Automated Podcast Transcription System

Problem Statement

- Podcasts are long; manually navigating content is difficult.
- Manual transcription is slow and error-prone.
- Topic identification requires additional human effort.

Solution:

- Automates transcription with ASR (Whisper).
- Segments topics automatically using semantic similarity & keyword extraction.
- Generates downloadable structured outputs.

System Architecture

Technology Stack

Component	Technology/Tool
ASR / Transcription	OpenAI Whisper Medium Model
Embedding / Segmentation	Sentence-Transformers (MiniLM)
Keyword Extraction	KeyBERT
Backend	Django REST API
Frontend	HTML / CSS / JavaScript (Single Folder)
Storage / Output	JSON & TXT downloads

Data Pipeline

- Audio Input: Upload WAV/MP3.
- Preprocessing: Convert to 16kHz WAV; split into 30s chunks.
- ASR Transcription: Whisper generates text + timestamps.
- Cleaning: Fix contractions, whitespace, punctuation.
- Topic Segmentation: Group semantically similar segments.
- Keyword Labeling: Extract top 3 keywords per topic.
- Output: Display on UI + download JSON/TXT.

Prompt Management

- LLM/AI prompts are internal only (no direct user input).
- Pre-defined, safe prompts for keyword extraction and labeling.
- Example:

  Extract top 3 keywords from the following segment: "Text segment..."

Handling Hallucinations & Misuse

- Whisper used for transcription: deterministic and accurate.
- Keyword extraction limits AI to factual content.
- Users cannot input arbitrary prompts: reduces risk of misuse.

Reducing Infrastructure Costs

- Whisper model loaded once in memory, chunk-wise processing.
- All keyword extraction done locally; no expensive cloud LLM calls.
- Temporary storage ensures minimal disk usage.

Human-in-the-Loop

- Users can review and correct topics.
- Downloaded JSON/TXT can be edited for:
- Correcting transcription errors
- Adjusting topic boundaries

How to Run

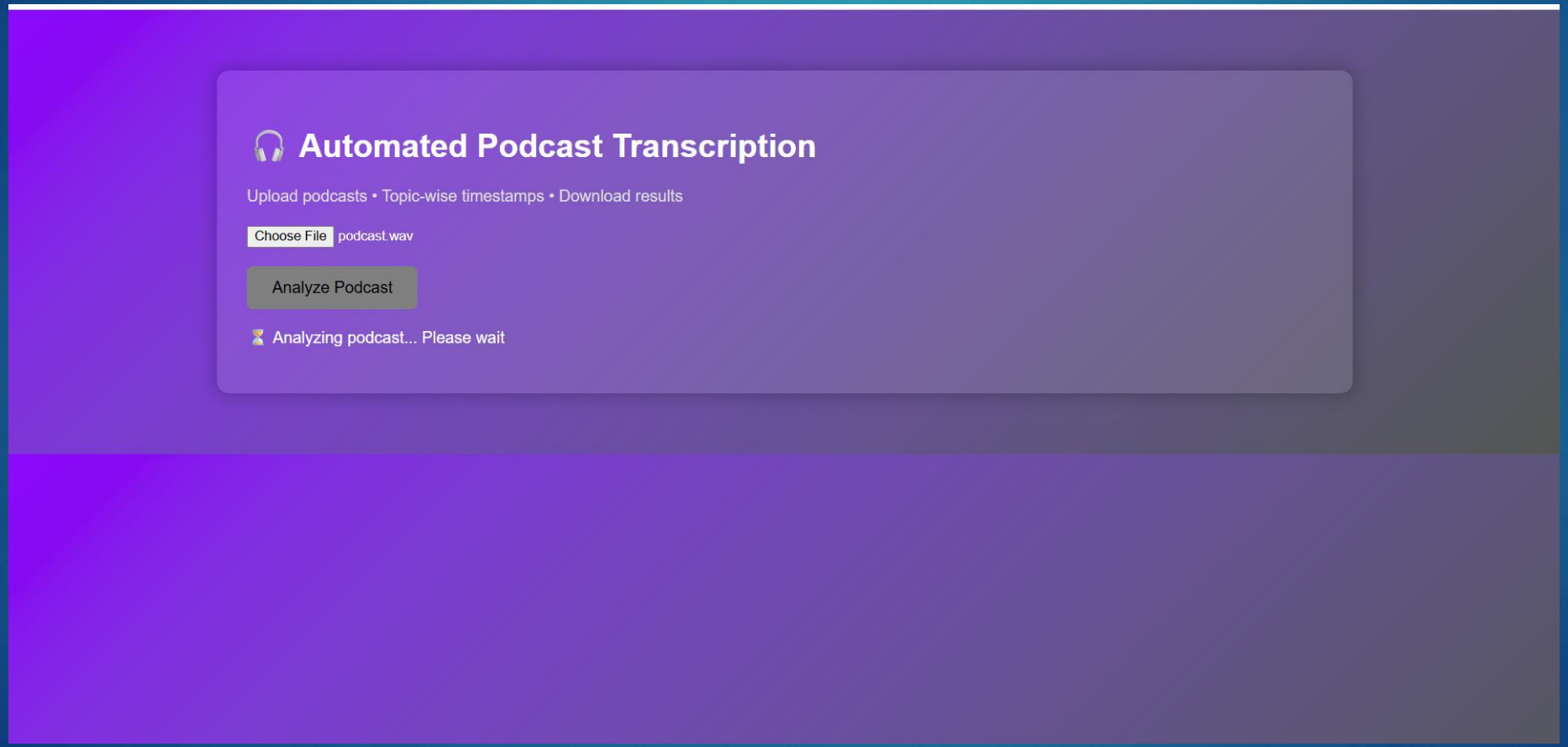
- Clone repository.
- Install dependencies:

  pip install -r requirements.txt
- Run Django server:

  python manage.py runserver
- Open frontend/index.html in browser.
- Upload podcast → Wait for transcription → Download JSON/TXT.

Demo / Screenshots

- Screenshot: File upload interface.



- Screenshot: Topic-wise transcription with keywords.