**Lab Exercise - Deploy Phi Model in Docker**

**Objective**

Download the microsoft/phi-1\_5 model using Hugging Face Transformers, expose a FastAPI endpoint for text generation, and deploy the service in Docker.

**Project Structure**

phi-docker/

├── app/

│ └── main.py

├── requirements.txt

├── Dockerfile

**Step 1: FastAPI App (app/main.py)**

from fastapi import FastAPI

from pydantic import BaseModel

from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline

import torch

app = FastAPI()

tokenizer = AutoTokenizer.from\_pretrained("microsoft/phi-1\_5")

model = AutoModelForCausalLM.from\_pretrained("microsoft/phi-1\_5", torch\_dtype=torch.float32)

generator = pipeline("text-generation", model=model, tokenizer=tokenizer)

class PromptRequest(BaseModel):

prompt: str

@app.post("/generate/")

def generate\_text(req: PromptRequest):

output = generator(req.prompt, max\_length=100, do\_sample=True, temperature=0.7)

return {"generated\_text": output[0]["generated\_text"]}

**Step 2: requirements.txt**

fastapi

uvicorn

transformers

torch

**Step 3: Dockerfile**

FROM python:3.10-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY app/ ./app

EXPOSE 8000

CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]

**Step 4: Build the Docker Image**

docker build -t phi-model-api .

**Step 5: Run the Docker Container**

docker run -p 8000:8000 phi-model-api

**Step 6: Test the API**

You can use curl:

curl -X POST http://localhost:8000/generate/ \

-H "Content-Type: application/json" \

-d '{"prompt": "The future of artificial intelligence is"}'

Or use Python:

import requests

response = requests.post("http://localhost:8000/generate/", json={"prompt": "The future of AI is"})

print(response.json())

**Expected Output**

A short continuation of the input text from the Phi model.

**Notes**

* The phi-1\_5 model may require sufficient memory (~1.5–2 GB).
* If deploying on GPU or in production, consider using torch\_dtype=torch.bfloat16 and setting device\_map="auto".
* You can use model caching by mounting Hugging Face cache (~/.cache/huggingface).