

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Technology Degree in

Information Technology

Balabharathi.P

Reg. No. 38120017



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI 600119, TAMILNADU

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under section 3 of UGC Act ,1956)

JEPPIAAR NAGAR,RAJIV GANDHI SALAI,CHENNAI-600119

www.sathyabama.ac.in

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work by **Balabharathi.P (Reg. No. 38120017)** who underwent the professional training on “Vedio editor” under our supervision from August 2021 to November 2021.

Internal Guide

Dr,R.Subhashini,M.E.,(Ph.D),

Head of the Department

Dr. R.SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I Balabharathi (**Reg.No. 38120017**) hereby declare that the Professional Training Report "**Vedio Editor**" done by me under the guidance of **Dr.R.Subhashini,M.E.,(Ph.D)** at **Sathyabama university** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D., Dean, School of Computing** and **Dr. R.SUBHASHINI M.E., Ph.D., Head of the Department, Dept. of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **R.Subhashini,M.E.,(Ph.D)**,for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

This project is about editing video for the user. This can be editing video for user wish. This runs in python.

The video can be edited by using the moviepy module. Whenever the user enters the video, the particular Video will be edited in this application.

By using the audio extracting, video merging function that is available in the moviepy module, each and every video will be edited in this application.

This video modify from old video which is given by user. The overall modules used in this project are moviepy. This is created new modify video using old video. Thus, the video is edited for each user.

TABLE OF CONTENTS

ABSTRACT

1 INTRODUCTION

- 1.1 PYTHON LANGUAGE INTRODUCTION
- 1.2 HISTORY OF PYTHON
- 1.3 PYTHON FEATURES
- 1.4 PYTHON MODULES

2 SYSTEM IMPLEMENTATION

- 2.1 TECHNOLOGIES USED
- 2.2 LANGUAGE USED
- 2.3 REQUIREMENTS
- 2.4 SUMMARY

3 CONCLUSION AND FUTURE WORK

- 3.1 SUMMARY
- 3.2 FUTURE WORK

REFERENCES

APPENDIX

- A.SOURCE CODE
- B.OUTPUT

1.INTRODUCTION

1.1 PYTHON LANGUAGE INTRODUCTION:

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation.

It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable.

It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions.

Python is Interpreted

Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive

You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented –**

Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language –**

Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

1.2 HISTORY OF PYTHON:

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

1.3 PYTHON FEATURES:

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP

- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java

1.4 PYTHON MODULES :

MOVIEPY

This Module is a video editing Module that is simple to use and written in pure python.

The Module can automate most of the building process for editing videos.

The Module edit the video by import Module from moviepy.editor package.

Syntax :: from moviepy.editor import * .

To install moviepy Module the following must be typed in command prompt.

--pip install moviepy.

SYSTEM IMPLEMENTATION

2.1 TECHNOLOGIES USED:

PYTHON 3.6
Moviepy

3.2 LANGUAGES USED:

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure.^[4] This feature is also sometimes termed the off-side rule.

3.3 REQUIREMENTS:

OPERATING SYSTEM:Window

RAM:4GB

MAXIMUM 2GB DATA

3.4 SUMMARY:

Python can serve as a scripting language for web applications, e.g, via Apache web server With Web Server Gateway Interface, a standard API has evolved to facilitate these applications.

Web frameworks like Django, Pylons, Pyramid, TurboGears, web2py, Tornado support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications. SQLAlchemy can be used as data mapper to a relational database.

Libraries such as NumPy, SciPy and allow the effective use of Python in scientific computing and provides domain-specific functionality.

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D animation packages such as 3ds Max, Cinema 4D, Light wave, Motion Builder, Soft image and musical notation programs like score writer and capella. It has also been used in several video games and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go. Python is also used in algorithmic trading and quantitative finance. Python can also be implemented in APIs of online brokerages that run on other languages by using wrappers.

4.CONCLUSION

4.1 SUMMARY:

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.

Other shells, including IDLE and IPython, add further abilities such as auto-completion, session state retention and syntax highlighting.

