Project Portfolio Entry

# Project 1: Text-to-Audio Bot

A Python-based tool that converts the content of a PDF into spoken audio using ElevenLabs voice synthesis.

## Overview

Project Name: Text-to-Audio Bot

Role: Full Stack Developer

Tech Stack: Python, PyPDF2, ElevenLabs API, dotenv, MoviePy

Problem Solved: Manually reading long PDF documents can be time-consuming. This bot automates PDF reading by converting it into audio.

Outcome / Impact: Successfully generates human-like voiceovers from PDFs. Can be used for news, articles, or accessibility tools for the visually impaired.

## Features

|  |  |
| --- | --- |
| Component | Description |
| main.py | Entry point; iterates through PDFs and triggers text-to-speech conversion. |
| pdf\_processor.py | Extracts and cleans text from PDFs. |
| voice\_synthesizer.py | Uses ElevenLabs API to convert cleaned text to realistic speech. |
| .env | Stores API keys securely. |
| config.py | Handles audio output directory settings. |
| requirements.txt | Dependency list for easy environment setup. |

## GitHub Repository

👉 [GitHub Repo Link Here]

## Resume-Ready Entry

Text-to-Audio Bot – PDF to Voice Automation

📍 Built a PDF-to-Audio pipeline using Python and ElevenLabs API.

- Engineered automated processing of news/article PDFs into high-quality `.mp3` audio files.

- Implemented OCR cleanup, API-based voice synthesis, and audio file management.

- Useful for accessibility, content listening on-the-go, or content repurposing.

# Project 2: Audio-to-Video Generator

This project converts an audio narration and PDF script into a synchronized, high-quality Instagram Reel-style video with captions and background music, fetched live from APIs.

## Overview

Project Name: Audio-to-Video Generator

Role: AI Workflow Engineer & Video Automation Developer

Tech Stack: Python, MoviePy, FFmpeg, Pixabay API, Freesound API, Pillow, PyPDF2

Problem Solved: Manual creation of video content from scripts is time-consuming. This bot automates the full process from narration to final Instagram Reel.

Outcome / Impact: Creates highly engaging, captioned video content dynamically using only a PDF script and generated narration, ideal for Shorts/Reels content creators.

## Features

|  |  |
| --- | --- |
| Component | Description |
| main.py | Entry point to trigger the full video generation pipeline. |
| video\_fetcher.py | Fetches random stock video and background music dynamically from Pixabay and Freesound APIs. |
| video\_maker.py | Core logic: syncs audio with script, adds subtitles, formats video to 9:16 aspect ratio, and exports final video. |
| utils.py | Selects random local video clips when web sources fail. |
| .env | Secures API keys for external services. |
| Oswald-SemiBold.ttf | Font used for subtitles in the video. |
| requirements.txt | Lists all Python dependencies for environment setup. |

## GitHub Repository

👉 [GitHub Repo Link Here]

## Resume-Ready Entry

Audio-to-Video Generator – Automated Video Creation for Reels

📍 Developed a video generator that transforms a narrated script into Instagram-style videos.

- Automatically fetches stock footage and background music via APIs (Pixabay, Freesound).

- Syncs subtitles line-by-line with narration using PDF script parsing and MoviePy.

- Formats vertical videos (9:16), adds animated captions, mixes narration and BGM, and renders final `.mp4` output.

# Project 3: Z-Score Arbitrage Monitor

A lightweight statistical monitoring system that tracks price differences in real-time across selected cryptocurrency exchanges using a custom spread and Z-score model.

## Overview

Project Name: Z-Score Arbitrage Monitor

Role: Backend Engineer & Quantitative Signal Developer

Tech Stack: Python, Pandas, NumPy, ccxt, PostgreSQL (optional), Telegram/Discord API

Problem Solved: Built a real-time analytics tool to identify rare pricing anomalies in crypto pairs across multiple exchanges.

Outcome / Impact: Generates concise signals based on statistical deviation, useful for manual or algorithmic monitoring of inefficiencies across markets.

## Features

|  |  |
| --- | --- |
| Component | Description |
| main.py | Runs the full engine to monitor crypto pairs across exchanges. |
| data\_feed.py | Collects mid-price data in real-time from multiple crypto exchanges. |
| stats\_engine.py | Maintains rolling spread window and computes Z-score for deviation checks. |
| signal\_manager.py | Applies filters and cooldown logic to determine meaningful signals. |
| alert\_system.py | Sends alerts to Discord, Telegram, or email (configurable). |
| storage.py | Optionally stores signals in PostgreSQL for later use or auditing. |
| config.py | Central configuration of pairs, thresholds, alert channels, and APIs. |

## GitHub Repository

👉 [GitHub Repo Link Here]

## Resume-Ready Entry

Z-Score Arbitrage Monitor – Crypto Price Anomaly Detection Bot

📍 Developed a real-time signal engine that tracks pricing deviations of crypto pairs across multiple exchanges.

- Used Z-score and spread reversion logic to filter high-confidence signals in fast-moving markets.

- Integrated multiple data sources (Binance, Kraken, etc.) via REST APIs with cooldown and alert control.

- Alerts published via Telegram/Discord for real-time monitoring of observed statistical outliers.

- Architected with a modular, event-driven design and scalable components (stats engine, data feed, storage).