```
print("E23CSEU2320")
print("LAB1")
```

```
E23CSEU2320
LAB1
```

```
!pip install diffusers
```

```
Requirement already satisfied: diffusers in /usr/local/lib/python3.12/dist-packages (0.36.0)
Requirement already satisfied: importlib_metadata in /usr/local/lib/python3.12/dist-packages (from diffusers) (8.7
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from diffusers) (3.20.3)
Requirement already satisfied: httpx<1.0.0 in /usr/local/lib/python3.12/dist-packages (from diffusers) (0.28.1)
Requirement already satisfied: huggingface-hub<2.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from diffu
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from diffusers) (2.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from diffusers) (2025
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from diffusers) (2.32.4)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.12/dist-packages (from diffusers) (0.7
Requirement already satisfied: Pillow in /usr/local/lib/python3.12/dist-packages (from diffusers) (11.3.0)
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (4.1
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (2
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffuse
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->diffusers) (3.11
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1.0
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>=
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from hugging
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-h
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.12/dist-packages (from importlib_metadata->dif
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests-
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->diffu
```

```
from diffusers import StableDiffusionPipeline
import torch
import os

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

# Create output folder
os.makedirs("synthetic_dataset", exist_ok=True)

prompts = [
    "A futuristic city at sunset, cyberpunk style",
    "A cute robot studying in a university classroom",
    "A realistic chest X-ray image of healthy lungs",
    "An Indian street food stall at night, cinematic lighting",
    "A peaceful mountain landscape with snow and pine trees"
]

# Generate images
for i, prompt in enumerate(prompts):
    image = pipe(prompt).images[0]
    image.save(f"synthetic_dataset/image_{i+1}.png")
```

```
Flax classes are deprecated and will be removed in Diffusers v1.0.0. We recommend migrating to PyTorch classes or
Flax classes are deprecated and will be removed in Diffusers v1.0.0. We recommend migrating to PyTorch classes or
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/to
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
model_index.json: 100%                                         541/541 [00:00<00:00, 65.1kB/s]

Fetching 15 files: 100%                                        15/15 [00:44<00:00,  2.88s/it]

preprocessor_config.json: 100%                                      342/342 [00:00<00:00, 19.4kB/s]

config.json: 100%                                          617/617 [00:00<00:00, 8.00kB/s]

scheduler_config.json: 100%                                       308/308 [00:00<00:00, 5.49kB/s]

config.json:       4.72k/? [00:00<00:00, 146kB/s]

special_tokens_map.json: 100%                                      472/472 [00:00<00:00, 10.7kB/s]

merges.txt:       525k/? [00:00<00:00, 11.8MB/s]

text_encoder/model.safetensors: 100%                                    492M/492M [00:36<00:00, 14.9MB/s]

safety_checker/model.safetensors: 100%                                   1.22G/1.22G [00:27<00:00, 44.1MB/s]

tokenizer_config.json: 100%                                       806/806 [00:00<00:00, 91.3kB/s]

vocab.json:       1.06M/? [00:00<00:00, 20.8MB/s]

config.json: 100%                                          547/547 [00:00<00:00, 27.4kB/s]

config.json: 100%                                          743/743 [00:00<00:00, 43.8kB/s]

unet/diffusion_pytorch_model.safetensors: 100%                              3.44G/3.44G [00:43<00:00, 224MB/s]

vae/diffusion_pytorch_model.safetensors: 100%                               335M/335M [00:39<00:00, 7.83MB/s]

Loading pipeline components...: 100%                                   7/7 [00:23<00:00,  3.23s/it]
`torch_dtype` is deprecated! Use `dtype` instead!
100%                                     50/50 [00:09<00:00,  6.91it/s]

100%                                     50/50 [00:07<00:00,  6.79it/s]

100%                                     50/50 [00:07<00:00,  6.65it/s]

100%                                     50/50 [00:07<00:00,  6.54it/s]

100%                                     50/50 [00:07<00:00,  6.32it/s]
```

```python
import os
from PIL import Image
import matplotlib.pyplot as plt

dataset_path = "synthetic_dataset"

images = sorted(os.listdir(dataset_path))

for img_name in images:
    img_path = os.path.join(dataset_path, img_name)
    img = Image.open(img_path)

    plt.imshow(img)
    plt.axis("off")
    plt.title(img_name)
    plt.show()
```

image_1.png

image_2.png

```
from diffusers import StableDiffusionPipeline
import torch
import os

model_id = "runwayml/stable-diffusion-v1-5"

pipe = StableDiffusionPipeline.from_pretrained(
    model_id,
    torch_dtype=torch.float16
).to("cuda")

pipe.enable_attention_slicing()  # reduces GPU memory usage

base_dir = "synthetic_dataset"

classes = {
    "pneumonia": "A realistic chest X-ray image showing pneumonia infection, medical radiology, high
    "normal": "A realistic chest X-ray image of healthy lungs, clear lungs, medical radiology",
    "covid": "A realistic chest X-ray image showing COVID-19 lung infection, ground glass opacity, m
}

images_per_class = 20


for class_name in classes.keys():
    os.makedirs(os.path.join(base_dir, class_name), exist_ok=True)
```

```
for class_name, prompt in classes.items():
    print(f"Generating images for: {class_name}")

    class_path = os.path.join(base_dir, class_name)

    for i in range(images_per_class):
        image = pipe(
            prompt,
            guidance_scale=7.5,
            num_inference_steps=50
        ).images[0]

        image_name = f"{class_name}_{i+1}.png"
        image.save(os.path.join(class_path, image_name))

        print(f"Saved: {image_name}")

print(" Medical synthetic dataset generation completed.")
```