

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DAY – 19

17 July 2025

LLaMA 3 Frontend Chat Application

Setup Instructions

Step 1: Run LLaMA 3 backend

If you use Ollama:

ollama run llama3

```
step@step-HP-ProDesk-400-G5-SFF:~$ ollama run llama3
>>> what is ai
A great question!

Artificial Intelligence (AI) refers to the development of computer systems that can perform tasks that would typically require human intelligence, such as:

1. **Learning**: AI systems can learn from data and improve their performance over time.
2. **Reasoning**: AI systems can draw conclusions based on given information and make decisions.
3. **Problem-solving**: AI systems can find solutions to complex problems.
4. **Perception**: AI systems can interpret and understand sensory information, such as images, speech, or text.

AI systems use various techniques, including:

1. **Machine learning**: AI systems learn from data and improve their performance over time.
2. **Deep learning**: A subset of machine learning that uses neural networks to analyze complex data.
3. **Natural language processing (NLP)**: AI systems can understand, generate, and process human language.
4. **Computer vision**: AI systems can interpret and understand visual information from images or videos.
```

Step 2: Basic Frontend to Connect to Ollama API

Create an i.html file to serve a basic web UI

```
step@step-HP-ProDesk-400-G5-SFF:~$ vim i.html
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>LLaMA 3 Chat</title>
  <style>
    body { font-family: sans-serif; margin: 20px; }
    #chat { width: 100%; height: 300px; border: 1px solid #ccc; padding: 10px; overflow-y: auto; white-space: pre-wrap; background: #f9f9f9; }
    #input { width: 80%; padding: 10px; }
    #send { padding: 10px; }
  </style>
</head>
<body>
  <h2>LLaMA 3 Chat (via Ollama)</h2>
  <div id="chat"></div><br>
  <input type="text" id="input" placeholder="Type a message..." />
  <button id="send">Send</button>

  <script>
    const chatBox = document.getElementById('chat');
    const input = document.getElementById('input');
    const sendBtn = document.getElementById('send');

    sendBtn.onclick = async () => {
      const userText = input.value.trim();
      if (!userText) return;

      appendMessage('You', userText);
      input.value = '';
      input.disabled = true;
      sendBtn.disabled = true;

      let aiMessage = '';
      appendMessage('LLaMA', '');

      const response = await fetch('http://localhost:11434/api/chat', {
        method: 'POST',
```

```

    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
      model: 'llama3',
      messages: [{ role: 'user', content: userText }],
      stream: true
    })
  });

  const reader = response.body.getReader();
  const decoder = new TextDecoder();

  while (true) {
    const { value, done } = await reader.read();
    if (done) break;
    const chunk = decoder.decode(value);
    const lines = chunk.split('\n').filter(line => line.trim());

    for (const line of lines) {
      try {
        const json = JSON.parse(line);
        const delta = json?.message?.content || '';
        aiMessage += delta;
        updateLastMessage('LLaMA', aiMessage);
      } catch (e) {
        console.error('Invalid JSON chunk:', line);
      }
    }
  }

  input.disabled = false;
  sendBtn.disabled = false;
  input.focus();
};

function appendMessage(sender, message) {
  const div = document.createElement('div');
  div.className = 'message';

