

How to Run Guide: Multi-Pass Lesson & Slide Generator

September 4, 2025

1 Prerequisites

- Linux/macOS/Windows with conda (Anaconda/Miniconda/Mamba).
- Python 3.11 (we'll create a fresh environment).
- An OpenAI API key with access to the models you configured.

2 Create and Activate a Conda Environment

Linux/macOS

```
1 # Create a clean environment
2 conda create -n edu-slides python=3.11 -y
3 conda activate edu-slides
```

Windows (PowerShell)

```
1 # Create a clean environment
2 conda create -n edu-slides python=3.11 -y
3 conda activate edu-slides
```

3 Install Python Dependencies

Install via **pip**

```
1 pip install --upgrade pip
2 pip install openai nest_asyncio
3 python -m pip install -U matplotlib
4 pip install numpy
```

4 Project Layout

Place files like this (reflecting the notebook workflow and generated outputs):

```

1 project_root/
2     final.ipynb                # main notebook you run
3     gen_images_from_slides.py  # provides _run_async(...)
4     viz_outputs/              # auto-created output root
5         explain_me_photosynthesis/ # one subfolder per input task
6             01_Definition_and_Purpose_of_Photosynthesis/
7                 photosynthesis_definition.py    # Matplotlib code (main)
8                 photosynthesis_definition.png    # rendered image
9             02_Photosynthesis_Chemical_Equation/
10                 photosynthesis_equation.py
11                 photosynthesis_equation.png
12             ... (more slides)
13         explain_me_projectile_motion/
14             01_.../
15             ...
16         summary_results.json      # (optional) global summary from
                                     generator

```

Naming conventions.

- **Task folders:** derived from the input prompt (e.g., `explain_me_photosynthesis`).
- **Slide folders:** two-digit index + brief title (e.g., `01_Definition_and_Purpose_of_Photosynthesis`).
- **Per-slide files:** one `.py` (Matplotlib code) and one `.png` image.

5 Configure Secrets and Models

1) Set your API key (do not hard-code)

Linux/macOS (bash):

```
1 export OPENAI_API_KEY="YOUR_NEW_SECURE_KEY"
```

Windows (PowerShell):

```
1 $env:OPENAI_API_KEY="YOUR_NEW_SECURE_KEY"
```

6 Run the Script

Jupyter/Notebook

Step 1: Open `final.ipynb` and select the `edu-slides` kernel (the conda env you created).

Step 2: In the first cell, set your inputs. Each item is a 2-tuple (`task`, `class`):

```

1 inputs = [
2     ("explain me photosynthesis", "High School"),
3     ("explain me projectile motion using cricket", "High School"),
4 ]

```

Step 3: Run the notebook top-to-bottom (Run All). Ensure `nest_asyncio` is installed and `OPENAI_API_KEY` is exported in the environment.

Step 4: On success, an output directory `viz_outputs/` is created in the project root. For each task there is a subfolder, and inside it one subfolder per slide, each containing:

- a .py file — the Matplotlib source (primary artifact)
- a matching .png image — the rendered figure

```
1 viz_outputs/<task_slug>/  
2     01_<Slide_Title>/  
3         <slide_name>.py  
4         <slide_name>.png  
5     ...
```

Step 5: To re-render or modify a single slide, change into that slide's folder and run the script:

```
1 cd viz_outputs/<task_slug>/01_<Slide_Title>/  
2 python <slide_name>.py
```