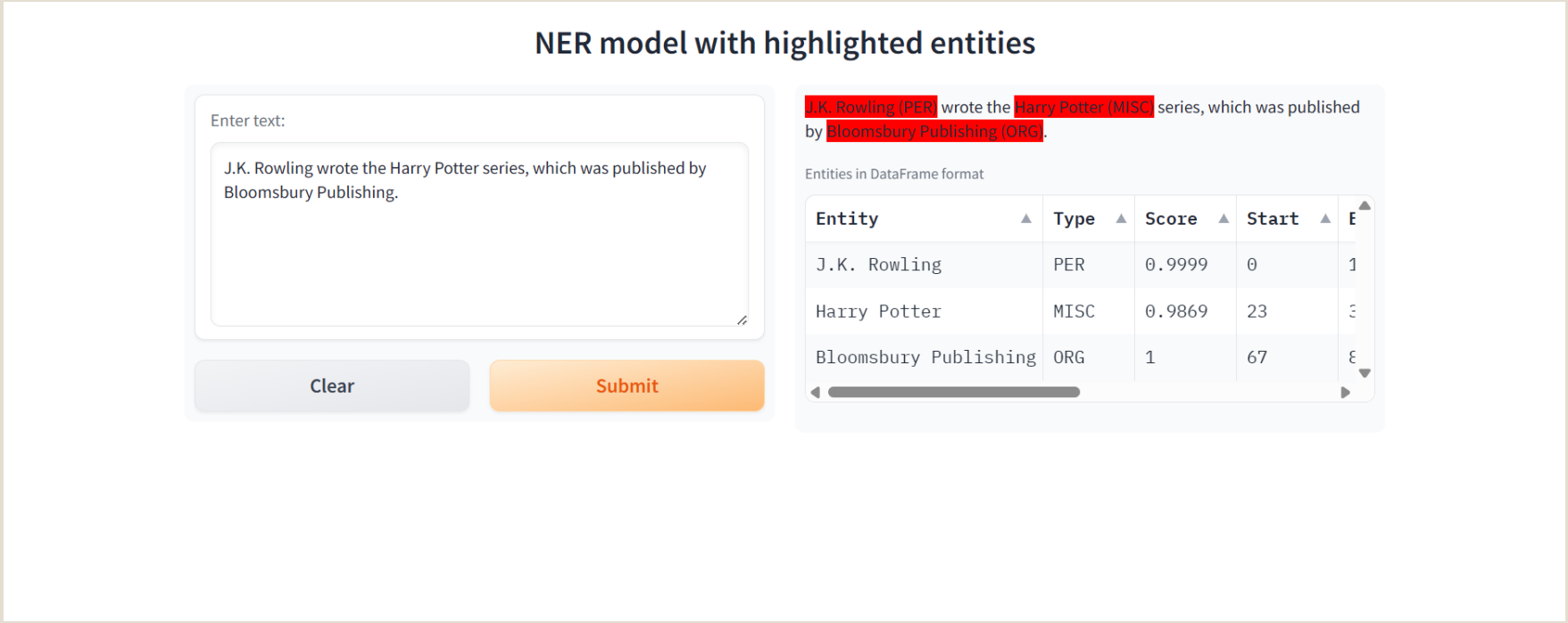
IMAGE GENERATION

Using Flux1 schnell with gradio locally

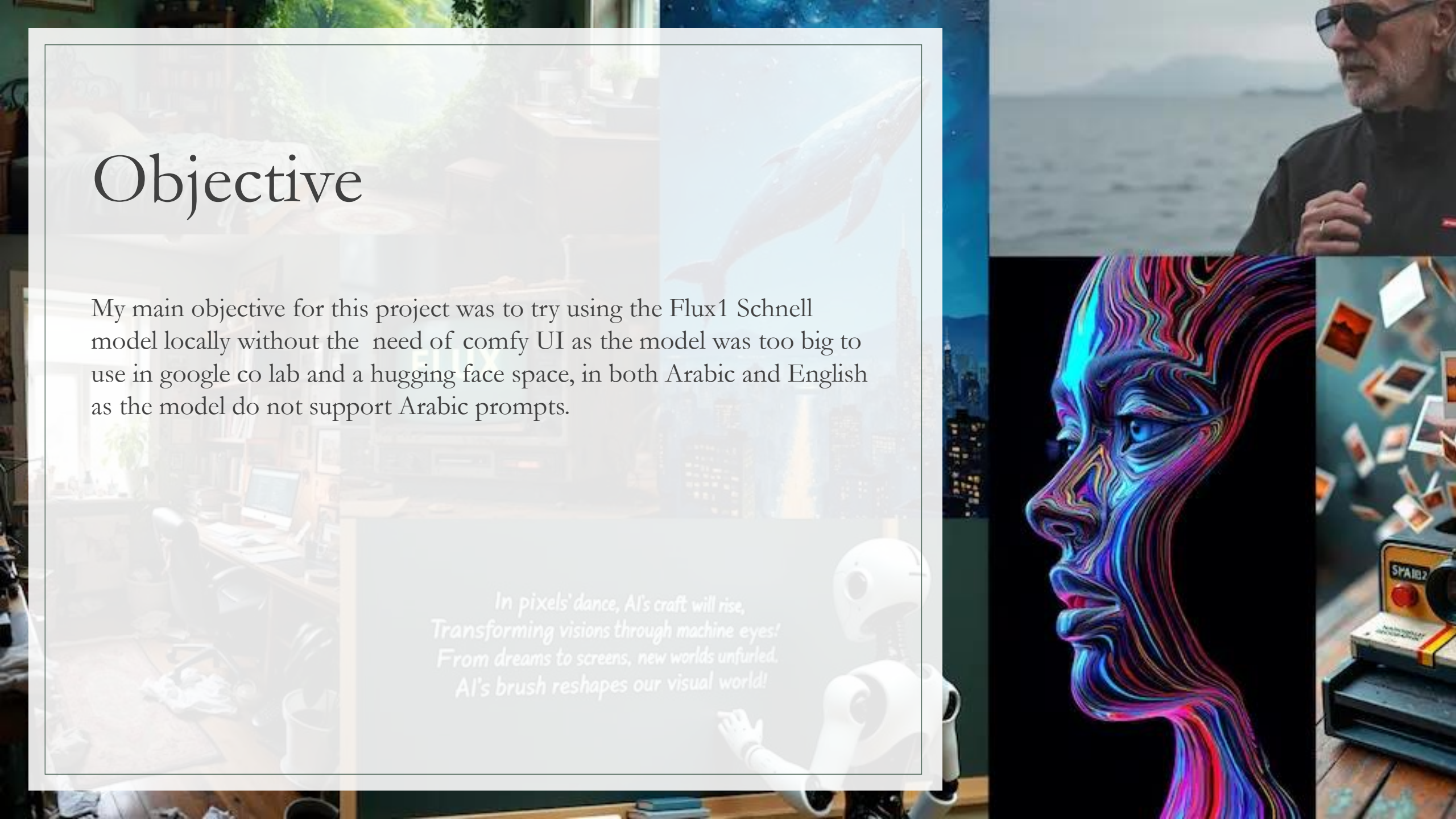
Previous project

A simple gradio interface with a NER model, the user will input a text with entities within it, and the outputs will be the same text with the entities highlighted by red, with a data frame that contain each sentence the entity was in, its type, score and more.



Objective

My main objective for this project was to try using the Flux1 Schnell model locally without the need of comfy UI as the model was too big to use in google co lab and a hugging face space, in both Arabic and English as the model do not support Arabic prompts.



What is flux.1 Schnell?

- Flux1 is the largest open-source image generation model developed by Black Forest Labs, the team who developed stable diffusion.
- There are three variations of this model, flux.1 dev which is the base model, flux.1 schnell which is a smaller version that operated faster, and flux.1 pro, a closed source version that can be only accessed through the API.
- Flux is highly regarded right now in the Ai image generation community, as it better than stable diffusion since it have enhanced image quality, advanced human anatomy and it is faster.

