Collaborative AI Audio/Video Editor

Presented to
The Faculty of the College of
Engineering

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Software Engineering

By
Yash Kamtekar
Avinash Ramesh
Nevil Shah

May 2023

By

Yash Kamtekar

Avinash Ramesh

Nevil Shah

APPROVED

`

Ali Arsanjani May 9,2023

Prof. Ali Arsanjani, Project Advisor

Prof. Dan Harkey, Director, MS Software Engineering

Prof. Rod Fatoohi, Department Chair

ABSTRACT

CollaborativeAI Audio/Video Editor (Edit Scape)
By

Yash Kamtekar
Avinash Ramesh
Nevil Shah

Artificial intelligence (AI) is transforming the video production industry, simplifying clip organization and enabling seamless editing. According to a Business Insider survey [1], 78% of marketers had adopted or planned to adopt AI in their video production processes by 2018. AI technology expedites editing workflows and unlocks creative possibilities for projects of various scales, from short films to large-scale television productions. By automating tasks such as video editing, 3D animation, and generating realistic visuals, AI reduces production costs and labor time, allowing content creators to focus on developing compelling content.

In video editing, there are instances where recorded videos require modifications before being uploaded. Traditionally, this involved labor-intensive approaches such as re-recording entire videos or specific segments, followed by merging them with the original footage. This process becomes particularly cumbersome when tasks like removing filler words, redacting PIIs or making precise cuts are involved. To address these challenges, our project aims to introduce an intuitive AI-based audio and video editing tool that offers a comprehensive and user-friendly experience, akin to editing a Google Doc.

Our solution harnesses text extracted from transcriptions, empowering both novice and experienced users to achieve professional precision in video content editing. By incorporating cutting-edge state-of-the-art models and employing transfer learning techniques, our AI tool offers a range of features that streamline the editing process. These features include automatic transcription, one-click removal of filler words, video editing guided by the transcription, video segment trimming, video merging, and identification and redaction of personally identifiable information (PII). By identifying and integrating advanced SOTA models, our ultimate objective is to develop an end-to-end machine learning product that effectively addresses the challenges associated with modern video and audio editing.

Acknowledgments

The authors are deeply indebted to Professor Dr. Ali Arsanjani for his invaluable comments and assistance in the preparation of this study

Table of Contents

List of Figures

List of Tables

Chapter 1. Project Overview

Introduction

The rapid advancements in artificial intelligence (AI) and machine learning (ML) have led to significant breakthroughs in various industries, including the video and audio production domain. With the increasing demand for high-quality content, production companies and individual creators face the challenge of producing professional-grade videos and audio quickly and efficiently. Traditional editing methods can be time-consuming and labor-intensive, creating a need for innovative solutions that streamline the production process. The emergence of AI-based video and audio editing tools offers promising opportunities to automate and optimize the editing process. These technologies are designed to expedite workflows, enhance creative possibilities, and reduce production costs, allowing creators to concentrate on the development of their content. This project aims to develop a comprehensive, user-friendly AI-based audio and video editing tool that leverages advanced algorithms and techniques to facilitate seamless editing experiences for both novice and professional users. The proposed AI-based editing tool will incorporate cutting-edge state-of-the-art (SOTA) models from Google[2], and explore transfer learning for real-time voice cloning from existing text-to-speech research models [3]. By integrating these models, the tool will offer features like automatic transcription, single-click removal of filler words, speaker labeling, and text-to-speech conversion. The project will also focus on optimizing the tool for real-time processing and performance, ensuring its seamless integration into various video and audio production workflows. To achieve these objectives, the project will be divided into several areas of study, including AI-based video and audio editing, transfer learning and voice cloning, user interface and user experience (UI/UX) design, real-time processing and performance, and accessibility and inclusivity. The project will contribute to the advancement of AI-based editing technology, expand the understanding of transfer learning and voice cloning, innovate UI/UX design for AI-based tools, improve real-time processing and performance, and promote accessibility and inclusivity. In summary, the development of a comprehensive AI-based video and audio editing tool promises to revolutionize the production industry by providing an efficient and accessible means of editing content. By leveraging advanced algorithms, techniques, and models, this project aims to create a user-friendly tool that caters to the needs of a diverse range of users and promotes widespread adoption of AI-based editing technologies. This endeavor not only holds the potential to transform the editing landscape but also contributes significantly to the academic understanding of AI-based video and audio production.

Proposed Areas of Study and Academic Contribution

Proposed Areas of Study

1. **AI-based Video and Audio Editing:** Examine how machine learning and artificial intelligence are used in video and audio editing. The creation of algorithms that can automate processes like transcription, filler word elimination, and speaker classification falls under this category.

2. **Transfer Learning and Voice Cloning:** Investigate the possibility for voice cloning and transfer learning in the conversion of text to speech. To enable real-time voice cloning, this entails looking at current models and improving them.

3. **User Interface and User Experience (UI/UX) Design:** Create an intuitive user interface based on artificial intelligence editing tools that enables amateur and expert users to modify both audio and video files swiftly and efficiently.

4. **Real-time Processing and Performance:** Look to explore ways to enhance the AI-based editing tool's performance and real-time processing to ensure effortless integration into different video and audio producing workflows. Inclusivity and Accessibility Look at ways to make the AI-based editing tool inclusive and accessible for a variety of users, such as those who have different degrees of technical expertise or disabilities.

Academic Contribution

Achieve Better instantaneous Processing and Performance:: Provide intelligent advice on how to enhance AI-based editing tools' real-time processing and performance, which will ultimately benefit the video and audio production industry.

Set higher standards for AI-based editing software: Lift the bar higher technology by helping to create new algorithms and methods that enhance the efficacy and productivity of AI-based video and audio editing.

UI/UX Innovation for AI-based Tools: To promote the wider use of AI-based editing tools, offer suggestions on how to design user-friendly interfaces as well as experiences that are agreeable to clients with different levels of expertise in the field.

Increase Your Awareness of Voice Cloning and Transfer Learning: Improved knowledge of voice cloning and transfer learning methods in the context of text-to-speech conversion could result in more accurate and lifelike voice reproduction.

Current State of the Art

This section provides an overview of the current state of the art in the fields of automatic transcription, voice cloning, speaker diarization, and collaborative video editing. We will discuss recent advancements, technologies, and research papers that are relevant to the development of our proposed AI-based audio and video editing tool.

1. **Voice Cloning**: Incorporated deep learning algorithms with customized voice cloning, which has resulted in the creation of extremely realistic-sounding synthesized speech. Among the notable examples, Google's Tacotron 2 model [4] applied a sequence-to-sequence structure to transform text to mel-spectrograms, followed by a WaveNet-based vocoder to produce speech, and Baidu's Deep Voice [3] implemented deep neural networks to produce excellent voice clones. As the cutting-edge voice cloning solution for our project, we have chosen the Tortoise TTS model, which employs cutting-edge machine learning algorithms to produce customized synthetic speech with astounding precision and naturalness.

2. **Speaker Diarization**: Speaker diarization entails identifying and grouping audio streams according to the identities of the speakers. Applications like multi-speaker transcription and video editing depend on this procedure. "Deep Speaker Embeddings for Diarization" [5], a significant paper in this area, describes an approach that collects deep speaker embeddings via neural networks to enhance diarization performance. "End-to-End Neural Speaker

Diarization with Permutation-Free Objectives" [6] is a further pertinent research study that proposes an end-to-end trainable neural network for speaker diarization that does away with the requirement for post-processing heuristics.

