LlamaFactory UI Approach Instructions

Step 1: Clone LLaMA Factory Repository

```
git clone https://github.com/hiyouga/LLaMA-Factory.git
cd LLaMA-Factory
```

Step 2: Install Requirements

```
pip install -r requirements.txt
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121
```

Step 3: Create Dataset

```
Path:
LLaMA-Factory/data/mydata.json

Example format:
[
  {
    "instruction": "Explain machine learning",
    "input": "",
    "output": "Machine learning is a method where computers learn from data."
  }
]
```

Step 4: Register Dataset

```
Open:
LLaMA-Factory/data/dataset_info.json

Add:
"mydata": {
    "file_name": "mydata.json",
    "columns": {
        "prompt": "instruction",
        "query": "input",
        "response": "output"
    }
}
```

Step 5: Launch WebUI

```
python src/webui.py

Open the link shown (usually http://127.0.0.1:7860)
```

Step 6: Select Model in UI

```
Example models:
```

```
TinyLlama/TinyLlama-1.1B-Chat-v1.0
or
google/gemma-2b-it
```

Step 7: Finetuning Settings

```
Finetuning type: LoRA
Quantization: 4-bit (QLoRA)
```

Step 8: Precision Settings

```
If BF16 not supported:
bf16 = False
fp16 = True
```

Step 9: Dataset Settings

```
Dataset name: mydata
Template: alpaca
Click: Preview Dataset
```

Step 10: Training Parameters

```
Batch size: 2
Gradient accumulation: 4
Learning rate: 2e-4
Max steps: 500
LoRA rank: 8
LoRA alpha: 16
Output directory: outputs/my-lora
```

Step 11: Start Training

```
Click Start Training.
Model adapter will be saved in outputs/my-lora
```

Step 12: Inference

```
Go to Chat tab.
Load base model + adapter.
Test your prompts.
```