Code walkthrough

- 1- I downloaded the model from hugging face using the huggingface_hub library and snapshot_download

```
from huggingface_hub import snapshot_download
snapshot_download(repo_id="black-forest-labs/FLUX.1-schnell",local_dir="C:\\codes\\flux1schnell")
```

- 2- then I downloading the needed libraries in a virtual environment to run the model locally.

-pytorch, transformers, accelerate, diffusers, sentencepiece, protobuf.

- 3- Import the needed libraries and call the model.

```
import torch
from diffusers import FluxPipeline
import os
from transformers import pipeline
import gradio as gr
from datetime import datetime
import PIL
import deepl

flux1 = FluxPipeline.from_pretrained("C:\\codes\\flux1schnell",torch_dtype=torch.bfloat16)
flux1.enable_sequential_cpu_offload()
```

Code walkthrough

- 4- Define a function that use the Flux1.scnell model and return an image path.

```python
def img_gen(prompt):

    image = flux1( #defining the pipe
        prompt,
        height=512,
        width=1024,
        guidance_scale=1.3,
        output_type="pil", #pillow
        num_inference_steps=4,
        max_sequence_length=256,
        #generator=torch.Generator("cpu").manual_seed(0) #seed(0) means e
        generator=torch.Generator("cuda").manual_seed(0) #This means gpu
    ).images[0] #access the first image

    image_path = "C:\\codes\\flux1schnell\\gallery" #the folder which the
    if not os.path.exists(image_path):
        os.mkdir(image_path) #if the folder is not found then create the

    curr_datetime = datetime.now().strftime('%Y-%m-%d %H-%M-%S') #the cur
    image_save_path =f"{image_path}\\{curr_datetime}.jpeg"
    image.save(image_save_path) #saving the image to the specified folder

    return image_save_path #return the image path to pass to gradio
```

Code walkthrough

5- A small function to translate Arabic prompts to English

```
def arabic_to_english(prompt):
    prompt = str(tranlator.translate_text(prompt, target_lang="EN-US"))
    to_english = img_gen(prompt)
    return to_english
```

