Deliverables:

This project presents a fully automated pipeline for generating educational YouTube Shorts using AI agents. It simulates how content teams can scale video production by automating topic research, scriptwriting, and video generation — all with minimal manual intervention.

To demonstrate feasibility, I've built the system using free-tier tools:

- **YouTube Data API (free)** for trending topic discovery
- **Together AI LLM (free)** for generating short voiceover scripts
- **Elai.io (free trial)** for creating avatar-based videos

Please note: As the solution relies on free trial versions and limited LLM access, the final video quality may not reflect the full potential of this system.

In addition to the working pipeline, I've also outlined 2–3 alternative approaches that could significantly improve performance, realism, and scalability if provided with full access to high-end tools (e.g., OpenAI GPT-4, Gemini Pro, or multimodal LLMs). These approaches are explained later in this document under the "Multiple Development Approaches" section.

This documentation covers:

- Project structure & setup
- Technology Used
- Architecture diagram
- Links
- Evaluation checklist
- Future enhancements

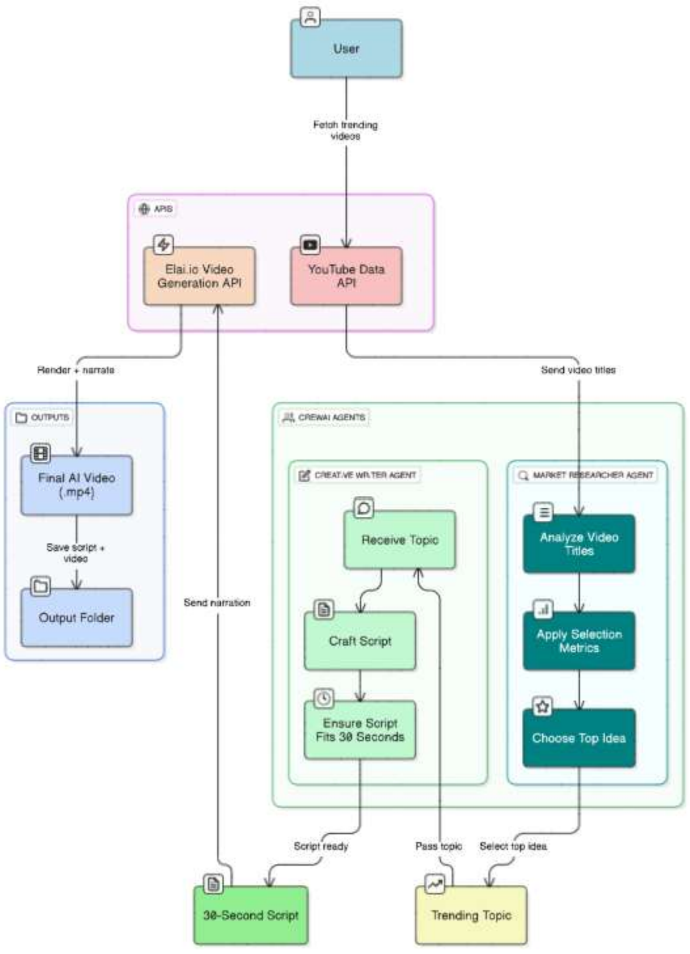
Technology Used :

Component	Tool / API
AI Orchestration	CrewAI (which works like humans)
Language Model	Together AI – LLaMA 3.3 70B Turbo
Video Generation	Elai.io API
Data Source	YouTube Data API v3

Project Structure:

```
youtube-content-automation/
│
├── main.py                   #  Main script: runs the full pipeline
├── config.py                 # API keys, model, and settings
├── .env                      # Stores secret API keys (not committed)
├── requirements.txt          #  Python dependencies
│
│ agents/                     # 🧠 AI agents using CrewAI
│   ├── __init__.py
│   ├── market_researcher.py  # → Finds trending YouTube topics
│   └── creative_writer.py    # → Writes 30s spoken scripts
│
├── tasks/                    #  Tasks assigned to agents
│   ├── __init__.py
│   ├── research_task.py      # → Assigns research job to agent
│   └── writing_task.py       # → Assigns script-writing task
│
├── tools/                    # 🔌 External API utilities
│   ├── __init__.py
│   ├── youtube_tools.py      # → YouTube Data API integration
│   └── elai_video_generator.py # → Elai.io video generation API
│
├── outputs/                  # 📤 Output folder for videos and scripts
│   └── run_YYYYMMDD_HHMMSS/  # Timestamped subfolder per run
│       ├── script.txt        # Generated voiceover script
│       └── final_video.mp4   # Final video with AI narration
```

Architecture Diagram:



Links :

1. Github Link: https://github.com/dhowfeekhasan/Youtube-Content-Automation
2. How to run the project: https://github.com/dhowfeekhasan/Youtube-Content-Automation?tab=readme-ov-file#how-to-run-the-project
3. Sample Output: https://app.elai.io/v/6887d178c670805733d4d462