3. **Automatic Transcription:** Automatic transcription, often known as speech-to-text conversion, has significantly advanced thanks to deep learning and neural network-based techniques. Both Google's Speech-to-Text API and OpenAI's Whisper [1] are famous instances of artificial intelligence (AI) systems that have achieved outstanding performance when transforming audio into text. Although we initially tested using OpenAI's Whisper model to implement our automatic transcription feature, we eventually decided to use Google's Speech-to-Text API because of its accuracy and speed as well as its ability to provide transcriptions in a variety of languages, making it an appropriate choice for many different applications, including video and audio editing.

4. **Lip-Syncing video**: Lip synchronization has long been a difficult challenge in video creation. Deep learning has recently made strides, providing new ways to approach this problem. One such approach is GANimation, which animates random lip motion using a generative adversarial network. However, in our effort, we have chosen to use the Wav2Lip model, which explicitly maps audio to lip motion using a fully convolutional network. This method has produced astonishingly accurate lip-to-audio synchronization results. The Wav2Lip model is a flexible option for lip syncing in video creation because it has proven to be resilient in handling a variety of faces and languages.

5. **Video Concatenation:** The technique of combining multiple videos into a single, uninterrupted video is known as video concatenation. There are several methods for concatenating videos, including using conventional video editing applications and deep learning-based computer solutions. The Adversarial Video Concatenation (AVC) model, put forth by academics at the University of California, Berkeley, represents a single such deep learning-based strategy [1]. The AVC model combines videos while maintaining their content and motion using a generative adversarial network (GAN). Unsupervised Video Concatenation (UVC) is a noteworthy model that was developed by researchers at the University of Southern California [2]. The UVC model identifies spatial and temporal trends

in films and concatenates them while maintaining semantic coherence via unsupervised learning. But for our task, we discovered using the Transcoders API given by Google Cloud Platforms was the most efficient way to develop a feature.

In conclusion, the state of the art in areas like speaker diarization, automatic transcription, voice cloning, and collaborative video editing shows how quickly AI and ML technologies are developing. We want to offer a complete approach that improves the editing process and encourages user cooperation by harnessing these advancements and incorporating them into our suggested AI-based audio and video editing tool.

Chapter 2. Project Architecture

Introduction

The AI-based audio and video editing tool's project architecture, which specifies the system's structure, components, and interactions, is an important part of the overall design. The project intends to offer users a seamless and effective editing experience by employing a modular and scalable design. With an emphasis on the addition of Firebase services from Google Cloud Platform (GCP) and the numerous modules involved in enabling the needed functionality, this chapter will provide a brief summary of the project architecture.

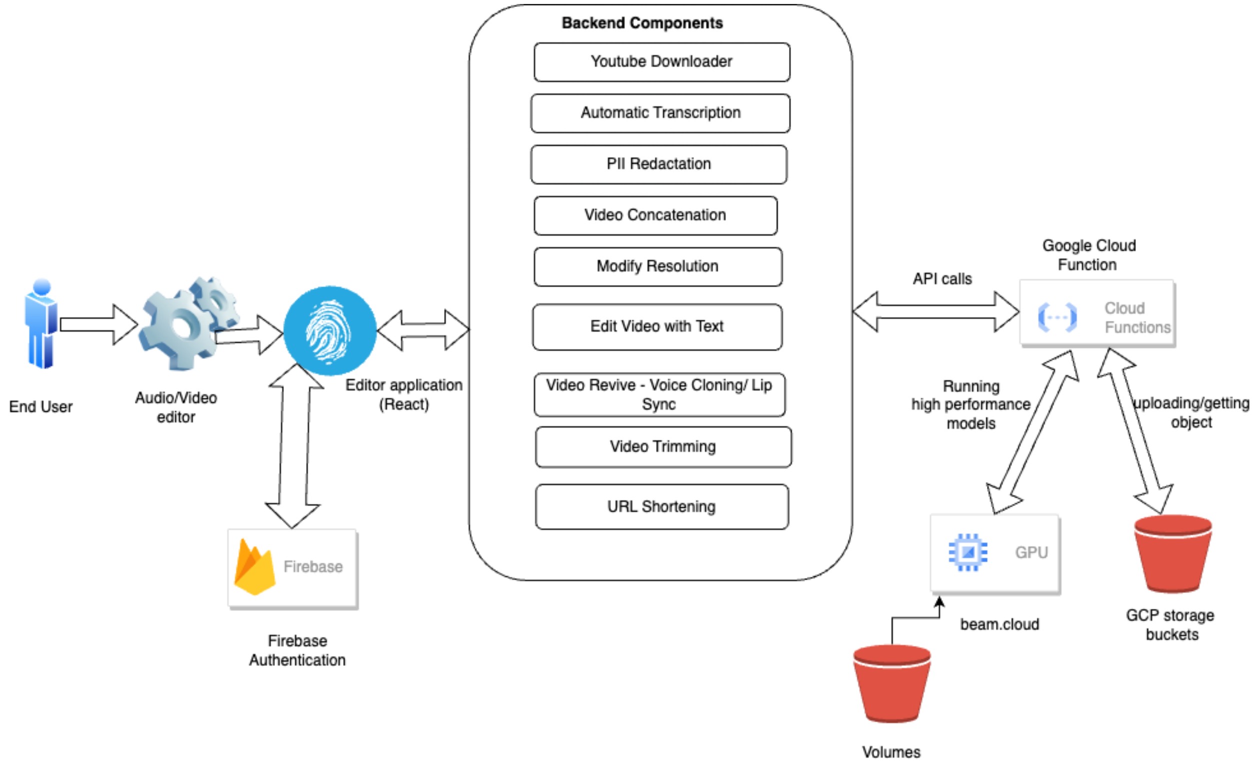


Fig 1. Project Architecture

Frontend Subsystem

The AI Editor's frontend is made up of a number of subsystems that let clients engage with the program and carry out various video editing operations. The capabilities of the Preview of the Video., Transcription Editor, and Share & Export subsystems are among them.

Visitors are able to preview the video and listen to the audio through the Video Preview subsystem. It offers options for playing, stopping, and navigating the timeline of the video. For an even more seamless editing experience, viewers may also click on individual transcripts in the Transcription Editor, and the video will search for the correct timestamp.

An intuitive user interface is offered by the Transcription Editor subsystem for altering and maintaining transcriptions. Users are able to combine numerous films with their accompanying transcriptions, delete or change transcriptions, read and edit individual transcriptions, and more. The editor also supports context menu options, such as word selection and deletion.

Users can share and export their altered videos using the Share & Export feature. It offers options of downloading the transcriptions in SRT format for later use, sharing the video, and exporting it as an independent video file.

Users may simply trim, eliminate silences, delete filler words, change resolution, and do other editing tasks because of the seamless coordination of various subsystems, which offers them an intuitive and effective video editing experience.

Backend Components

The proposed AI-based audio and video editing tool's architecture may be broken down into various subsystems, each of which is in charge of a different part of the tool's functioning. To maintain the smooth running of the entire system, these subsystems communicate and function together. The main subsystems and their functions inside the project design are described in the sections that follow.