6- Examples for gradio in both English and Arabic

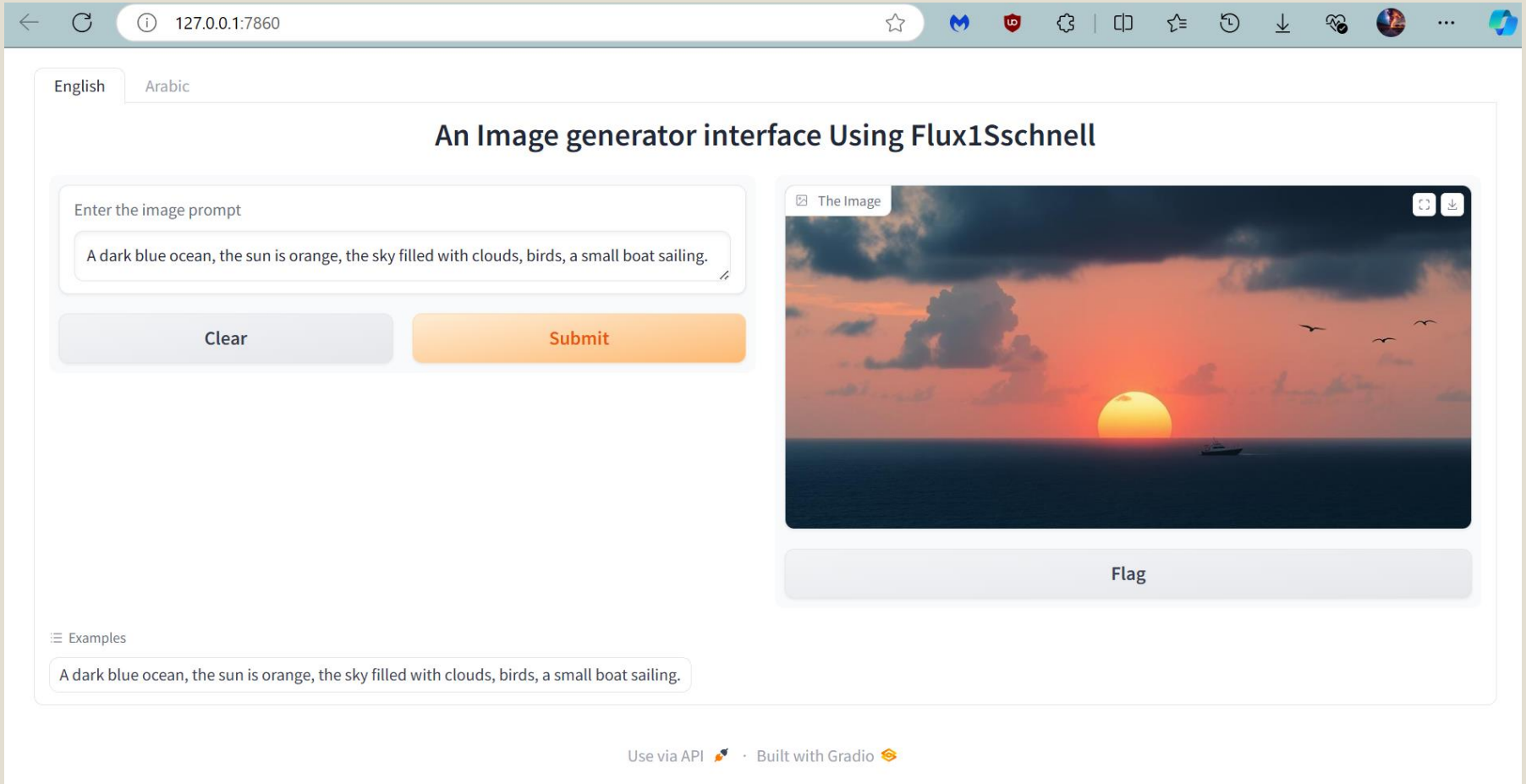
```
example1= ["A dark blue ocean, the sun is orange, the sky filled with clouds, birds, a small boat sailing."]
example2= ["حديقة مليئة بالزهور الملونة، السماء زرقاء والشمس منيرة، فراشات ملونة حول الزهور"]
```

