## **🎯 Project Title**

**Python + PyQt Media Player using MPV with Remote-Controlled Video & Overlay Ads (Debian 11)**

**I need a GUI application which should have a mpvplayer inside and run based on the REST api based inputs! Assume it’s similar to TV playback ! I should be able to run different types of ads on the screen(text/gif) and ad breaks(tv style ad break video ad)!**

## **📌 Objective**

Deliver a fully working **Python-based media player** that uses:

* **MPV** for video playback
* **PyQt** for fullscreen GUI layout
* **REST APIs** to control playback, overlays, and ad injection

The player runs on **Debian 11 (ARM64 or x86)** and is designed for **TV-style media presentation**. The system must **leverage MPV’s native capabilities** (playlist, resume) instead of re-implementing them.

You are **not responsible** for integrating the program into boot flow or kiosk setup — just build a clean, self-contained application with a REST API interface.

## **🧱 Tech Stack (Must Use)**

* **Python 3.x**
* **PyQt5 or PyQt6** – for layout/UI control
* **MPV** – launched via subprocess or controlled with python-mpv
* **Flask** or **FastAPI** – for API server
* **Target OS**: Debian 11 (on ARM64 SBC like RK3566/3588)

## 

## **🔧 Core Features**

### **🎥 1.**

### **Main Movie Playback**

* Accept a **folder path** as input (e.g., /media/usb/videos/)
* Launch MPV with that path and flags:  
  + --save-position-on-quit
  + --input-ipc-server=/tmp/mpvsocket
* Let MPV:  
  + Auto-create a playlist
  + Autoplay on launch
  + Resume from last position after power loss or app restart

⚠️ Do **not** implement custom playlist or resume logic. Use MPV’s internal features.

### **⏩ 2.**

I need UI buttons for next and fastforward, and those should be auto hidden buttons !

## **📺 3.**

## **Ad Playback (2 Types)**

### **🟨 A. Overlay Ads (Non-interrupting: Image or Text)**

* Show scrolling or static banners **while the movie plays**
* Ad overlays must appear at **bottom or side** of the screen
* Main movie must **resize dynamically** to accommodate overlay

#### **✅ API:**

| POST /show-overlay {  "position": "bottom", *// or "side"*  "type": "image" | "text",  "content": "/path/to/image.png" *// or a text string* } |
| --- |

* For text: scroll or render as a fixed banner
* For image: display for 10 seconds (configurable)

#### **✅ API:**

#### **POST /hide-overlay**

Hides the current overlay.

### **🟥 B. Interrupt Ads (Full-Screen Video Ads)**

* Interrupt current movie
* Play an ad video full screen
* Resume the movie after ad finishes

#### **✅ API:**

| POST /play-interrupt-ad {  "file": "/path/to/ad\_video.mp4" } |
| --- |

* Movie must pause
* Ad plays in full screen
* Movie resumes automatically after ad ends

## **📦 Deliverables**

* Fully working **Python application** with:  
  + main.py or player.py
  + PyQt GUI with dynamic layout support
  + REST API for all controls
  + MPV invocation logic
* requirements.txt and README.md
* Test folder with a few dummy videos and an ad
* Proper layout for:  
  + Fullscreen movie
  + Resized movie with bottom/side ads
* All interactions via API — **no keyboard, no GUI buttons**