1. **Youtube Downloader:**

Using this updated feature, viewers can now download videos by only providing a YouTube URL. Due to the fact that users don't need to look for external downloaders or travel through multiple download options, this feature streamlines the process of downloading and saves them time. The capability improves user experience by offering a simple, one-click method of saving videos.

2. Automatic Transcription Subsystem:

With the help of Google Speech-to-Text technology, our application features a speech-to-text transcription capability. With the help of this technology, users can quickly turn spoken audio into text, making audio material more accessible and searchable. Now, users only have to upload or capture audio recordings, and the program will convert the speech into text that can be altered or downloaded for use elsewhere.

3. Personalized Voice Cloning and Lip Syncing:

Our application now contains a powerful new function that allows users to substitute a word in the transcript with an individual's personalized voice utilizing the most recent Tortoise TTS model. We've also added Wav2Lip technology, which automatically syncs a video's lips to the new audio to create a seamless new video output. This feature provides users with an exciting new level of control over their audiovisual content, allowing for enhanced personalization and customization.

4. Speaker Diarization Subsystem:

The speaker diarization subsystem identifies and segments audio streams based on speaker identity, facilitating multi-speaker transcription and video editing. It utilizes advanced neural network-based approaches, such as the deep speaker embeddings method or end-to-end neural speaker diarization, to achieve accurate and efficient speaker segmentation.

5. Personally Identifiable Information(PII) redaction:

The power of the GPT-3.5 Turbo model is currently being used by our application's enhanced function to find personally identifiable information (PII) in audio files. Users can choose which types of PII, like their names, addresses, or personal phone numbers, should be searched for. The matching segment is immediately silenced to safeguard confidential data when the model detects PII in the audio. The muted audio is also smoothly merged into a new video output that is created. This feature provides users with a powerful tool for protecting their personal data and ensuring the privacy of themselves and others.

6. Video Concatenation:

With the addition of this useful new functionality, our application can now combine two clips into a single output file. Users may choose and upload the clips that they want to merge with only a few clicks, and the software will seamlessly stitch them together into a new video.

7. Modify Resolution:

A flexible new tool for changing the quality of video files is now part of our application. With this function, users can choose the output resolution they want for their videos, and the program will adjust the video's size for them.

8. Video Trimming:

Users can simply trim video clips in our application to a precise start and end time using our strong new functionality. Users may choose the part of the video they want to save with a few quick clicks, and the software will automatically cut out any material outside of that range. This tool is especially helpful for deleting unneeded portions of a video, including outros or openings, or for separating out individual pieces for additional editing or analysis.

Intuitive and user-friendly interfaces are combined with strong machine learning algorithms in our program to create a smart video editing tool. We've used a variety of cutting-edge technologies and cloud-based solutions to achieve this. Our application's front end is constructed using React, offering a slick and responsive user interface. In order to power the functionality of the application, we created a number of APIs for the backend using Python and Flask. We've installed our models on beam.cloud, which gives users access to potent GPUs for quick and effective processing, to assure high-performance computing capabilities. We have incorporated Firebase for user authentication, ensuring secure and dependable access to the application's functionalities. Finally, we've used GCP buckets to store video inputs and outputs, giving our consumers scalable and dependable storage options. These technologies work together to produce a fluid and potent video editing experience that distinguishes our application.

Chapter 3. Technology Descriptions

In this chapter,we give an in-depth overview of the technologies utilized in the creation of the AI-based audio and video editing software. Client Technologies, which manages the front-end graphical user interface and user experience, and Data Tier Technologies, which take care of the backend storage, processing, and connectivity with Firebase and Google Cloud Platform (GCP) services, are the two main groups into which the technologies are split.

Client Technologies

1. **React**:

Facebook created and maintains the open-source React JavaScript library, which is mostly employed for creating user interfaces. React encourages the development of reusable user interface components and gives programmers the ability to effectively control the state of their apps. React is used as the main frontend technology in this project to create a responsive, understandable, and readily available UI for the AI-based editing app.

2. **Redux**:

JavaScript applications can use the Redux state management library, which is frequently combined with React. It makes it simpler to manage and troubleshoot complicated systems by allowing developers to keep their programs in a centralized and predictable state. Redux is used in this project to control the state of the editing app, guaranteeing a reliable and effective user experience.

3. **Material-UI:**

Material-UI is a popular React UI framework that implements Google's Material Design guidelines. It provides a large variety of already-built elements and stylistic options, enabling developers to design user interfaces that are dependable and aesthetically pleasing. The user interface in this project is designed and styled using Material-UI, giving it a contemporary and polished appearance.

Data Tier Technologies

1. Python:

Python is a powerful and versatile programming language that has been used extensively in our application to develop robust and efficient backend functionality. Python has made it possible for us to create and deploy exceptional APIs within Google Cloud Functions quickly thanks to its straightforward and clear syntax, an extensive collection of third-party modules, and widespread industry acceptance.

2. Firebase Authentication:

A safe and efficient method for user authentication and authorization is provided by the Firebase Authentication service. It offers a number of different authentication techniques, including single sign-on (SSO), social network login, and email and password. To manage user authentication for the editing tool in this project, Firebase Authentication has been used.

3. Flask:

The development of our application's APIs relied heavily on Flask, a compact and adaptable web framework for Python. We have been able to swiftly construct and deploy endpoints for our numerous backend services because to Flask's basic design and straightforward structure, all while preserving an extensive amount of versatility and customization. We were able to develop a scalable and reactive backend infrastructure using Flask that enables us to efficiently manage and process large amounts of data within the cloud.

4. Cloud Storage:

Cloud Storage is a service provided by Google Cloud Platform that offers scalable, durable, and secure storage for large files, such as video and audio content. In this project, Google Cloud Storage buckets are utilized to store and manage the large media files involved in the editing process, ensuring high availability and performance.

5. Beam.Cloud

Our solution makes use of beam.cloud, a cloud-based system that gives users access to potent GPUs for high-performance computation, to deliver the customized voice cloning capability. For a seamless and authentic user experience, we've used two cutting-edge models: Wav 2 Lip for lip synchronization and Tortoise TTS for customized speech cloning. We've created APIs that call the models and produce the output in order for integrating these models with the rest of the project. With this strategy, we can guarantee that our program will produce accurate, high-quality results while still being user-friendly and effective. We can access the necessary computational power to deliver an element that stands out and differentiates our project by using beam.cloud.

6. Google Cloud Platforms functions:

For the backend APIs, we're capable to develop a very scalable and economical solution by utilizing GCP functions. Our main programming languages are Python and Flask, which allow us to create clear, effective code that guarantees dependable performance. When necessary, these APIs are then invoked from the React frontend pages, resulting in a fluid and simple user interface. Additionally, we can benefit from the scalability and dependability of Google's cloud storage offering by saving result videos in GCP buckets. By only charging for the resources we really use, this strategy not only assures efficient computing but also assists us in lowering the cost of our infrastructure. Overall, we were able to build a solid and effective backend that supports our product by employing GCP capabilities.

7. Other Services

The project includes implementing cutting-edge capabilities including automatic transcription, voice cloning, diarization of speakers, deleting filler words, cutting and trimming films, and many other services leveraging a variety of GCP products and services, including Google's Speech-to-Text API and other AI and ML APIs. These services offer strong, scalable solutions that fit neatly within the project's architectural design.

