Multi Language Video

# Project overview

The project's objective is to create an AI-powered platform that lets users upload amateur or personal videos and instantly makes them available in a variety of languages, complete with natural-sounding, high-quality dubbing that resembles the expert standards of services like Netflix.  
  
To improve accessibility, global reach, and user experience, viewers will be able to easily choose their preferred language, and the video will play with synchronized voiceovers in that language.

# Core Concept

While professional studios invest heavily in multilingual dubbing for entertainment content, there is currently **no accessible solution** that offers the same experience for user-generated videos. This project aims to democratize high-quality multilingual dubbing by combining the latest advancements in:

* **Speech recognition (ASR)**
* **Neural machine translation (NMT)**
* **Voice synthesis and voice cloning**
* **Audio-video synchronization**

# Key Features

* **Video Uploading**: Users can upload personal or informal videos (e.g., vlogs, lectures, social content).
* **Auto Transcription**: The system extracts speech from the video using advanced speech-to-text models.
* **Multilingual Translation**: Transcripts are translated into target languages with near-human quality.
* **Voice Generation**: Target-language audio is generated using realistic AI voice models, optionally preserving speaker identity through voice cloning.
* **Audio-Video Sync**: The new voice track is time-aligned with the original video for natural playback.
* **Language Selection Interface**: Viewers can switch between available languages via an intuitive player (e.g., dropdown menu).

# Use Cases

* Social media creators expanding to global audiences
* Remote teams sharing content across language barriers
* Education and training content localization
* Personal storytelling and family video sharing in multilingual families

# Milestone 1 – Proof of Concept (MVP)

Create a working prototype with a single video processed in 2–3 languages. The prototype will demonstrate:

* End-to-end processing from input video to multilingual output
* Natural AI voice dubbing
* A simple video player with language switching functionality

## Technologies Involved

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | |  | | --- | |  |  |  | | --- | | **Technology Stack** | |
| Speech-to-Text | OpenAI Whisper, Google Speech API |
| Translation | DeepL API, Google Translate API, OpenAI GPT |
| Text-to-Speech (TTS) | ElevenLabs, PlayHT, Azure TTS, OpenAI TTS |
| Voice Cloning (Optional) | ElevenLabs, Resemble.ai |
| Audio-Video Merging | ffmpeg, moviepy (Python) |
| Frontend Video Player | React.js / HTML5 with language toggle |
| Backend (optional) | Python (Flask, FastAPI), Node.js, Django |
| Hosting (optional) | Firebase, Vercel, or local prototype |

## Backend

### Django (Python)

Django will handle:

* Video uploads and storage
* Processing pipeline control (transcription, translation, TTS)
* API endpoints (using Django REST Framework)
* User authentication (if needed)
* Database (e.g., storing video metadata, language versions, etc.)

## Frontend:

#### React.js

React will power your user interface:

* Upload form (drag and drop, file selection)
* Video player with a **language switch dropdown**
* Playback controls and progress bar
* Responsive UI for desktop and mobile

### Working with React and Django

|  |  |  |
| --- | --- | --- |
| **Layer** | **Tool** | **Responsibilities** |
| Frontend | React.js | User Interface (upload, select language, play video) |
| API Layer | Django REST | Communicates between frontend and backend |
| Backend | Django | Processing logic, file handling, TTS, translations |
| Media Pipeline | Python libs | ffmpeg, moviepy, OpenAI/DeepL APIs, TTS tools |
| Database | PostgreSQL | Stores videos, metadata, language options |
| Storage | Local / S3 | Stores videos and generated dubbed versions |

### Why This Stack is a Great Fit for Your Project

|  |  |
| --- | --- |
| **Benefit** | **Explanation** |
| Seamless API Integration | React can easily talk to Django via REST APIs. |
| Python Ecosystem | It is already using AI tools (Whisper, TTS) in Python — Django fits perfectly. |
| Scalable & Secure | Django handles file processing, permissions, and user accounts reliably |
| Modern Frontend Experience | React gives you Netflix-style video controls, dynamic UI, and smooth playback |
| Easy Deployment | It can host both on platforms like Vercel (frontend) and Heroku/Render (backend). |

### Sample Workflow

1. User uploads a video via React UI
2. React calls Django REST API → /api/upload
3. Django stores the video and begins processing (transcribe, translate, TTS)
4. Processed audio files are saved and synced to the original video
5. Django sends back a list of available languages
6. React shows a language toggle + video playback using <video> and custom controls