In [2]:
```
# Required installations
!pip install --upgrade diffusers transformers -q
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.7/43.7 kB 2.4 MB/s eta
0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.6/2.6 MB 40.4 MB/s eta 0:
00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.5/9.5 MB 69.2 MB/s eta 0:
00:00
```

This cell installs the necessary Python libraries needed for the project, specifically the diffusers and transformers libraries. These libraries are essential for loading and working with the pre-trained models used in this project, such as Stable Diffusion v2 for image generation and GPT-2 for text generation.

In [3]:
```
# Necessary imports
from pathlib import Path
import torch
import pandas as pd
import numpy as np
from diffusers import StableDiffusionPipeline
from transformers import pipeline, set_seed
import matplotlib.pyplot as plt
import cv2
import os
import logging
```

```
The cache for model files in Transformers v4.22.0 has been updated. Mig
rating your old cache. This is a one-time only operation. You can inter
rupt this and resume the migration later on by calling `transformers.ut
ils.move_cache()`.

0it [00:00, ?it/s]
```

Here, we import the various Python libraries that are crucial for the project's functionality. These include standard libraries like os and logging for managing files and logging information, as well as more specialized libraries like torch for deep learning, diffusers for handling the image generation model, and transformers for the text generation model. Additionally, matplotlib and cv2 are imported for image processing and display purposes.

In [4]:
```
# Configuration settings
class Config:
    device_type = "cuda"
    random_seed = 42
    torch_gen = torch.Generator(device_type).manual_seed(random_seed)
    num_steps = 35
    model_id = "stabilityai/stable-diffusion-2"
    output_image_size = (400, 400)
    guidance_factor = 9
    text_model_id = "gpt2"
    dataset_size = 6
    max_text_length = 12
    output_dir = "generated_images"
    log_file = "generation.log"
```

This cell defines the Config class, which contains all the configuration settings for the project. These settings include the device type (CUDA for GPU usage), random seed for reproducibility, model IDs, and various parameters that control the image generation process, such as the number of steps, output image size, and guidance scale. These configurations are crucial for ensuring consistency and reproducibility throughout the project.

In [5]:
```
# Setup logging
logging.basicConfig(filename=Config.log_file, level=logging.INFO, format
```

This cell initializes the logging mechanism, which is used to record key events during the execution of the project. The log file (generation.log) will store messages related to the image generation process, such as the prompts used and the paths to the generated images. This is particularly useful for debugging and reviewing the output.

In [6]:
```
# Load the image generation model
diffusion_pipeline = StableDiffusionPipeline.from_pretrained(
    Config.model_id, torch_dtype=torch.float16,
    revision="fp16", use_auth_token='hf_wfduPRhyWmvNoYAYZPaSgCWqtozjoIoZ
    guidance_scale=Config.guidance_factor
)
diffusion_pipeline = diffusion_pipeline.to(Config.device_type)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.p
y:89: UserWarning: 
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your setti
ngs tab (https://huggingface.co/settings/tokens), set it as secret in y
our Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to ac
cess public models or datasets.
  warnings.warn(

model_index.json:   0%|          | 0.00/511 [00:00<?, ?B/s]

/usr/local/lib/python3.10/dist-packages/diffusers/pipelines/pipeline_lo
ading_utils.py:219: FutureWarning: You are loading the variant fp16 fro
m stabilityai/stable-diffusion-2 via `revision='fp16'` even though you
can load it via `variant=`fp16`. Loading model variants via `revision
='fp16'` is deprecated and will be removed in diffusers v1. Please use
`variant='fp16'` instead.
  warnings.warn(
vae/diffusion_pytorch_model.safetensors not found

Fetching 12 files:   0%|          | 0/12 [00:00<?, ?it/s]

pytorch_model.bin:   0%|          | 0.00/681M [00:00<?, ?B/s]

text_encoder/config.json:   0%|          | 0.00/624 [00:00<?, ?B/s]

unet/config.json:   0%|          | 0.00/900 [00:00<?, ?B/s]

tokenizer/special_tokens_map.json:   0%|          | 0.00/460 [00:00<?,
?B/s]

tokenizer/tokenizer_config.json:   0%|          | 0.00/815 [00:00<?, ?
B/s]

tokenizer/merges.txt:   0%|          | 0.00/525k [00:00<?, ?B/s]

tokenizer/vocab.json:   0%|          | 0.00/1.06M [00:00<?, ?B/s]

diffusion_pytorch_model.bin:   0%|          | 0.00/1.73G [00:00<?, ?B/
s]

diffusion_pytorch_model.bin:   0%|          | 0.00/167M [00:00<?, ?B/s]

vae/config.json:   0%|          | 0.00/602 [00:00<?, ?B/s]

scheduler/scheduler_config.json:   0%|          | 0.00/345 [00:00<?, ?
B/s]
```

```
Keyword arguments {'use_auth_token': 'hf_wfduPRhyWmvNoYAYZPaSgCWqtozjoI
oZnU', 'guidance_scale': 9} are not expected by StableDiffusionPipeline
and will be ignored.

Loading pipeline components...:   0%|          | 0/5 [00:00<?, ?it/s]

/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils
_base.py:1601: FutureWarning: `clean_up_tokenization_spaces` was not se
t. It will be set to `True` by default. This behavior will be depracted
in transformers v4.45, and will be then set to `False` by default. For
more details check this issue: https://github.com/huggingface/transform
ers/issues/31884 (https://github.com/huggingface/transformers/issues/31
884)
  warnings.warn(
An error occurred while trying to fetch /root/.cache/huggingface/hub/mo
dels--stabilityai--stable-diffusion-2/snapshots/d75b612d366d802b1753960
de862a9270c8d55f1/vae: Error no file named diffusion_pytorch_model.safe
tensors found in directory /root/.cache/huggingface/hub/models--stabili
tyai--stable-diffusion-2/snapshots/d75b612d366d802b1753960de862a9270c8d
55f1/vae.
Defaulting to unsafe serialization. Pass `allow_pickle=False` to raise
an error instead.
An error occurred while trying to fetch /root/.cache/huggingface/hub/mo
dels--stabilityai--stable-diffusion-2/snapshots/d75b612d366d802b1753960
de862a9270c8d55f1/unet: Error no file named diffusion_pytorch_model.saf
etensors found in directory /root/.cache/huggingface/hub/models--stabil
ityai--stable-diffusion-2/snapshots/d75b612d366d802b1753960de862a9270c8
d55f1/unet.
Defaulting to unsafe serialization. Pass `allow_pickle=False` to raise
an error instead.
```