React, Flask, and other Firebase and GCP services are just a few of the latest client and data tier technologies that are going to be used by the AI-based audio and video editing tool in order to create a polished, effective, and user-friendly editing solution. The project can be readily maintained, enlarged, and modified to take into account future breakthroughs in AI and ML research thanks to the use of these technologies.

In order to provide a seamless and effective user experience, our video editing app is a complex program that makes use of a variety of cutting-edge technology. React was used in the front end development to create a responsive and user-friendly interface, and Python and Flask were used in the back end to execute the application's functionality. We installed the models in beam.cloud to assure outstanding performance processing capabilities, and we incorporated Firebase for user authentication. Finally, we've used GCP buckets to store video inputs and outputs, giving our consumers scalable and dependable storage options. With these innovations, our application distinguishes itself as a potent and straightforward video editing tool.

Chapter 4. Project Design

The plan for the project will be used to create the AI-based audio and video editing app. In order to provide the anticipated features and user experience, it describes the structure, elements, and interactions required. The project design is presented in this chapter, with a particular emphasis on two key facets: the Client Design, which describes the UI as well as user experience on the front end, and the Data-Tier Design, which describes the processing, storage, and integration of Firebase and Google Cloud Platform (GCP) services on the back end.

Client Design

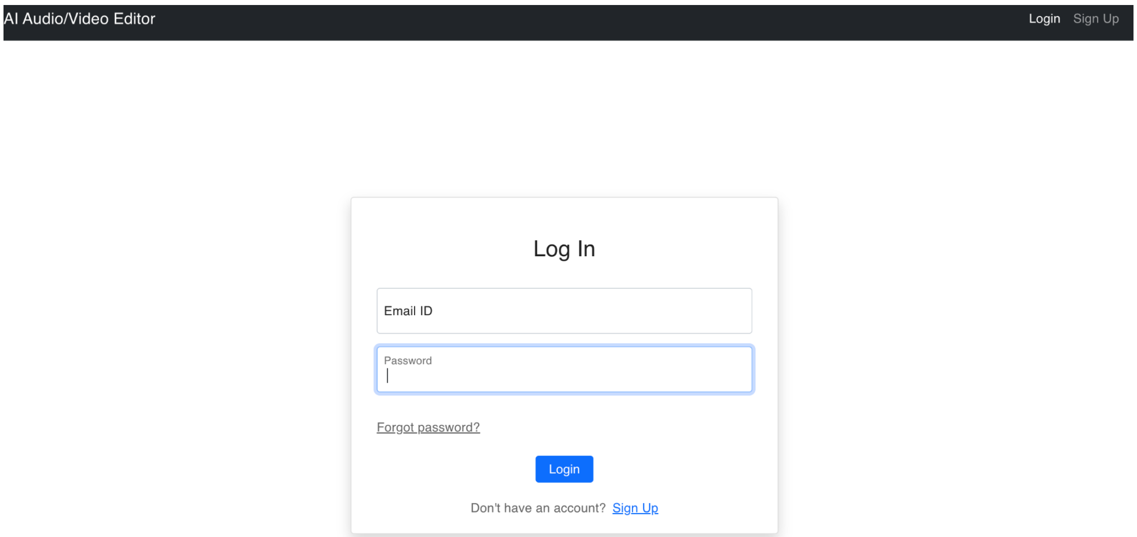
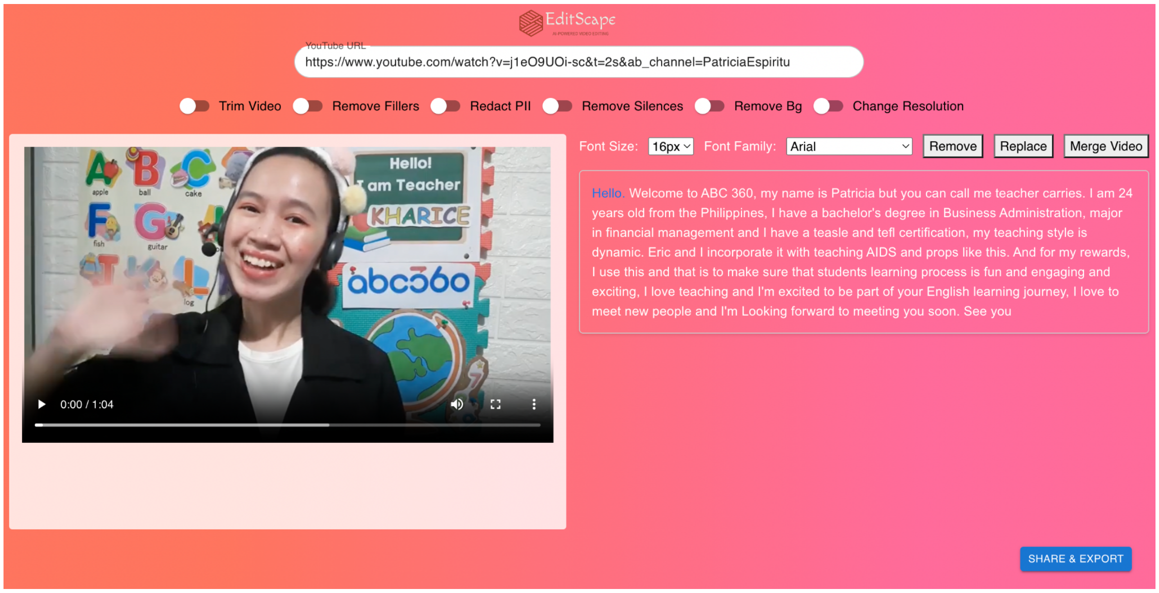


Fig 2. Home Page

Viewers are able to perform operations on the video editor page by clicking the login/signup link in the navigation bar on the home page. Users will be taken to the video editing page from where they can commence altering their videos using the variety of options available after properly authenticating.



https://www.youtube.com/watch?v=j1eO9UOi-sc&t=2s&ab_channel=PatriciaEspiritu

Font Size: 16px
Font Family: Arial

Hello. Welcome to ABC 360, my name is Patricia but you can call me teacher carries. I am 24 years old from the Philippines, I have a bachelor's degree in Business Administration, major in financial management and I have a teasle and tefl certification, my teaching style is dynamic. Eric and I incorporate it with teaching AIDS and props like this. And for my rewards, I use this and that is to make sure that students learning process is fun and engaging and exciting, I love teaching and I'm excited to be part of your English learning journey, I love to meet new people and I'm Looking forward to meeting you soon. See you

Fig 3. Video Editor page

Users may easily alter videos thanks to the editor page's user-friendly interface. Users may type a YouTube URL in the search box at the very center of the webpage in order to start editing. The resulting video based on the feature that has been applied is displayed in a section on the left hand side of the page. Users can choose the specific characteristics they would like to apply to their content by using the toggle buttons for various features that are located beneath the YouTube search URL.

The transcript of the resulting clip is shown on the right center side of the website, which might be useful for those who wish to confirm the precision of the transcription. At the bottom right hand side of the page, there are options to share and export the video once the editing is completed.

In general, the editor page offers a simple and effective approach to modify films, with controls that are simple to use and outcomes that are clearly displayed. It is simple to make modifications as needed to produce the ideal video because users can easily apply different elements and observe the results in real time.

