1. Text-Based Short Video Generation Using AI

```
!pip install -q diffusers transformers accelerate torch torchvision safetensors imageio imageio[ffmpeg]


from diffusers import DiffusionPipeline
import torch
from IPython.display import Video, display
import numpy as np
import imageio

model_id = "damo-vilab/text-to-video-ms-1.7b"
pipe = DiffusionPipeline.from_pretrained(model_id, torch_dtype=torch.float16, variant="fp16")
pipe = pipe.to("cuda")  # Use GPU


prompt = "A serene landscape with a flowing river and birds flying overhead."
result = pipe(prompt, num_frames=16)
frames = result.frames[0]
video_path = "/content/generated_video.mp4"
imageio.mimsave(video_path, [np.array(f) for f in frames], fps=8)

print("Video generated and saved at:", video_path)
display(Video(video_path, embed=True, width=560))
```

1. Image-Based Short Video Generation Using AI

```
# STEP 1: Install required libraries
!pip install -q diffusers transformers accelerate torch torchvision safetensors imageio imageio[ffmpeg]

# STEP 2: Import dependencies
from diffusers import StableVideoDiffusionPipeline
import torch
from PIL import Image
import numpy as np
import imageio
from google.colab import files  # for uploading files

# STEP 3: Upload your image
```

```
print("Please upload an image file (jpg/png).")
uploaded = files.upload()
image_path = list(uploaded.keys())[0]  # get uploaded file name
image = Image.open(image_path).convert("RGB")
image = image.resize((512, 512))  # resize for model

# STEP 4: Load the video diffusion model
model_id = "stabilityai/stable-video-diffusion-img2vid-xt"
pipe = StableVideoDiffusionPipeline.from_pretrained(model_id, torch_dtype=torch.float16,
variant="fp16")
pipe = pipe.to("cuda")  # use GPU

# STEP 5: Generate a short video
result = pipe(image, num_frames=6)  # 6 frames for low memory

frames = result.frames[0]

# STEP 6: Save and display video
video_path = "/content/generated_video.mp4"
imageio.mimsave(video_path, [np.array(f) for f in frames], fps=8)

print("Video generated and saved at:", video_path)
from IPython.display import Video, display
display(Video(video_path, embed=True, width=560))
```