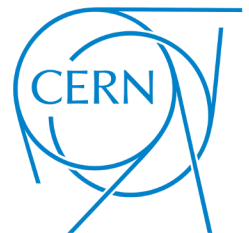


Summer Student **Work Project Report**

Tourpetas Georgios
10th August, 2018

Supervisor: Maria Dimou / *IT Department*





Introduction

CERN Document Server (CDS)

The CERN Document Server (CDS) is the institutional repository that provides acquisition, search and collaborative tools to manage collections of documents produced at CERN. The collections include HEP documents, Multimedia documents, Bulletins, Administrative (Directorate, HR, Finance) documents, EU Project documents (OpenAire, EMI), Library and Project document types.

Part of the Multimedia documents are photos and lecture videos, organised in *Collections*. Specifically videos of CERN lectures are recorded in two distinct channels, results of the *lecturer* and *material* captures.

Aveditor

This project aims at building a web application (Aveditor - <http://cern.ch/aveditor>), to edit, split and merge video channels in the Academic Training video collection in CDS. The result of this work will be a new unique video with *lecturer/material* periodic flip, or small *lecturer* image embedded in the *material* image. It will be used for uploading to the dedicated YouTube channel CERN Lectures and be part of the playlist CERN Academic Training. Also for editing self-made educational videos which describe existing procedures/services and their features and tools to the users.

The Development Process

Code Structure Overview

```
Aveditor/
├── common/
├── secret/
├── services/
│   ├── celery_tasks.py
│   └── ffmpeg.py
├── static/
│   ├── css/
│   └── js/
├── templates/
├── views/
├── main.py
├── upload_form.py
├── settings.py
└── Dockerfile
```

Below is a summary of the contents of each directory:

- common/
Authentication files
- secret/
hmm.. secret
- celery_tasks.py
Necessary functions for server communication
- ffmpeg.py
All the ffmpeg commands that are needed for editing, splitting and merging video channels are implemented here
- static/
Contains the static web files (CSS stylesheets and JavaScript script files) each one in its respective directory
- templates/
Contains the Jinja template files.
- views/
Contains the Flask view files. Each view is a function that “decorates” an HTML Jinja template file by passing it data
- upload_form.py
- settings.py
Contains all the necessary configuration

The Web Interface

The web interface provides a user friendly way of exploiting the power of the implemented ffmpeg commands for editing, splitting and merging video channels.



Create a new video

Camera video URL: ?
Enter an https:// URL:

Slides URL: ?
Enter an https:// URL:

Select intro and outro ?

- ☒ Academic Training
- ☐ E-Learning
- ☐ Other

Period of speaker's presentation ?

From: To:

Period of questions ?

From: To:

Picture in Picture ?

- ☒ continuous
- ☐ for 20 seconds every 100 seconds
- ☐ only camera
- ☐ only slides

Go!



Conclusion

I hope that this tool will make a small contribution in order to make Cern Academic Lectures available to a wider audience, thanks to the popularity of Youtube.

Maria, Marek and Rene thank you for your patience and your valuable help.

