```
Start coding or generate with AI.


!pip install flask pyngrok diffusers transformers accelerate --quiet

from flask import Flask, render_template_string, request
from pyngrok import ngrok
from diffusers import StableDiffusionPipeline
import torch, os, uuid, random

# HTML Template
HTML_TEMPLATE = """
<!DOCTYPE html>
<html>
<head>
<title>Text to Image</title>
<style>
body {
    background: linear-gradient(135deg, #1f1c2c, #928dab);
    color: white; text-align: center;
    font-family: Arial, sans-serif;
}
h1, h2.glow {
    animation: glowAnim 1s ease-in-out infinite alternate;
}
input {
    padding: 10px; width: 60%; font-size: 16px; border-radius: 5px; border: none;
}
button {
    padding: 10px 20px; font-size: 16px;
    background: linear-gradient(90deg, #00c6ff, #0072ff);
    border: none; cursor: pointer; color: white; border-radius: 5px;
}
img {
    margin-top: 20px; max-width: 256px; margin-right: 10px;
    border: 3px solid white;
    opacity: 0; transition: opacity 1.2s ease-in-out;
}
.button {
    padding: 8px 16px;
    font-size: 14px;
    background: #0072ff;
    border: none;
    color: #fff;
    border-radius: 25px;
    cursor: pointer;
    transition: 0.3s ease;
}
button:hover {
    background: #0056cc;
    transform: scale(1.05);
}

#loading {
    display: none; margin-top: 20px; font-size: 18px; font-weight: bold;
}
.loader-bar {
    width: 60%; height: 12px; background: #444; border-radius: 6px;
    margin: 10px auto; overflow: hidden;
}
.loader-bar-fill {
    width: 0%; height: 100%; background: linear-gradient(90deg, #00c6ff, #0072ff);
    animation: loadAnim 3s linear infinite;
}
@keyframes loadAnim {
    0% { width: 0%; }
    50% { width: 100%; }
    100% { width: 0%; }
}
@keyframes glowAnim {
    from { text-shadow: 0 0 5px #ff0000; }  /* Red Glow */
    to { text-shadow: 0 0 20px #ff0000; }   /* Red Glow Stronger */
}
.tip-box {
    margin-top: 15px; font-size: 14px; color: #fffb;
    font-style: italic;
}
</style>
<script>
let tips = [
    "🎨 Did you know? The first camera took 8 hours to capture a photo!",
    "✨ Pro Tip: Imagination has no limits!",
    "🖌️ Fun Fact: Van Gogh only sold 1 painting while alive.",
    "⚡ Creativity is intelligence having fun.",
    "🌌 Every pixel here is born from AI magic."
];
let tipIndex = 0;

function showLoading() {
    document.getElementById("loading").style.display = "block";
    changeTips();
}

function changeTips() {
    document.getElementById("tipText").innerText = tips[tipIndex];
    tipIndex = (tipIndex + 1) % tips.length;
    setTimeout(changeTips, 2000);
}
</script>
```

```
</head>
<body>
<h1 class="glow">🖼️ AI Image Creator</h1>
<form method="POST" onsubmit="showLoading()">
    <input name="prompt" placeholder="Enter prompt..." required>
    <button type="submit">Generate</button>
</form>
<div id="loading">
    <div id="typingText">✨ Creating your masterpiece...</div>
    <div class="loader-bar"><div class="loader-bar-fill"></div></div>
    <div class="tip-box" id="tipText"></div>
</div>
{% if prompt_text %}
    <h2>Prompt: "{{ prompt_text }}"</h2>
{% endif %}
{% if image_urls %}
    <h2 class="glow">🤗 Generated Images:</h2>
    {% for url in image_urls %}
        <div style="display:inline-block;">
            <img src="{{ url }}" onload="this.style.opacity=1">
            <a href="{{ url }}" download="image_{{ loop.index }}.png" class="download-btn">Download {{ loop.index }}</a>
        </div>
    {% endfor %}
{% endif %}
</body>
</html>
"""

# Flask app
app = Flask(__name__)

# Load model
model_id = "dreamlike-art/dreamlike-diffusion-1.0"
pipe = StableDiffusionPipeline.from_pretrained(
    model_id, torch_dtype=torch.float16, use_safetensors=True
).to("cuda")

pipe.enable_attention_slicing()

@app.route("/", methods=["GET", "POST"])
def index():
    image_urls = []
    prompt_text = None
    if request.method == "POST":
        prompt = request.form.get("prompt")
        if prompt:
            prompt_text = prompt
            with torch.inference_mode():
                images = pipe(
                    prompt,
                    num_inference_steps=20,
                    height=448,
                    width=448,
                    num_images_per_prompt=3
                ).images
            os.makedirs("static", exist_ok=True)
            for img in images:
                filename = f"{uuid.uuid4().hex}.png"
                filepath = os.path.join("static", filename)
                img.save(filepath)
                image_urls.append("/" + filepath)
    return render_template_string(HTML_TEMPLATE, image_urls=image_urls, prompt_text=prompt_text)

# Ngrok setup
ngrok.set_auth_token("PLEASE ADD NGROK TOKEN")
public_url = ngrok.connect(5000)
print("Public URL:", public_url)

app.run(port=5000)
```

Start coding or generate with AI.

Start coding or generate with AI.