YouTube Automation Tool Checklist

Steps to build and implement your own YouTube content automation pipeline.

Based on I Built A Full Youtube Automation Tool With Cursor Ai Auto Upload Shorts Descriptions More by Casey Builds It

---

1. Core Automation Setup

Establish the foundational components for your YouTube automation tool.

- ■ Set up a Python-based automation pipeline.
- ■ Integrate transcript generation from local video files.
- ■ Implement AI content generation for descriptions and comments.
- ■ Configure YouTube Data API for automated video uploads.
- ■ Develop Shorts generation using FFmpeg and FFprobe.
- ■ Ensure Shorts are cropped and formatted for vertical viewing.
- ■ Add blurred backgrounds to Shorts clips.

2. Transcription Module

Focus on accurately capturing audio from your videos.

- ■ Choose and integrate a transcription tool (e.g., faster-whisper).
- ■ Ensure the tool generates transcripts directly from local video files.
- ■ Verify transcript accuracy for subsequent AI processing.
- ■ Test transcription with various audio qualities and accents.

3. AI Content Generation

Leverage AI to create engaging and SEO-friendly text content.

- ■ Select an AI model for content generation (e.g., Claude).
- ■ Input video transcripts into the AI model.
- ■ Generate SEO-friendly descriptions for long-form videos.
- ■ Generate initial comments for each video upload.
- ■ Generate unique descriptions for each Short.
- ■ Generate unique comments for each Short.
- ■ Experiment with AI prompts for optimal content quality.

4. Video Upload Automation

Streamline the process of publishing videos to YouTube.

- ■ Integrate with the YouTube Data API.
- ■ Configure API credentials securely.
- ■ Automate the pushing of videos live.

- [ ] Set up metadata fields for automated uploads (title, tags, etc.).
- [ ] Test the upload process with draft and published videos.

5. Shorts Creation and Formatting

Automate the creation of short-form video content.

- [ ] Implement logic to slice long-form videos into multiple Shorts.
- [ ] Utilize FFmpeg and FFprobe for video manipulation.
- [ ] Develop smart cropping to follow on-screen action.
- [ ] Format each Short clip for vertical aspect ratio.
- [ ] Generate separate metadata (description, comment, thumbnail) for each Short.
- [ ] Automate the posting of each generated Short.

6. User Interface and Management

Build a user-friendly interface for managing your automation tool.

- [ ] Develop a Graphical User Interface (GUI) for the Python script.
- [ ] Create a video management dashboard.
- [ ] Allow users to pull up existing YouTube videos within the dashboard.
- [ ] Enable editing of titles and thumbnails via the dashboard.
- [ ] Implement functionality to regenerate content (descriptions, comments) after upload.

7. Advanced Features & Future Development

Consider additional functionalities and ongoing improvements.

- [ ] Integrate blog auto-publishing for videos.
- [ ] Aim to auto-publish videos to dedicated website pages.
- [ ] Continuously refine the smart cropping algorithm.
- [ ] Explore options for automated thumbnail generation.
- [ ] Investigate tools for managing multiple YouTube channels.

Resources & Further Reading

Use these pages while working through the checklist:

- Casey Builds It

https://caseybuildsit.com/

- Casey Builds It

https://caseybuildsit.com/the-files/

- Downloads

https://caseybuildsit.com/downloads/

- Casey Builds It

https://caseybuildsit.com/make-contact/

Casey Builds It | -
- | caseybuildsit.com
Automate your YouTube workflow to save time and focus on content creation.