        messages = data.get('messages', [])
```

```
        payload = {
```

```
            "model": MODEL_NAME,
```

```
            "messages": messages,
```

```
            "temperature": 0.7,
```

```
            "max_tokens": 800
```

```
}
```

```
headers = {  
    "Authorization": f"Bearer {TOGETHER_API_KEY}",  
    "Content-Type": "application/json"  
}
```

```
response = requests.post(  
    "https://api.together.xyz/v1/chat/completions",  
    json=payload,  
    headers=headers  
)
```

```
response.raise_for_status()
```

```
ai_message = response.json()["choices"][0]["message"]["content"]  
return jsonify({"response": ai_message})
```

```
except Exception as e:  
    return jsonify({"error": str(e)}), 500
```

```
if __name__ == '__main__':  
    app.run(debug=True, port=5000)
```