1. **Generate a short creative story based on a given theme using a pre-trained GPT-2 model.**

Solution:

```
!pip install transformers

from transformers import pipeline

generator = pipeline("text-generation", model="gpt2")

prompt = "A brave astronaut lands on Mars and discovers"

result = generator(prompt, max_length=60, num_return_sequences=1)

print("Generated Story:\n")

print(result[0]['generated_text'])
```

2. **Generate an image based on a descriptive text prompt using a pre-trained Stable Diffusion model.**

```
!pip install diffusers transformers accelerate safetensors torch

from diffusers import StableDiffusionPipeline
import torch

pipe = StableDiffusionPipeline.from_pretrained(
    "runwayml/stable-diffusion-v1-5",
    torch_dtype=torch.float16
).to("cuda")

prompt = "A beautiful sunset over a calm lake with mountains in the background"

image = pipe(prompt).images[0]

image.show()

image.save("sunset_image.png")
```

3. **Generate Creativity with AI: Stories and Visuals from a Single Prompt**

```
!pip install transformers diffusers accelerate safetensors torch

from transformers import pipeline
from diffusers import StableDiffusionPipeline
from IPython.display import display
import torch
```

```
prompt = "A futuristic city with flying cars and glowing skyscrapers"

story_generator = pipeline("text-generation", model="gpt2", no_repeat_ngram_size=3)
story = story_generator(f"Write a short story about: {prompt}", max_length=120,
num_return_sequences=1)[0]['generated_text']

print("📖 Generated Story:\n")
print(story)

pipe = StableDiffusionPipeline.from_pretrained(
    "runwayml/stable-diffusion-v1-5",
    torch_dtype=torch.float16
).to("cuda")

image = pipe(prompt).images[0]
display(image)
image.save("generated_future_city.png")

print("\n mage saved as 'generated_future_city.png'")
```

4. Generate a Story from an Image Using AI

```
!pip install transformers torch pillow

from transformers import BlipProcessor, BlipForConditionalGeneration, pipeline

from PIL import Image

from google.colab import files



print("Upload an image (jpg/png):")

uploaded = files.upload()

image_path = list(uploaded.keys())[0]

image = Image.open(image_path)

processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")

model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-
base")

inputs = processor(image, return_tensors="pt")
```

```
caption_ids = model.generate(**inputs)

caption = processor.decode(caption_ids[0], skip_special_tokens=True)

print("\n Image Caption:", caption)

story_generator = pipeline("text-generation", model="gpt2")

story_prompt = f"Write a short story about this: {caption}"

story = story_generator(story_prompt, max_length=80,
num_return_sequences=1)[0]['generated_text']

print("\n Generated Story:\n", story)
```

5. Generate a descriptive caption for a given image to explain what is happening in it.

```
!pip install transformers torch pillow

from transformers import BlipProcessor, BlipForConditionalGeneration

from PIL import Image

from google.colab import files

print("Upload an image (jpg/png):")

uploaded = files.upload()

image_path = list(uploaded.keys())[0]

image = Image.open(image_path)


processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")

model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-
base")


inputs = processor(image, return_tensors="pt")

caption_ids = model.generate(**inputs)

caption = processor.decode(caption_ids[0], skip_special_tokens=True)

print("\n Image Caption:")
```

```
print(caption)
```