Summer Fellowship Report

On

Spoken Tutorial

Submitted by

Ajay Tripathi

Under the guidance of

Prof.Kannan M. Moudgalya

Chemical Engineering Department

IIT Bombay

June 20, 2018

Acknowledgment

I, Ajay Tripathi, a FOSSEE intern of the Spoken-Tutorial Project is overwhelmed in all humbleness and gratefulness to acknowledge my deep gratitude to all those who have helped me accomplish the tasks assigned to me.
I wholeheartedly thank **Mrs. Nancy Varkey** for selecting me to be a part of this valuable project and for providing constant motivation to do better.
I am very thankful to my mentors **Miss. Kirti Ambre** & **Mr. Saurabh Adhikary** for their valuable suggestions. They were always there to show the right track when i needed help. With help of their brilliant guidance and encouragement, i was able to complete the assigned tasks.
Last but not the least, I wholeheartedly thank all the other colleagues working in different projects for helping me evolve better with their support & advices.

With Regards,

Ajay Tripathi
(Amity University, Noida)

Contents

Chapter 1: Problem #1

Introduction

Spoken-Tutorial provides tutorial in 22 regional languages. To achieve this tutorials are dubbed and video is stored at the server as 22 different videos, hence increasing storage space. It also posed a problem for offline module of the website, as the users had to download all the videos making the package size quite large.

Possible Solutions

- Compress Video Files
- Change File Format (Some formats store same information in lesser space than other formats)
- Store only one video with multiple audio files.
- Reduce the quality of the videos.

Applied Solution

Changing video format from ogv to webm in order to make sure that the videos are smaller in size. Saving only one video and have audio for all the languages and playing them synchronously on the browser. These steps help us reduce the size that needs to be downloaded from the `*\cdcontent*` page.

Results

Course	**Languages**	**Current**	**Solution**
Blender	Assamese, Bengali, English, Gujarati, Hindi, Kannada, Marathi, Nepali, Punjabi, Tamil, Urdu	1.6 GB	790.8 MB
Biopython	English	55.9 MB	55.4 MB
LibreOffice Suite Math	Assamese, Bengali, English, Gujarati, Hindi	255.8 MB	143.4 MB

As it can be noted from the table, when downloading the tutorial in multiple languages several hundred MBs can be reduced. The more the number of languages selected, then more data is saved in the proposed system.

Test Cases

Task	Status
Upload System: Uploading audio & video tutorials as a contributor and getting it verified by administrator, domain reviewer, quality reviewer and publishing the video to check if the new system works properly.	**Working**
Script System: Checking if the script is correctly synced and the questions from forums appear on the correct timestamp. Integrity of the previous system while asking a new question is also maintained.	**Working**
Deployment: The new code has been deployed to the beta server and is functional.	**Working**

Challenges Faced

In the process of implementing the solution, some interesting challenges were faced and solved as mentioned below:

Changing the current video and audio format

While making silent video and audio files from the existing video files, conversion on files can cause unexpected loss in quality and duration if not done carefully (some of the challenges faced are mentioned below). In cases, File size may also expand exponentially. The server currently contains .ogv and .mp4 format videos. The best conversion is noted from .mp4 to .webm(video) and .ogg(audio). Hence, the contributors will be uploading the video in `*.mp4*` and the other language contributors will upload `*.ogg*` files.

Shortening of Video

When converting the video from ogv to silent ogv video, the length of the video would be lost by 2-5 seconds. This caused problems in video / audio synchronization. This problem was solved by using mp4 files and converting them to webm & instead of removing audio, the audio of the video was filtered by `-af volume 0.0` option in ffmpeg.

Corrupt .ogv videos

In the process of conversion, some of the ogv files, example *Biopython* course gets corrupted losing video quality. This is solved by using using *`.mp4`* instead of the *`.webm`* file format.

Changing Script Format

`.srt` is no more supported by the video.js player (used in the website). Currently, all the scripts are stored in *`.srt`* format, these files needed to be converted to the format supported by the player (*`.vtt`* format).

Changing database structure

After the implementation of the new system, the information in *`video`* column & *`audio`* column need to be changed. The videos do not have language, now, they only contain a *`-Video`* at the end of the name. audio file's path are stored in a separate column.

Synchronization of audio and video

With the help of mozilla's popcorn.js and standard Web APIs, the audio and video are delivered in synchronization.

Chapter 1: Problem #2

Testimonials are send by the users as text / audio / video. The existing system allows only text testimonials to be displayed, the audio and video testimonials are not displayed on the website.

Possible Solutions

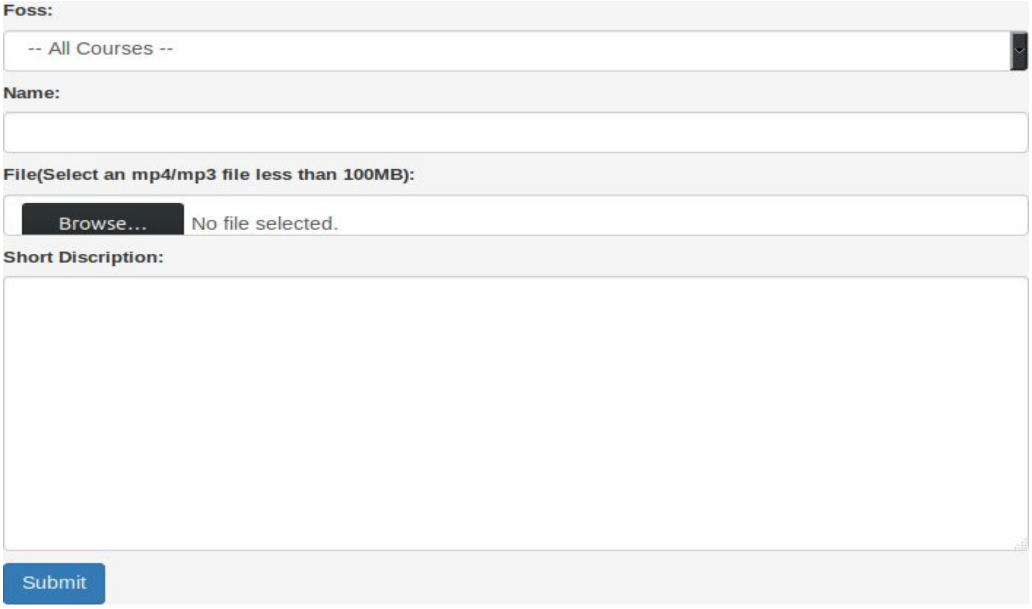
- Uploading all the audio / video testimonial on the server.
- Asking user to upload on their youtube channel and sending the link which can then be embedded on the website.