In this cell, the Stable Diffusion v2 model is loaded using the StableDiffusionPipeline from the diffusers library. The model is downloaded and configured to use half-precision floating-point (fp16) for efficiency, and it is moved to the GPU for faster processing. This step prepares the model to generate images based on text prompts.

```
In [ ]:
```

In [7]:
```
# Load the text generation model for prompt generation
prompt_generator = pipeline("text-generation", model=Config.text_model_i
set_seed(Config.random_seed)
```

```
config.json:   0%|          | 0.00/665 [00:00<?, ?B/s]

model.safetensors:   0%|          | 0.00/548M [00:00<?, ?B/s]

generation_config.json:   0%|          | 0.00/124 [00:00<?, ?B/s]

tokenizer_config.json:   0%|          | 0.00/26.0 [00:00<?, ?B/s]

vocab.json:   0%|          | 0.00/1.04M [00:00<?, ?B/s]

merges.txt:   0%|          | 0.00/456k [00:00<?, ?B/s]

tokenizer.json:   0%|          | 0.00/1.36M [00:00<?, ?B/s]

Hardware accelerator e.g. GPU is available in the environment, but no `
device` argument is passed to the `Pipeline` object. Model will be on C
PU.
```

This cell initializes the GPT-2 model, which is used to generate text prompts for the image generation process. The pipeline function from the transformers library is used to set up the text generation model. The random seed is set for consistency, ensuring that the same text prompts can be generated across different runs.

In [14]:
```
# Function to generate an image based on a text prompt
def create_image_from_text(text_prompt, pipeline_model):
    result_image = pipeline_model(
        text_prompt, num_inference_steps=Config.num_steps,
        generator=Config.torch_gen,
        guidance_scale=Config.guidance_factor
    ).images[0]

    resized_image = result_image.resize(Config.output_image_size)
    return resized_image
```

In [15]:
```
# Function to generate prompts using GPT-2
def generate_prompts(prompt_seed, num_prompts):
    prompts = prompt_generator(
        prompt_seed,
        max_length=Config.max_text_length,
        num_return_sequences=num_prompts
    )
    return [p['generated_text'] for p in prompts]
```

In [17]:
```
# Function to generate and save multiple images
def generate_and_save_images(prompts, model, save_dir):
    if not os.path.exists(save_dir):
        os.makedirs(save_dir)

    for idx, prompt in enumerate(prompts):
        image = create_image_from_text(prompt, model)
        image_path = os.path.join(save_dir, f"image_{idx+1}.png")
        image.save(image_path)
        logging.info(f"Generated image for prompt: '{prompt}' and saved
```

In [18]:
```
# Function to display a grid of generated images
def display_images(image_dir, num_images_per_row=3):
    images = [cv2.imread(os.path.join(image_dir, img)) for img in os.lis
    num_images = len(images)
    num_rows = (num_images // num_images_per_row) + int(num_images % num

    plt.figure(figsize=(15, 5 * num_rows))
    for i in range(num_images):
        plt.subplot(num_rows, num_images_per_row, i + 1)
        plt.imshow(cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB))
        plt.axis('off')
    plt.show()
```

Here, we define several key functions:

create_image_from_text: Generates an image from a given text prompt using the Stable Diffusion model.

generate_prompts: Generates multiple text prompts using the GPT-2 model.

