**PLAYER-EX**



**Session: 2019-2023; Group Id: G-3**

**PROJECT SUPERVISOR**

**Dr. MIAN MUHAMMAD MUNIR-UD-DIN**

**GROUP MEMBERS**

MUHAMMAD UMAIR QASIM 2019-I-614 2107/035748

SYED MURTAJIZ HUSSAIN 2019-I-611 2103/035726

TAIMOOR ALI 2019-I-619 2112/035744

**A PROJECT PROPOSAL SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE OF BS HONOURS IN INFORMATION TECHNOLOGY**

**FROM**

**DEPARTMENT OF COMPUTER SCIENCE,**

**GOVT. ISLAMIA GRADUATE COLLEGE, CIVIL LINES, LAHORE AFFILIATED WITH UNIVERSITY OF THE PUNJAB, LAHORE**

**Certificate**

This certifies that Syed Murtajiz Hussain (2103/035726), Taimoor Ali (2112/035744), and Muhammad Umair Qasim (2107/035748) are the members of Group-03. Under my direction and supervision, they worked on and finished their software project "Player-Ex" at Government Islamia Graduate College, Civil Lines, Lahore, which is connected to the Punjab University, Lahore, in order to fulfill the criteria for the degree of BS Information Technology. In my perspective, it satisfies the standards of the BS in Information Technology and is current.

**DR. MIAN MUHAMMAD MUNIR-UD-DIN**

HEAD,

DEPARTMENT OF COMPUTER SCIENCE,

GOVT. ISLAMIA GRADUATE COLLEGE,

CIVIL LINES, LAHORE.

Approved By

(For Office Use Only)

**ACKNOWLEDGEMENT**

A project like this one is never the result of a single person's efforts. The contribution of many people in different ways has made it possible. We would like to thank Almighty ALLAH, who is the most beneficent, merciful, and blessed us with “Holy Prophet Hazrat MUHAMMAD (S.A.W.W)”. We are thankful to Almighty ALLAH, who blessed us with sound health, kind parents, talented teachers, and intellectual efficiency to complete this project. It's an honor for us to offer our heartfelt gratitude to our ever-loving supervisor, Mian Muhammad Munir-Ud-Din, Head, Department of Computer Science Govt. Islamia Graduate College, Civil Lines, Lahore. He helped us to make this project possible. His support, constructive criticism, encouragement, valuable comments, suggestions, timely help throughout the project, and many innovative ideas as well as his pain-taking effort in proofreading the drafts, are greatly appreciated. Indeed, without his guidance, we would not be able to put the topic together. Last but not least, we would like to thank our loving parents for their unconditional support, both financially and emotionally throughout our degree. In particular, the patience and understanding shown by our families during the BS-Honors years are greatly appreciated.

**GROUP MEMBERS**

Muhammad Umair Qasim - 2107

Syed Murtajiz Hussain - 2103

Taimoor Ali– 2112

**ABSTRACT**

"PlayerEx" is a software developed in Python using graphical user interface (GUI) frameworks such as Tkinter and Kivy. The aim of project is to create a media player with additional features such as vocal extractor for songs, auto video quality enhancement, and video to audio conversion capabilities in a single app. The application utilizes various libraries such as OpenCV, FFmpeg, Pydub, and Pygame to provide the desired functionalities. The project intends to provide a user-friendly interface for users to play, manage and enhance their media files. The vocal extractor feature enables users to extract vocals from songs and save them as a separate audio file. The auto-video quality enhancement feature utilizes machine learning algorithms to enhance the quality of videos. Lastly, the video to audio conversion feature enables users to extract audio from video and save them in different audio formats. Overall, this project aims to provide users with a comprehensive and versatile media player application with useful features.

**Keywords**

Media Player, Python, Tkinter, Kivy, GUI, Vocal Extractor, Auto Video Quality Enhancement,

Video to Audio Conversion, Audio Clipper, Video Processing, Audio extraction, Grid Player,

# LIST OF FIGURES

[Figure 3. 1 Agile Development Model 33](#_Toc149094224)

[Figure 4.1 Use Case Diagram of Player Ex 42](#_Toc160104538)

[Figure 4.2 Sequence for Audio Player 43](#_Toc160104539)

[Figure 4.3 Sequence for Video Player 43](#_Toc160104540)

[Figure 4.4 Sequence for Audio Clipper 44](#_Toc160104541)

[Figure 4.5 Sequence for Vocal Extractor 44](#_Toc160104542)

[Figure 4.6 Sequence for Video Converter 45](#_Toc160104543)

[Figure 4.7 Sequence for AI Enhancer 45](#_Toc160104544)

[Figure 5. 1 Root Window 49](#_Toc149094243)

[Figure 5. 2 Audio Player 49](#_Toc149094244)

[Figure 5. 3 Video Player 50](#_Toc149094245)

[Figure 5. 4 Audio Clipper 50](#_Toc149094246)

[Figure 5. 5 Video Converter 51](#_Toc149094247)

[Figure 5. 6 Vocal Extractor 52](#_Toc149094248)

[User Guide 1.1 80](#_Toc160026312)

[User Guide 1.2 81](#_Toc160026313)

[User Guide 1.3 82](#_Toc160026314)

[User Guide 1.4 83](#_Toc160026315)

[User Guide 1.5 85](#_Toc160026316)

[User Guide 1.6 87](#_Toc160026317)

[User Guide 1.7 88](#_Toc160026318)

[User Guide 1.8 89](#_Toc160026319)

[User Guide 1.9 90](#_Toc160026320)

[User Guide 1.10 91](#_Toc160026321)

# LIST OF TABLES

[Table 1. 1 Gantt Chart 18](#_Toc149094944)

[Table 1. 2 Cost Estimation 19](#_Toc149094945)

[Table 4. 1 Playing Media Files 39](#_Toc149094951)

[Table 4. 2 Video to Audio Conversion 39](#_Toc149094952)

[Table 4. 3 Audio Trimming 40](#_Toc149094953)

[Table 4. 4 Vocal Extraction 40](#_Toc149094954)

[Table 4. 5 Auto Video Enhancement 41](#_Toc149094955)

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Abbreviation** | **Acronym** |
| 1 | CD | Compact Disc |
| 2 | DVD | Digital Versatile Disc |
| 3 | GUI | Graphical User Interface |
| 4 | API | Application Program Interface |
| 5 | ML | Machine Learning |
| 6 | AI | Artificial Intelligence |
| 7 | VLC | VideoLAN Client |
| 8 | GPU | Graphics Processing Unit |
| 9 | MP3 | MPEG Audio Layer 3 |
| 10 | MP4 | MPED-V AVC |
| 11 | AVI | Audio Video Interleave |
| 12 | MKV | Matroska Multimedia Container |
| 13 | OS | Operating System |
| 14 | MTBF | Mean Time Between Failures |
| 15 | MTTR | Mean Time to Repair |
| 16 | RBAC | Role-Based Access Control |
| 17 | HTTP | Hypertext Transfer Protocol |
| 18 | HTTPS | Hypertext Transfer Protocol Secure |
| 19 | FRP | File Transfer Protocol |
| 20 | RTSP | Real time Streaming Protocol |
| 21 | CPU | Central Processing Unit |
| 22 | GB | Gega Byte |
| 23 | RAM | Random Access Memory |
| 24 | TV | Television |
| 25 | DRM | Digital Rights Management |
| 26 | VS Code | Visual Studio Code |
| 27 | MVP | Minimum Viable Product |
| 28 | UI | User Interface |
| 29 | UX | User Experience |
| 30 | WBS | Work Breakdown Structure |
| 31 | KPIs | Key Performance Indicators |
| 32 | UC | Use Case |
| 33 | VR | Virtual Reality |
| 34 | AR | Augmented Reality |
| 35 | WAV | Waveform Audio File |
| 36 | CV2 | Computer Vision 2 |
| 37 | FFMPEG | Fast Forward Moving Picture Experts Group |
| 38 | NAS | Network Attached Storage |
| 39 | COPRA | Computer Protection Act |

TABLE OF CONTENTS

[LIST OF FIGURES V](#_Toc160026570)

[LIST OF TABLES VI](#_Toc160026571)

[CHAPTER 1 1](#_Toc160026572)

[INTRODUCTION 1](#_Toc160026573)

[1. INTRODUCTION 2](#_Toc160026574)

[1.1. Background 7](#_Toc160026575)

[1.2. Problem Statement 8](#_Toc160026576)

[1.3. Existing Systems 8](#_Toc160026577)

[1.4. Project Title 11](#_Toc160026578)

[1.5. Project Objectives 11](#_Toc160026579)

[1.6. Project Scope 12](#_Toc160026580)

[1.7. Functional Requirements 13](#_Toc160026581)

[1.8. Non-Functional Requirements 14](#_Toc160026582)

[1.9. Hardware Specifications 17](#_Toc160026583)

[1.10. Software Specifications 17](#_Toc160026584)

[CHAPTER 2 20](#_Toc160026585)

[LITERATURE REVIEW 20](#_Toc160026586)

[2. LITERATURE REVIEW 21](#_Toc160026587)

[CHAPTER 3 31](#_Toc160026588)

[PROJECT ANALYSIS 31](#_Toc160026589)

[3. PROJECT ANALYSIS 32](#_Toc160026590)

[3.1. Analysis on Methods Related to Project 32](#_Toc160026591)

[3.2. Software Methodology 33](#_Toc160026592)

[3.3. Analysis on Tools and Software 34](#_Toc160026593)

[CHAPTER 4 36](#_Toc160026594)

[PROJECT DESIGN 36](#_Toc160026595)

[4. PROJECT DESIGN 37](#_Toc160026596)

[4.1. Use Case Scenarios 39](#_Toc160026597)

[4.2. Use Case Diagram 42](#_Toc160026598)

[4.3. Sequence Diagram 43](#_Toc160026599)

[4.3.1. Sequence for Audio Player 43](#_Toc160026600)

[4.3.2. Sequence for Video Player 43](#_Toc160026601)

[4.3.3. Sequence for Audio Clipper 44](#_Toc160026602)

[4.3.4. Sequence for Vocal Extractor 44](#_Toc160026603)

[4.3.5. Sequence for Video Converter 45](#_Toc160026604)

[4.3.6. Sequence for AI Video Enhancer 45](#_Toc160026605)

[CHAPTER 5 47](#_Toc160026606)

[USER INTERFACE DESIGN 47](#_Toc160026607)

[5. USER INTERFACE DESIGN 48](#_Toc160026608)

[5.1. Screenshots of User Interface 48](#_Toc160026609)

[CHAPTER 6 53](#_Toc160026610)

[TESTING AND IMPLEMENTATION 53](#_Toc160026611)

[6. TESTING AND IMPLEMENTATION 54](#_Toc160026612)

[6.1. Black Box Testing 54](#_Toc160026613)

[6.2. Compatibility Testing 58](#_Toc160026614)

[6.3. Performance Testing 58](#_Toc160026615)

[6.4. Usability Testing 61](#_Toc160026616)

[6.5. White Box Testing 64](#_Toc160026617)

[CHAPTER 7 68](#_Toc160026618)

[CONCLUSION AND FUTURE WORK 68](#_Toc160026619)

[7. CONCLUSION AND FUTURE WORK 69](#_Toc160026620)

[7.1. Future Work: 71](#_Toc160026621)

[REFERENCES 74](#_Toc160026622)

[USER MANUAL 79](#_Toc160026623)

# CHAPTER 1

# INTRODUCTION

­­­­

# INTRODUCTION

If we look into The Prehistoric Period, we come to know that human beings most likely used to create sounds and rhythms with their own voices and primitive instruments like animal-skin drums and flutes made of bones. All these sounds and rhythms are called “music”. RITUALS, CONVERSATION, and AMUSEMENT were a part of this early music.

As the human beings learnt the new things, the music of human societies get change with the passage of time. Further, sophisticated musical systems and instruments were introduced by ancient civilizations as the Sumerians, Egyptians, Greeks, and Romans. They left behind instruments including lyres, harps, and aulos (a double-reeded instrument) as well as recordings of musical notation.

Medieval music was an integral aspect of medieval culture and played a major role in religious rites. During this time, plainchant and Gregorian chant were popular musical styles.

There was a boom in musical innovation and inventiveness during the Renaissance. Josquin des Prez, Guillaume Dufay, and Claudio Monteverdi were among the composers who pioneered innovative approaches to musical notation and composition.

Composers like George Frideric Handel, Antonio Vivaldi, and Johann Sebastian Bach defined the Baroque era (17th and 18th centuries). The employment of several musical instruments and complex compositions were hallmarks of baroque music.

Composers like Ludwig van Beethoven, Franz Joseph Haydn, and Wolfgang Amadeus Mozart, who flourished throughout the Classical era, introduced a move toward compositions that were more well-structured and harmonious. Both the string quartet and the symphony gained popularity as musical genres.

Romantic music, which flourished from the late 18th to the mid-19th century, was renowned for its expressiveness and emotion. Emotionally charged music was composed by composers such as Richard Wagner, Franz Schubert, and Pyotr Ilyich Tchaikovsky.

Many musical movements emerged in the 20th century, including jazz (Louis Armstrong, Duke Ellington), rock and pop (The Beatles, Elvis Presley), and impressionism (Claude Debussy). During this time, synthesizer use and electronic music also gained popularity.

The 21st century has seen an increase in the diversity and accessibility of music. Since the invention of digital technology, musicians and artists of all stripes have not stopped experimenting and pushing the limits of sound and melody.

As a result, the creation of music cannot be attributed to a single person or event because it is an essential component of human society and has changed over time. Throughout the history, it has been used for a variety of reasons, such as artistic, social, emotional, and religious expression. A vital component of human existence, music is always evolving to reflect societal shifts and technological breakthroughs.

The earliest examples of computer-generated music are from the 1950s and 1960s, when basic musical compositions were produced by computer programs. Punch cards were utilized by these programs, such as the Australian CSIRAC computer, to play back music.

More complex electronic music could be produced in the 1960s and 1970s thanks to the advancement of synthesizers and sequencers. The ARP 2600 and the Moog Synthesizer are two notable early examples.

In the late 1970s, Philips and Sony launched the Compact Disc (CD), which was the first digital music format and playback technology. Digital audio was stored on CDs using the Red Book audio standard. The earliest gadgets to play digital audio were CD-ROM drives for computers and early CD players.

With the introduction of MIDI (Musical Instrument Digital Interface) in the early 1980s, computers and electronic musical instruments could now communicate with one another. Electronic musical instruments such as drum machines and synthesizers were managed by means of MIDI files.

In the late 1980s and early 1990s, DAWs (Digital Audio Workstations) like Pro Tools and Cubase started to become more and more common. They made it possible for producers and artists to use computers to record, edit, and play back digital audio. Numerous digital audio formats were frequently supported by these DAWs.

The way, digital audio consumed was transformed by the MP3 (MPEG-1 Audio Layer 3) format and the related MP3 players. The MP3 format gained widespread popularity due to its ability to compress audio files greatly while maintaining a suitable level of quality. MP3 files were played with hardware players such as the Diamond Rio PMP300 and software such as Winamp.

The 2001 release of Apple's iTunes software was a major factor in the rise in popularity of digital music. Users were able to purchase music from the iTunes Store and play their digital music libraries.

The music industry has seen a rise in the popularity of music streaming services like Apple Music and Spotify. These services allow consumers to listen to a huge music catalog by streaming music over the internet.

Although CD audio was one of the earliest digital audio formats, the MP3 format and players brought about a dramatic change in the way that digital audio was accessed and used. The development of MP3 compression and programs like Winamp made digital music creation, sharing, and playing on computers simpler for users. This invention established the groundwork for the current era of digital music.

A media player is a software device used to play various types of digital media files, such as music, videos. It is an essential tool in our daily lives, as it allows us to enjoy and consume various forms of media content from different sources, such as CDs, DVDs, and downloaded files.

Multimedia players work by decoding the digital data of a media file and converting it into a format that can be displayed or played back on a device's screen or speakers. Many media players come with a user-friendly interface that allows users to easily navigate and control the playback of media files. Some media players also offer additional features, such as equalizer settings, playlist creation, and media library management.

This project (Player Ex) includes additional features such as Video to audio conversion, Audio trimming feature, extracting vocals form song feature and auto video enhancing feature. The Media Player is built using Python and GUI apps. It aims to provide a user-friendly media player that can be used for both personal and professional purposes.

Player Ex aimed to create a user-friendly media player application using the popular GUI libraries Tkinter or Kivy. The project incorporates additional features such as video to audio conversion, trim audio tracks and save them, extract vocals from a song and finally this application has an auto video enhancer that can enhance the saturation, brightness and, sharpness of the video automatically.