Applied Solutions

User can send the file to the administrator the way they prefer, that file can be processed and uploaded from the administrator’s end. This allows freedom for the users to send the testimonial and the processing can be done in the spoken office.

Results

The form to be filled to upload the testimonial is shown in (fig. i), it is accessible from the `testimonial/new/` page and after the testimonials are uploaded, they are displayed on the series page (fig. 2) and testimonial page (fig. 3).



Foss:

-- All Courses --

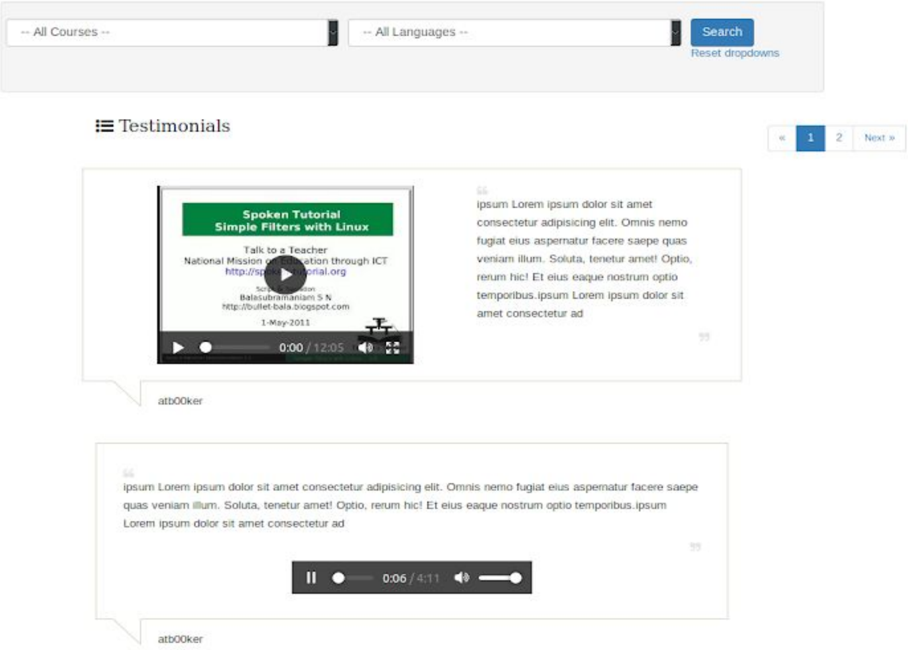
Name:

File(Select an mp4/mp3 file less than 100MB):

No file selected.

Short Discription:

(fig. 1 - Upload Form)



-- All Courses --

-- All Languages --

Search

Reset dropdowns

Testimonials

« 1 2 Next »

Spoken Tutorial
Simple Filters with Linux
Talk to a Teacher
National Mission on Education through ICT
http://spoken-tutorial.org
Balasubramaniam S N
http://bullet-bala.blogspot.com
1-May-2011
0:00 / 12:05

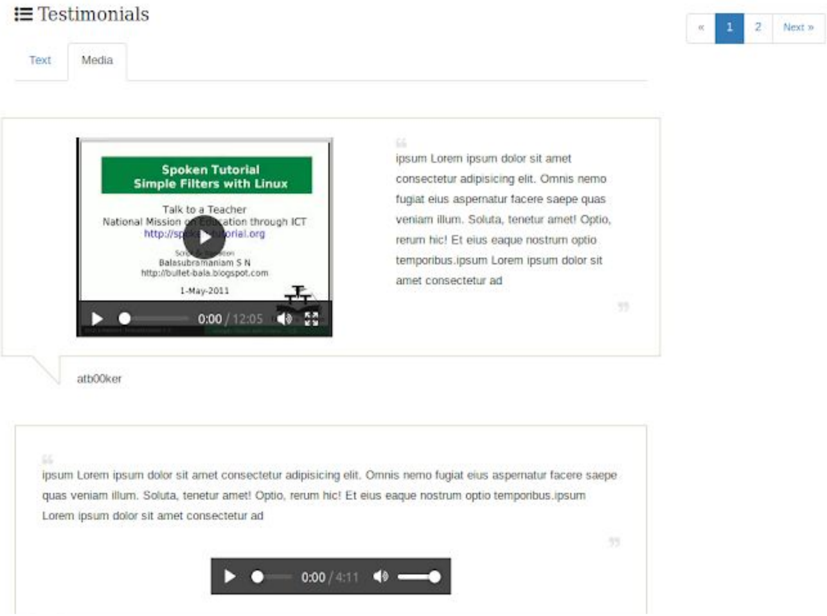
ipsum Lorem ipsum dolor sit amet consectetur adipisicing elit. Omnis nemo fugiat eius aspernatur facere saepe quas veniam illum. Soluta, tenetur amet! Optio, rerum hic! Et eius eaque nostrum optio temporibus.ipsum Lorem ipsum dolor sit amet consectetur ad

atb00ker

ipsum Lorem ipsum dolor sit amet consectetur adipisicing elit. Omnis nemo fugiat eius aspernatur facere saepe quas veniam illum. Soluta, tenetur amet! Optio, rerum hic! Et eius eaque nostrum optio temporibus.ipsum Lorem ipsum dolor sit amet consectetur ad

atb00ker

(fig. 2 - View Tutorials on `series` page)