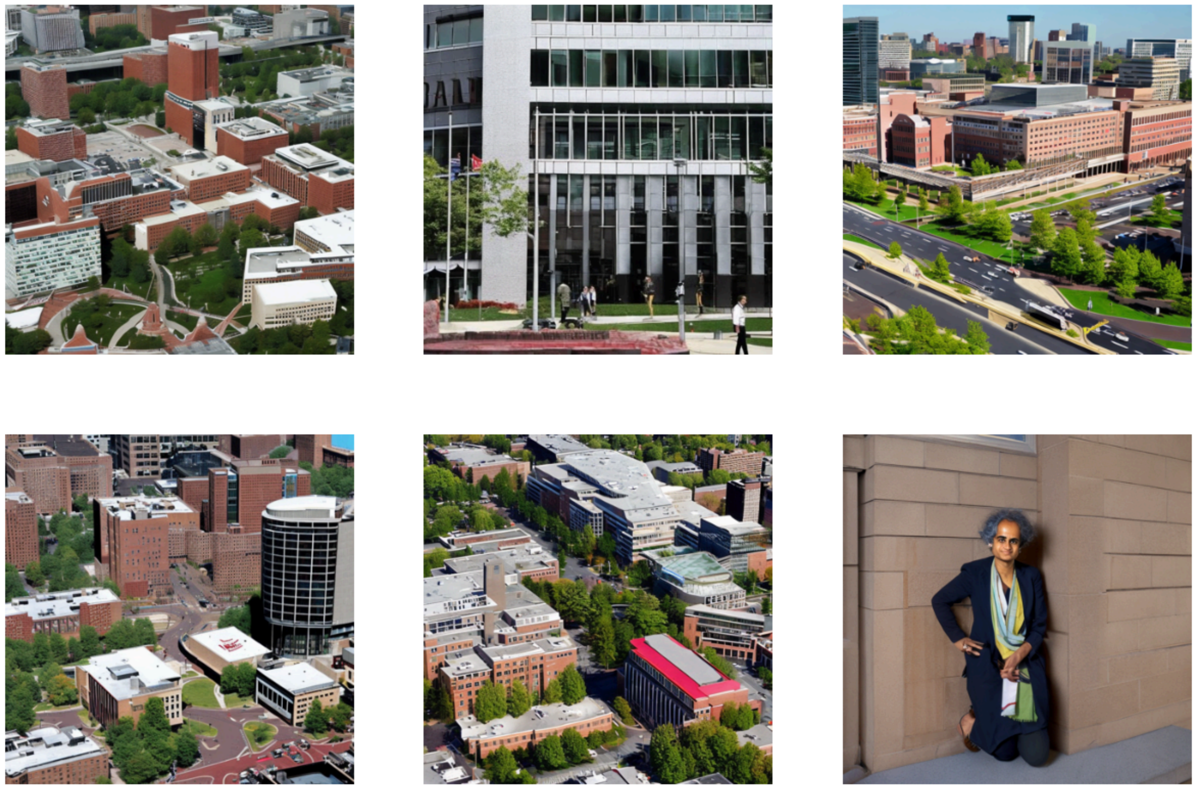
generate_and_save_images: Generates and saves images based on the generated prompts.

display_images: Displays the generated images in a grid format.

These functions are the core of the project, handling the generation, saving, and visualization of images.

In [20]:
```python
# Main function to generate prompts, create images, and display them
def main():
    # Step 1: Generate prompts
    initial_prompt = "Northeastern University"
    prompts = generate_prompts(initial_prompt, Config.dataset_size)

    # Step 2: Generate and save images
    generate_and_save_images(prompts, diffusion_pipeline, Config.output_

    # Step 3: Display generated images
    display_images(Config.output_dir)

# Execute the main function
main()
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

  0%|          | 0/35 [00:00<?, ?it/s]

  0%|          | 0/35 [00:00<?, ?it/s]

  0%|          | 0/35 [00:00<?, ?it/s]

  0%|          | 0/35 [00:00<?, ?it/s]

  0%|          | 0/35 [00:00<?, ?it/s]

  0%|          | 0/35 [00:00<?, ?it/s]
```



This cell contains the main function, which orchestrates the entire image generation pipeline. It first generates text prompts based on an initial seed, then creates and saves the corresponding images, and finally displays the results in a grid format. Executing this function runs the entire workflow, from text prompt generation to image visualization.

These output cells display the images generated by the Stable Diffusion model based on the prompts provided by GPT-2. Each image represents a different interpretation of the prompt "Northeastern University," showcasing the diversity and creativity of the model's outputs. The images are displayed in a grid format, making it easy to compare the different outputs side by side.

```
In [ ]:
```