Code walkthrough

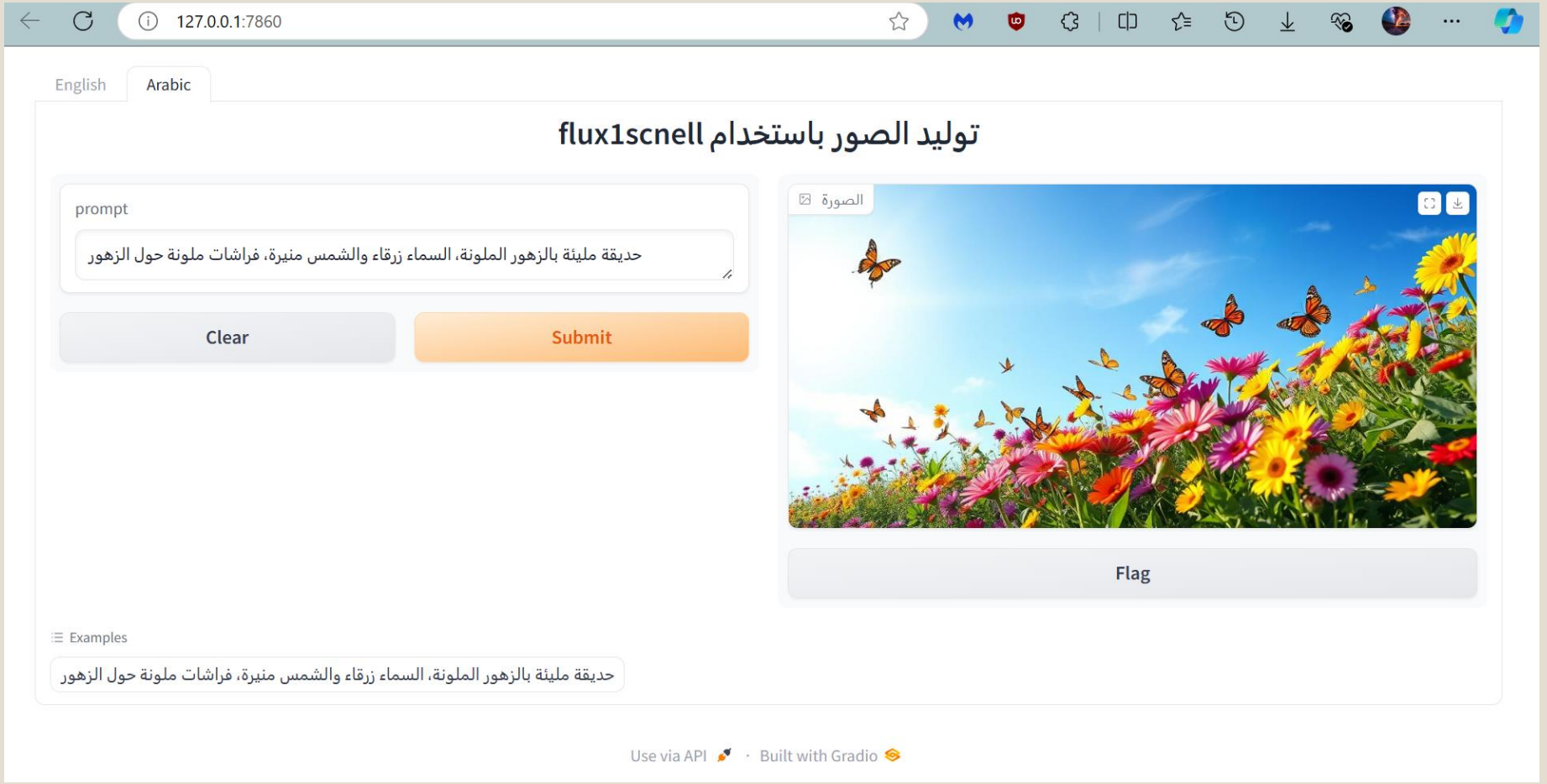
Finally, the gradio interfaces implementation

```
english_interface = gr.Interface(
    fn=img_gen,
    inputs=gr.Textbox(label= "Enter the image prompt"),
    outputs=gr.Image(label="The Image", type='filepath'),
    title = "An Image generator interface Using Flux1Sschnell",
    examples= example1
)

Arabic_interface = gr.Interface(
    fn = arabic_to_english,
    inputs = gr.Textbox("أدخل وصف الصورة"),
    outputs=gr.Image(label="الصورة", type='filepath'),
    title = "flux1scnell توليد الصور باستخدام ",
    examples= example2

)

gr.TabbedInterface(
    [english_interface, Arabic_interface],["English","Arabic"]
).launch()
```

Results


127.0.0.1:7860
English
Arabic
An Image generator interface Using Flux1Sschnell
Enter the image prompt
A dark blue ocean, the sun is orange, the sky filled with clouds, birds, a small boat sailing.
Clear
Submit
The Image
Flag
Examples
A dark blue ocean, the sun is orange, the sky filled with clouds, birds, a small boat sailing.
Use via API
Built with Gradio

Results


127.0.0.1:7860
English
Arabic
توليد الصور باستخدام flux1scnell
prompt
حديقة مليئة بالزهور الملونة، السماء زرقاء والشمس منيرة، فراشات ملونة حول الزهور
Clear
Submit
الصورة
Flag
Examples
حديقة مليئة بالزهور الملونة، السماء زرقاء والشمس منيرة، فراشات ملونة حول الزهور
Use via API
Built with Gradio

Measures done to include Arabic

For Arabic, since the model does not work with Arabic, I decided to use DeepL translation API to translate Arabic prompts to English, and the only reason I did not use a hugging face model for this is me having an API key before handed.

```
tranlator = deepl.Translator("c8[redacted]x")
```

Putting Arabic prompt without translating it will result in this:

Guide used

Installing Flux.1 Schnell without comfy UI:

Link to GitHub repo


Thank
you!