Testimonials

« 1 2 Next »

Text | Media

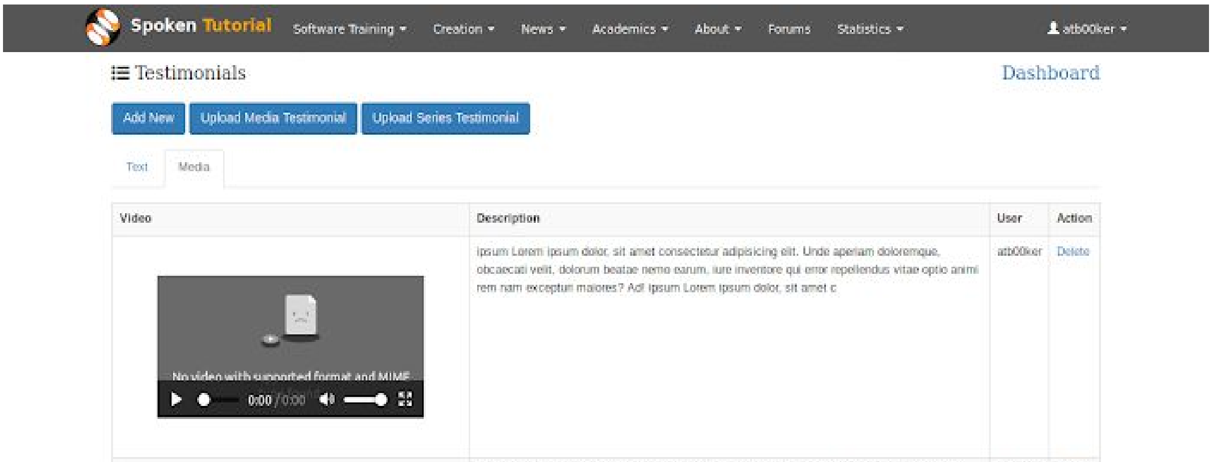
Spoken Tutorial
Simple Filters with Linux
Talk to a Teacher
National Mission on Education through ICT
http://spoken-tutorial.org
Balasubramaniam S N
http://bullet-bala.blogspot.com
1-May-2011
0:00 / 12:05

ipsum Lorem ipsum dolor sit amet consectetur adipisicing elit. Omnis nemo fugiat eius aspernatur facere saepe quas veniam illum. Soluta, tenetur amet! Optio, rerum hic! Et eius eaque nostrum optio temporibus.ipsum Lorem ipsum dolor sit amet consectetur ad

atb00ker

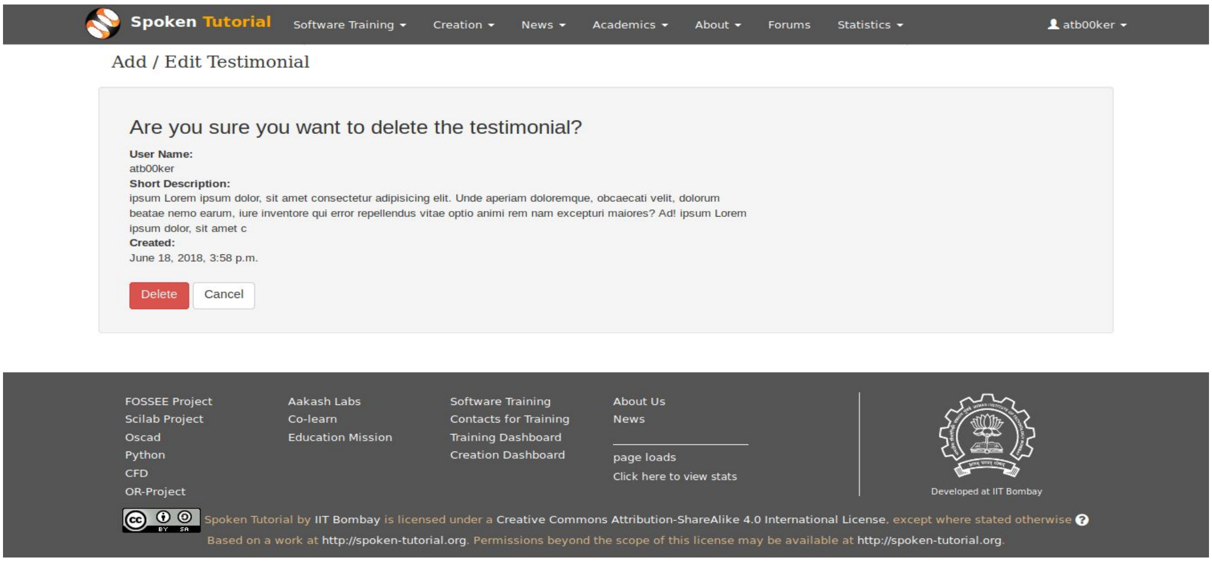
ipsum Lorem ipsum dolor sit amet consectetur adipisicing elit. Omnis nemo fugiat eius aspernatur facere saepe quas veniam illum. Soluta, tenetur amet! Optio, rerum hic! Et eius eaque nostrum optio temporibus.ipsum Lorem ipsum dolor sit amet consectetur ad

(fig. 3 - View Tutorials on `testimonial` page)



Video	Description	User	Action
	ipsum Lorem ipsum dolor, sit amet consectetur adipisicing elit. Unde aperiam doloremque, obcaecati velit, dolorum beatae nemo earum, iure inventore qui error repellendus vitae optio animi rem nam excepturi maiores? Ad! ipsum Lorem ipsum dolor, sit amet c	atb00ker	Delete

(fig. 4 - Media Testimonial review page for the administrator)



Add / Edit Testimonial

Are you sure you want to delete the testimonial?

User Name:
atb00ker

Short Description:
ipsum Lorem ipsum dolor, sit amet consectetur adipisicing elit. Unde aperiam doloremque, obcaecati velit, dolorum beatae nemo earum, iure inventore qui error repellendus vitae optio animi rem nam excepturi maiores? Ad! ipsum Lorem ipsum dolor, sit amet c

Created:
June 18, 2018, 3:58 p.m.