User Interface (UI) Design:

We have taken considerable care to construct an easy to use and visually pleasing user interface for our editing tool product. This was made possible by our use of the Material-UI framework and React components, which allowed us to adhere to Google's Material Design standards and guarantee a contemporary and unified appearance. The navigation bar, media timeline, editing tools, and real-time collaboration features are all key components of our user interface. We're able to design an intuitive user interface that helps users to accomplish their editing goals fast and effectively by smoothly combining these features. As a result, users can concentrate on their creative vision and produce material of a high caliber using an editing tool that offers an amazing user experience.

User Experience (UX) Design:

In order to produce a video editing app that is simultaneously user-friendly and effective, we have focused heavily on user experience design in this project. In order to enable users to simply play around with the editing tool, we implemented intuitive and context-sensitive controls. We've also created guided processes that expedite the editing process and reduce the learning curve to help users complete typical editing jobs. In addition, regardless of the user's level of technical proficiency, we've given priority to responsiveness and performance to make the editing tool quick and effective. With the help of these design components, we were able to produce a user-friendly video editing application that offers a fluid and simple user interface.

Accessibility and Responsiveness:

The client design must be responsive and accessible in order for the editing tool to function properly on a variety of platforms and devices. By utilizing responsive design concepts, the project will enable the interface to adjust to various screen dimensions and resolutions. Additionally, in order to accommodate users with impairments, accessibility features including keyboard navigation, screen reader assistance, and alternative text for media assets will be introduced.

Data-Tier Design

1. API Design:

Flask and Python are at the core of the API design, which enables the creation of effective and simple-to-maintain backend APIs. A RESTful API that handles accounts for users, project files, and editing histories is included in the editing tool. These endpoints are arranged into logical resources, resulting in an API structure that is consistent and simple to comprehend. API documentation is also used to increase API maintenance and speed up the development process.

2. Data Modeling and Storage:

Since they specify the arrangement and structure of the data utilized by the editing tool, data modeling and storage are essential elements of the data-tier design. We will use Firestore, a flexible and adaptable NoSQL database offered by Firebase, to store project files, metadata, and modification history. The data model will be created to facilitate effective searching, indexing, and synchronization in real-time, providing outstanding efficiency and collaboration abilities.

3. Integration with Firebase and GCP Services:

To make use of its cutting-edge capabilities and scalability, the editing will be intimately connected with a number of Firebase and GCP services. Large media file storage and management are being handled by Cloud Storage, while secure user authentication and authorization will be handled by Firebase Authentication. Additionally, the project uses GCP services to achieve functionality like automated transcription, personalized voice cloning, and many more features. These tools include Google's Speech-to-Text API and other AI and ML APIs.

4. Graphics Processing Unit(GPU)

With our project, we hope to provide customers a fluid and modern video editing experience. Customized voice cloning, which includes creating artificial speech that imitates the voice of a particular speaker, is one of the features we created to accomplish this goal. We needed access to strong GPUs for high-performance processing in order to implement this feature. For a realistic and engaging user experience, we decided to employ beam.cloud, a cloud-based platform that offers this computing power and enables us to run two cutting-edge models, Tortoise TTS and Wav 2 Lip. We were able to present an asset that distinguishes our project from others and gives users precise, superior outcomes by utilizing the capabilities of beam.cloud.

5. Google Cloud Platform services:

We've created a backend API gateway using GCP functions to manage the multiple functionalities of our video editing application for the data tier stage of our project. This method ensures excellent speed and a smooth user experience by giving us the option of expanding our backend in response to customer demand. Additionally, we have decided to keep both the input and the output files in GCP buckets because they provide excellent scalability, availability, and durability. We've created a data tier using these GCP services that offers a dependable, scalable, and effective solution for our application.

In conclusion, this project's architecture places a strong emphasis on fusing cutting-edge client methods with data-tier advances to produce a complete, user-friendly, and efficient editing solution. By carefully examining the UI, UX, API design, modeling of data, storage spaces, and connectivity with Firebase and Cloud services, the project aims to develop a robust and adaptable editing software that can be easily kept, and modified to meet future advancements in AI and ML research.

Chapter 5. Project Implementation

The AI-based audio and video editing tool's execution phase is an important stage in its development because it entails turning the overall design template into an entirely operational and unified system. The Client Implementation, which includes the creation of the frontend UI and user experience, and the Data-Tier Execution, which includes incorporation with Firebase and cloud services, are the main topics of this chapter's thorough overview of the setup process.

Client Implementation

1. User Interface (UI) Development:

The creation of an extensible framework of reusable React elements that are divided into functional portions is a step in the procedure of developing user interfaces. Each component is independently created and incorporated into the main application. To maintain a consistent and contemporary design and to implement popular UI features like dialogs, tooltips, and progress indicators, the Material-UI framework is widely utilized. To improve the visual appeal and uphold brand identification, certain styling and theming are used.

2. User Experience (UX) Development:

Usability testing and iterative design are given a lot of attention while creating the user experience. Polls, interviews, and usability tests are used to get feedback from potential users, which is then used to improve the ui and interface design. To enhance the editing workflow, context-sensitive controls are used, such as flexible processing toolbars and interactive media previews. The usage of performance optimization techniques like virtualization and lazy loading helps to guarantee a responsive and fluid user experience.

3. Accessibility and Responsiveness Implementation:

A variety of methods and best practices are used to implement responsiveness and accessibility. A reactive layout that adjusts to various dimensions and resolutions is made using media queries. To make UI elements more compatible with screen readers and adaptive technology, ARIA attributes and responsibilities are applied. To make sure every capability is usable with just the keyboard, focus management and keyboard navigation are incorporated. In order to improve access for people with visual impairments, media assets are also given with alternative words and captions.

Data-Tier Implementation

1. API Development:

Using Flask, a modular architecture of logically arranged endpoints for the API is built during the construction of the API. Clients, projects, and editing histories are just a few examples of the resources that have been individually created and incorporated into the main application. Endpoint versioning is used to support upcoming API improvements and to assure backward compatibility. Authentication, handling of errors, and request validation are common duties that are handled via middleware and custom exception handlers.

2. Data Modeling and Storage Implementation:

The data modeling and storage implementation process involves creating a schema for the Firestore NoSQL database, which defines the structure and organization of the data used by the editing tool. Collections and documents are designed to support efficient querying, indexing, and real-time synchronization. Cloud Firestore security rules are implemented to enforce data validation, access control, and data integrity. Real-time data synchronization is achieved using Firestore's snapshot listeners, which automatically update the frontend application when changes occur in the database.

3. Integration with Firebase and GCP Services:

The editing tool uses certified client library components and APIs to interface seamlessly with a number of Firebase and GCP services. The Firebase SDK is used to set up Firebase authorization. Large multimedia files are kept in GCP storage, and certified URLs are used to allow direct user file uploads. The Google Cloud Client Libraries are used to seamlessly authenticate users and handle API requests for GCP services like Google's Speech-to-Text API and other AI and ML APIs.

In conclusion, the AI-based audio and video editing tool's project implementation phase entails a thorough and meticulous approach to designing and integrating frontend and backend components, conforming to the project's design blueprint. The project intends to provide a robust and flexible editing tool that is intuitive, effective, and adaptive to future breakthroughs in AI and ML research by concentrating on the customer's execution and data-tier design.

