AI YouTube Copilot – End-to-End Technical Documentation

1. About the Project

AI YouTube Copilot is an end-to-end, fully local, open-source AI system designed to assist AI/ML content creators in producing **high-quality YouTube Shorts and Long-Form technical videos**. The system is optimized for:

- Explaining AI/ML concepts clearly (student & interviewer level)
- Generating structured scripts for shorts and long videos
- Reading and understanding complete GitHub repositories (code + docs)
- Producing clean architecture and flow diagrams
- Operating with **zero paid APIs** and **offline-first design**

The system runs efficiently on a **16 GB RAM + RTX 3050 GPU** setup using quantized local LLMs.

---

2. Key Objectives

- Automate AI/ML content creation without compromising technical depth
- Bridge the gap between **code → explanation → video-ready content**
- Enable creators to scale content creation while maintaining accuracy
- Serve as a strong portfolio-grade project demonstrating real-world LLM engineering

---

3. System Capabilities & Features

3.1 Dual Content Modes

A. Shorts Mode (30–60 seconds)

Designed for fast, high-impact content.

Features: - Hook generation (attention grabber) - Simple definitions (beginner-friendly) - Visual explanation planning - Diagram generation (Mermaid / PlantUML) - Teleprompter-style scripts - Tool & concept explainers

Typical use cases: - AI facts - Definitions ("What is RAG?", "What is Fine-tuning?") - Tool explanations - Concept breakdowns

---

B. Long Video Mode (10–30 minutes)

Designed for deep technical walkthroughs.

Features: - GitHub repository ingestion (all files & code) - Architecture explanation - Module-wise code walkthrough - Interview-level explanations - Demo flow planning - Chapter & timestamp generation - Research paper simplification

Typical use cases: - Project showcases - System design explanations - Research breakdowns - Tutorial-style videos

---

4. Technology Stack (100% Free & Local)

4.1 Large Language Models (LLMs)

Managed using **Ollama** (local inference).

Primary models: - **Llama 3 8B (Q4_K_M)** – Main reasoning & explanations - **Mistral 7B Instruct** – Code understanding & walkthroughs - **Phi-3 Medium** – Fast generation for shorts

Only one model is loaded at a time to optimize VRAM usage.

4.2 Embedding & Retrieval (RAG)

- **Embedding Model:** `nomic-embed-text`
- **Vector Database:** ChromaDB (local)

Purpose: - Accurate repository understanding - Semantic search across code & docs - Reduced hallucination

---

4.3 Code & Repository Analysis

- `git clone` (shallow clone)
- Tree-sitter (multi-language parsing)
- Python AST analysis
- Dependency graph extraction

Supported file types: - `.py`, `.js`, `.ts`, `.md`, `.json`, `.yaml`, `.yml`

4.4 Diagram Generation

- **Mermaid.js** – Flowcharts & architectures
- **PlantUML** – Component & sequence diagrams

All diagrams are text-based for clarity, version control, and reuse.

4.5 User Interface

- **Streamlit** (lightweight, fast)

Tabs: - Mode Selection - Shorts Generator - Long Video Generator - Repo Analyzer - Diagrams - Export

---

5. Full System Architecture

5.1 High-Level Flow

User → UI → Controller → (Repo Ingestor | Content Planner) → RAG → LLM → Output Generator

---

5.2 Core Components

Controller

- Manages user mode selection
- Orchestrates pipeline execution

Repo Ingestor

- Clones GitHub repo
- Reads files
- Filters relevant content

File & Code Parser

- Splits content into logical chunks
- Extracts structure & metadata

RAG Engine

- Embeds chunks
- Retrieves relevant context
- Supplies grounded input to LLM

Content Planner

- Builds video outlines
- Chooses explanation depth

Explanation Engine

- Generates multi-level explanations
- Adapts tone (shorts vs long videos)

Diagram Generator

- Produces Mermaid / PlantUML code

---

6. Retrieval-Augmented Generation (RAG) Pipeline

1. Repository files are chunked logically
2. Each chunk is embedded locally
3. Relevant chunks are retrieved per query
4. Retrieved context is passed to LLM
5. LLM generates grounded explanations

This ensures: - High factual accuracy - Minimal hallucination - Interview-grade explanations

---

7. Explanation Levels

The system supports three explanation depths:

- **Beginner Level:** Intuition-based, analogies
- **Student Level:** Technical clarity, step-by-step
- **Interviewer Level:** Design choices, trade-offs, scalability

The user selects the desired level per output.

---

8. Performance Optimization (RTX 3050)

- Quantized models (Q4)
- GPU-only inference
- Lazy file loading
- Cached embeddings
- Context window control
- Progressive summarization

This ensures smooth performance on 16 GB RAM systems.

---

9. Project Structure

```
ai-youtube-copilot/
├── core/
│   ├── controller.py
│   ├── planner.py
│   ├── explainer.py
│   ├── diagrammer.py
│
├── ingest/
│   ├── github_loader.py
│   ├── file_parser.py
│   ├── code_analyzer.py
│
├── rag/
│   ├── chunker.py
│   ├── embedder.py
│   ├── retriever.py
│
├── models/
│   ├── llm_manager.py
│
├── ui/
│   ├── app.py
│
├── outputs/
│
└── README.md
```

---

10. Development Process

Phase 1: Core Setup

- Local LLM setup (Ollama)
- UI skeleton
- Repo ingestion

Phase 2: RAG & Parsing

- Chunking strategy
- Embedding + retrieval
- Code structure extraction

Phase 3: Content Intelligence

- Shorts planner
- Long video planner
- Explanation depth control

Phase 4: Diagrams & Polish

- Architecture diagrams
- Export formats
- Performance tuning

---

11. Usage Guide

Shorts Mode

1. Select Shorts Mode
2. Enter topic or tool name
3. Choose explanation level
4. Generate script & diagrams

Long Video Mode

1. Select Long Video Mode
2. Provide GitHub repo URL
3. Choose explanation depth
4. Generate outline, walkthrough, diagrams

---

12. Outputs

- Teleprompter scripts
- Video outlines
- Mermaid / PlantUML diagrams
- Interview talking points
- Demo flow plans

All outputs are exportable as Markdown or text.

---

13. Why This Project Matters

- Demonstrates real-world LLM system design
- Combines AI + education + productivity
- Fully offline and cost-free
- Directly powers a YouTube AI/ML channel

- Strong differentiator in interviews and portfolios

---

14. Future Enhancements (Optional)

- Auto YouTube metadata generation
- B-roll suggestions
- Multi-language support
- Fine-tuned local models
- Shorts-to-carousel conversion

AI YouTube Copilot is not just a tool—it is a complete **AI-powered content engineering system** built for creators who want accuracy, depth, and scalability without relying on paid APIs.