Spoken Tutorial by IIT Bombay is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. except where stated otherwise
Based on a work at http://spoken-tutorial.org. Permissions beyond the scope of this license may be available at http://spoken-tutorial.org.

(fig. 5 - Delete Testimonial page)

Test Cases

Task	**Status**
Create Text Testimonial: Making a new testimonial on the after the implementation of the new code to check the compatibility.	**Working**
Create Video / Audio Testimonial: Creating a new video / audio testimonial and checking for uploaded tutorial on view page.	**Working**
View Testimonials in Series Page	**Working**
View Testimonials in Testimonials Page	**Working**

Delete Audio / Video Testimonial: Deleting tutorial removes it from the database.	Working
Deployment: The new code has been deployed to the beta server and is functional.	Working

Note: events/tests.py contains tests for the module.

Design Considerations

Assumptions and Dependencies

- Python 2.7
- Django == 1.8

Packages & Technology Used

- **FFmpeg**: A complete, cross-platform solution to record, convert and stream Audio and Video.

- **HTML**: It is the main markup language for displaying web pages and other information that can be displayed in a web browser.

- **CSS**: (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language.

- **JavaScript**: It is a prototype-based scripting language that is dynamic, weakly typed and has first-class functions and is mainly used for validation etc.

- **Python**: High-Level Scripting Language for Back-End coding and for server side programming.Python features a dynamic types system and automatic memory management and supports multiple programming paradigms.

- **Django**: Python framework for web application development. It follows MVC(Model View Controller) structure for managing the models and controlling the views

- **MySQL**: It is the world's most used open source relational database management that runs as a server providing multi-user access to a number of databases.

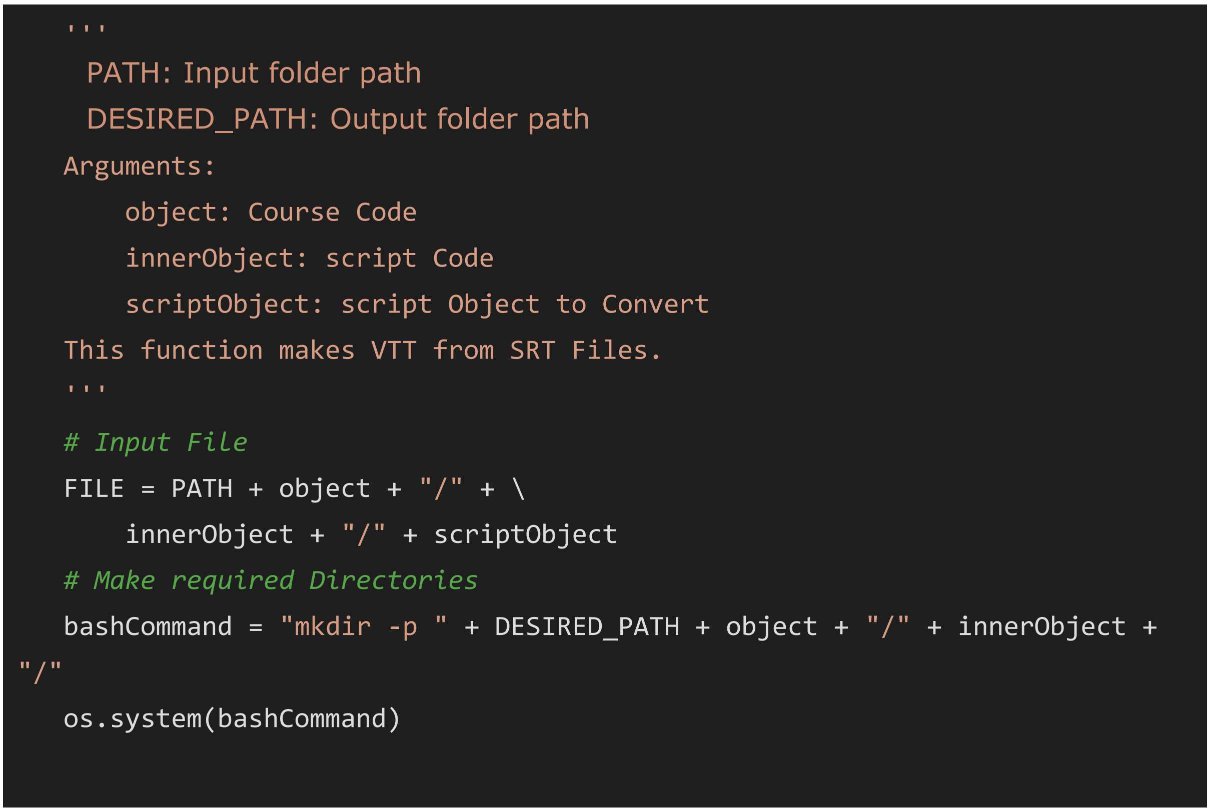
- **Popcorn.js**: it is Mozilla's HTML5 video and media library for the open web. It allows web developers, filmmakers, artists, designers and others to easily create timeline based web productions.

Conversions

- *`.srt`* to *`.vtt`*

Conversion of scripts from *`.srt`* to *`.vtt`* is done with the help of python script.