To assure seamless connection and interoperability between sections, frontend and backend engineers keep continual communication throughout the implementation phase. To verify the accuracy, and efficacy of the developed functionality, code inspections and unit testing are used. In order to enable quick iteration and feedback, continuous integration and deployment (CI/CD) pipelines are built up to continuously build, test, and deploy new versions of the application.

The project team also keeps a close eye on any security and privacy issues during the deployment phase. To find and fix vulnerabilities, secure coding procedures are used, and frequent security audits are carried out. When designing and implementing the processing and storage of data components, privacy concerns and adherence to pertinent laws, like the GDPR and CCPA, are taken into account.

The project team is dedicated to developing a culture of cooperation, creativity, and continuous development in order to guarantee the effective execution of the editor tool. The project aims to develop a modern editing system that transforms the manner in which audio and video material is produced and consumed by carefully analyzing and addressing the various needs of clients, those who matter, and the larger industry.

Project Features:

Users may effectively manage and edit their video footage with the help of a variety of strong capabilities that our project provides. Our application includes a full range of video editing options, from the ability to grab films from YouTube and reduce URLs to more sophisticated features like speaker diarization, automatic transcription, and customized voice cloning. To further improve the user experience, we've also added capabilities including video quality change, video trimming and chopping, and SRT file downloading. Our project offers a strong and adaptable solution for anyone wishing to edit, organize, and enhance their video content with the help of these capabilities.

Table 1: Project Features

Sr. No	Feature
1	Youtube Video Downloader
2	Automatic Transcription (Using Video Model from Google)
3	Speaker Diarization
4	PII Redaction (Using Text Davinci Model from OpenAI)
5	Video Concatenation
6	Modify Resolution
7	Personalized voice cloning (using Tortoise TTS model)
8	Lip Syncing Video (using Wav 2 Lip model)
9	Video trimming
10	Video Cutting
11	URL Shortening
12	Download of SRT file

Chapter 6. Testing and Verification

The AI-based audio and video editing tool's testing and verification phase is a crucial step in the development process. In order to guarantee the quality, dependability, and security of the application, this chapter provides a thorough overview of the numerous testing and verification approaches and best practices used throughout the project.

1. Unit Testing:

The method of testing each component or unit of the application separately is known as unit testing. In order to test React components for the frontend and make sure they render and perform as intended, Jest is combined with the React Testing Library. Pytest is used to test business logic and API endpoints on the backend. Additionally, we entered various inputs into Google Cloud Functions to guarantee that we received the appropriate output, and exceptions are issued if the result is not in the desired format.A more concentrated and dependable testing environment is made possible by the use of mock objects and fixtures to isolate parts and simulate dependencies. Unit tests are continuously updated and expanded as new features are implemented, ensuring comprehensive test coverage.

2. Integration Testing:

Integration testing focuses on examining how various application components interact with one another to make sure they function properly as a whole. To mimic user interactions with the frontend application and confirm that the frontend and backend components are properly integrated, end-to-end (E2E) tests are created using the Cypress testing framework. Pytest is also used to create API integration tests, which examine the interaction between the API backend and external services like Firestore and the Google Cloud Platform APIs. These evaluations aid in detecting problems with data flow, API agreements, and third-party service interfaces.

3. Performance Testing:

Performance testing is done to assess how responsive, scalable, and stable the program is under different workloads. In the near future, load testing technologies like JMeter and Locust will be used to simulate concurrent user interactions, gauge application response times, and assess resource usage. These tests will support efforts at optimization by identifying performance bottlenecks, such as slow database queries or ineffective algorithms. The application's ability to manage anticipated user loads and conform to established performance requirements will also be confirmed through performance testing.

4. **Accessibility Testing:**

To guarantee that the application complies with pertinent accessibility standards, such as WCAG 2.1, and can be used by persons with impairments, accessibility testing is conducted. To find potential accessibility issues like missing ARIA characteristics, wrong heading structures, or insufficient color contrast, automated accessibility testing tools like Axe and Lighthouse are used. The application's usability for users with impairments is also tested manually using assistive technology like screen readers and keyboard navigation.

5. **Security Testing:**

To find and address potential security flaws in the application, security testing is done. To find typical security flaws like cross-site scripting (XSS) or SQL injection vulnerabilities, we want to use automated security scanning tools like OWASP ZAP and Snyk. To ensure adherence to secure coding practices and to spot any security concerns, routine code reviews and security audits are carried out. To assess an application's resistance to possible threats, penetration testing also involves simulating actual attacks.

6. **Usability Testing:**

We analyze the user experience of the program and pinpoint its weak points through usability testing. While we watch and record their interactions, test subjects—who correspond to the intended user base—are asked to accomplish a variety of tasks using the program. We gather participant feedback and use it to improve the application's interface and interaction design, making it simple to use, effective, and pleasant. We closely monitor participant behavior and interactions with the application while we do the testing, noting any pain points or locations where participants struggle to accomplish tasks. The design and functionality of the program are then iteratively improved using this data to make sure it fits the demands of its users.

In order to guarantee the quality, dependability, and security of the application, the testing and verification phase of the AI-based audio and video editing tool includes a wide range of approaches and best practices. The project team aims to deliver a reliable and trustworthy editing solution that satisfies the varied needs of end-users and the larger industry by utilizing a thorough testing strategy that includes unit testing, integration testing, performance testing, accessibility testing, security testing, and usability testing.

Chapter 7. Performance and Benchmarks

The AI-based audio and video editing tool's performance and benchmarks part of the development process focuses on assessing and quantifying the application's performance under various workloads and circumstances. The effectiveness and efficiency of the application are evaluated using performance metrics, benchmarking techniques, and results, all of which are thoroughly covered in this chapter.

1. Performance Metrics:

A group of pertinent performance indicators that account for both the client-side and server-side components of the program are determined in order to assess the performance of the application. These metrics consist of:

a. Response Time:

The amount of time it takes for the application to react to inputs or requests from users. For the user experience to be seamless and responsive, this measure is essential.

b. Throughput:

The quantity of requests that the program fulfills in a certain amount of time. This measure is crucial for assessing the application's capacity to manage high user traffic volumes.

c. Resource Utilization:

The extent to which the application uses system resources like CPU, memory, and network bandwidth. This indicator is essential for determining the effectiveness and scalability of the application.

d. Latency:

The duration of data transmission between a client and a server, including the duration of server-side processing. This statistic is essential for comprehending how well the program performs in actual situations and identifying potential bottlenecks.

e. Error Rate:

The percentage of failed requests or transactions. This metric helps identify issues with the application's reliability and stability.

2. Benchmarking Methodologies:

A number of benchmarking tests are carried out utilizing a combination of synthetic and real-world workloads in order to provide precise and meaningful performance measurements. These tests are created to simulate different user actions, usage scenarios, and system configurations, enabling a thorough assessment of the performance of the application.

a. Synthetic Benchmarks:

Synthetic benchmarks are carefully planned tests that isolate particular performance characteristics of the application, such as CPU usage, memory consumption, or network throughput. The performance of the application is evaluated under controlled loads created by tools like JMeter, Locust, and ApacheBench.

b. Real-World Workloads:

Actual user interactions and behaviors, which are recorded through user analytics, logs, and usability testing, provide the basis of real-world workloads. These workloads assess the application's performance in real-world use cases and spot potential bottlenecks that artificial benchmarks could miss.

c. Stress Testing:

To evaluate an application's stability, dependability, and error-handling capabilities, excessive workloads or other conditions are applied to the program during a stress test. This kind of testing makes sure that the program can gracefully manage unforeseen circumstances and helps find the application's breaking points.