Evaluation Checklist:

Content Quality & Performance Assurance

- Using top-performing YouTube topics as input (via Data API)
- Enforcing LLM prompt constraints (spoken-only, <90 words, no filler)

Automation Depth

- YouTube Data API – Topic scraping, trend filtering
- Together AI LLM – Voiceover script generation
- Elai.io API – Avatar narration and video rendering

Code Quality

- Modular (clearly separated into agents/, tasks/, tools/)
- Configurable (config.py, .env, constants)
- Readable, well-documented, and uses CrewAI best practices
- Includes .gitignore and .env.example for clean collaboration

Some other approaches

Approach 1: Separate LLMs per Agent (High Accuracy & Speed)

Instead of sharing a single LLM across all agents, each agent can be assigned its own LLM instance.
This allows specialized, parallel processing, improving accuracy and speed for tasks like research and scriptwriting.

Example: In the current version, both the market researcher and scriptwriter share the same LLM.
A more optimized setup would give each agent its own LLM — allowing smoother execution in large-scale pipelines.

Approach 2: CrewAI with Multi-Agent Workflow + Realistic LLMs (Scalable Teamwork)

CrewAI enables a collaborative workflow where multiple agents handle distinct tasks like research, writing, and production.
Though CrewAI defaults to OpenAI, it can be enhanced using powerful models like

Gemini or OpenAI GPT-4, which are capable of generating more realistic and meaningful video content.

Insight: Using advanced LLMs with multimodal capability (like OpenAI's GPT-4 or Gemini Pro Vision) can lead to near-human scriptwriting and avatar interactions, making videos more engaging and production-ready.

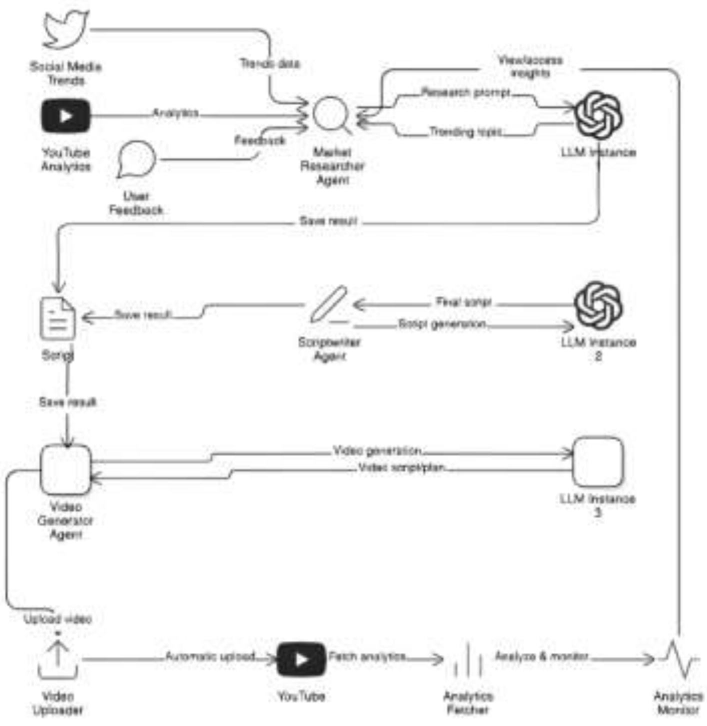
Approach 3: Social Platform Integration for Trend Discovery *(User-Driven Insights)*

Instead of depending solely on the YouTube API for trending topics, this approach proposes integrating social platforms like Instagram, LinkedIn, and Twitter (X) to gather real user engagement signals and crowd-sourced content insights.

- Instagram Reels & Hashtags: Analyze viral educational reels, hashtags, or explore pages to extract trending themes.
- LinkedIn Polls & Comments: Collect responses from students and professionals on topics they struggle with or are interested in.
- Twitter/X Trends: Use hashtags, tweet frequency, and replies to identify current educational conversations.

Notes:

- Requires access to developer accounts (Meta, LinkedIn, Twitter)
- Must comply with data privacy policies and platform scraping rules
- Involves additional effort in data cleaning, prompt shaping, and trend scoring


Social Media Trends
Trends data
YouTube Analytics
Analytics
User Feedback
Feedback
Market Researcher Agent
View/access insights
Research prompt
Trending topic
LLM Instance
Save result
Script
Save result
Scriptwriter Agent
Final script
Script generation
LLM Instance 2
Save result
Video Generator Agent
Video generation
Video script/plan
LLM Instance 3
Upload video
Video Uploader
Automatic upload
YouTube
Fetch analytics
Analytics Fetcher
Analyze & monitor
Analytics Monitor