The following can be used on any *`.srt`* file to convert to *`.vtt`*.



```
'''
 PATH: Input folder path
 DESIRED_PATH: Output folder path
Arguments:
    object: Course Code
    innerObject: script Code
    scriptObject: script Object to Convert
This function makes VTT from SRT Files.
'''
# Input File
FILE = PATH + object + "/" + \
    innerObject + "/" + scriptObject
# Make required Directories
bashCommand = "mkdir -p " + DESIRED_PATH + object + "/" + innerObject +
"/"
os.system(bashCommand)
```

```
    # Make the file
    output = "WEBVTT\n\n"
    with open(FILE) as fout:
        line = fout.readline()

        while line:
            if "-->" in line:
                output = output+line[:8]+".000 --> "
                output = output+line[13:21]
                output = output+".000"+"\n"
            else:
                output = output+line
            line = fout.readline()

    with open(FILE[:-4]+".vtt", "a") as fout:
        objectList = output.split("\n")
        for object in objectList:
            fout.write(object)
            fout.write('\n')
```

- `*.mp4*` to `*.webm*`

Conversion of videos from `*.mp4*` to `*.webm*` is done with the help of FFmpeg.

```
ffmpeg -y -i  <MY_AWESOME_INPUT>.mp4 -vcodec libvpx -af 'volume=0.0' -max_muxing_queue_size 1024 -f webm <MY_AWESOME_OUTPUT>-Video.webm
```

- `*.mp4*` to `*.ogg*`

Conversion of videos from `*.mp4*` to `*.ogg*` is done with the help of FFmpeg.

```
ffmpeg -y -i <MY_AWESOME_INPUT>.mp4 -vn -acodec libvorbis <MY_AWESOME_OUTPUT>.ogg
```

- `*video*` column & `*audio*` column

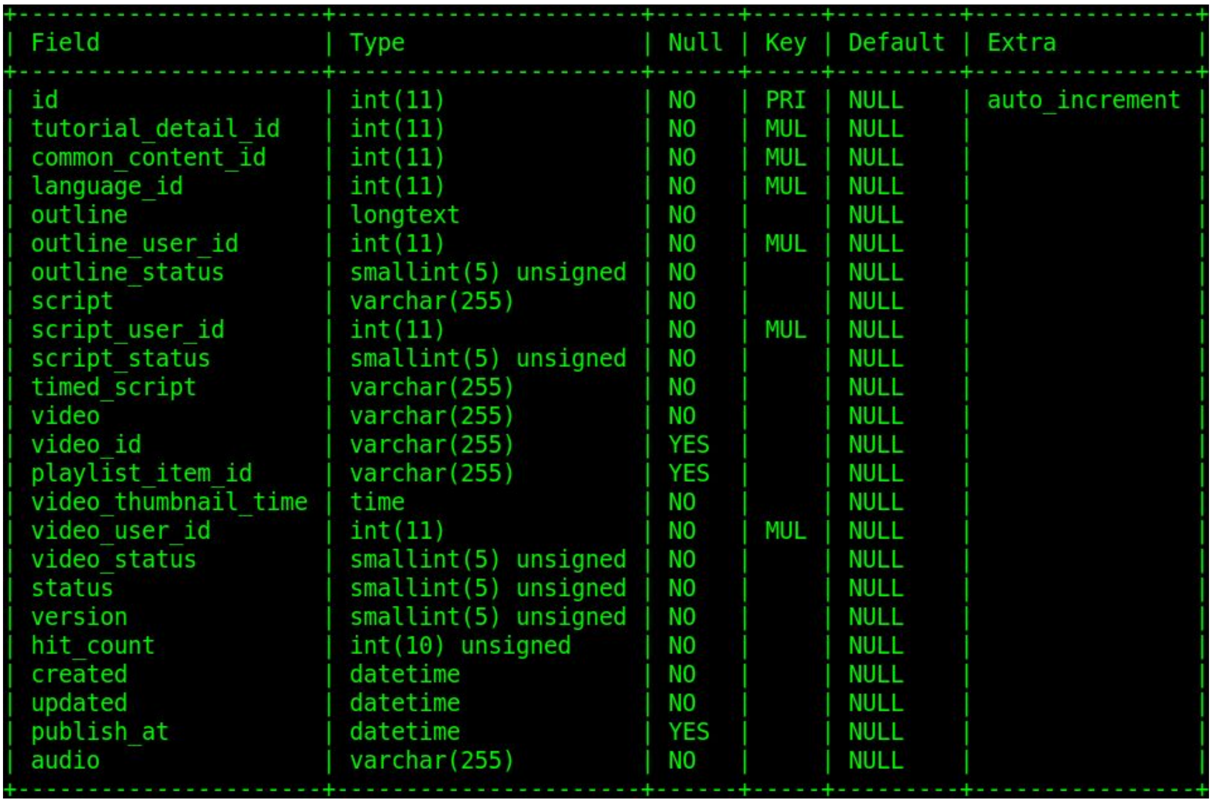
  `*video*` column & `*audio*` column in creation_tutorialresource table. Audio and Video Information for new system can be changed with script:

```
        '''
        Script to change path of the video saved in accordance to
        the new file system & populate audio path column for
existing rows.
        '''
        # Get data from database
        try:
            # Check if the `audio` field exists in the database.
            field_exist_check =
TutorialResource._meta.get_field('audio')
            existing_video_list =
TutorialResource.objects.filter(video__gt="").values_list("pk",
"video", "language_id__name")
            for row in existing_video_list:
                audio_name  = row[1].rsplit('.')[0]
                # To ensure database is not corrupted on running
this script again by mistake.
                if audio_name.rsplit('-',1)[1] != "Video":

TutorialResource.objects.filter(pk=row[0]).update(audio=audio_name
+".ogg", video=audio_name.rsplit(row[2])[0]+"Video.webm")
        except FieldDoesNotExist:
            print ("1001: No changes have occured", error)
        except Exception as error:
            print ("1002:", error)
```

