**Project Objective**

The goal of this project is to automate the production and publishing of a YouTube video using multiple AI agents that perform specific tasks. The idea is to have an end-to-end system where each agent is responsible for one part of the workflow—from researching trending topics and generating a script outline, to creating a voiceover, video, thumbnail, and related metadata—culminating in an automated (or semi-automated) upload process that requires only your final approval.

**Detailed Workflow**

1. **Content Research Agent**
   * **Objective:**  
     Fetch trending topics (from Google Trends) to determine a relevant subject for the video.
   * **What We Did:**  
     We attempted to fetch trending topics using the pytrends library. However, due to a 404 response from Google, we built in a fallback that uses a default topic (e.g., "Introduction to AI").
   * **Output:**  
     A CSV or DataFrame of trending topics, which defaults to a preset topic when live data isn't available.
2. **Scriptwriting & Outline Agent**
   * **Objective:**  
     Generate a video script outline based on the trending topic, including cues (e.g., “[PAUSE]”) for pacing during narration.
   * **What We Did:**  
     We built a script that reads the trending topic (or uses the default) and creates a text file (script\_outline.txt) with a structured outline (Introduction, Background, Analysis, Conclusion), with “[PAUSE]” markers inserted.
   * **Output:**  
     A text file with the script outline.
3. **Voiceover/Audio Agent**
   * **Objective:**  
     Generate a voiceover for the script. Initially, we used the macOS “say” command, then experimented with Coqui TTS.
   * **What We Did:**  
     We integrated Coqui TTS (using the VCTK-based VITS model) and tried different speaker IDs to achieve a deep, male voice reminiscent of Peter Coyote or Morgan Freeman. We also tweaked text preprocessing to adjust pacing. However, we encountered challenges: the voice often sounded robotic or ended up female despite our settings, and tuning the parameters to get exactly the desired tone remains a challenge.
   * **Output:**  
     A generated audio file (voiceover.wav) that we can further evaluate and tweak.
4. **Video Creation & Editing Agent**
   * **Objective:**  
     Combine the voiceover and the script outline into a video. This agent creates a simple black background video, overlays the script as subtitles, and syncs the voiceover audio.
   * **What We Did:**  
     We used ffmpeg (via the ffmpeg-python library) to build a video from the generated assets. The final video is saved as final\_video.mp4.
   * **Output:**  
     A video file (final\_video.mp4).
5. **Content Planning & Calendar Agent**
   * **Objective:**  
     Schedule the video for publication, typically setting a future publish date and recording it in a content calendar file.
   * **What We Did:**  
     We generated a calendar text file (content\_calendar.txt) that schedules the video (e.g., for tomorrow at 12:00 PM).
   * **Output:**  
     A content calendar file.
6. **Thumbnail & Graphic Design Agent**
   * **Objective:**  
     Generate a thumbnail image for the video.
   * **What We Did:**  
     Using Python’s Pillow library, we created a simple thumbnail image with text overlay and saved it as thumbnail.jpg.
   * **Output:**  
     A thumbnail image file.
7. **SEO & Metadata Optimization Agent**
   * **Objective:**  
     Generate metadata (title, description, tags) for the video to optimize its discoverability on YouTube.
   * **What We Did:**  
     We built a script that outputs a JSON file (metadata.json) with predefined metadata based on the video topic.
   * **Output:**  
     A metadata JSON file.
8. **Social Media & Promotion Agent**
   * **Objective:**  
     Generate a social media post to promote the video.
   * **What We Did:**  
     A script was created to output a text file (social\_post.txt) containing a promotional message.
   * **Output:**  
     A text file with a simulated social media post.
9. **Analytics & Performance Agent**
   * **Objective:**  
     Simulate gathering analytics data (views, watch time, etc.) for the video.
   * **What We Did:**  
     A script was built that generates an analytics report (analytics\_report.json) with simulated metrics.
   * **Output:**  
     An analytics report file in JSON format.
10. **Quality Assurance (QA) Agent**
    * **Objective:**  
      Run a check on all the generated assets to ensure that everything is present and meets basic quality criteria.
    * **What We Did:**  
      The QA agent simply prints a status for each asset (script, voiceover, video, thumbnail, etc.).
    * **Output:**  
      A console output indicating that all assets passed quality assurance.
11. **High-Level Manager Agent**
    * **Objective:**  
      Consolidate the outputs from all agents into a single report, providing a comprehensive view of the entire workflow.
    * **What We Did:**  
      A script was built that reads the outputs from all agents and creates a consolidated report (consolidated\_report.txt).
    * **Output:**  
      A consolidated report file.
12. **Video Publishing Agent**
    * **Objective:**  
      Automate the final publishing process by scheduling the video upload and prompting for your final approval 3 hours before the scheduled publish time.
    * **What We Did:**  
      A publishing agent was developed that calculates the scheduled time and reminder time, waits until the reminder, and then prompts for approval before simulating an upload.
    * **Output:**  
      A simulated upload process and a consolidated report on the publishing status.

**Challenges Encountered**

* **Fetching Real-Time Data:**  
  Google Trends API requests (via pytrends) returned 404 errors. We implemented fallbacks to default topics to ensure continuity.
* **Voiceover Quality:**  
  Initially, the macOS "say" command and early attempts with Coqui TTS did not yield a satisfactory, deep male voice. Tweaking the model and parameters (including speaker IDs) is still challenging, and the desired voice remains elusive.
* **Python Version and Package Compatibility:**  
  Python 3.13 caused issues with some packages, so we switched to Python 3.11 to maintain compatibility with Coqui TTS and other libraries.
* **Automation vs. Manual Intervention:**  
  While the goal is full automation, some components (like voiceover quality and script timing) require fine-tuning. We built in fallback mechanisms and a final approval stage to minimize your intervention while still allowing for manual review when needed.
* **Integration Complexity:**  
  Bringing together many different agents (each using different libraries and tools) has been challenging, especially ensuring that file paths, data formats, and timings align perfectly.

**Overall Structure and Objective**

* **Objective:**  
  To create a fully automated workflow that generates, reviews, and publishes a YouTube video with minimal manual intervention.
* **Workflow Structure:**
  1. **Content Research** (Fetch trending topic or use default)
  2. **Scriptwriting & Outline** (Generate script with pause cues)
  3. **Voiceover/Audio** (Generate narration using TTS; later replaced by your narration if needed)
  4. **Video Creation & Editing** (Combine voiceover and script into a video)
  5. **Content Planning** (Schedule the video publication)
  6. **Thumbnail Creation** (Generate a thumbnail)
  7. **SEO & Metadata** (Create metadata for the video)
  8. **Social Media Promotion** (Generate a social media post)
  9. **Analytics & Performance** (Simulate video analytics)
  10. **Quality Assurance** (Check that all assets exist)
  11. **High-Level Management** (Consolidate all outputs)
  12. **Video Publishing** (Automate the final upload process after final approval)
* **Next Steps:**  
  Now that all agents are built and tested individually, the idea is to have a master orchestrator script that triggers all these agents sequentially, allowing you to review the final outputs and give a final approval (especially for publishing). This orchestrator makes the whole workflow run like a well-oiled machine with minimal manual intervention.