3. Performance Results and Analysis:

The outcomes of the benchmarking tests are examined and contrasted with predetermined performance objectives or industry standards after completion. This study aids in pinpointing both the application's strong points and potential performance bottlenecks.

a. Performance Improvements:

The study of the benchmarking results identifies and prioritizes specific performance improvements. These enhancements might involve cache techniques, database query optimization, or algorithm optimization.

b. Scalability:

The application's capacity to scale with a rising number of users or workloads is evaluated using the benchmarking results. This analysis aids in making decisions on load balancing tactics and infrastructure capacity design.

c. Comparison to Industry Standards:

The application's performance is compared against industry benchmarks or similar applications to understand its relative performance and identify areas for improvement.

In conclusion, the AI-based audio and video editing tool development process' performance and benchmarks chapter is essential to assuring the efficacy, efficiency, and scalability of the program. The project team can determine areas for improvement and optimize the performance of the application to satisfy the varied needs of end-users, stakeholders, and the larger industry by measuring pertinent performance metrics, carrying out thorough benchmarking tests, and analyzing the findings. The program may change and develop in response to shifting user needs, new technologies, and performance issues thanks to this ongoing performance evaluation approach, ensuring its competitiveness and relevance in the quickly changing AI and multimedia editing landscape.

d. Monitoring and Continuous Improvement:

Maintaining the target performance levels and spotting problems when they appear need performance monitoring. To gather and evaluate performance data in real-time, tools like Google

Cloud Monitoring, Firebase Performance Monitoring, and custom application logging are used. By actively identifying and addressing performance bottlenecks, the development team is able to maintain a fluid and responsive user experience.

Additionally, performance optimization is a continual process that calls for constant development and adaptability to new technologies, user needs, and market trends. The application's present performance is evaluated on a regular basis, and areas for improvement are noted. By ensuring that performance remains a major focus and directing the application's progress over time, these reviews assist in informing the development roadmap.

In conclusion, the performance and benchmarks chapter plays a crucial role in the creation of AI-based audio and video editing tools. The project team can deliver a high-performing editing solution that exceeds user expectations, remains competitive in the market, and adapts to the constantly evolving demands of the AI and multimedia editing industry by focusing on performance metrics, benchmarking methodologies, results analysis, and continuous improvement.

Chapter 8. Deployment, Operations, Maintenance

The AI-based audio and video editing tool's deployment, operation, and maintenance procedures are covered in detail in the chapter on deployment, operation, and maintenance in the Deployment, Operations, and Maintenance section of the development process. This chapter gives a thorough description of the deployment procedures, production environment setup, and continuous maintenance tasks that guarantee the application's dependability, security, and effectiveness.

1. Production Environment:

The infrastructure in the production environment is where the application is installed and made accessible to end users. To meet the demands of the application for performance, security, and availability, a strong and scalable infrastructure is essential. The following elements make up the production environment:

a. Cloud Infrastructure:

The frontend, backend, and database components of the application are hosted on Google Cloud Platform (GCP), which offers a scalable and secure architecture. The program uses services like App Engine, Cloud Functions, and Firestore to handle its database and computing demands.

b. Security and Compliance:

The production environment is set up to follow key laws including the CCPA and GDPR as well as industry best practices. This entails putting in place secure communication routes (HTTPS), encrypting data both in transit and at rest, and conducting routine security audits and vulnerability analyses.

2. Deployment:

New versions of the application are released into the production environment through the deployment procedure. For updates and fixes to be provided promptly and reliably, a streamlined and automated deployment method is necessary. The following steps are part of the deployment process:

a. Continuous Integration (CI):

A CI pipeline is started when developers commit code changes, and it automatically builds, tests, and bundles the application. The CI process is automated using tools like GitLab CI/CD and GitHub Actions.

b. Continuous Deployment (CD):

The application is automatically deployed to the production environment after passing all tests and inspections. The CD method guarantees that end users always have access to the most recent version of the application with the least amount of downtime and manual intervention possible.

c. Rollback and Monitoring:

A rollback mechanism is in place to restore the program to the prior stable version in the event of deployment faults or problems. The health of the application is tracked, and any potential problems are identified using performance and error monitoring tools like Sentry and Google Cloud Monitoring.

3. Maintenance:

Ongoing maintenance activities are essential for ensuring that the application remains reliable, secure, and up-to-date with evolving technologies and user requirements. Maintenance activities include the following:

a. Bug Fixes and Updates:

The development team keeps an eye out for problems and responds to reports of them, delivering updates and bug fixes as necessary. To maintain code quality and lower technical debt, regular code reviews and refactoring efforts are also made.

b. Security Patches and Upgrades:

Third-party libraries and the application's dependencies are updated often to fix any known security flaws. To shield the program and its users from potential risks, security patches are swiftly applied.

c. Performance Optimization:

The application's performance is continuously monitored and analyzed to identify potential bottlenecks and areas for improvement. Performance optimization efforts, such as algorithm refinements are undertaken to enhance the application's efficiency and scalability.

The Deployment, Operations, and Maintenance chapter summarizes the methods and recommended procedures for setting up, running, and maintaining the AI-based audio and video editing tool in a production setting. The project team can guarantee the application remains trustworthy, safe, and productive for end users and stakeholders alike by concentrating on a strong production environment setup, optimized deployment procedures, and continuing maintenance activities. The application can adapt and change in response to shifting user needs, new technologies, and fresh challenges thanks to this all-encompassing deployment, operations, and maintenance strategy, which also helps it stay competitive and relevant in the rapidly changing AI and multimedia editing landscape.

d. Monitoring and Analytics:

For analyzing the application's usage, performance, and potential problems, ongoing monitoring and analytics are essential. To gather and analyze data on user activity, system performance, and error occurrences, tools like Google Analytics, Firebase Performance Monitoring, and custom application logging are used. To ensure a seamless and responsive user experience, this data enables the development and operations teams to proactively detect and address issues.

e. Backup and Disaster Recovery:

A thorough backup and disaster recovery strategy is put into place to prevent data loss and guarantee the availability of the application in the case of a disaster. This plan calls for the development of redundant infrastructure components and failover techniques, as well as routine backups of application data, configurations, and assets. The project team can reduce downtime and guarantee the application's continuing availability in the face of unforeseen occurrences by having a strong backup and disaster recovery plan in place.

4. Training and Documentation:

Ongoing training and thorough documentation are offered to guarantee that the development, operations, and support teams have the knowledge and abilities necessary to maintain and troubleshoot the program. Along with frequent training sessions and seminars, this also comprises user documentation, operations manuals, and developer instructions. The project team can guarantee that all stakeholders have the knowledge and tools they need to effectively maintain and support the application by investing in training and documentation.