Database Manipulation

- \`creation_tutorialresource\`

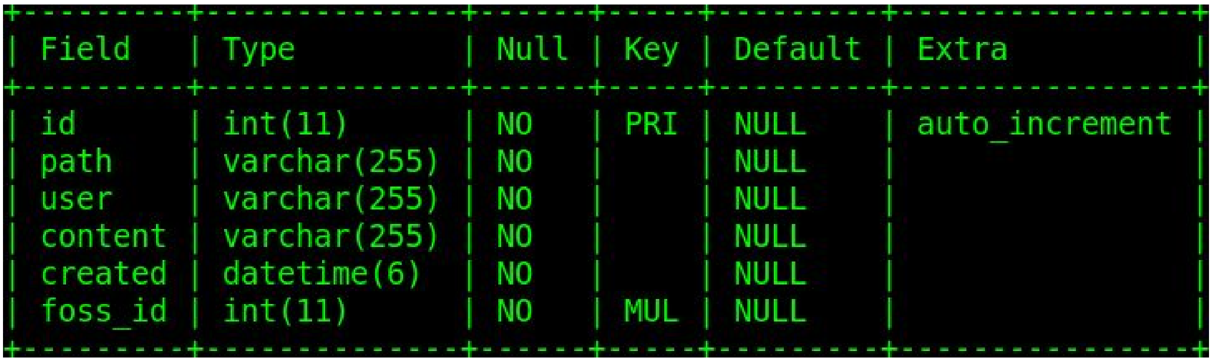


```
+----------------------+----------------------+------+-----+---------+----------------+
| Field                | Type                 | Null | Key | Default | Extra          |
+----------------------+----------------------+------+-----+---------+----------------+
| id                   | int(11)              | NO   | PRI | NULL    | auto_increment |
| tutorial_detail_id   | int(11)              | NO   | MUL | NULL    |                |
| common_content_id    | int(11)              | NO   | MUL | NULL    |                |
| language_id          | int(11)              | NO   | MUL | NULL    |                |
| outline              | longtext             | NO   |     | NULL    |                |
| outline_user_id      | int(11)              | NO   | MUL | NULL    |                |
| outline_status       | smallint(5) unsigned | NO   |     | NULL    |                |
| script               | varchar(255)         | NO   |     | NULL    |                |
| script_user_id       | int(11)              | NO   | MUL | NULL    |                |
| script_status        | smallint(5) unsigned | NO   |     | NULL    |                |
| timed_script         | varchar(255)         | NO   |     | NULL    |                |
| video                | varchar(255)         | NO   |     | NULL    |                |
| video_id             | varchar(255)         | YES  |     | NULL    |                |
| playlist_item_id     | varchar(255)         | YES  |     | NULL    |                |
| video_thumbnail_time | time                 | NO   |     | NULL    |                |
| video_user_id        | int(11)              | NO   | MUL | NULL    |                |
| video_status         | smallint(5) unsigned | NO   |     | NULL    |                |
| status               | smallint(5) unsigned | NO   |     | NULL    |                |
| version              | smallint(5) unsigned | NO   |     | NULL    |                |
| hit_count            | int(10) unsigned     | NO   |     | NULL    |                |
| created              | datetime             | NO   |     | NULL    |                |
| updated              | datetime             | NO   |     | NULL    |                |
| publish_at           | datetime             | YES  |     | NULL    |                |
| audio                | varchar(255)         | NO   |     | NULL    |                |
+----------------------+----------------------+------+-----+---------+----------------+
```

For the implementation of the solution(one video multiple audio), a new column is added in the creation_tutorialresource table named \`*audio*\`.

- \`*audio*\` stores the path to the audio file for the record
- \`*video*\` stores the path to the video file for the record.

- \`events_mediatestimonials\`

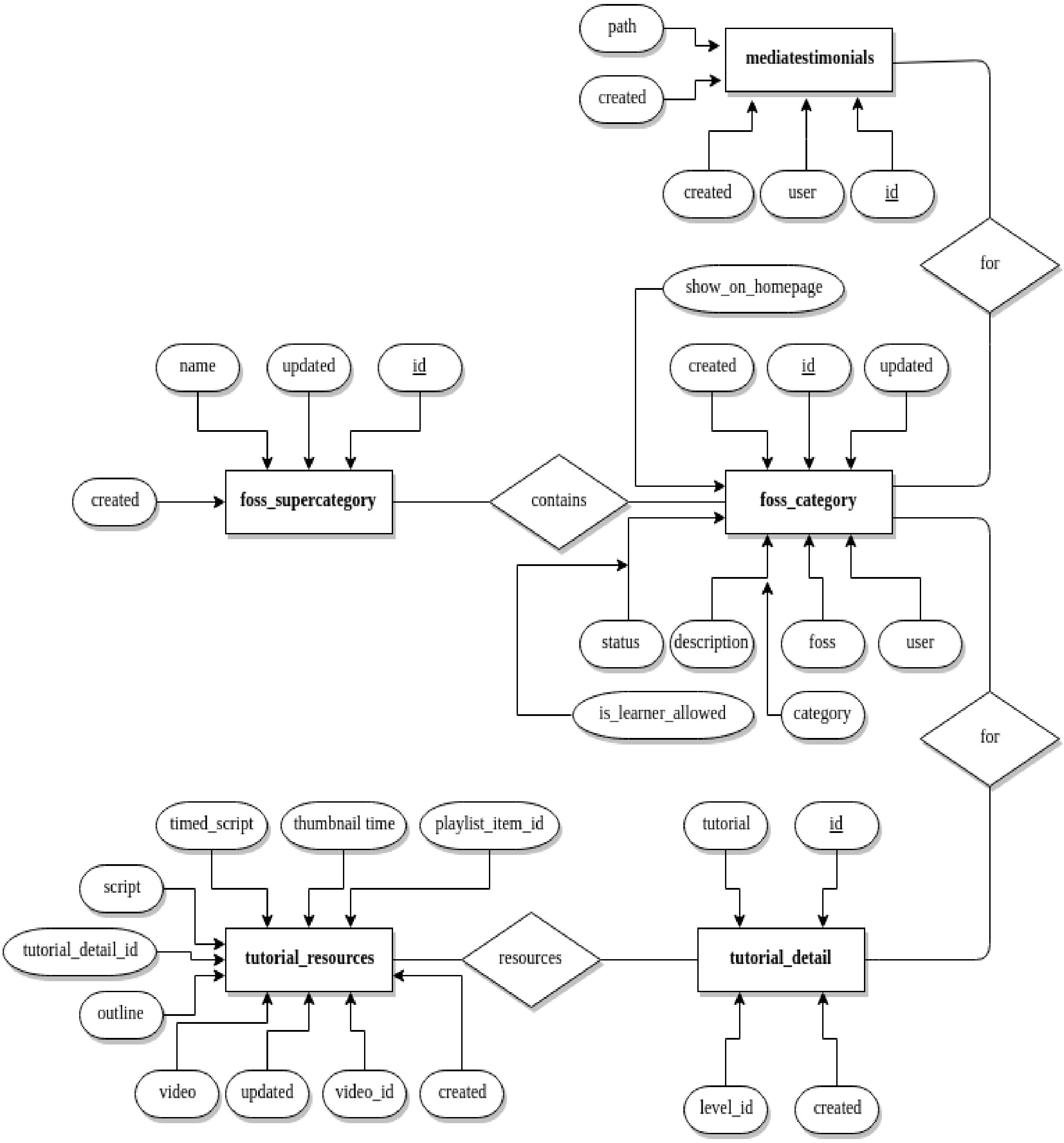


```
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| id      | int(11)      | NO   | PRI | NULL    | auto_increment |
| path    | varchar(255) | NO   |     | NULL    |                |
| user    | varchar(255) | NO   |     | NULL    |                |
| content | varchar(255) | NO   |     | NULL    |                |
| created | datetime(6)  | NO   |     | NULL    |                |
| foss_id | int(11)      | NO   | MUL | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
```