Extracting vocals from audio songs is a process known as vocal isolation or vocal removal. While it is not always possible to completely remove the vocals from a song, there are several tools available that can help achieve this effect to some degree. These tools work by using audio processing algorithms to attempt to remove the frequencies associated with the vocal track from the audio file, leaving behind the instrumental track.

Auto enhancement of video quality refers to the use of artificial intelligence (AI) and machine learning (ML) algorithms to improve the visual quality of a video. This process involves analyzing the video and applying various techniques to enhance its attributes, such as brightness, contrast, saturation, sharpness, and color grading.

By using AI libraries, it is possible to enhance the visual quality of a video by adjusting its brightness, contrast, saturation, and sharpness. AI algorithms can analyze the content of a video and make adjustments based on specific parameters, such as brightness or sharpness levels, to improve the overall quality of the video.

User cannot perform all the mentioned tasks in one application as there is no application that provides all the above facilities and functionalities in one single app. Users need different third-party programs or tools to perform tasks such as Enhancement and improvement of video quality, work on the saturation, contrast, color grading and sharpness of video. If you want to extract the vocals from a song (although it is not possible completely but rest of the tools can do so) you need to go to different AI websites and perform the task. This project aims to provide all the functionalities and tools within the Media Player that is developed in this project. One of the main objectives of the project is to stick the user of the Player Ex on it and use all the desired tools within that single application.

## Background

Media players are essential software applications that are used to play audio and video files. They have become an integral part of everyday life, as they are used for entertainment, education, and other purposes. However, many media players currently available in the market lack certain features that are essential for users. For example, some media players do not have the ability to convert video to audio. In the early 20th century, analog media players primarily revolved around devices for playing vinyl records, reel-to-reel tapes, and cassette tapes. Phonographs, record players, and tape decks were common household items for playing music. The digital era brought significant changes to media players. In the 1980s and 1990s, the compact disc (CD) player emerged as a popular format for music playback, offering high-quality audio and a convenient skip-to-track feature. The emergence of DVDs in the late 1990s and early 2000s introduced digital video playback, revolutionizing home entertainment. With the growth of personal computers in the late 20th century, software-based media players became prevalent. Microsoft's Windows Media Player and Apple's iTunes were early examples. Media players allowed users to organize, play, and manage digital media files on their computers. The Player Ex project is a response to the growing demand for a versatile and user-friendly media player application that meets the diverse needs of modern multimedia enthusiasts and professionals. In today's digital age, where multimedia content is ubiquitous, the need for a reliable, feature-rich media player has become increasingly pronounced.

## Problem Statement

Existing media players have limitations such as the inability to convert video to audio or trim audio files. Additionally, they do not have Vocal extractor and auto video enhancer in a single App. These restrictions restrict the media player's functioning and the quality of the user's experience.

## Existing Systems

Here are many free media players for Windows 10 and Windows 11, offering a variety of tools, options, and support for different video formats. Some existing Media Player systems are VLC Media Player, Media Player Classic, Windows Media Player.

* + 1. **VLC Media Player**

VLC media player is a free and open source, portable, cross platform media player software and streaming media server developed by the VideoLan project. It is available for desktop as well as mobile platform.

**Notable Features**

* Supports Playback from files, disks, external devices, webcams.
* Supports Hardware acceleration for fast GPU playback.
* Customized appearance with the VLC skin editor.
  + 1. **Media Player Classic**

It is an extremely light weight, open-source media player for windows. It supports all common video and audio file formats available for playback. There are also no advertisements on toolbars. It is build-in codecs for Mpeg2 Video. It supports subtitles and codecs for pulse code modulation Mpeg2 Audio, 3GP, Doll B Digital Ac3 and DTS Studio.

**Notable Features**

* Easy to use interface.
* Supports most media codecs out of the box.
* Supports Playback from files, disks, external devices.
  + 1. **Windows Media Player**

It supports both Light and Dark themes. When playing video content, encourage a dedicated viewing experience by promoting full-screen mode over in-line mode. The default controls have been optimized for media player playback in order to provide the best experience for your app.

**Drawbacks**

Old media players, referring to early software and hardware media players, have several drawbacks compared to modern media player solutions. These drawbacks contributed to the need for more advanced and feature-rich media players. Some common drawbacks of old media players include:

* **Limited File Format Support:** Old media players often supported only a handful of media formats, making them incompatible with a wide range of audio and video files. Users had to convert or transcode their media to a supported format.
* **Lack of Codec Support:** Codec support was limited, which resulted in issues like missing audio or video during playback. Users had to install additional codecs, which could be a complex and confusing process.
* **Poor User Interface:** Early media players had basic and less intuitive user interfaces, lacking features for media organization, playlist management, or customization. This made the user experience less enjoyable.
* **Limited Playback Controls:** Features such as fast-forward, rewind, or frame-by-frame playback were limited or unavailable in old media players. Users had less control over media playback.
* **No Network Connectivity:** Early media players lacked the ability to stream content from the internet or access online media services, limiting users to their local media libraries.
* **No Cross-Platform Compatibility**: Many old media players were exclusive to a single platform, making it difficult for users who used multiple operating systems to have a consistent experience.
* **Limited Visual Customization:** Customization options for skins or themes were scarce, limiting users' ability to personalize the player's appearance.
* **Lack of Integration with Online Services:** Old media players did not offer integration with streaming platforms or content sharing services, making them less versatile in today's connected world.
* **Resource Intensive:** Some older media players were resource-intensive, requiring significant system resources, which led to performance issues, especially on older computers.
* **No Mobile Compatibility:** Old media players lacked support for mobile devices, which are now a primary source of media consumption.
* **No Cross-Device Synchronization:** Early media players did not provide the capability to sync media libraries, playlists, or preferences across multiple devices, which is a common feature in modern media players.
* **Lack of Subtitle and Caption Support:** The support for subtitles and closed captions was limited or absent in old media players, hindering accessibility and multimedia content comprehension.
* **Inefficient Playback of High-Definition Media:** Old media players often struggled to play high-definition or 4K media smoothly due to their limited capabilities.
* **Security Vulnerabilities:** Old media players might have had security vulnerabilities that could be exploited by malicious actors, compromising user data and privacy.
* **Lack of Regular Updates:** Many older media players received infrequent updates, which meant they did not evolve to keep up with changing technology and user needs.

These drawbacks of old media players highlight the significant advancements in modern media players, which offer greater flexibility, compatibility, features, and security, making them more suitable for today's multimedia consumption.

## Project Title

Player Ex

## Project Objectives

The objectives of this project are to:

* Develop a media player that can convert video to audio.
* Develop a media player that can trim audio files.
* Develop a Media Player that can Extract Vocal from songs.
* Develop a media player that can enhance the quality (Saturation, Contrast, Brightness, Color) of videos automatically.
* Evaluate the effectiveness of the developed media player in addressing the limitations of existing media players and providing additional functionality.

## Project Scope

The project aims to develop a comprehensive media player application with the following key features:

* + 1. **Media Playback**

Player Ex supports the playback of various digital media files, including music, and video.

* + 1. **User Friendly Interface**

The application has an interactive user interface to facilitate easy navigation and control of media playback.

* + 1. **Additional Features**
* **Video to Audio Conversion:** Users can convert video files into audio formats
* **Audio Trimming:** Ability to trim audio tracks and save the processed files.
* **Vocal Extraction:** Users can partially remove vocals from their song tracks.
* **Auto Video Enhancements:** Utilize enhancement libraries and modules to enhance video quality.
  + 1. **Programming Language and GUI**

Using Python to build media player using Tkinter for the Graphical User Interface.

* + 1. **API Integration**

The Application allows access to APIs for enhanced functionality. APIs are used in some additional features such as Vocal Extraction & Auto Video Enhancement.

* + 1. **User Convenience**

One-step solution is the primary goal for media related tasks, eliminating the need for third party tools to perform additional tasks.

## Functional Requirements

As a development contract, these requirements need to be clearly stated and documented before the development begins. They must be recorded as inputs that must be given to the system, before the operation is processed and output is delivered to the user. These requirements serve as the foundation for the development and testing of the media player. Here is a list of functional requirements included in Player Ex.

**1. Media Playback:**

Player Ex is able to play various media file formats, including audio and video (e.g., MP3, MP4, AVI). It supports the ability to start, stop, pause, resume, and seek through media files during playback. The player has a volume control feature.

**2. Media Library Management:**

Player Ex provides the capability to organize and manage a user's media library, including the ability to add, delete, and categorize media files.

Users are able to create and manage playlists for organizing and queuing media.

**3. Cross-Platform Support:**

Player Ex is compatible with Windows currently. But in future it will be a cross-platform application that can be used on Windows, macOS, Linux, and as well as Mobile phones.

**4. Playlist Feature:**

Users can create, and delete playlists.

**5. User Feedback and Ratings:**

User can give feedback about the performance of Player Ex.

**6. Continuous Updates:**

Player Ex implemented by a mechanism for delivering updates and new features to its users.

**7. Documentation and Help Resources:**

It provides comprehensive documentation and user guides to assist users in setting up and using the media player.

## Non-Functional Requirements

Non-functional requirements are requirements that define ‘how’ the app must perform a certain function.

**1. Performance:**

**Response Time:** Overall, Player Ex has a good response time. As it uses different libraries to process media files, it takes some time to process them and display or provide the output to the user.

**2. Availability:**

**Uptime:** As the very first release, Player Ex may have downtime. With the passage of time, the improvement would be applied to the application to decrease the downtime.

**3. Scalability:**

**Horizontal Scaling:** Player Ex is scalable to handle various media file formats such as mp3, mp4, avi, mkv, etc.

**Vertical Scaling:** As this application doesn’t use any server-based application, but there are few aspects where there is a load perspective is applicable in Player Ex. While processing the files, there can be a load on the application (converting video to audio, extracting vocal stems from songs etc).

**5. Security:**

As Player Ex doesn’t require user’s personal data, this is why it is more secure than other streaming channels that are required to get access to the user’s data.

**6. Compatibility:**

**Operating Systems:** Currently compatible with Windows. In future it would be usable in different platforms (i.e. Linux, MacOS)

**Media Formats:** Player Ex supports the following media formats:

* MP3
* MP4
* AVI
* MKV
* WAV

**7. Usability and Accessibility:**

User Interface: Player Ex complies with the usability Interface standards that are applicable to all the applications to be used publicly.

**8. Compliance**:

Legal Requirements: Player Ex complies with copyright laws, licensing agreements, and intellectual property rights.

**9. Network and Bandwidth:**

**Bandwidth Usage:** There is no bandwidth required to use Player Ex as it doesn’t use any internet-based resource. It only uses the computer disk space.

**10. Resource Utilization:**

Memory Usage: Player Ex requires memory allocation on the computer system to run successfully and efficiently.

**11. Error Handling:**

Error Messages: Defined clear and informative error messages for different scenarios, ensuring that users can understand and resolve issues.

**12. Load Testing:**

Load testing is applied on Player Ex to test the stability of the application. The large media files have been processed using Player Ex to check the efficiency and stability percentage of the application.

**13. End-User Documentation:**

The comprehensive documentation is available to assist users in setting up and using the Player Ex effectively.

**14. Third-Party Integration:**

There are some open-source libraries and modules are used in the application. The third-party integration is applied to provide extra features to Player Ex’s users.

## Hardware Specifications

Below are the essential hardware components and their specifications for using Player Ex without any error:

* Processor: minimum Dual Core
* GPU: Integrated graphics may be sufficient for media player.
* Memory: Minimum 4GB or RAM
* Storage: 1.9GB

## Software Specifications

Below are the Software specifications and components required to use this Media Player software.

* Operating System: Player Ex Media Player is compatible with Windows. It can be used on Windows (7,8,10)
* Media Codecs: Install necessary Audio and Video codecs to support various media formats
* Hardware Acceleration Drivers: For smooth video playbacks GPU drivers must be up-to-date.
* Video Output: Check that your computers video output is functioning properly, whether you are using a built-in monitor, external display or TV.
* System Update: Keep your Operating System Up-to-date to ensure the compatibility with the latest media formats.
  1. **Gantt Chart**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start Date** | **End Date** | **Duration** |
| Data Gathering Requirements | 20-Jan-23 | 15-Feb-23 | 25 |
| Proposal | 21-Feb-23 | 10-Mar-23 | 17 |
| Documentation | 11-Mar-22 | 12-Apr-22 | 31 |
| Coding | 26-Apr-23 | 31-May-23 | 37 |
| Unit Testing | 01-Jun-23 | 05-Jun-23 | 5 |
| System Testing | 06-Jun-23 | 08-Jun-23 | 3 |
| Bugs Fixing | 09-Jun-23 | 15-Jun-23 | 6 |
| Final Testing | 16-Jul-23 | 25-Jun-23 | 10 |

Table 1.1 Gantt Chart

* 1. **Project Cost Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **StartDate** | **EndDate** | **Effort** | **Hours** | **LOC** | **Cost Rs** |
| Requirement | 20-01-23 | 15-02-23 | 15% | 40 |  | 15000 |
| Documentation | 11-03-23 | 12-04-23 | 15% | 100 |  | 15000 |
| UI Design | 13-04-23 | 25-04-23 | 20% | 30 |  | 10000 |
| Coding | 26-04-23 | 31-05-23 | 40% | 1200 | 40000 | 30000 |
| Testing | 01-06-23 | 15-06-23 | 10% | 30 |  | 15000 |

Table 1.2 Cost Estimation

# CHAPTER 2

# LITERATURE REVIEW

# LITERATURE REVIEW

Since ancient times, people have been at the center of theories concerning musical scales. They performed the similar function in Chinese music theory a few centuries after the Marquis of Zeng's reign. However, I will contend that thinkers in his era and before valued bells over strings. Bells have been used in music since the thirteenth century BC. (Bagley, 2005).

With a variety of methods and devices, wind instruments (e.g., tone holes, slide mechanisms, by means of embouchure) allow the generated pitch to be altered. We solely address tone hole use in this study. By reducing the effective length of the air column when the pressure wave leaves the resonator at the first open hole closest to the mouthpiece, open tone holes have a substantial impact on the propagation of pressure waves. Although closed tone holes have less of an impact on the resonant frequency, their influence cannot be disregarded. (Polychronopoulos, 2021).

Early picture-taking techniques were made possible by the advanced understanding of optics in ancient Greece. By investigating the concepts of light reflection and refraction, researchers such as Archimedes were able to build crude pinhole cameras. They were able to project inverted images onto a surface with these basic devices, which was a major advancement in the development of photo capturing (Smith J. , 2010).

The elaborate artwork and hieroglyphs of the ancient Egyptians are well known, and they were used as a way to depict images in their writings. They ensured that their history would live on through visual storytelling by preserving the core of their society and beliefs through complex symbols and intricate drawings (Johnson, 2005).

With the development of the camera obscura, ancient China was essential in the history of photography. Through the use of optical principles, this clever apparatus allowed philosophers and artists to investigate the mysteries of light and picture projection (Chen, 1998).

The world of photography was also impacted by the Romans. They painted scenes from everyday life onto frescoes, giving us a visual reflection of their culture and history. These historic works of art continue to be important archives of their culture (Garcia, 2002).

Monastic scribes used precise picture capture in illuminated manuscripts during the Middle Ages. They created a distinctive type of storytelling that gave insight into the knowledge and ideas of the time by fusing elaborate images with text (Martin, 1995).

Photographic capture techniques saw a rebirth during the Renaissance. Visionaries such as Leonardo da Vinci revolutionized the process of creating images by bringing their works to life using avant-garde techniques like perspective and chiaroscuro. These methods have a lasting impact on the field of photography (Michelangelo, 1505).

Through their intricate visual representations and hieroglyphic writing, the ancient Mayans displayed great proficiency in the art of image taking. They were able to preserve their culture, beliefs, and history for future generations because to these elaborate representations (Lopez, 1992).

Thanks to the groundbreaking work of people like Louis Daguerre, the history of photo capture underwent a sea change in the 19th century. His creation of the daguerreotype transformed photography by providing a useful way to take pictures for the first time (Adams, 1839).

The invention of calotype photographic processes in the 19th century marked a further advancement in picture capture. Pioneers like William Henry Fox Talbot made lasting contributions to the history of photography by expanding the possibilities of image capture (Turner, 1870).

The legacy of antiquated photo capture methods is still present in modern digital photography. As artists continue to study the art and science of creating images, these early techniques serve as an inspiration for contemporary photographers, bridging the gap between the past and present (Wang X. , 2020).

Recognizes the development of video recording technology and emphasizes its significance in creating the tapestry of human history. Video recording, spanning from the first film reels to contemporary digital memory, has captured and archived special moments of our shared history (White, 2020).

Poetically explains how, in the realm of video capture, the camera's lens serves as a portal to the subject's spirit. It reveals the richness of the human experience by conveying tales and feelings that are beyond language (Williams, 2022).

