DeepScaleR Projects

Created by	R Riddhi Gupta
URL	https://github.com/riddhi-gupta-ai

▼ Project - 1 AI-Powered Math Solver with DeepScaleR

- Use DeepScaleR to build an AI-powered math solver that can handle complex math problems such as algebra, calculus, and linear algebra.

Step by Step Through:

▼ Step - 1 Install and Use DeepScaleR to Solve Complex Math Problems

1. Install Ollama : https://ollama.com/
2. Pull DeepScaleR Model :

```
ollama pull deepscaler
```

3. Run and Test DeepScaleR Model :

```
ollama run deepscaler
```

Now, It's Running you can ask anything with the model

- To Get Commands Help : Type `/?`
- To quit : Type `/bye`

▼ Step - 2 Implementing a Web-Based Math Assistant using Python and Gradio.

- Use Gradio : It is a Free Open source, Easy to use Python Library that helps you create user interfaces or UIs for machine learning models, APIs and Python Functions.

1. Create a Project Folder - DeepScaleR/Project1
2. Open it in terminal or VS Code terminal

```
pip install gradio
pip install ollama
```

3. Create an app.py file

```
import gradio as gr
import ollama

# Function to process user queries
def solve_math_problem(problem):
    response = ollama.chat(model='deepscaler', messages = [{'role'
    return response['message']


# Define Gradio Interface
interface = gr.Interface(
    fn = solve_math_problem,
    inputs = gr.Textbox(label="Enter a Math Problem"),
    outputs = gr.Textbox(label="Solution"),
    title = "AI-Powered Math Solver",
    description = "Ask any Math Problem, and Deepscaler will provid

# Launch the app
interface.launch()
```

4. Run the file

5. Go to the link
6. Test the App
7. Customize it accordingly

▼ Project 2 : AI Chatbot using DeepScaleR and Ollama

- Handle General Purpose Conversations, and we will deploy it as an API while ensuring low-latency responses.
- We have to create a Fine Tuned Chatbot model, By default DeepScaleR is optimized for mathematical Reasoning, To enhance its conversational ability, we will fine tune it with better chat style responses.

▼ Step 1 : Building a Chatbot for General-Purpose Conversation

1. Create a new Folder Deepscaler / Project2
2. Open in VS Code
3. Create a Modelfile

```
FROM deepscaler
SYSTEM "You are a helpful and friendly AI chatbot. Respond in a co
```

4. Open the Folder with terminal

```
ollama create deepscaler-chat -f Modelfile
```

```
ollama run deepscaler-chat
```

Now, you can work with it like a chatbot.

▼ Step 2 : Deploying DeepScaleR in an API using Fast API

1. Be in the same folder terminal

```
pip install fastapi uvicorn
```

2. Create a file - chatbot_api.py inside project2 folder

```
from fastapi import FastAPI
import ollama

# Initializing an app
app = FastAPI()

# Create the route endpoint
@app.get("/")
def home():
    return {"message": "Welcome to the DeepScaleR Chatbot API"}

# Create another endpoint for query
@app.get("/chat")
def chat(message:str):
    response = ollama.chat(model='deepscaler-chat', messages=[{'r
    return {'response':response['message']}
```

to run the above code for testing

```
uvicorn chatbot_api:app --host 0.0.0.0 --port 8000
```

- if you copy the url and say /chat?message="how are you"
- it will give you the data

▼ Step 3 : Create a Web Based Chatbot UI with Gradio

1. Create new file - chatbot_ui.py

```
import gradio as gr
import ollama

# Function to interact with the DeepscalerR Chatbot
def chat_with_bot(user_message):
    response = ollama.chat(model='deepscaler-chat', messages=[{'r
    return response['message']
```

```
# Define Gradio Chatbot Interface
chatbot_ui = gr.Interface(
    fn = chat_with_bot,
    inputs = gr.Textbox(label="Chat Message"),
    outputs = gr.Textbox(label="Bot Response"),
    title = "AI-Chatbot using DeepScaler",
    description = "Chat with an AI Chatbot powered by Deepscaler a
    )

if __name__ == '__main__':
    chatbot_ui.launch()
```

- Run this and Customize it accordingly.