

# lessoncode4

app\_web\_basic.py

## Basic Web-Based AI Chat App (Gradio + Ollama)

This Python script creates a **simple web app** using **Gradio** that allows users to interact with the **DeepSeek 1.5B AI model** via the **Ollama CLI**. The app provides a **user-friendly interface** where users can enter a prompt and receive a response.

## Code Explanation

### 1 Importing Required Modules

```
import gradio as gr
import subprocess
```

- **gradio** → Used to create the web interface.
- **subprocess** → Allows running terminal commands to interact with **Ollama**.

### 2 Function to Run AI Model

```
def run_ollama_prompt(prompt, model="deepseek-r1:1.5b"):
    """Runs a prompt using Ollama CLI and returns the response."""
    command = ["ollama", "run", model, prompt]

    try:
        result = subprocess.run(command, capture_output=True, text=True, c
heck=True, encoding="utf-8")
        return result.stdout.strip()
    except subprocess.CalledProcessError as e:
        return f"Error running Ollama: {e}"
```

- This function **runs Ollama** in the terminal with the specified AI model.
- **Captures and returns the AI's response.**

- **Handles errors** if the command fails.

### 3 Connecting AI with the Web App

```
def generate_response(prompt):  
    return run_ollama_prompt(prompt)
```

- This function **links the AI model** with the Gradio interface.
- It **takes user input** and **returns the AI-generated response**.

### 4 Creating the Web Interface

```
with gr.Blocks() as app:  
    gr.Markdown("# DeepSeek 1.5B Chat App")  
  
    with gr.Row():  
        prompt_input = gr.Textbox(label="Enter your prompt", placeholder="Type something...")  
  
    with gr.Row():  
        submit_button = gr.Button("Generate Response")  
  
    response_output = gr.Textbox(label="Response", interactive=False)  
  
    submit_button.click(fn=generate_response, inputs=prompt_input, outputs=response_output)
```

#### ♦ What This Does:

- **Displays a title** using `gr.Markdown("# DeepSeek 1.5B Chat App")`.
- **Creates a text input box** for the user to type a prompt.
- **Adds a "Generate Response" button** to trigger AI processing.
- **Displays AI responses** in a non-editable output box.
- **Connects UI elements:** When the button is clicked, `generate_response()` is called.

### 5 Launching the App

```
app.launch()
```

- Starts the **Gradio web app** on a local server.
- 

## **How to Run the App**

### **1** Ensure Ollama and Gradio are installed:

```
pip install gradio
```

### **2** Run the script:

```
python app_web_basic.py
```

### **3** Open the generated link in a web browser.

### **4**

**Start chatting with the AI!**

---

## **Example Usage**

1. Open the web app.
  2. Type a question: "**What is the capital of France?**"
  3. Click "**Generate Response**".
  4. The response box shows: "**Paris**".
-