As well as standard desktop integrated development environments (see Wikipedia's "Python IDE" article), there are Web browser-based IDEs; SageMath (intended for developing science and math-related Python programs); PythonAnywhere, a browser-based IDE and hosting environment; and Canopy IDE, a commercial Python IDE emphasizing scientific computing.

4.2 FUTURE WORK:

This project has really been faithful and informative. It has made us learn and understand the many trivial concepts of Python Language. As we have used python Tkinter as a GUI it provides various controls, such as buttons, labels and text boxes to build a userfriendly application and IntelliJ IDE as a platform of carrying codes.

The fast growing use of internet confirms the good future and scope of the proposed project.

Finally I learnt a valuable lifelong lesson about the improvements and working on python .

REFERENCES

www.google.com

www.python.org

APPENDIX

A. SOURCE CODE

```
from moviepy.editor

import *
choice = int(input("Enter 1 for audio_extract 2 for video merging"))

if choice==1:

    video_mp4 = input("Enter video files's name")

    audio_mp3 = input("Enter file name for the convertor audio (without extension)")
```

```
clip = VideoFileClip(video_mp4)
audio = clip.audio
audio.write_audiofile(audio_mp3+".mp3")
clip.close()
audio.close()
elif choice==2:
    input1 = input("video 1")
    input2 = input("video 2")
    video_1 = VideoFileClip(input1)
    video_2 = VideoFileClip(input2)
    final_video = concatenate_videoclips([video_1,video_2])
    final_video.write_videofile("final_video.mp4")
else:
    print("Enter the correct choice")
```

OUTPUT:

INPUT FOR AUDIO EXTRACTING:

```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help

Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

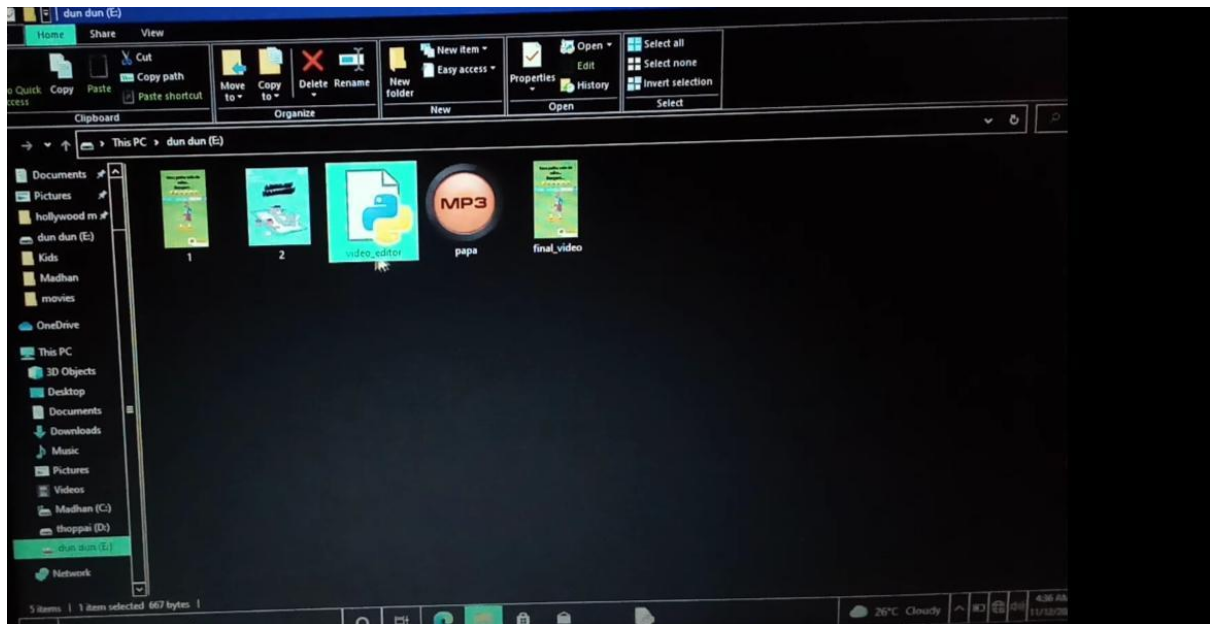
===== RESTART: E:\video to audio.py =====
Enter video files's name:bala.mp4
Enter file name for the convertor audio (without extension)bha
MoviePy - Writing audio in bha.mp3
chunk: 0% | 0/268 [00:00<?, 71t/s, now=None]chunk: 1% | 3/268 [00:01<02:38, 1.67it/s, now=None]chunk: 20% | 54/268 [00:01<00:05, 38
.84it/s, now=None]chunk: 38% | 102/268 [00:02<00:02, 76.50it/s, now=None]chunk: 56% | 149/268 [00:02<00:00, 115.97it/s, now=None]chunk: 71% |
190/268 [00:02<00:00, 151.60it/s, now=None]chunk: 85% | 228/268 [00:02<00:00, 176.33it/s, now=None]chunk: 100% | 268/268 [00:02<00:00, 206
.40it/s, now=None]
>>> |
MoviePy - Done.
```

```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help

Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

===== RESTART: E:\video_editor.py =====
Enter your choice1
Enter video files's name:bala.mp4
Enter file name for the convertor audio (without extension)bbhh
MoviePy - Writing audio in bbhh.mp3
chunk: 0% | 0/536 [00:00<?, 71t/s, now=None]chunk: 1% | 2/536 [00:01<07:14, 1.17it/s, now=None]chunk: 5% | 28/536 [00:01<00:24, 21
.05it/s, now=None]chunk: 9% | 48/536 [00:01<00:12, 38.39it/s, now=None]chunk: 12% | 65/536 [00:02<00:08, 52.93it/s, now=None]chunk: 15% | 82/536 [00:02<00:06, 67.50it/s, now=None]chunk: 24% | 128/536 [00:03<00:03, 132.64it/s, now=None]chunk: 35% | 189/536 [00:03<00:02, 180.28it
/s, now=None]chunk: 39% | 204/536 [00:03<00:01, 211.85it/s, now=None]chunk: 48% | 259/536 [00:03<00:01, 235.68it/s, now=None]chunk: 50% | 268/536 [00:03<00:01, 243.68it/s, now=None]chunk: 75% | 402/536 [00:03<00:00, 157.07it/s, now=None]chunk: 87% | 468/536 [00:04<00:00, 127.31it/s, now=None]chunk: 96% | 517/536 [00:04<00:00, 95.15it/s, now=None]chunk: 99% | 530/536 [00:04<00:00, 88.63it
/s, now=None]
>>> 2
>>>
MoviePy - Done.

===== RESTART: E:\video_editor.py =====
Enter your choice2
video lbala.mp4
MoviePy - Building video final_video.mp4.
MoviePy - Writing audio in final_videoTEMP_MPY_wvf_snd.mp3
chunk: 0% | 0/729 [00:00<?, 71t/s, now=None]chunk: 0% | 2/729 [00:01<11:12, 1.08it/s, now=None]chunk: 0% | 3/729 [00:02<07:23, 1.46it/s, n
ow=None]chunk: 1% | 4/729 [00:02<05:15, 2.30it/s, now=None]chunk: 5% | 36/729 [00:02<00:15, 36.57it/s, now=None]chunk: 7% | 51/729 [00:02<0
0:14, 45.46it/s, now=None]chunk: 9% | 64/729 [00:02<00:13, 50.27it/s, now=None]chunk: 10% | 75/729 [00:02<00:12, 52.30it/s, now=None]chunk: 12% | 87/729 [00:03<00:12, 51.94it/s, now=None]chunk: 13% | 97/729 [00:03<00:10, 55.93it/s, now=None]chunk: 15% | 110/729 [00:03<00:10, 55.93it/s, now=None]chunk: 16% | 120/729 [00:03<00:10, 59.43it/s, now=None]chunk: 18% | 134/729 [00:03<00:09, 63.27it/s, now=None]chunk: 21% | 154/729 [00:04<00:08, 67.16it/s, now=None]chunk: 25% | 185/729 [00:04<00:07, 72.52it/s, now=None]chunk: 27% | 199/729 [00:04<00:06, 83.91it/s, now=None]chunk: 2
>>>
MoviePy - Writing video final_video.mp4
>>>
MoviePy - Done.
```

Thank you...