**Setup instructions for both GPU and CPU environments**

**1. System Requirements**

**For GPU Setup**

* **NVIDIA GPU** with CUDA support
* **NVIDIA Driver** installed (CUDA >= 11.0)
* **Docker** with GPU support via nvidia-docker
* **CUDA Toolkit** installed on the host system
* **NVIDIA Container Toolkit** (to enable GPU support in Docker containers)

**For CPU Setup**

* **CPU-based system** (no special hardware requirements)
* **Docker** (CPU-only version)

**General Requirements (For both GPU and CPU):**

* **Docker** installed on your system
* **Git** installed to clone repositories
* Sufficient memory to download and run the models (~4GB for Stable Diffusion, ~2GB for SAM)

**2. Docker Setup**

**Step 1: Install Docker**

1. **Install Docker**:
   * **For GPU users**: Follow the [NVIDIA Docker Installation Guide](https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html).
   * **For CPU users**: Follow the standard Docker Installation Guide.
2. **Check Docker Installation**: Run the following command to verify Docker is installed correctly:

**CMD:**

*docker --version*

**Step 2: Install NVIDIA Docker Toolkit (For GPU)**

For GPU-enabled systems, install nvidia-docker to use the GPU in Docker containers:

**CMD:**

*sudo apt-get install -y nvidia-container-toolkit*

*sudo systemctl restart docker*

To test that Docker can access your GPU, run:

**CMD:**

*docker run --rm --gpus all nvidia/cuda:11.0-base nvidia-smi*

This should display information about your GPU and installed drivers.

**3. Clone the Project**

Clone the project repository:

bash

Copy code

git clone https://github.com/your-repo/text-to-image-app.git

cd text-to-image-app

The folder structure should look like this:

Copy code

deep\_edge/

├── client/

├── model/

├── server/

└── Dockerfile

**4. Configure the Dockerfile**

The provided Dockerfile is already designed to handle both GPU and CPU setups. Make sure the Dockerfile is configured as shown:

**Dockerfile for GPU/CPU:**

**Dockerfile**

# Use the base image for GPU/CPU support

FROM python:3.9-slim

# Set working directory inside the container

WORKDIR /app

# Copy project files

COPY ./client /app/client

COPY ./model /app/model

COPY ./server /app/server

# Install system dependencies

RUN apt-get update && apt-get install -y \

git \

libgl1-mesa-glx \

&& rm -rf /var/lib/apt/lists/\*

# Install PyTorch and other dependencies

RUN pip install --no-cache-dir \

flask \

torch \

torchvision \

diffusers \

opencv-python-headless \

pillow \

matplotlib \

transformers \

clip-by-openai \

segment-anything \

flask-cors

# Expose the port for Flask

EXPOSE 5000

# Command to run the Flask app

CMD ["python", "server/app.py"]

**5. Build the Docker Image**

**For GPU Setup (Use nvidia runtime):**

Build the Docker image:

**CMD:**

*docker build -t text-to-image-app-gpu .*

Run the Docker container with GPU support:

**CMD:**

docker run --gpus all -p 5000:5000 text-to-image-app-gpu

**For CPU Setup:**

Build the Docker image:

**CMD:**

*docker build -t text-to-image-app-cpu .*

Run the Docker container:

**CMD:**

docker run -p 5000:5000 text-to-image-app-cpu

**6. Access the Application**

Once the container is running, you can access the application from your web browser:

* URL: http://localhost:5000

You should see the **Text-to-Image Generator** interface where you can enter text prompts and generate images.

**7. Troubleshooting**

**For GPU Users:**

* Ensure that you have installed NVIDIA drivers and CUDA correctly.
* Verify the nvidia-docker installation by running:

**CMD:**

*docker run --rm --gpus all nvidia/cuda:11.0-base nvidia-smi*

**Common Issues:**

* **Out of memory error**: Make sure you have sufficient RAM and disk space. If running on a GPU, check your GPU memory usage.
* **Slow performance on CPU**: CPU environments will take longer to process models like Stable Diffusion. Consider using a GPU for faster results.

**8. Optional: Setup a Virtual Environment (Local Development)**

If you want to run the project locally without Docker (for development):

1. **Create a virtual environment**:

**CMD:**

python3 -m venv venv

source venv/bin/activate

1. **Install dependencies**:

**CMD:**

pip install -r requirements.txt

1. **Run the Flask app**:

**CMD:**

python server/app.py

1. Access the app at http://localhost:5000.

With these instructions, you can set up the project in both GPU and CPU environments using Docker!