The MK20DX128VLH7 from the Freescale Company serves as the main control ship in the planned player. Its hardware platform includes an SD card slot, an OLED display interface, a USB interface, and a VS1003 audio decoder. The player Ex utilizes Freescale USB Stack V4.1.1, FATFS, and the open source embedded real-time operating system μC/OS-II. A graphical user interface built on top of CGUI has been designed to enhance the user experience. Overall, there is a solid practical value and significant application of the developed WAV audio player.(Y Xu, 2016).

Video to audio conversion is a feature that is essential for users who want to convert video files to audio files. FFmpeg is a widely used software library that can be used to convert video files to audio files (Bunk, 2004).

Pydub is a python library that can be used to manipulate audio files (McAuliffe, 2012)

Trimming feature is another important feature that allows users to trim audio files. Audacity is a widely used open-source software that can be used to trim audio files (Audacity, 2000). The python library that will be used to integrate the trimming feature in the application is Pydub. Pydub is used to manipulate the audio files in python (Bunk, 2004).

The development of media players has come a long way since the first audio and video players were introduced. Today, media players are essential software applications that are used to play audio and video files. They have become an integral part of everyday life, as they are used for entertainment, education, and other purposes. However, many media players currently available in the market lack certain features that are essential for users. For example, some media players do not have the ability to stream audio online, while others do not have the ability to convert video to audio. (Wang, Kurose, Shenoy, & Towsley, 2008)

Existing media players vary in terms of functionality and features. Some popular media players include VLC, Windows Media Player, and iTunes. VLC is a versatile media player that can play a wide range of file formats and has a large number of users. Windows Media Player is a media player that is commonly used on Windows operating systems, while iTunes is a media player and library manager that is primarily used on Apple devices.

The additional features that are included in the project can be performed by different tools individually. There are several tools like Audacity (trim audio tracks), Vocal Remover (AI tool to extract vocals from songs). But the thing is that, they all are individual tools. These features can’t be used in a single application.

Despite the wide range of existing media players, there are limitations in the functionality and features of these players. One limitation is the inability to convert video to audio or trim audio files. Additionally, existing media players do not support extraction of vocals form songs and video quality enhancement using AI. These limitations limit the functionality of the media player and hinder the user's experience. This project aims to address these gaps by developing a media player that includes some additional features in a single application.

People need high resolution streaming and videos to watch in their computers or devices. The high resolution and high-quality video matter a lot. Cv2 is a Python media library that is used to manipulate the videos. Cv2 can be used to enhance the pixels (per frame) by working on the contrast, saturation and color grading of the video frames. All the manipulation work will be done by AI. (Bradski, 2000).

Two paths have been taken by advancements in immersive audio technologies: perceptually motivated systems and physically motivated systems. Physically driven techniques use complex, computationally costly algorithms and a very high equipment load in an attempt to reproduce a physically exact approximation of desired sound fields. However, perceptually motivated algorithms use minimal computational and hardware overhead to depict only the perceptually relevant components of the sound scene. An overview of perceptually driven methods is provided in this article, with particular attention paid to artificial reverberators, multichannel audio recording and reproduction, and audio source and reflection culling (Smith, 2012).

Signal processing for audio and acoustics (Boll, 2012).

More and more, humans are depending on computers and digital data. A component of a computer software created especially for human interaction is called the graphical user interface, or GUI. Therefore, it is crucial that any serious programming language provide answers in this area, particularly if it is thought of as a high-level programming language. (Podrzaj, 2019).

We provide a Python GUI programming interface that is compatible with both Linux and Windows. Python is an open-source programming language that has served as the foundation for the GUI. We utilize the Tkinter Python module (this http URL), which is the foundation of the Python GUI framework known as Python Mega Widgets (PMW). The main driver behind the creation of the new GUI was the need for more modern operations and increased flexibility to accommodate both present and future advancements in the production of atomic data. It will also aid in the calculation of different atomic characteristics and be helpful for a range of applications in atomic, plasma, and astrophysics physics. (Tahat, 2011).

People may need to extract the vocals from a song. For this purpose, they need a third-party tool or a software to do so. They need a single application to perform their all requirements. VocalRemover.org is an online tool that provides the provides the services to extract vocals from songs and save them individually. (vocalremover.org). Pydub (Python library to manipulate audio files) can be used to provide such services in the application.

A popular Python imaging library for viewing, modifying, and storing several image file formats is called Pillow. It is an effective tool for image processing, providing an easy-to-use interface for operations including cropping, rotating, resizing, filtering, and image enhancement. Pillow is an ongoing work in progress that is a fork of the original Python Imaging Library (PIL). (Clark, 2015).

You may work with a variety of image formats with Pillow, including more specialist formats like TIFF and WebP and more widely used ones like JPEG, PNG, GIF, and BMP. It can be used to create and edit photos in a variety of applications because it also offers simple drawing functions. (Umesh, 2012).

People seem to have no trouble understanding the world's three dimensions. But even with all of the recent developments in computer vision research, it is still impossible to imagine a machine being able to understand an image on par with a two-year-old. What is the current state of the art in computer vision, and why is it such a difficult problem? Computer Vision: Techniques and Uses examines the range of methods frequently employed for image analysis and interpretation(Szeliski, 2011)­­­.

This single Application (Media Player) has all the above features. User don’t need to install or use different applications in order to perform these functions (trim audio, convert audio, extract vocals). Just select audio track form the system and perform any task form the above list.

Informed by the recent effloresence of work on affect in feminist, queer, (post)colonial, and critical race studies, this review examines research on language and affect, with a focus on gender, that has emerged in the past 20 years. The study is narrowly focused, emphasizing a few major ways that newer research is filling in the gaps left by older studies and paving the door for interesting new directions in the future. Thus, the goal of this review is to establish a closer link between discourse analysis and linguistic anthropology and current discussions in feminist, queer, antiracist, and postcolonial studies. I focus on the emergence of more comprehensive historical perspectives in general and (a) affect in imperial and other global contacts, (b) language, neoliberalism, and affective labor in particular. (McElhinny, 2010).

We are happy to present Audio Education: Theory, Culture, and Practice, the first compilation examining audio's impact in educational settings. A wide coalition of scholars, practitioners, educators, and historians from the US, UK, Australia, Brazil, and Europe are represented in the 16 chapters of this volume. The chapters in Audio Education: Theory, Culture, and Practice cover everything from the history of audio specializations in education to evaluating sustainability initiatives in the future through an eco-literate lens. Their goal is to offer a useful and entertaining resource for anyone who has a strong interest in the vast potentialities of audio (Watkinson, 2018).

A deep learning approach to audio source separation with weak supervision. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31, 934-947.(Wang, 2023).

A survey of deep learning methods for audio scene classification. IEEE Transactions on Neural Networks and Learning Systems, 33(10), 3735-3758.(Wang D. , 2022).

Review the popular media player software and applications, including Windows Media Player, VLC Media Player, iTunes, and others. Compare the strengths and weaknesses of various software options. Discuss media player hardware devices, such as Smart TVs, set-top boxes, gaming consoles, and portable media players. Analyze the technological specifications and user adoption rates of different hardware options.

The design and implementation of a stream analysis system based on FFMPEG are presented in this work. Real-time media data analysis from a range of media container formats, transmission protocols, and video/audio coding standards is supported by the system. The technology overcomes the drawbacks of conventional stream analysis systems, which are better suited for analyzing Internet media streams because they only support a single data format. The paper first presents the system's structure. The FFMPEG-based Decoding Section is then implemented in the paper. In the interim, it presents the decoding procedure and key data structures of FFMPEG. The Analysis Section is implemented in the paper at the end. A more sophisticated stream analysis system can be built upon the methodology described in this work. (Jiang, 2013).

The Open-Source Computer Vision Library, or OpenCV for short, is a popular and incredibly flexible open-source computer vision and machine learning software library. Originally created by Intel, it is being kept up to date by a development community. Applications in computer vision, image processing, and machine learning are the target markets for OpenCV. (Bradski G. , 2008)

A deep learning approach to speech enhancement in noisy environments. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 1019-1032.(Liu, 2021).

Explore how media players have adapted to the rise of streaming services. Discuss the integration of media players in smart TVs and other streaming-capable devices. Analyze how media players have influenced multimedia consumption habits and trends. Examine research on the impact of media players on binge-watching, media multitasking, and other behaviors.

Identify the challenges and issues facing media players, such as security concerns, DRM, and compatibility. Discuss potential future directions and innovations in media player technology. Summarize the key findings from the literature review. Provide insights into the current state of media players and their evolving role in the media landscape. Suggest areas for further research and development in the field of media players.

Provide an overview of media players and their significance in today's digital age. Explain the purpose and scope of the literature review. Outline the structure of the review. Categorize media players into different types, such as hardware-based, software-based, and web-based media players. Highlight the distinguishing features and functions of each type.

With today's reasonably priced digital video cameras, the majority of people may now access the capability of digital recording. Regrettably, this has led to a commensurate rise in the range of file types and codecs accessible. Certain codecs are less constrained by proprietary license constraints and more efficient than others. Therefore, being able to convert between several file formats is quite helpful since it allows you to utilize the format that you are most comfortable with rather than being limited to a single one. (Tomar, 2006).

# CHAPTER 3

# PROJECT ANALYSIS

# PROJECT ANALYSIS

## Analysis on Methods Related to Project

In this chapter the method of development of “Player Ex” have been discussed. The Player Ex is the system that is used for entertainment purpose. This allows the user to play the media files and edit the media files i.e. convert the file formats, enhance the videos and more.

* + 1. **Programming Structure**

The development of Media Player is an important role between user and application, it should be accurate and efficient. The Programming structure used for the Player Ex Development is Agile Development Model.

* + 1. **GUI Construction**

GUI stands for Graphical User Interface that is used to display the information about the application. GUI is an important factor for any application to interact with it. ADOBE PHOTOSHOP, ADOBE ILLUSTRATOR and FIGMA is used for GUI construction.

* + 1. **Testing**

To ensure that everything in the app is working properly and everything is according to user need, test performance has to be done upon the systems functionality. For the testing of this project VS Code and Git bash terminal is used. Git bash is the terminal that is used to run or test the python code.

## Software Methodology

Player Ex has a range of features to implement, such as video-to-audio conversion, audio trimming, vocal extraction, and auto video enhancement. To successfully develop this project, Agile Development model is used.

* + 1. **Agile Development Model**

Agile development is well-suited for projects with evolving requirements and a need for continuous integration of new features. Here is how Agile is adapted for Player Ex

The Agile Development Model is an iterative and incremental approach to software development that prioritizes collaboration, adaptability, and customer feedback.



Figure 3.1 Agile Development Model

* + - 1. **Planning**

The project is divided into smaller, manageable tasks and features. After dividing the tasks, they are prioritized them based on importance and complexity. For example, start with building the core media player functionality and then move on to additional features.

* + - 1. **Design**

Design the User Interface of the application. All the front pages, screens and all that stuff that will show some information to the end user designed after planning phase.

* + - 1. **Iterative Development**

It started with a minimum viable product (MVP) that includes the basic media player functionality. As project progress, add one feature at a time, thoroughly test it, and release incremental updates.

* + - 1. **Frequent Testing**

Team regularly tested the application to ensure that each feature works as expected. This is crucial for maintaining the overall quality of the application.

* + - 1. **User Feedback**

Gather feedback from users as new features or updates release. This feedback loop will help refining and improving the application based on user needs and preferences.

## Analysis on Tools and Software

Following tools and software are used to develop Player Ex.

* + 1. **Figma**

Figma is a cloud-based design and prototyping tool used by designers, developers, and teams to create user interfaces (UI) and user experiences (UX) for digital products such as websites, mobile apps, and more. Figma is known for its collaborative features, real-time editing, and accessibility from any device with an internet connection. It has gained popularity in the design and development community for its ease of use and ability to streamline the design and development process.

* + 1. **Visual Studio Code**

Visual Studio Code (VS Code or simply Code) is a free, open-source code editor developed by Microsoft. It is a highly popular and widely used code editor among developers, thanks to its versatility, performance, and an extensive ecosystem of extensions and plugins. VS Code is designed to be lightweight yet feature-rich, making it suitable for various programming languages and development tasks, including Python development.

* + 1. **Python**

Python can be used for Media Player development by leveraging various libraries and frameworks to create a custom media player application. One commonly used library for building graphical user interfaces (GUIs) in Python is Tkinter.

# CHAPTER 4

# PROJECT DESIGN

# PROJECT DESIGN

Project design is a critical phase in the project management process that involves creating a detailed plan and blueprint for the entire project, from initiation to completion. The design phase sets the interface for the project, outlining its scope, objectives, activities, timeline, resources, and deliverables. Here are the key components and steps in Player Ex project design:

**1. Project Objectives:**

* Develop a media player that can convert video to audio.
* Develop a media player that can trim audio files.
* Develop a Media Player that can Extract Vocal from songs.
* Develop a media player that can enhance the quality (Saturation, Contrast, Brightness, Color) of videos automatically.
* Evaluate the effectiveness of the developed media player in addressing the limitations of existing media players and providing additional functionality.

**2. Scope Definition**

The project aims to develop a comprehensive media player application with the following key features:

* Media Playback: Play various digital media file formats (i.e. mp3, mp4, mkv etc).
* User Friendly Interface: Player Ex aims to provide a supportive and user-friendly interface.
* Additional Features: The project aims to provide some additional features to its users (i.e. MP4-MP3 converter, Video enhancement, Vocal separation, Audio clipping etc).

**3. Risk Assessment:**

Conducted a thorough risk assessment to identify potential project risks and developed strategies for risk mitigation and management.

**4. Work Breakdown Structure (WBS):**

Here is the Work Breakdown Structure (WBS) of Player Ex.

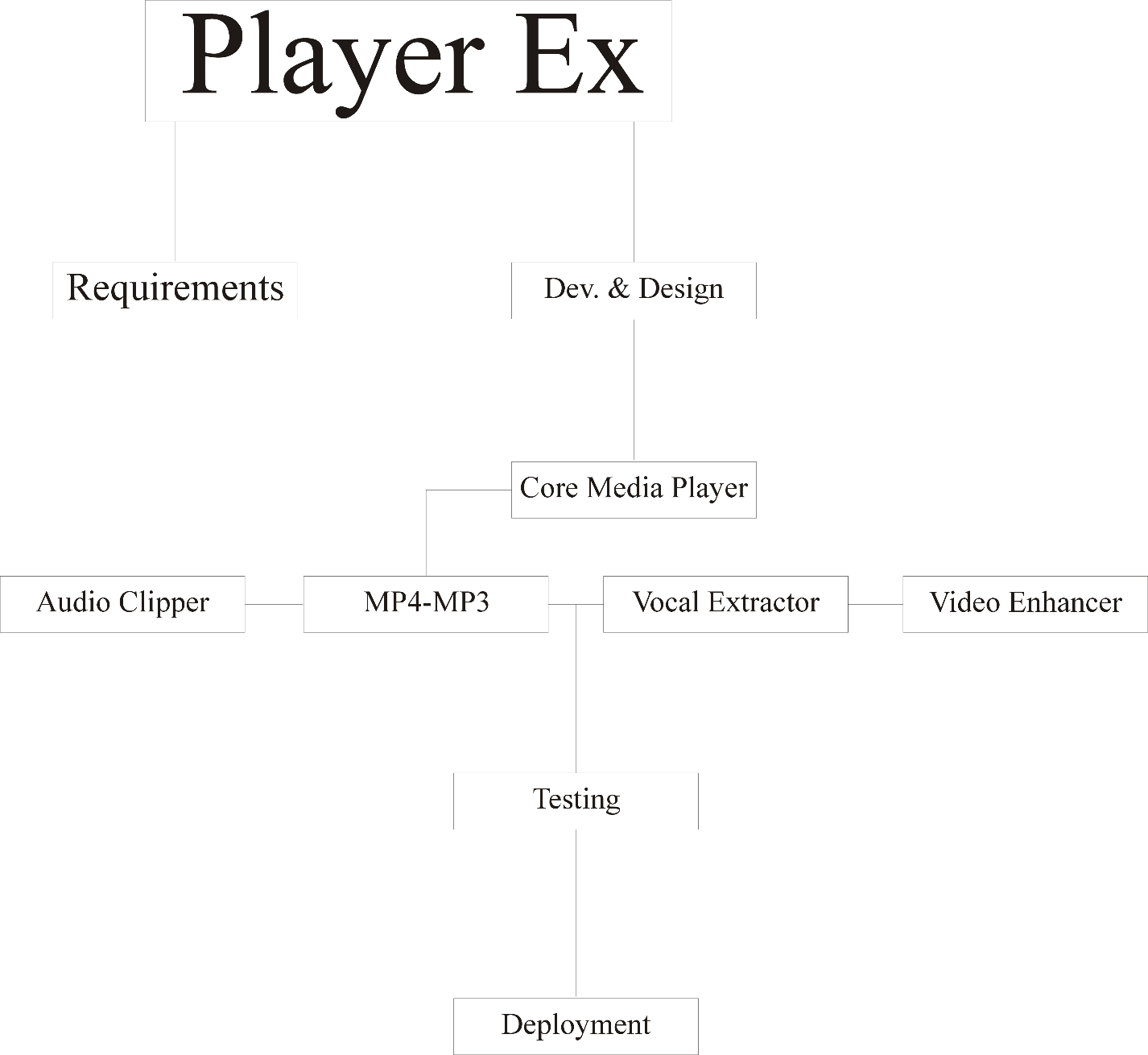


Figure 4.0 Work Breakdown Structure of Player Ex

**9. Project Documentation:**

Developed project documentation that will help users to learn how to use Player Ex effectively and easily.