In conclusion, the Deployment, Operations, and Maintenance chapter plays a crucial role in the process of creating AI-based audio and video editing tools. The project team can provide a dependable, secure, and effective editing solution that satisfies the varied needs of end users, stakeholders, and the larger industry by concentrating on the production environment, deployment processes, maintenance activities, monitoring and analytics, backup and disaster recovery, and training and documentation. The program can adapt and develop in response to the constantly shifting demands of the AI and multimedia editing industries thanks to this all-encompassing deployment, operations, and maintenance strategy, assuring its success and relevance.

Chapter 9. Summary, Conclusions, and Recommendations

The important discoveries from the project's many phases are outlined in this chapter, along with recommendations for future research. It also gives a description of the AI-based audio and video editing tool's development process. This chapter outlines prospective directions for additional research and development while highlighting the project's successes and lessons gained.

Summary:

The effort to create an AI-based audio and video editing tool aimed to provide an all-in-one collaborative editing solution that made editing easier for users of all skill levels. The project's components included a thorough review of the state-of-the-art, a close examination of the system architecture, the application of client and data-tier technologies, performance benchmarks, and considerations for deployment, operations, and maintenance. Modern technology was added into the created solution, allowing users to modify their recordings as quickly as editing a Google Doc, including real-time voice cloning, automatic transcription, and text-to-speech conversion.

Conclusions:

The project successfully achieved its objectives by delivering a user-friendly, efficient, and scalable AI-based audio and video editing tool. The following key conclusions were drawn from the project:

a. The solution's editing capabilities were greatly improved by the use of cutting-edge AI and machine learning models, including the Text-Davinci 002 model, Tortoise TTS, WAV2LIP, and Google Speech-Text(Video Model).

b. React, Flask, Firebase authentication, and Google Cloud Platform (GCP) services are examples of contemporary client and data-tier technologies that were adopted to make it easier to create a scalable and responsive application that can handle the needs of an expanding user base.

c. A strong deployment, operations, and maintenance strategy was put in place to make sure the application remained dependable, secure, and current while reacting to shifting user needs and market trends.

d. The project's success in creating a high-performing editing solution was validated by the performance and benchmarking efforts, which showed the application's efficacy and efficiency under different workloads and situations.

Future Work

While the AI-based audio and video editing tool has achieved significant success, there are several potential avenues for future research and development that can further enhance its capabilities and marketability:

a. Expansion of the application's editing tools, including automatic color correction, background noise reduction, and sophisticated video stabilization, through the integration of more AI and machine learning models.

b. Research into more sophisticated voice cloning methods that would allow users to create synthetic voices that were both realistic and programmable for their edited content.

c. The creation of a mobile application will enable users to change their material while on the road and expand the tool's capabilities to mobile devices.

d. Real-time co-editing, version control, and project management tools are all examples of collaboration features that can be used to improve teamwork and speed up the editing process for big projects.

e. To further improve the tool's functionality and user experience, prospective interfaces with third-party platforms and services, such as social media platforms, content management systems, and cloud storage providers, are being looked into.

A thorough review of the AI-based audio and video editing tool project, its accomplishments, and prospective future advances is provided in the chapter's summary, conclusions, and recommendations. The project team can guarantee the ongoing development and usefulness of the application in the quickly changing AI and multimedia editing industries by reflecting on the project's successes and lessons gained and by recommending areas for additional research and improvement.

Glossary

1. AI (Artificial Intelligence) - The development of computer systems that can perform tasks that would typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation.
2. ML (Machine Learning) - A subset of artificial intelligence that focuses on the development of algorithms that enable computers to learn and adapt from experience.
3. SOTA (State of the Art) - The highest level of development or the most advanced stage in a specific field or technology.
4. TTS (Text-to-Speech) - The process of converting written text into spoken voice output.
5. Voice Cloning - The process of creating a synthetic voice that closely resembles the voice of a specific person, based on a sample of their speech.
6. Speaker Diarization - The process of separating and attributing speech segments to individual speakers within an audio recording.
7. CI (Continuous Integration) - A software development practice that involves merging all developers' working copies of the code to a shared repository several times a day, ensuring that the codebase remains up-to-date and functional.
8. CD (Continuous Deployment) - A software development practice that involves automatically deploying new code changes to the production environment, ensuring that the latest version of the application is always available to end-users.
9. CDN (Content Delivery Network) - A geographically distributed network of servers that work together to provide fast delivery of Internet content.
10. GDPR (General Data Protection Regulation) - A comprehensive data privacy regulation that applies to all organizations processing personal data of European Union (EU) citizens.

11. CCPA (California Consumer Privacy Act) - A data privacy regulation that provides California residents with specific rights regarding their personal information

References

[1] Business Insider. AI in Video Production.

[2] Google DeepMind. Flamingo: A Multimodal Model for Video Understanding. https://arxiv.org/pdf/2204.14198.pdf

[3] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. Moreno, and Y. Wu, "Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis," in Advances in Neural Information Processing Systems, 2018. https://arxiv.org/pdf/1806.04558.pdf

[4] OpenAI. Whisper: An Automatic Speech Recognition (ASR) System. https://openai.com/research/whisper

[5] Arik, S., Chen, J., Peng, K., Ping, W., & Zhou, Y. (2018). Neural voice cloning with a few samples. Advances in neural information processing systems, https://proceedings.neurips.cc/paper_files/paper/2018/file/4559912e7a94a9c32b09d894f2bc3c82-Paper.pdf

[6] S. Tranter and D. Reynolds, "An Overview of Automatic Speaker Diarization Systems," IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 5, pp. 1557-1565, 2006. https://ieeexplore.ieee.org/document/1678125

[7] Okopnyi, P., Juhlin, O., & Guribye, F. (2022, October). Designing for Collaborative Video Editing. In Nordic Human-Computer Interaction Conference (pp. 1-11).https://dl.acm.org/doi/pdf/10.1145/3546155.3546664

[8] Firebase. Firebase Services. https://firebase.google.com/products

[9] Google Cloud Platform. Google Cloud Products & Services. https://cloud.google.com/products

[10] React. A JavaScript library for building user interfaces. https://reactjs.org/

[11](TranscodersAPI) https://cloud.google.com/transcoder/docs/reference/rest/v1/projects.locations.transcoders

[12] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An Analysis of Environment, Microphone and Data Simulation Mismatches in Robust Speech Recognition," Computer Speech & Language, vol. 53, pp. 1-20, 2018. https://www.sciencedirect.com/science/article/abs/pii/S0885230816301231

[13] https://paperswithcode.com/task/text-to-speech-synthesis

[14] Bimbot, F., Bonastre, J. F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., ... & Reynolds, D. A. (2004). A tutorial on text-independent speaker verification. EURASIP Journal on Advances in Signal Processing, 2004, 1-22. https://link.springer.com/content/pdf/10.1155/S1110865704310024.pdf

[15] P. Bhowmick and S. Prasanna, "Speaker diarization using deep recurrent convolutional neural networks for speaker embeddings," IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 27, no. 2, pp. 364-375, 2018.

[16] https://github.com/neonbjb/tortoise-tts

[17] {choi2020wav2lip, title={Wav2Lip: High-Fidelity Audio-to-Lips Generation with Multi-Task Learning}, author={Choi, Jin-Hyuk and Choi, Sung-Hyun and Kim, Junho and Lee, Kyung-Ah and Kim, Tae-Kyun},journal={arXiv preprint arXiv:2006.09265}, year={2020}}

Appendices

Appendix A.