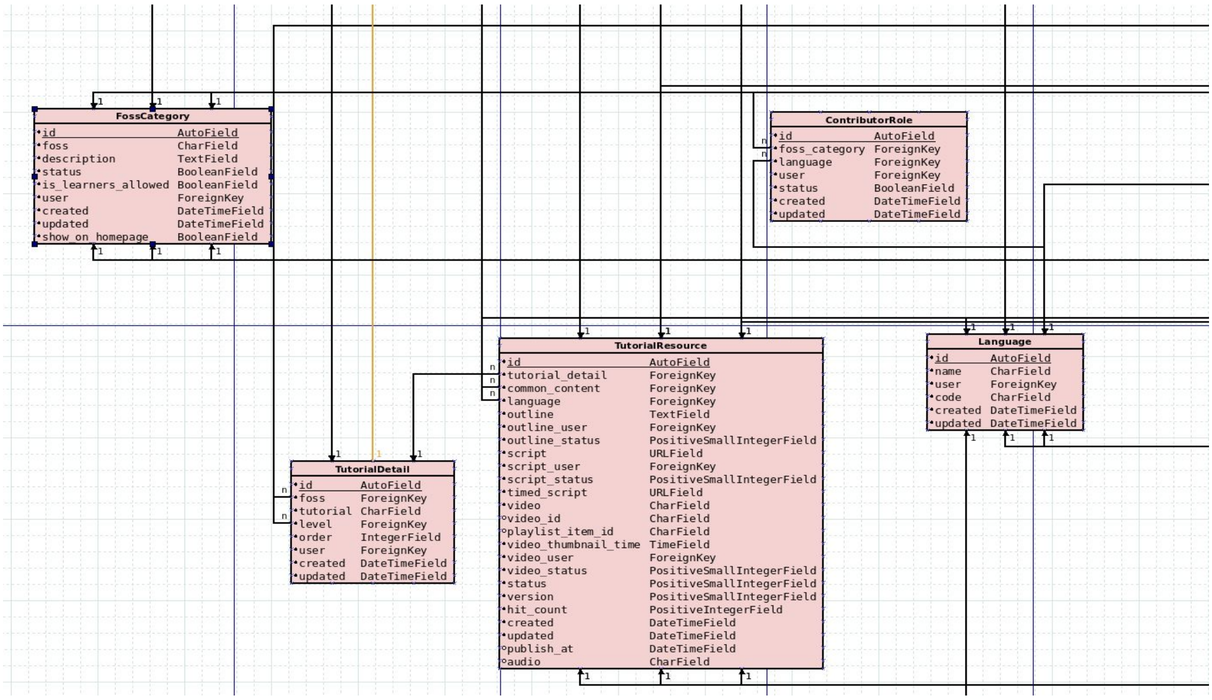
- \`id\` stores the id of media testimonial
- \`path\` stores path to the testimonial

- `user` stores the name of the user who send the testimonial
- `content` stores the text content description of the testimonial
- `created` stores the date & time of creation of the record
- `foss_id` contains a foreign key relation to the foss published for users.

ER Diagram



Schema Diagram



FossCategory	
•id	AutoField
•foss	CharField
•description	TextField
•status	BooleanField
•is_learners_allowed	BooleanField
•user	ForeignKey
•created	DateTimeField
•updated	DateTimeField
•show_on_homepage	BooleanField

ContributorRole	
•id	AutoField
•foss_category	ForeignKey
•language	ForeignKey
•user	ForeignKey
•status	BooleanField
•created	DateTimeField
•updated	DateTimeField

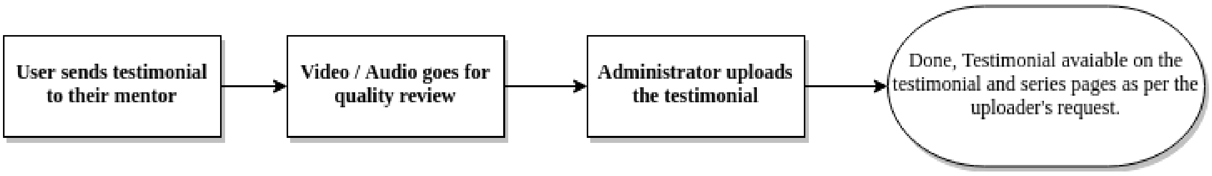
TutorialDetail	
•id	AutoField
•foss	ForeignKey
•tutorial	CharField
•level	ForeignKey
•order	IntegerField
•user	ForeignKey
•created	DateTimeField
•updated	DateTimeField