## Use Case Scenarios

* + 1. **Playing Media Files**

|  |  |
| --- | --- |
| UC\_ID | UC\_1 |
| **Name** | Playing Media files |
| **Actor** | User |
| **Main Flow** | 1. Launch the app 2. Select a media file 3. Click “Play” |

Table 4.1 Playing Media Files

* + 1. **Video to Audio Conversion**

|  |  |
| --- | --- |
| UC\_ID | UC\_2 |
| **Name** | Video to Audio Conversion |
| **Actor** | User |
| **Main Flow** | 1. Launch the app 2. Go to “Features” section 3. Click “MP4/MP3” 4. Select a Video File 5. Click “convert” |

Table 4.2 Video to Audio Conversion

* + 1. **Audio Trimming**

|  |  |
| --- | --- |
| UC\_ID | UC\_3 |
| **Name** | Audio Trimming |
| **Actor** | User |
| **Main Flow** | 1. Launch the app 2. Go to “Features” section 3. Click and open “Audio Clipper/Trimmer” 4. Select Audio File (.wav recommended) 5. Hit the “Create Clip” button |

Table 4.3 Audio Trimming

* + 1. **Vocal Extraction**

|  |  |
| --- | --- |
| UC\_ID | UC\_4 |
| **Name** | Vocal Extraction |
| **Actor** | User |
| **Main Flow** | 1. Launch the app 2. Open “Vocal Extraction” by clicking it in “Features” section 3. Select a Song File (.wav recommended) 4. Hit the “Create Clip” button |

Table 4.4 Vocal Extraction

* + 1. **Auto Video Enhancement**

|  |  |
| --- | --- |
| UC\_ID | UC\_5 |
| **Name** | AI Video Enhancement |
| **Actor** | User |
| **Main Flow** | 1. Launch the app 2. Go to “Features” section 3. Click “AI Video Enhancer” 4. Select a Song File (.wav recommended) 5. Hit the “Create Clip” button |

Table 4.5 Auto Video Enhancement

## Use Case Diagram

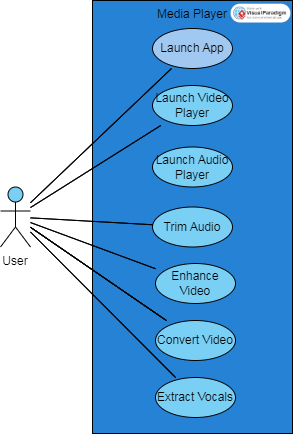


Figure 4.1 Use Case Diagram of Player Ex

## Sequence Diagram

### Sequence for Audio Player

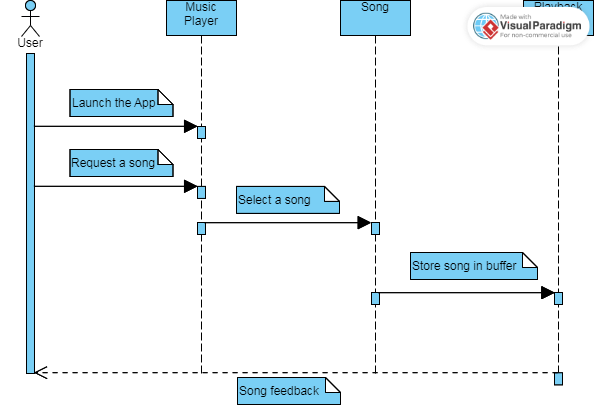


Figure 4.2 Sequence for Audio Player

### Sequence for Video Player

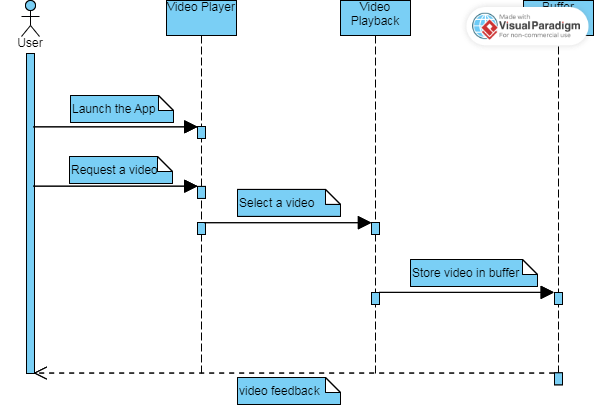


Figure 4.3 Sequence for Video Player

### Sequence for Audio Clipper

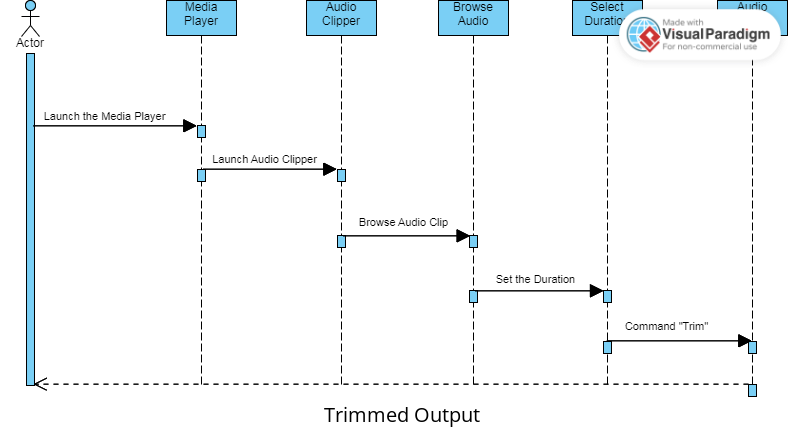


Figure 4.4 Sequence for Audio Clipper

### Sequence for Vocal Extractor

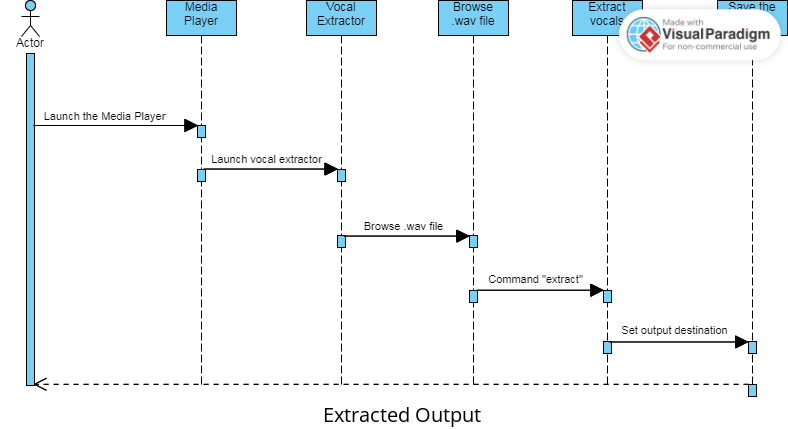


Figure 4.5 Sequence for Vocal Extractor

### Sequence for Video Converter

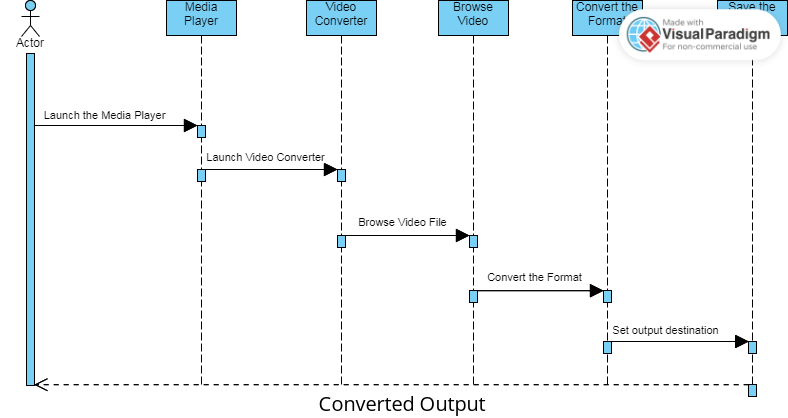


Figure 4.6 Sequence for Video Converter

### Sequence for AI Video Enhancer

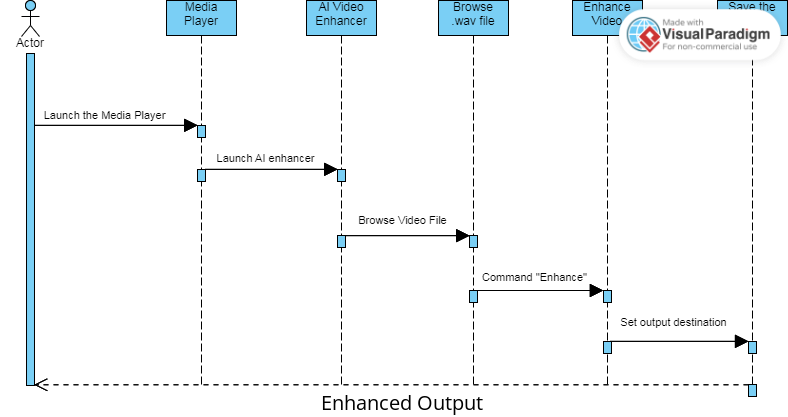


Figure 4.7 Sequence for AI Enhancer

# CHAPTER 5

# USER INTERFACE DESIGN

# USER INTERFACE DESIGN

This Chapter includes the appearance (User Interface Design) of the Player Ex. The basic Media Player UI includes a root window having six buttons. The List of Buttons is as follows:

* There are some steps of database design
* Audio Player
* Video Player
* Audio Clipper
* MP4 to MP3 Converter
* Vocal Extractor
* Auto Video Enhancer

There are two sections of the root window. In the left section, all the buttons are placed, and in the right section, there is a logo of the Player Ex. Left section is representing the buttons that refers to the separate window for every button i.e. Audio Player, Video Player etc have their own separate window.

## Screenshots of User Interface

All the screenshots of the User Interface of Media Player, are embedded below. Every screenshot/window is representing every separate window for every separate task.

* + 1. **Root Screen**

****

Figure 5.1 Root Window

* + 1. **Audio Player Screen**

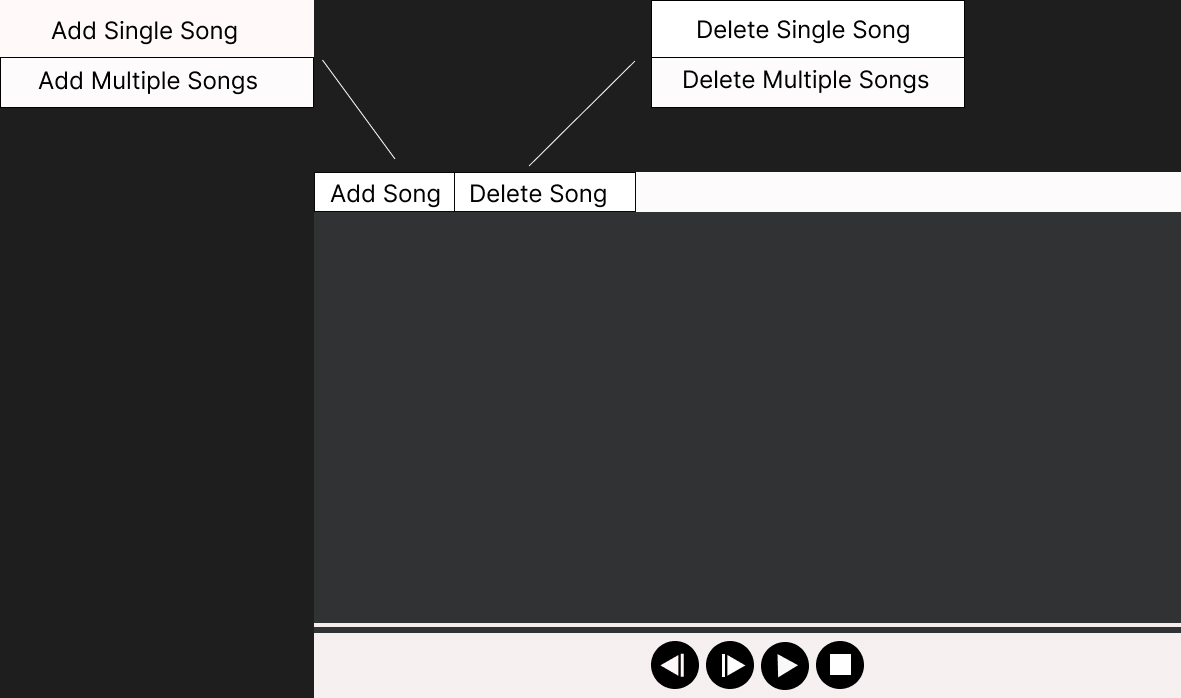
****

Figure 5.2 Audio Player

* + 1. **Video Player Screen**

****

Figure 5.3 Video Player

* + 1. **Audio Clipper Screen**

****

Figure 5.4 Audio Clipper

* + 1. **MP4/MP4 Converter Screen**

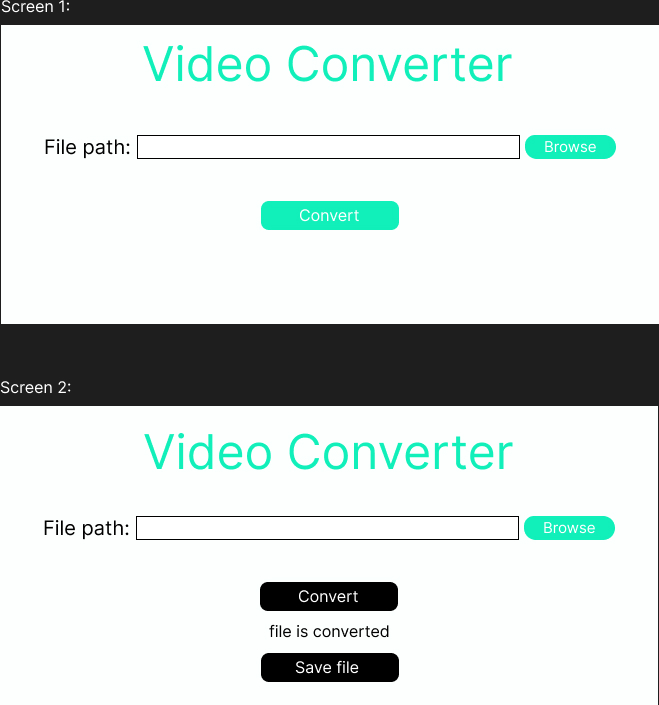
****

Figure 5.5 Video Converter

* + 1. **Vocal Extractor Screen**

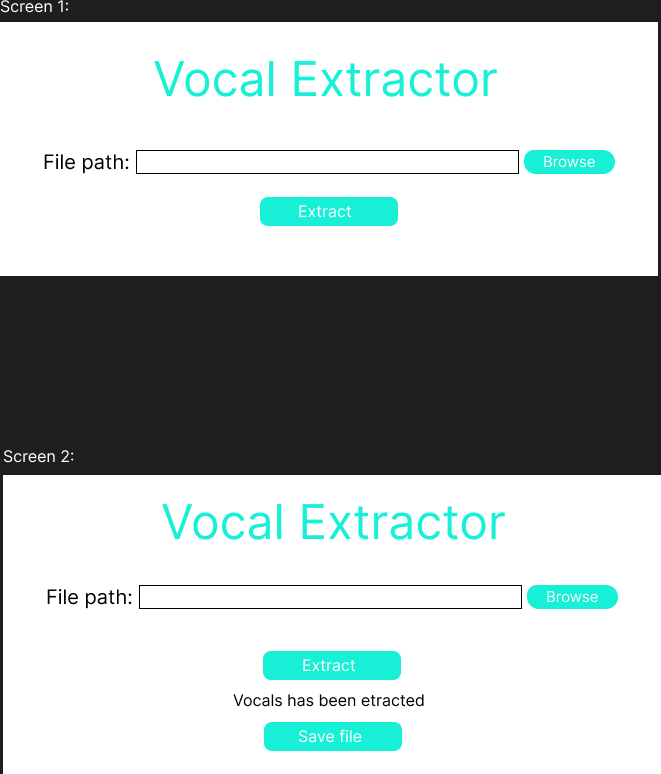
****

Figure 5.6 Vocal Extractor

# CHAPTER 6

# TESTING AND IMPLEMENTATION

# TESTING AND IMPLEMENTATION

## Black Box Testing

Black box testing for Player Ex involves testing the software without knowledge of its internal code or structure. The goal is to assess its functionality and behavior from a user's perspective. Black box testing is a type of software testing where the main goal is to assess a system's functionality without looking at its core code or architecture. Black box testing is crucial to the development of media players since it guarantees that the player fulfills its intended requirements and performs as intended from the user's point of view. Black box testing is used in media player development in the following ways:

* **Understand Requirements**: Begin by gathering and understanding the requirements for the media player. This includes knowing the supported formats, features, user interfaces, and expected behaviors.
* **Test Planning**: A plan that outlines the scope of testing, objectives, test cases, and test scenarios. Make sure to cover different aspects of the media player, such as playback, user interfaces, file format support, and so on.
* **Test Design**: Design test cases that cover a wide range of scenarios, including both typical and edge cases. File formats, playback controls, screen resolutions, network conditions, and error handling are tested.
* **Installation and Setup**: Installed the player EX on a test machine or virtual environment. Make sure it is properly configured and any required plugins or codecs are installed.

