Lecture 37 & 38: Audio Transcription Model Speech to Text & Speech to Text Part 2

What is Audio Transcription?

- **Definition**: Audio transcription is the process of converting **spoken audio** into **written text** using AI models.
- **Purpose**: Makes audio content searchable, readable, and useful for captions or subtitles.
- **Spring AI** provides a built-in **OpenAI Audio Transcription Model** for this task.

Why Transcription?

- Platforms like **YouTube**, **Udemy**, etc., use this for **auto-generated captions**.
- Improves **accessibility** for learners.
- Useful for **note-taking**, **subtitles**, and **content indexing**.

Controller Setup:

- Create a separate controller for handling **audio requests**.
- Example: AudioGenController as a **RestController**.
- **Speech to Text (STT)** → Convert audio to text.

Code Example: Speech to Text:

```java
@RestController
public class AudioGenController {

    private OpenAiAudioTranscriptionModel audioModel;

    public AudioGenController(OpenAiAudioTranscriptionModel audioModel) {
        this.audioModel = audioModel;
    }

    @PostMapping("api/stt")
    public String speechToText(@RequestParam MultipartFile file) {
        return audioModel.call(file.getResource());
    }
}
```

Explanation:

- MultipartFile file → Accepts uploaded audio.
- audioModel.call(file.getResource()) → Sends audio to the model and returns transcribed text.
- Endpoint: **POST /api/stt**.

Fix Errors:

- **Payload Too Large**
  - By default, Spring supports only **1 MB** uploads.
  - Fix: Add in application.properties:

```
spring.servlet.multipart.max-file-size=20MB
spring.servlet.multipart.max-request-size=20MB
```

Summary:

- **Speech to Text** is implemented using **the OpenAI Audio Transcription Model**.
- Setup involves:
  - A REST Controller.
  - MultipartFile for audio input.
  - audioModel.call() for conversion.
- Real-world use: captions, notes, accessibility, searchable transcripts.