```

```

    div.innerHTML = `<strong>${sender}</strong> <span>${message}</span>`;
    chatBox.appendChild(div);
    chatBox.scrollTop = chatBox.scrollHeight;
  }

  function updateLastMessage(sender, newText) {
    const messages = chatBox.getElementsByClassName('message');
    if (!messages.length) return;
    const last = messages[messages.length - 1];
    const span = last.querySelector('span');
    span.textContent = newText;
    chatBox.scrollTop = chatBox.scrollHeight;
  }
</script>
</body>
</html>

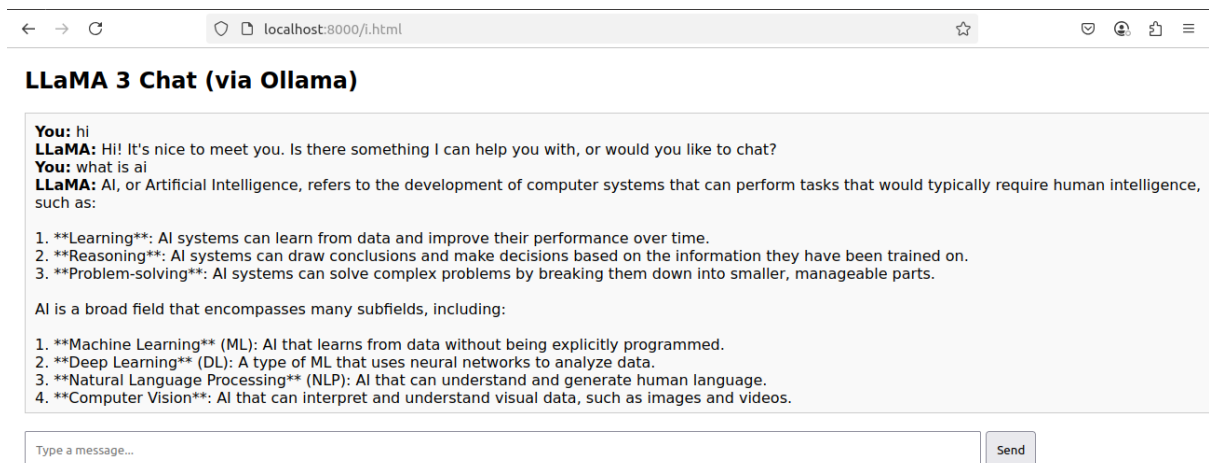
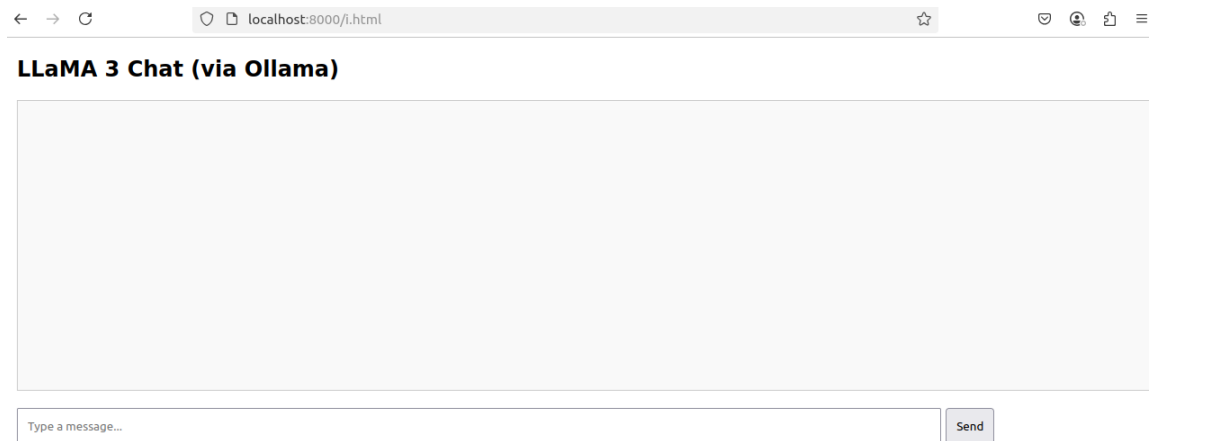
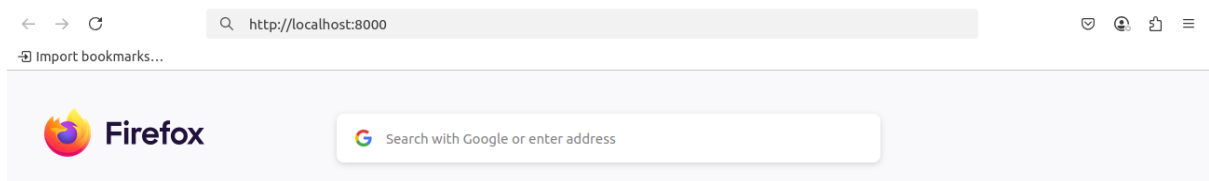
```

Step 3: Serve Frontend on localhost

Use a simple local server

```
step@step-HP-ProDesk-400-G5-SFF:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [18/Jul/2025 09:01:02] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Jul/2025 09:01:04] "GET /i.html HTTP/1.1" 200 -
```

Then go to:



Name : Aditi Tangri

URN : 2302460

CRN : 2315004