**1. Functional Testing:**

Playback Functionality: Check to see if the media player can smoothly play a variety of media formats, including audio and video.

Basic Regulators: Check out the play, pause, stop, rewind, fast forward, and volume controls, among other important features.

Choose Your Media Sources Wisely: Make sure the player can access files locally, on CDs, DVDs, streaming services, and external devices, among other sources.

**2. Compatibility Testing:**

File Format Support: Ensure that there are no compatibility problems with a variety of widely used media file formats (such as MP3, MP4, AVI, and MKV) that the media player supports.

Operating Systems: To make sure the player functions as intended, test it on several operating systems, including Windows, macOS, and Linux.

Mobile Platforms: If appropriate, check the media player's compatibility across a range of mobile platforms, such as iOS and Android.

**3. User Interface (UI) Testing:**

GUI Functionality: Check that the buttons, menus, and navigational elements of the player's graphical user interface function as they should.

Resolution and Screen Size: To make sure the user interface (UI) is still responsive and aesthetically pleasing, test the player on a range of screen sizes and resolutions.

Accessibility: Check to see if the player conforms with accessibility guidelines, which include keyboard navigation and screen reader compatibility.

**4. Performance Testing:**

Media Loading Time: Measure the time it takes for the media player to load and start playing media files.

Resource Usage: Assess the player's CPU and memory usage, particularly when handling high-definition videos.

Streaming Performance: Test the player's performance with streaming media and analyze buffering times.

**5. Usability Testing:**

User Experience (UX): Evaluate the overall user experience, considering aspects like intuitiveness, ease of use, and user satisfaction.

Error Handling: Assess how the media player handles errors, such as file not found, corrupted media files, or network issues.

**6. Playlist and Library Management:**

Verify that the media player can create, edit, and manage playlists effectively. Ensure the player can organize and search through large media libraries efficiently.

**7. Integration and Connectivity Testing:**

Test the player's ability to integrate with external devices, such as streaming devices, headphones, or smart TVs. Check network connectivity features like streaming from the internet and sharing media across a network.

**8. Security and Privacy Testing:**

Test the player's ability to integrate with external devices, such as streaming devices, headphones, or smart TVs. Check network connectivity features like streaming from the internet and sharing media across a network.

**9. Regression Testing:**

Continuously conduct regression testing to ensure that new features and bug fixes do not introduce new issues or disrupt existing functionality.

**10. Performance Under Stress:**

Conduct stress testing to evaluate how the media player handles heavy workloads, such as opening multiple files simultaneously or rapid, repeated user actions.

**11. Exploratory Testing:**

Encourage testers to explore the media player and identify unexpected issues or unusual behavior. Black box testing should be conducted by skilled QA testers who can approach the media player as end-users would, examining its functionality, usability, and performance without knowledge of the internal code. This comprehensive testing approach helps ensure that the media player functions reliably and provides a positive user experience.

**Black Box Testing in Project:**

* Examine and understand the requirements and specifications of the project application.
* Identify valid and invalid inputs with expected outputs to check that the system detects them properly.
* Create test cases with different test scenarios and inputs..

## Compatibility Testing

Compatibility testing deals with delivering a software product so that it works well across different available versions, configurations, and platforms based on user requirements. It is one of the nonfunctional requirements and hence focuses on delivering consistent and high-quality performance. Compatibility testing helps ensure complete customer satisfaction as it checks whether the application performs or operates as expected for all the intended users across multiple platforms. Compatibility Testing is a type of project testing to check whether your project is capable of running on different hardware, operating systems, applications, and network environments.

* .Check the media player quality and versions of the project.
* Check the media player is according to customers' requirements.
* Check the media player is running on compatible systems and is acceptable for every OS..

## Performance Testing

Performance testing is a non-functional software test used to evaluate how well an application performs. More precisely, performance testing measures an application’s stability, speed, scalability, and responsiveness under specific workloads. In particular, performance testing aims to evaluate several metrics such as browser, page, and network response times, server request processing times, number of acceptable simultaneous users, CPU memory consumption, and the number and type of errors that arise when the application is being used.

* + 1. **Identify Performance Metrics**
* Playback quality (e.g., video and audio quality).
* Responsiveness to user inputs (e.g., play, pause, seek).
* Resource utilization (CPU, memory, network bandwidth).
* Load handling (simultaneous users or streams).
  + 1. **Test Environment Setup**

Set up a test environment that closely resembles the production environment, including the hardware and software configurations.

* + 1. **Test Data Preparation**

Prepare a wide range of media files (different formats, resolutions, bitrates, and durations) to cover a variety of scenarios.

* + 1. **Performance Testing Types**
* Load Testing: Determine how the media player performs under expected load conditions, such as multiple users simultaneously accessing and playing media.
* Stress Testing: Push the media player beyond its limits to identify where it might fail or perform poorly.
* Scalability Testing: Assess how well the media player scales.
* Endurance Testing: Continuously use the media player over an extended period to check for resource leaks or degradation over time.
  + 1. **Performance Testing Tools**

Utilize performance testing tools to simulate user interactions and gather data. Tools like JMeter, Gatling, Apache Benchmark, or custom scripts can be useful.

* + 1. **Performance Metrics Monitoring**

Continuously monitor performance metrics during the tests. Use logging and monitoring tools to gather data, such as CPU and memory usage, response times, error rates, and network throughput.

* + 1. **Reporting and Analysis**

Analyze the test results and create comprehensive reports. Identify bottlenecks, issues, and areas for improvement. Consider the user experience and media quality.

* + 1. **Tuning and Optimization**

Make necessary adjustments and optimizations based on the findings. This might involve improving resource management, optimizing code, or upgrading hardware.

* + 1. **Retesting**

After making changes, re-run the performance tests to ensure that the issues have been addressed and the media player performs better.

* + 1. **Compliance Test**

Depending on your industry or region, you may need to conduct compliance testing to ensure the media player adheres to legal and regulatory requirements.

## Usability Testing

Usability testing of a media player is essential to evaluate how well users can interact with the software and to identify any issues or areas for improvement in terms of user experience. Usability testing in media player development is essential to ensure that the player is user-friendly, intuitive, and meets the needs and expectations of its intended audience. Usability testing focuses on evaluating the user experience, interface design, and overall ease of use. Here's how usability testing can be conducted in media player development:

**1. Define Clear Objectives:**

Determine the specific goals and objectives of the usability testing. What aspects of the media player's usability do you want to evaluate? Examples may include ease of navigation, media playback controls, or the intuitiveness of playlists.

**2. Create User Personas:**

Develop user personas that represent the target audience for the media player. This will help testers identify with the users' characteristics and preferences.

**3. Select Test Participants:**

Recruit a diverse group of test participants who match the user personas. Aim for a mix of ages, backgrounds, and familiarity with media players to ensure comprehensive feedback.

**4. Design Test Scenarios and Tasks:**

Create realistic usage scenarios and tasks that participants will perform during the testing. For example:

* Play a specific song or video file.
* Create a playlist and add multiple songs.
* Adjust the volume and equalizer settings.
* Stream a video from an online source.
* Ensure that scenarios cover a wide range of typical user interactions.

**5. Develop Testing Materials:**

Prepare any necessary materials, such as test scripts, questionnaires, or surveys that will be used during the testing process.

**6. Conduct the Usability Test:**

Hold individual or group usability testing sessions with participants. Encourage them to think aloud as they interact with the media player. Observe how participants navigate the user interface, perform tasks, and accomplish their goals.

**7. Collect Qualitative and Quantitative Data:**

Gather both qualitative data (e.g., participant feedback, comments, and observations) and quantitative data (e.g., time taken to complete tasks, success rates, and errors).

**8. Analyze Test Results:**

Evaluate the collected data to identify usability issues and areas where improvements are needed. Common issues may include difficulties in finding features, confusing interface elements, or unclear error messages.

**9. Prioritize and Document Findings:**

Prioritize the identified usability issues based on their impact on the user experience. Document these findings, including screenshots or video recordings of user interactions.

**10. Iterative Design and Development:**

Share usability test findings with the development team. Collaborate to address the identified issues and make necessary design and functionality improvements. Conduct multiple iterations of usability testing as the player evolves, ensuring that user feedback is incorporated into the development process.

**11. Post-Test Surveys and Questionnaires:**

After the usability testing, provide participants with surveys or questionnaires to collect their feedback on the overall experience, satisfaction, and suggestions for further improvements.

**12. Validate the Design Changes:**

After making changes based on usability testing results, validate the design improvements with follow-up testing to confirm that the issues have been resolved and that the user experience has improved.

**13. Continuous Feedback Loop:**

Implement a continuous feedback loop for usability by encouraging users to provide feedback during the beta testing phase and after the media player's release. This ensures ongoing improvement and user satisfaction.

Usability testing in media player development is an iterative process that helps ensure that the player is user-centric and aligned with user expectations. By incorporating feedback from real users, developers can create a media player that is easy to use and provides a positive user experience.

## White Box Testing

White box testing, also known as structural or glass-box testing, involves examining the internal structure of a software application to ensure its correctness, efficiency, and reliability. In the context of a media player, white box testing would focus on verifying the correctness of the underlying code and its interaction with the media player's functionality. White box testing of a media player is essential for ensuring that the software functions correctly, efficiently, and securely. It complements other types of testing, such as black box testing (which assesses functionality from a user's perspective), to provide a comprehensive evaluation of the media player's quality and reliability.

White box testing in media player development involves examining the internal structure and code of the software to assess its functionality, security, and reliability. This type of testing is also known as structural or glass-box testing and is typically conducted by developers or quality assurance engineers who know the software's architecture. Here's how white box testing can be applied in media player development:

**1. Code Analysis:**

Review the source code of the media player to identify potential vulnerabilities or issues in the codebase. Pay attention to coding standards, proper error handling, and security practices.

**z. Code Coverage Analysis:**

Use code coverage tools to measure how much of the code is executed during various test cases. This helps ensure that all code paths are tested.

**3. Security Testing:**

Conduct security assessments to identify and mitigate vulnerabilities. Test for potential security issues like buffer overflows, injection attacks, and unauthorized access.

**4. Performance Testing:**

Evaluate the performance of the media player by analyzing the code to identify bottlenecks, memory leaks, and resource consumption issues.

**5. Data Flow Analysis:**

Analyze how data flows within the media player to ensure that sensitive information is handled securely and that data is not unintentionally exposed.

**6. Stress Testing:**

Assess the media player's code for how it handles heavy workloads, concurrent users, and large media files. Ensure that the software performs reliably under stress.

**7. Error Handling and Exception Testing:**

Examine how the media player's code handles errors, exceptions, and edge cases. Ensure that it provides informative error messages and gracefully recovers from failures.

**8. API and Interface Testing:**

Verify that the code implementing APIs and interfaces (e.g., media file parsers, streaming protocols) is working correctly and that it adheres to standards.

**9. Code Review and Static Analysis:**

Perform code reviews to identify design and coding issues. Use static analysis tools to automatically identify potential problems in the code, such as code smells and style violations.

**10. Regression Testing:**

Continuously conduct white box testing as changes are made to the media player code. This helps ensure that new code additions do not introduce issues or vulnerabilities.

**11. Compatibility Testing:**

Verify that the media player's code is compatible with different platforms and environments, including various operating systems and hardware configurations.

**12. Code Optimization:**

Identify areas in the code where optimization is possible, such as reducing computational complexity or memory usage.

**13. Dependency Analysis:**

Review the dependencies of the media player code to ensure that third-party libraries and components are up to date and secure.

**14. Code Quality Metrics:**

Use code quality metrics to assess the maintainability, readability, and complexity of the code. Maintainability factors into how easy it is to update and maintain the software in the long run.

**15. Documentation Review:**

Examine the code documentation to ensure it is accurate, complete, and well-maintained. Code comments and documentation are essential for other developers to understand and work with the code.

White box testing is a valuable aspect of media player development to ensure code quality, security, and reliability. It helps identify and address issues at the code level and is often conducted alongside black box and usability testing to provide comprehensive coverage of the media player's functionality and user experience.

# CHAPTER 7

# CONCLUSION AND FUTURE WORK

# CONCLUSION AND FUTURE WORK

Taking a look on the functionality, design, and performance, the world of media players is explored. A wide range of topics has been covered, from the fundamental concepts to advanced features, usability considerations, and various testing methodologies. Here are the key takeaways:

Project begins by establishing a strong foundation, understanding what media players are, like how they work, and their role in enabling the consumption of multimedia content. Emphasizing the role, it plays in user experience and the effective consumption of multimedia content by exploring User Interface Design.

**Project Objectives:**

The objectives of this project are to:

* Develop a media player that can convert video to audio.
* Develop a media player that can trim audio files.
* Develop a Media Player that can Extract Vocal from songs.
* Develop a media player that can enhance the quality (Saturation, Contrast, Brightness, Color) of videos automatically.

**Challenges and Limitations**

During the development of the Player Ex, various challenges met. Some of the challenges are following

* Faced various bugs and errors while creating vocal extractor feature. The implementation of the resources (libraries for audio stem separation i.e. OpenCv) was a tricky task.
* After that, encountered some bugs in audio clipper feature while setting-up the audio segmentation libraries (ffmpeg).
* Different challenges faced, but the major challenge was to compile the standalone executable file of the application.

**Acknowledgments:**

A project like this one is never the result of a single person's efforts. The contribution of many people in different ways has made it possible. We would like to thank Almighty ALLAH, who is the most beneficent, merciful, and blessed us with “Holy Prophet Hazrat MUHAMMAD (S.A.W.W)”. We are thankful to Almighty ALLAH, who blessed us with sound health, kind parents, talented teachers, and intellectual efficiency to complete this project. It's an honor for us to offer our heartfelt gratitude to our ever-loving supervisor, Mian Muhammad Munir-Ud-Din, Head of Department of Computer Science at Govt. Islamia Graduate College, Civil Lines, Lahore. He helped us to make this project possible. His support, constructive criticism, encouragement, valuable comments, suggestions, timely help throughout the project, and many innovative ideas as well as his pain-taking effort in proofreading the drafts, are greatly appreciated. Indeed, without his guidance, we would not be able to put the topic together. Last but not least, we would like to thank our loving parents for their unconditional support, both financially and emotionally throughout our degree. In particular, the patience and understanding shown by our families during the BS-Honors years are greatly appreciated.

## Future Work:

Every project has its future objectives to maintain and update the application to stand in the market. Player Ex has also some future objectives to provide its users updated application.

**Future Features and Enhancements**

* Develop enhanced stem separation feature that can provide various tracks after separating.
* Provide updated UI interface to tackle with the audio clipper feature. Starting and ending time stamp can be selected using UI pointers.
* Deploy the application on the marketplace and get sales of the application, so more users can use the Player Ex features.

**Performance Optimization:**

In future, the performance optimization will be applied on the Player Ex. So the users of Player Ex can use the interactive and well managed application.

**Compatibility and Platform Expansion:**

Currently Player Ex is just for Windows OS. In future it will be available for cross platforms. Player Ex could be used on different platforms (i.e. Windows, Linux, Mac).

**Security and Privacy Enhancements:**

Player Ex will be providing security and data privacy to its users. User data will be secure, it will not be used for unethical purposes.