TutorialResource	
•id	AutoField
•tutorial_detail	ForeignKey
•common_content	ForeignKey
•language	ForeignKey
•outline	TextField
•outline_user	ForeignKey
•outline_status	PositiveSmallIntegerField
•script	URLField
•script_user	ForeignKey
•script_status	PositiveSmallIntegerField
•timed_script	URLField
•video	CharField
○video_id	CharField
○playlist_item_id	CharField
•video_thumbnail_time	TimeField
•video_user	ForeignKey
•video_status	PositiveSmallIntegerField
•status	PositiveSmallIntegerField
•version	PositiveSmallIntegerField
•hit_count	PositiveIntegerField
•created	DateTimeField
•updated	DateTimeField
○publish_at	DateTimeField
○audio	CharField

Language	
•id	AutoField
•name	CharField
•user	ForeignKey
•code	CharField
•created	DateTimeField
•updated	DateTimeField

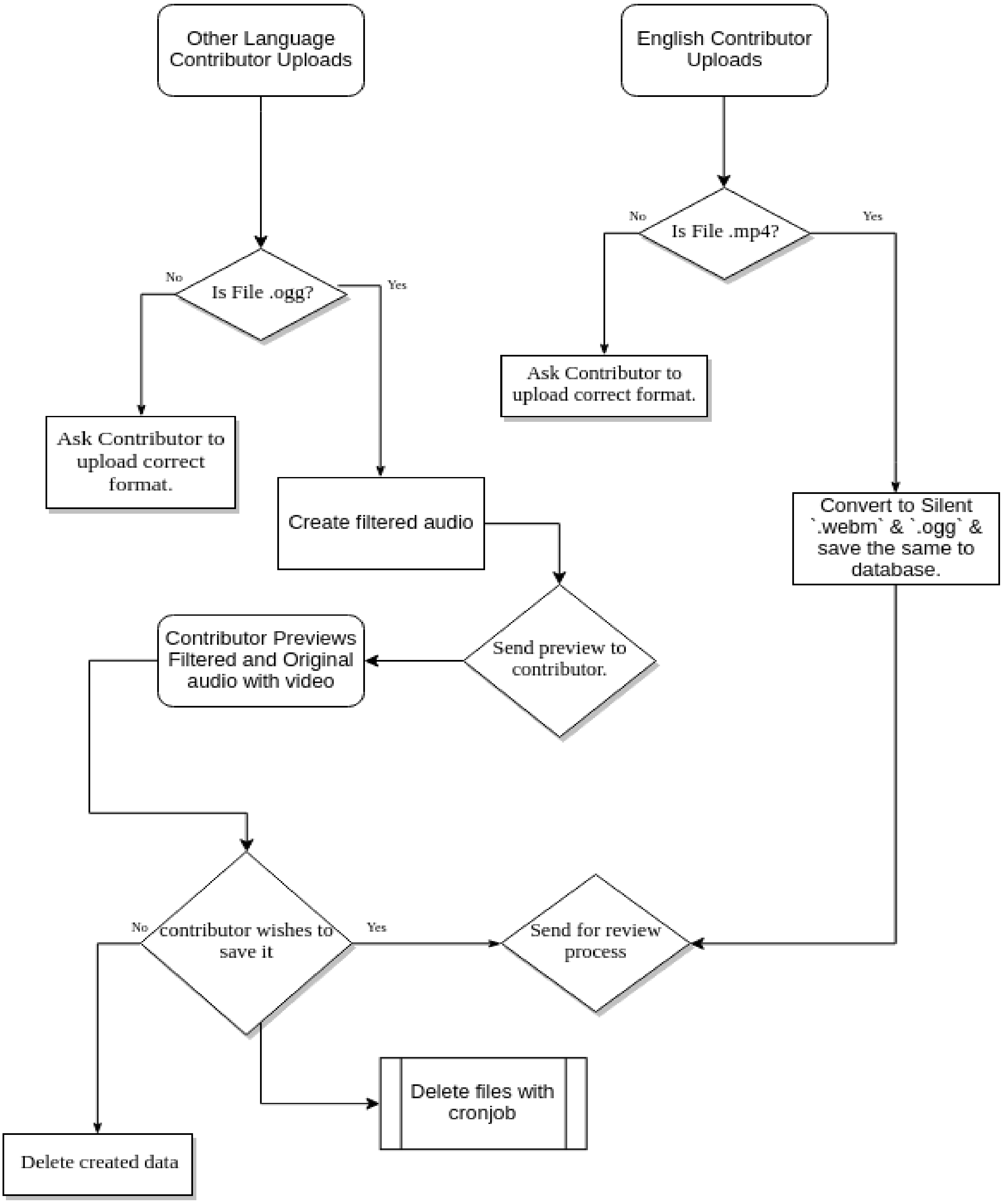
Sequence Diagram

1. The sequence diagram for enabling media testimonials is as followed:
    - Users will to send tutorials to spoken office where the processing of the testimonial can take place, this includes quality check for content and video / audio, once this is done, testimonial can go live.



User sends testimonial to their mentor
Video / Audio goes for quality review
Administrator uploads the testimonial
Done, Testimonial avaiable on the testimonial and series pages as per the uploader's request.

2. The sequence diagram for enabling one video multiple audio task is as followed:
    - Since action for english audio uploaders and other language uploaders is different, the diagram covers the path for both type of uploaders.
    - Diagram assumes audio filtering method is in place.


Other Language Contributor Uploads
English Contributor Uploads
No
Is File .mp4?
Yes
No
Is File .ogg?
Yes
Ask Contributor to upload correct format.
Ask Contributor to upload correct format.
Create filtered audio
Convert to Silent `.webm` & `.ogg` & save the same to database.
Contributor Previews Filtered and Original audio with video
Send preview to contributor.
No
contributor wishes to save it
Yes
Send for review process
Delete files with cronjob
Delete created data

Reference

- **Popcorn.js:** https://github.com/mozilla/popcorn-js
- **FFmpeg package:** http://ffmpeg.org/ffmpeg.html