# REFERENCES

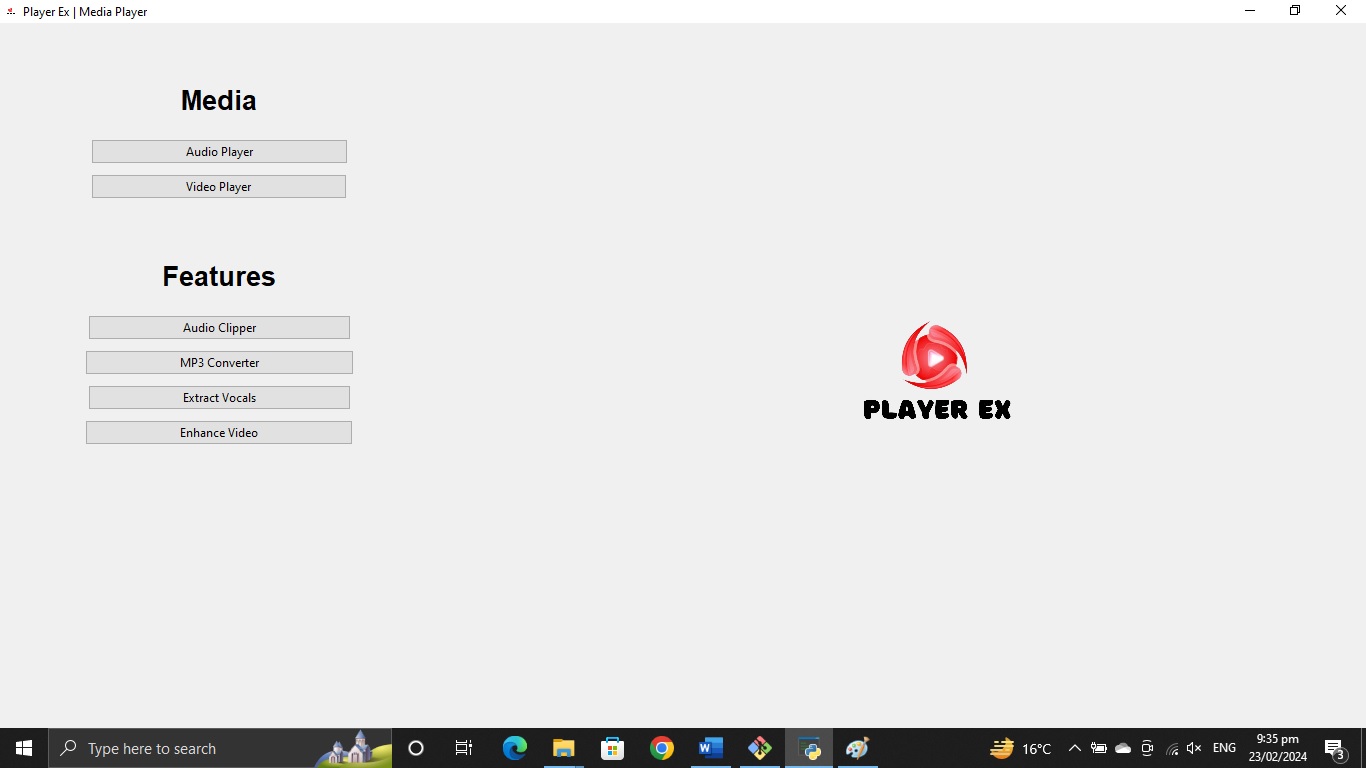
1. Anderson, L. (2018). A Comprehensive Guide to Python Programming. Python Press.
2. A. Gallardo Antolín, R. San Segundo Hernández, UPM-UC3M system for music and speech segmentation (2010).
3. Brown, E. (2017). Audio Processing in Python: A Practical Guide. Audio Engineering Society Journal, 65(3), 198-212.
4. Carter, P. (2023). Multimedia File Format Analysis. Digital Media and Entertainment Research, 20(4), 315-330.
5. C. -H. Wu, Y. -H. Chiu, C. -J. Shia, C. -Y. Lin, Automatic segmentation and identification of mixed-language speech using delta-BIC and LSA-based GMMs. IEEE Trans. Audio Speech Lang. Process.14(1), 266–276 (2006).
6. Davis, K. (2023). User Experience Design for Multimedia Applications. HCI & Usability Review, 17(3), 211-225.
7. D. Castán, A. Ortega, A. Miguel, E. Lleida, Audio segmentation-by-classification approach based on factor analysis in broadcast news domain. EURASIP J. Audio Speech Music. Process.2014(1), 34 (2014).
8. D. de Benito-Gorron, A. Lozano-Diez, D. T. Toledano, J. Gonzalez-Rodriguez, Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset. EURASIP J. Audio Speech Music. Process.2019(1), 9 (2019).
9. Flask.(2023). A lightweight Python web framework. Retrieved from https://flask.palletsprojects.com/en/2.1.x/
10. FFmpeg.(2023). A complete, cross-platform solution to record, convert and stream audio and video. Retrieved from https://ffmpeg.org/
11. Garcia, R. (2019). Real-Time Streaming of Media Files. Streaming Technology Conference Proceedings, 72-86.
12. Gonzalez, R. C., & Woods, R. E. (2018). Digital image processing (4th ed.). Pearson Education.
13. Gonzalez, C. (2017). Audio Analysis and Signal Processing. Audio Engineering Journal, 62(5), 342-356.
14. H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, Mixup: beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017).
15. Hall, D. (2020). Mobile User Interface Design for Media Player Apps. Mobile Interaction Review, 15(3), 289-304.
16. Harris, M. (2016). Visual Design Principles for Multimedia Applications. Multimedia & Visual Arts, 12(4), 317-330.
17. Jähne, B. (2013). Digital image processing: Concepts, algorithms, and applications (7th ed.). Springer Science & Business Media.
18. J. Foote, in IEEE International Conference on Multimedia and Expo (ICME). Automatic audio segmentation using a measure of audio novelty, (2000), pp. 452–455. https://doi.org/10.1109/icme.2000.869637.
19. Johnson, A. (2021). User-Centered Interface Design for Media Players. Media & Design Journal, 6(3), 115-132.
20. Kivy documentation.(2023). Retrieved from https://kivy.org/doc/stable/
21. Krippendorff, K. (2013). Content analysis: An introduction to its methodology. Sage publications.
22. Librosa.(2023). A Python Library for audio and music analysis in Python. Retrieved from https://librosa.github.io/
23. Lee, S. (2018). Machine Learning for AI-Based Video Enhancement. IEEE Transactions on Multimedia, 23(7), 1809-1822.
24. Martinez, P. (2023). Cloud Computing and Media Player Applications. CloudTech Magazine, 21(6), 45-58.
25. Mitchell, L. (2022). Cloud-Based Media Storage Solutions. Cloud Computing Trends, 28(6), 68-82.
26. NIST, The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan, (Melbourne, 2009).
27. Opencv.(2023). A Python library for image and video processing in Python. Retrieved from https://opencv.org/
28. Python Software Foundation.(2023).Python programming language. Retrieved from https://www.python.org/
29. PyQt.(2023).Python bindings for the Qt cross-platform application and UI framework. Retrieved from https://www.riverbankcomputing.com/software/pyqt/
30. Pydub.(2023). Manipulate audio with an simple and easy high-level interface. Retrieved from https://pydub.com/
31. Pygame documentation. (2023). Retrieved from https://www.pygame.org/docs/
32. Perez, R. (2016). Advanced User Interaction in Media Player Applications. Human-Computer Interaction Research, 5(3), 211-226.
33. Robinson, S. (2023). Playlist Management Strategies in Media Players. Music Technology Research, 8(2), 135-148.
34. Reed, D. (2021). Python for Multimedia Development. Python Developer's Magazine, 15(1), 39-52.
35. Rodriguez, H. (2017). User Authentication in Media Player Applications. Security & Privacy Journal, 10(5), 53-68.
36. R. Huang, J. H. Hansen, Advances in unsupervised audio classification and segmentation for the broadcast news and NGSW corpora. IEEE Trans. Audio Speech Lang. Process.14(3), 907–919 (2006).
37. Smith, J. (2022). Multimedia Processing: Concepts and Applications. Publisher.
38. S. Cerdà, J. Albert, A. Giménez Pastor, J. Andrés Ferrer, J. Civera Saiz, A. Juan Císcar, Albayzin evaluation: the PRHLT-UPV audio segmentation system.
39. Szeliski, R. (2011). Computer vision: Algorithms and applications (2nd ed.). Springer Science & Business Media.
40. Smith, J. (2010). Media Players and Their Features. New York: Random House.
41. T. Theodorou, I. Mporas, N. Fakotakis, An overview of automatic audio segmentation. Int. J. Inf. Technol. Comput. Sci. (IJITCS). 6(11), 1–9 (2014).
42. Tkinter.(2023).Python GUI programming. Retrieved from https://docs.python.org/3/library/tkinter.html
43. Turner, B. (2022). Security in Multimedia Applications. Cybersecurity Journal, 7(4), 277-291.
44. Taylor, J. (2019). Visualization Techniques in Media Players. Visual Computing Journal, 7(4), 279-293.
45. Turner, M. (2016). Audio and Video Streaming Protocols. Journal of Streaming Technologies, 8(2), 129-143.
46. White, G. (2021). Database Design for Multimedia Storage. Database & Information Systems Journal, 14(1), 42-56.
47. Williams, A. (2019). AI-Based Video Enhancement Techniques. International Conference on Artificial Intelligence, 129-145.
48. Williams, R. (2019). AI-Based Video Enhancement Techniques. International Conference on Artificial Intelligence, 129-145.
49. Watkinson, J. (2018). The art of digital audio (6th ed.). Focal Press.
50. Summary and Notable Features of Existing Systems (VLC, Media Player Classic) <https://www.makeuseof.com/tag/top-5-free-media-players-for-windows/>

# USER MANUAL

­­­

User Manual or User Guide to use Player Ex is given below. If a layman uses Player Ex, the complete guide to that person is explained with detailed screenshots.

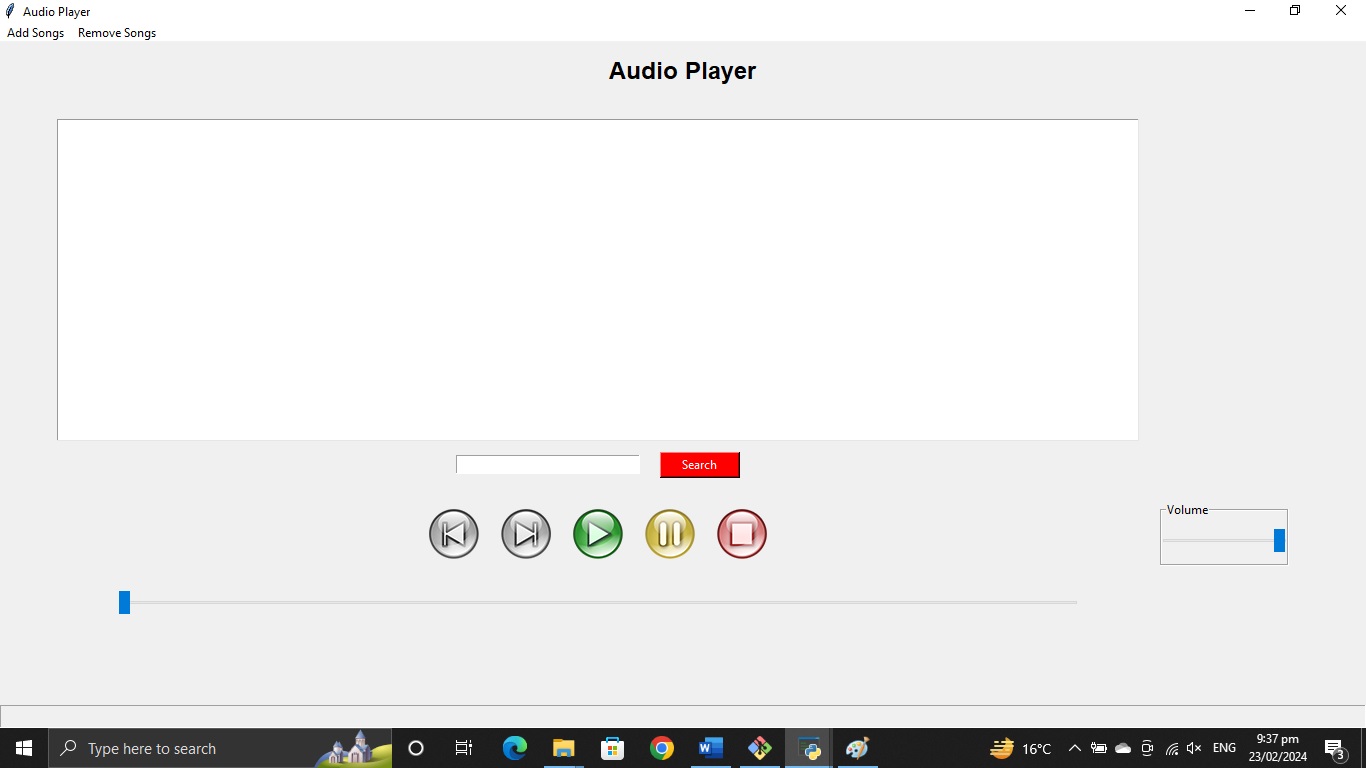
**Root Window Guide:**

****

User Guide 1.1 Root Window

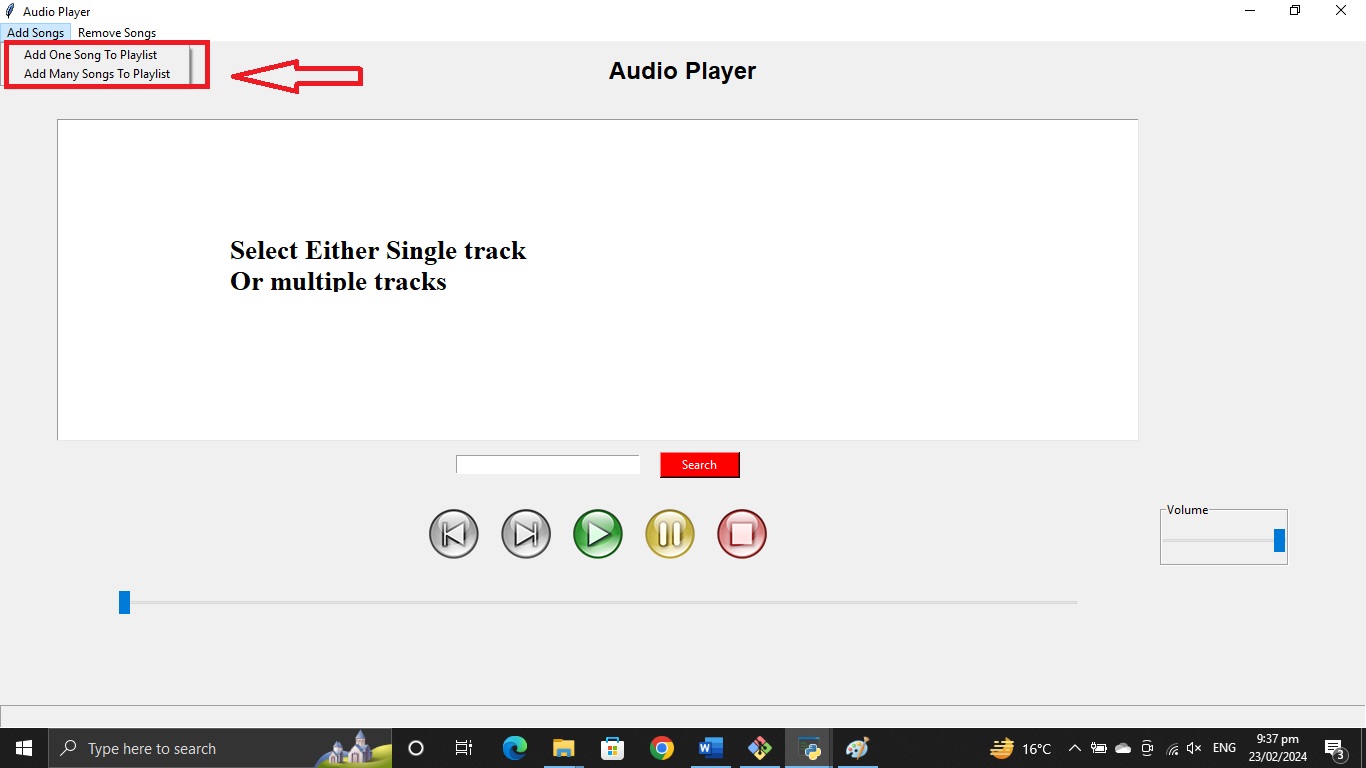
* Open Player Ex App
* The Above “Root” window will appear
* Choose the feature of your choice
* It will pop-up a new window for every feature
* Click “Audio Player”

**Audio Player Guide:**

****

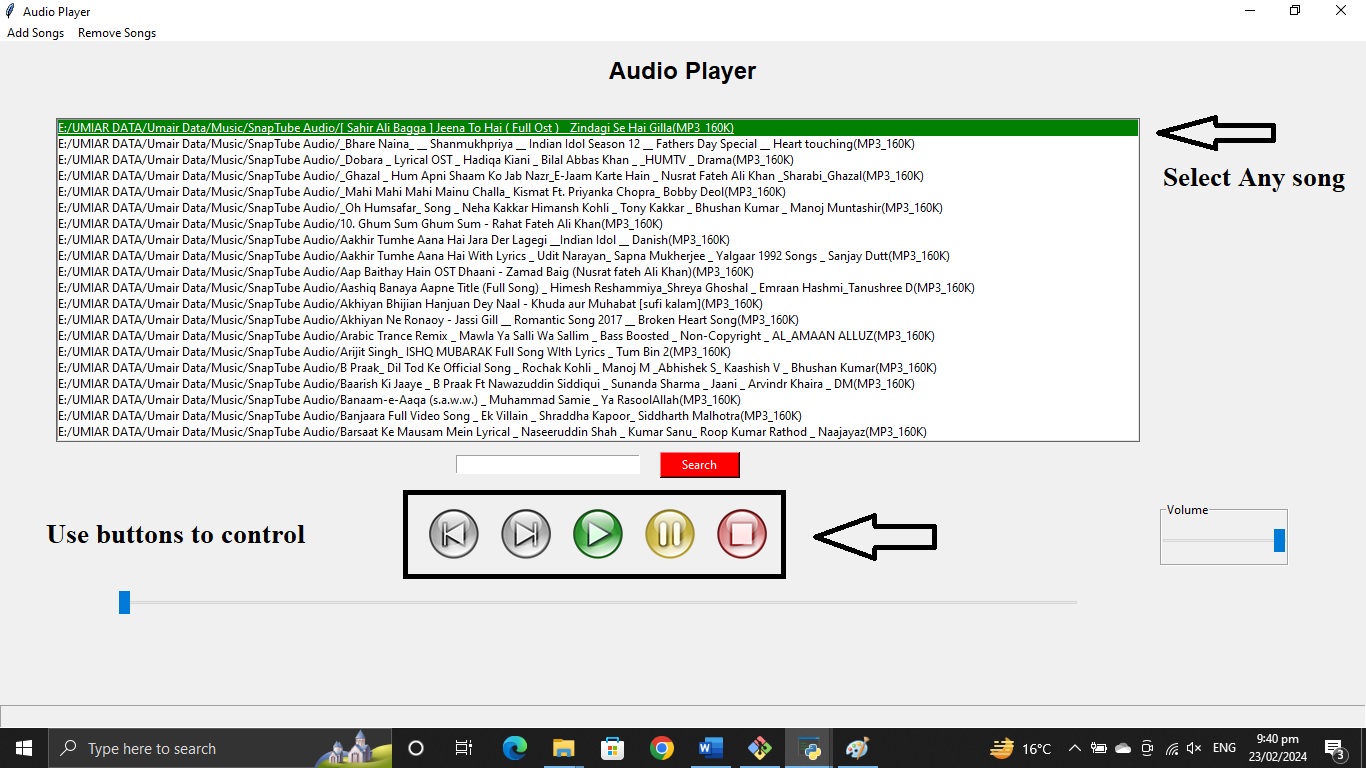
User Guide 1.2 Audio Player

* After clicking “Audio Player” the above window will open
* There is a button named “Add a song”
* Click on that button to add your song to playlist

****

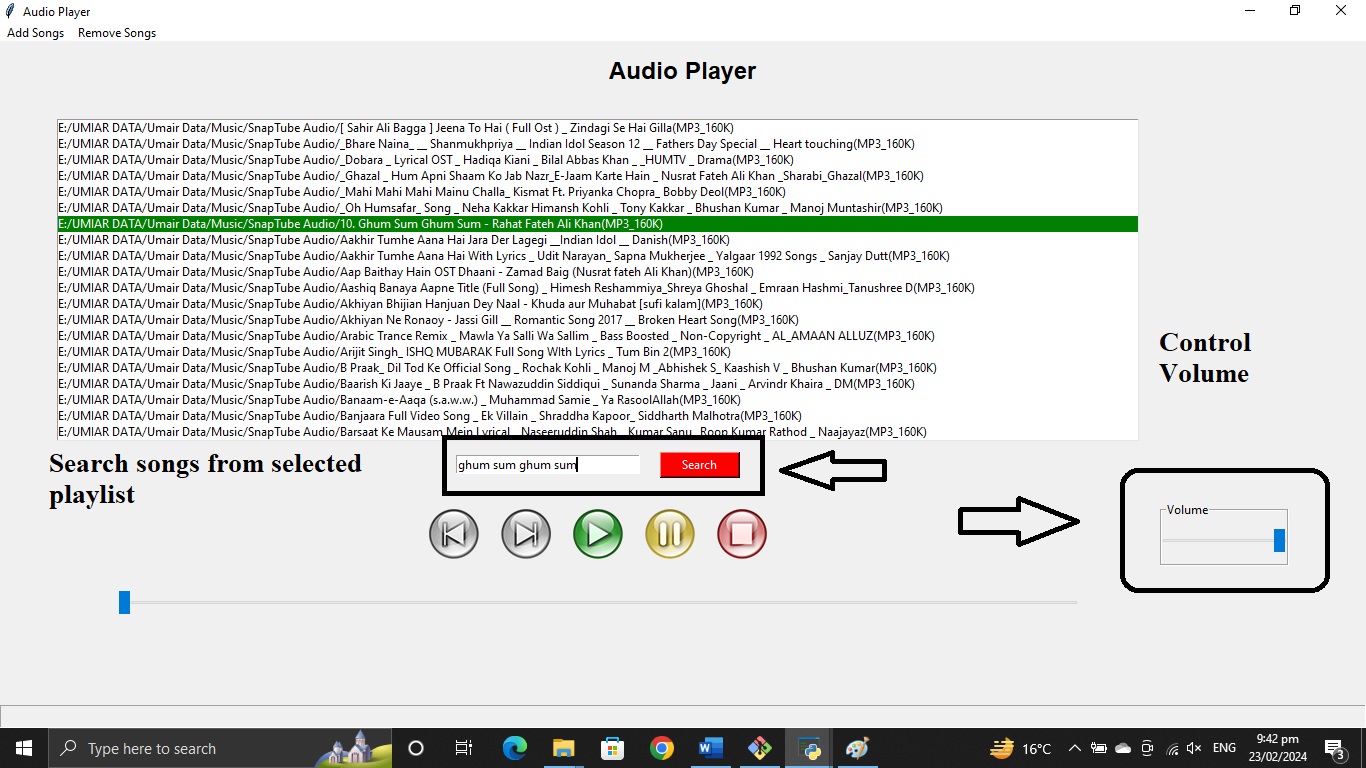
User Guide 1.3 Upload Tracks

* You can add single or multiple songs to the playlist.
* Click on “Add many songs to Playlist”
* Select the list of songs and click “Open”
* After clicking “Open” the playlist will be updated and the black screen of Audio Player window
* Select any of the song from the playlist and click on “Play” button

****

User Guide 1.4 Audio Player Controls

* “Play” button will play the selected song.
* “Pause” button will pause the running song.
* Click “Previous” button to play the previous of the selected song.
* Click “Next” button to play the next of the selected song.
* “Stop” button will finish the Playback.
* Control the volume of the song by dragging the volume bar up or down.
* Timeline bar is to control the flow of the song.
* You can search the songs from playlist you’ve selected.

****

User Guide 1.7 Audio Player Controls

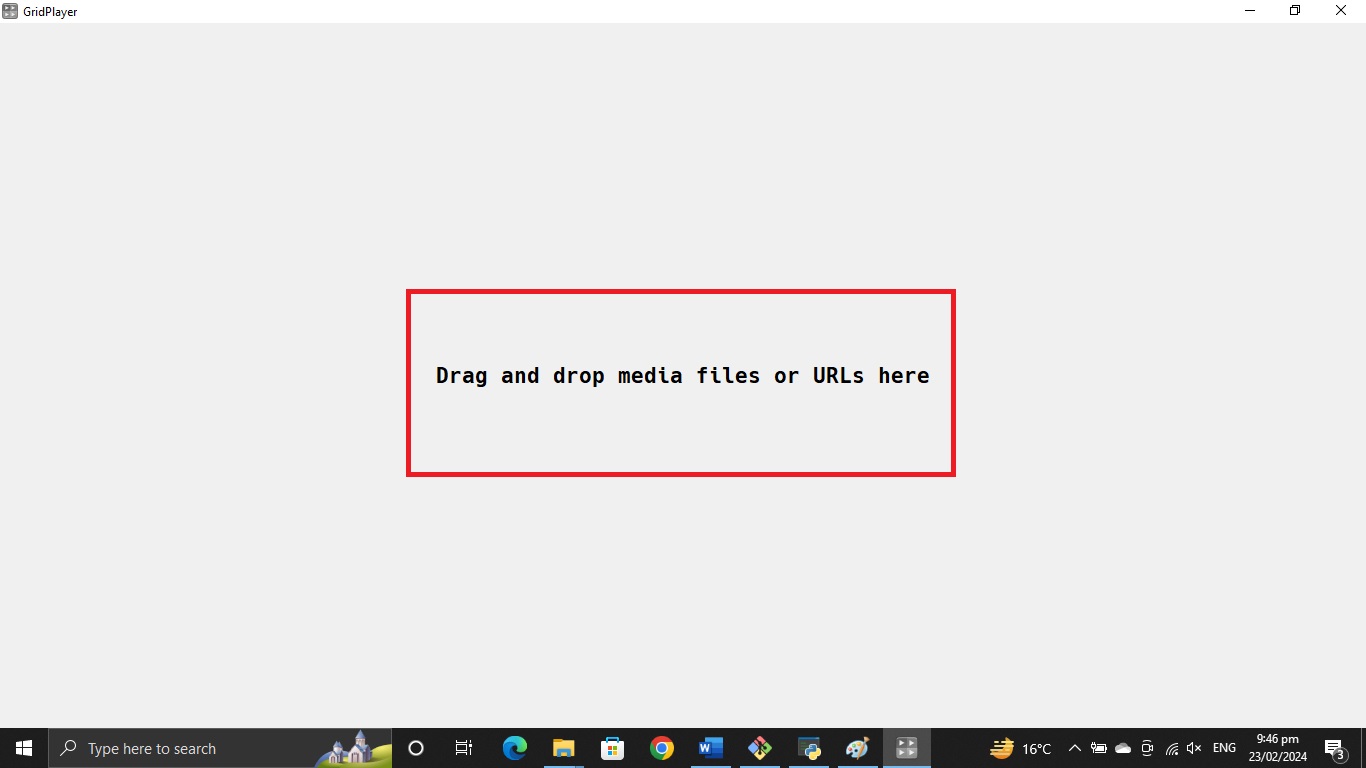
* After all, clear the playlist

****

User Guide 1.7 Deleting Playlist

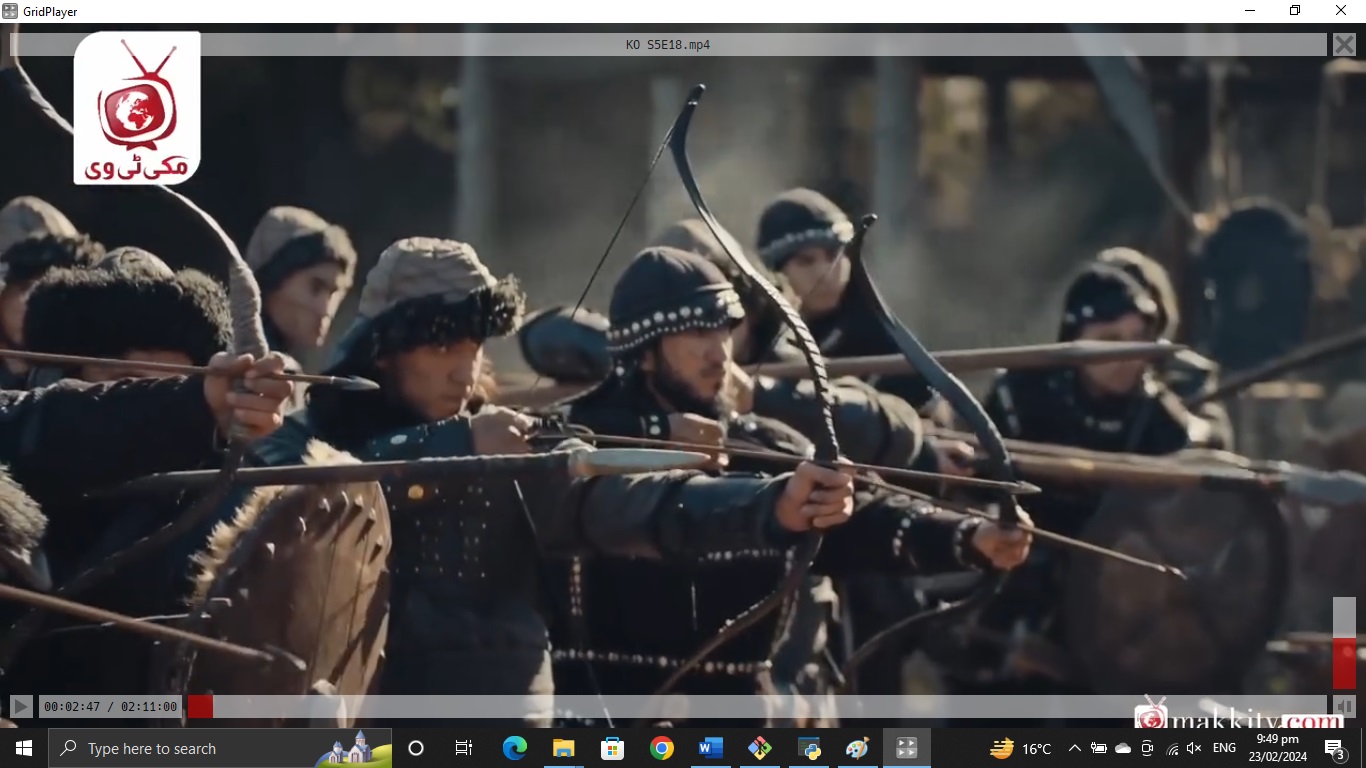
* Click on “Delete Songs”
* Delete song one by one or delete all the songs at one click.

**Video Player Guide:**

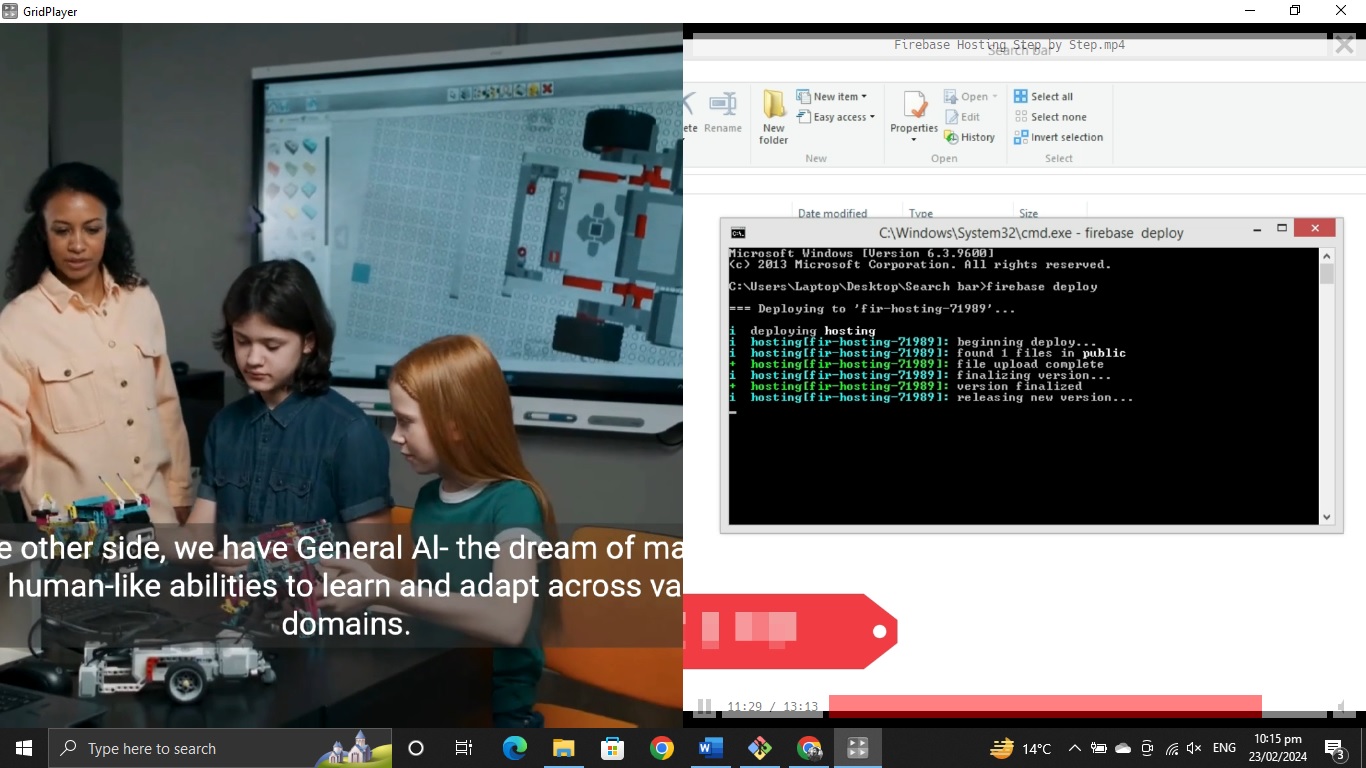
****

User Guide 1.5 Video Player

* Drop your video files/URL’s in the video player area.
* It will start playing the video.
* You can control the flow of video by using the timeline bar.
* Also control the volume.
* Click on the screen to pause/resume the video.

****

* You can play more than one video side-by-side in the GridPlayer.
* Switch between grid video by double click on the video screen

****

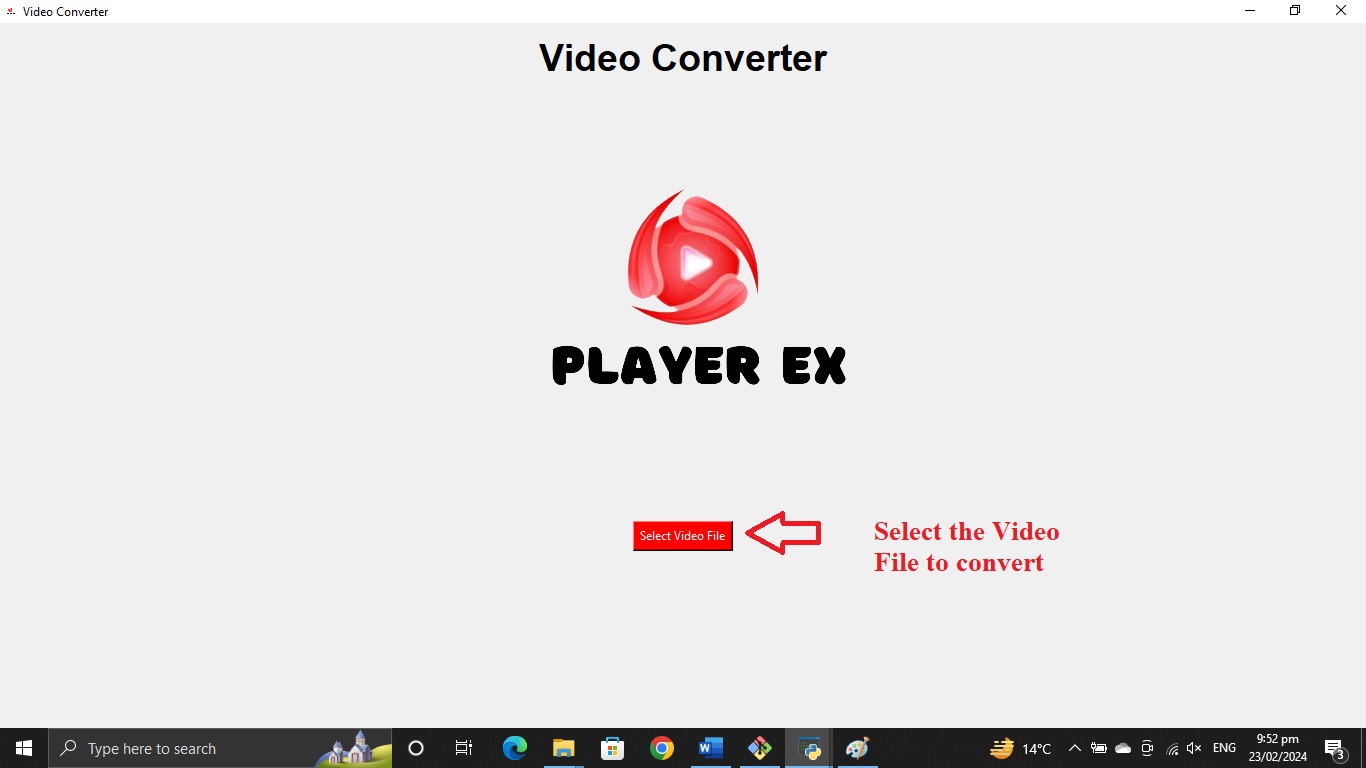
**Audio Clipper Guide:**



User Guide 1.6 Audio Clipper

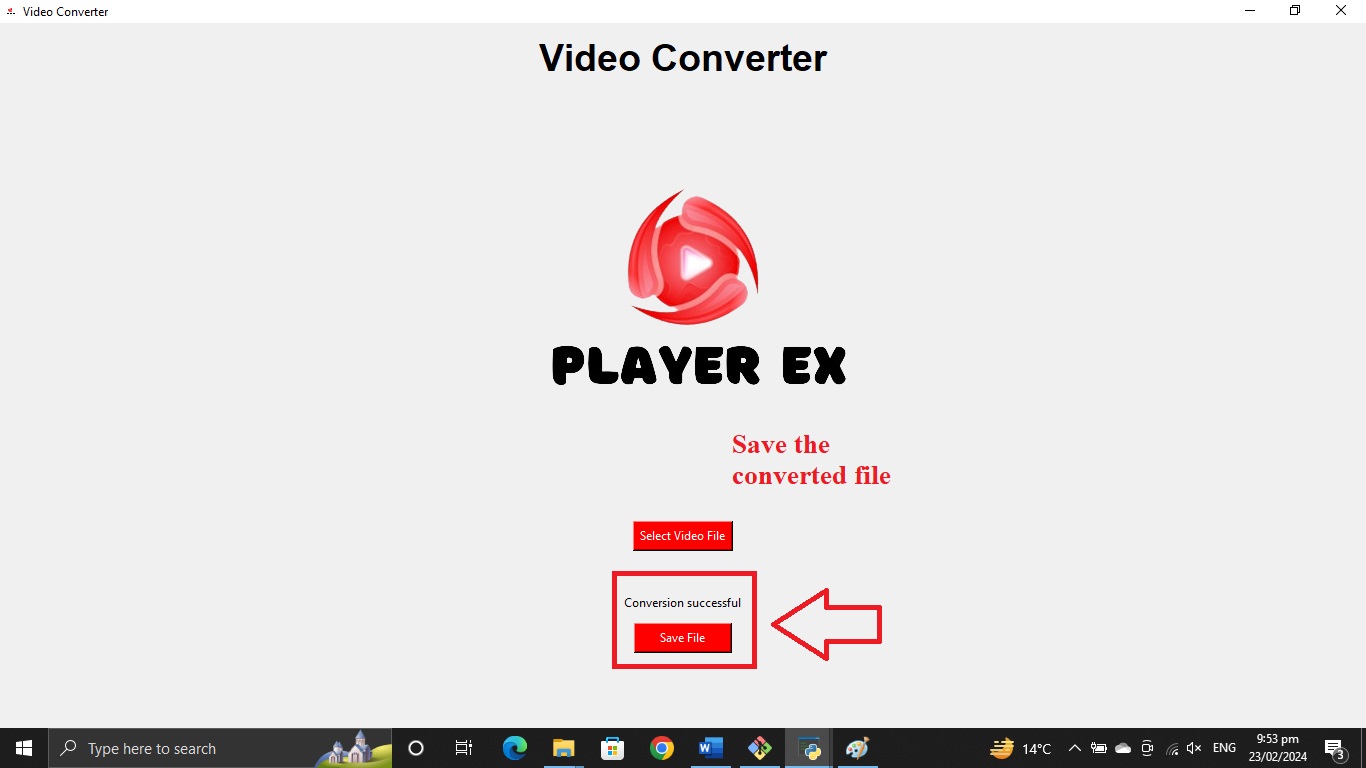
* Click “Browse” button to select the audio file to make a clip
* After clicking on browse, a dialog box will appear.
* Select the audio file from the system and open that file
* After uploading the file, set your “Start Time” and “End time” to make the clip
* After setting-up the time duration, click on “Create Audio Clip”
* After making the clip, a new dialog box will appear
* Save the clip in the system directory.

**MP4 to MP3 Converter Guide:**

****

User Guide 1.7 Video Converter

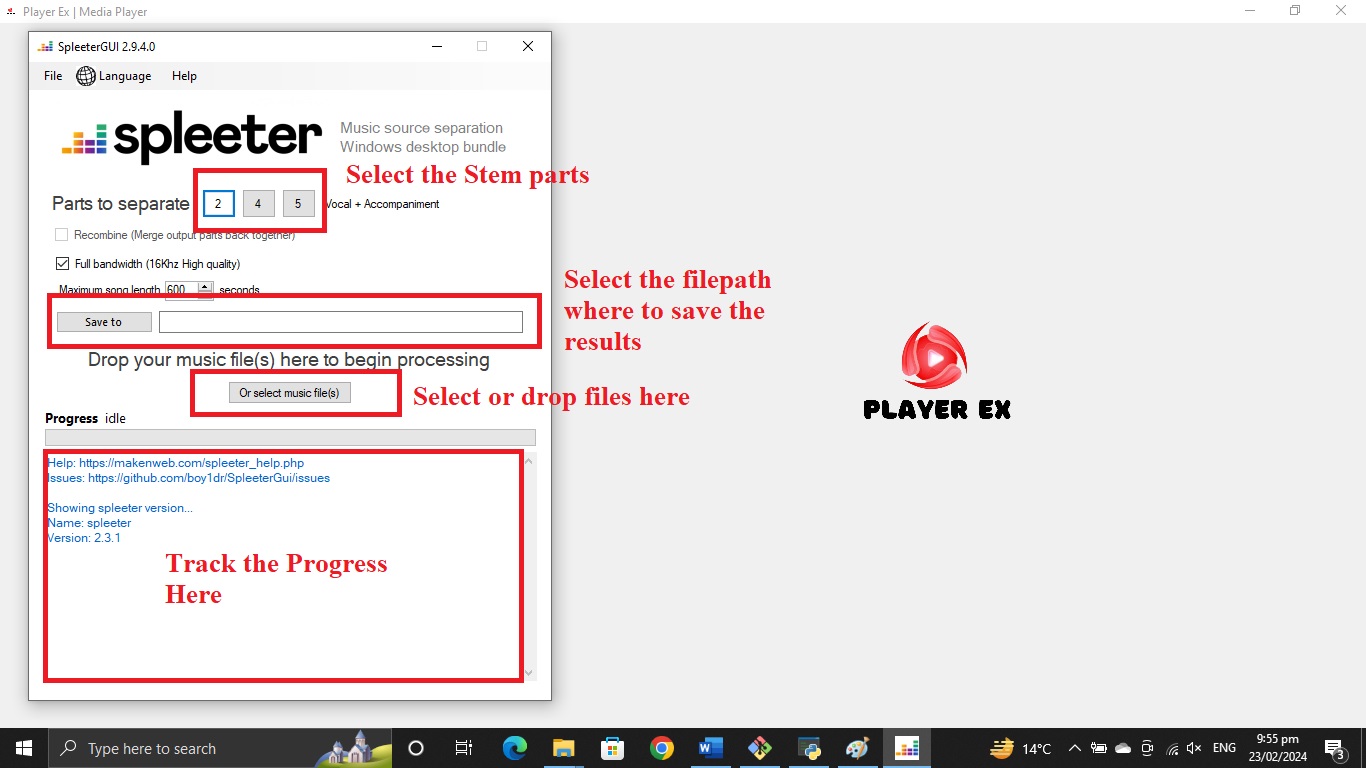
* To convert MP4 file into MP3 file, click on “Select Video File” button
* After that a dialog box will appear
* Select the video to convert it into MP3 format
* After selecting the file, click “Open”
* It will be converted into MP3 format



User Guide 1.8 Save Output Video Converter

* When the process is completed, a message will appear “File is Converted”
* After that, click “Save File” button
* Give the location to the system to store the converted file on computer system.

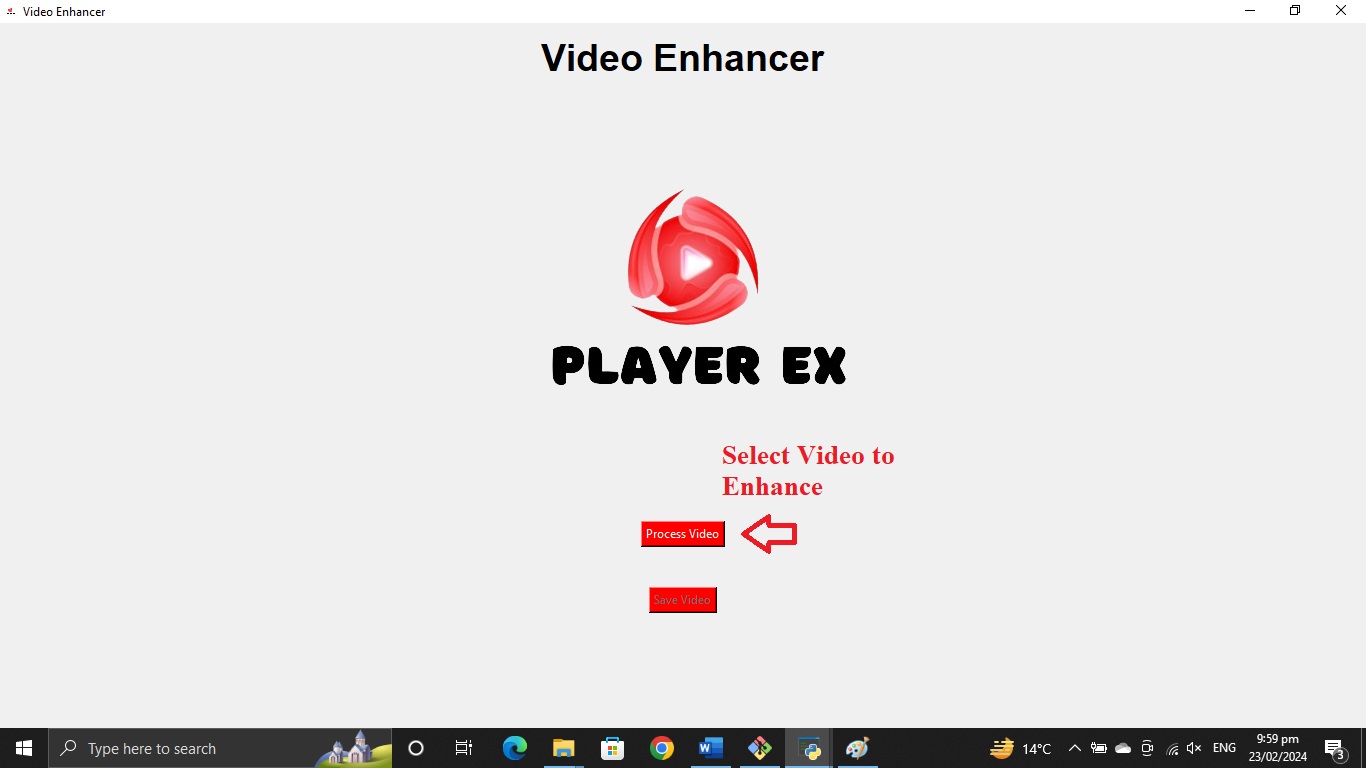
**Vocal Extractor Guide**



User Guide 1.9 Vocal Extractor

* Open the Vocal Extractor from PlayerEx.
* You’ll find a UI window of Spleeter.
* Select/drop your files here.
* Select the stem parts you need to extract from your songs.
* Select the file path where you want to save the results.
* Track your progress in the below mentioned section.
* After your work is done, you can see the result folder in the selected filepath.

**Video Enhancer Guide**



User Guide 1.10 Video Enhancer

* Click Enhance video button in the PlayerEx.
* You’ll find Video Enhancer UI window.
* Here you can select the video that you want to enhance.
* Click on “Process Video” button, you’ll find a dialog box where you can select the video.
* After selecting the video, the processing will be started.
* When the process is complete, a dialog box will appear.
* Save the processed video on your location.