VOCO BUDDY
[Text-To-Speech]

A PROJECT REPORT

Submitted By

Tanu Sharma
(University Roll No-2000290140125)

Rajat Deol
(University Roll No-2000290140096)

Himanshu Tomar
(University Roll No- 2000290140051)

Under the Supervision of
Dr. Akash Rajak
(Professor)
KIET Group of Institutions

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(May – 2022)

DECLARATION

We hereby declare that the work presented in this report entitled "**Voco Buddy**", was carried out by **Rajat Deol**, **Tanu Sharma** and **Himanshu Tomar**. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. We have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: Rajat Deol
Roll. No.: (2000290140096)
Course: Master of Computer Application

Name: Tanu Sharma
Roll. No.: (2000290140125)
Course: Master of Computer Application

Name: Himanshu Tomar
Roll. No.: (2000290140051)
Course: Master of Computer Application

(Candidate Signature)

CERTIFICATE

Certified that Rajat Deol (2000290140096) Tanu Sharma (2000290140125) and Himanshu Tomar (2000290140051) has carried out the project work presented in this report entitled “Voco Buddy” for the award of Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University, Lucknow under my supervision. The report embodies result of original work, and studies are carried out by the student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University.

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Akash Rajak

Professor

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Ajay K Shrivastava
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

There are about millions of blind and visually impaired people worldwide. These people are not able to see and hear anything. Disability of visual text reading has a huge impact on the quality of life for visually disabled people. Though many several devices have been designed for helping visually disabled to see objects using an alternating sense such as sound and touch, the development of text reading device is still at an early stage. Some existing systems for text recognition are typically limited either or require user assistance or may be of high cost. Therefore, some low-cost system must be developed that will enable to automatically locate and read the text aloud to visually impaired persons. In this paper we will study various ideas to recognize the text character and convert it into speech signal along with some applications of TTS systems. This paper also contains the various terms and concepts of text to speech conversion systems.

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide, **Dr**. **Akash Rajak**, for providing me with the right guidance and advice at the crucial junctures and for showing me the right way. I extend my sincere thanks to our respected Head of the Department, **Dr. Ajay Kumar Shrivastava**, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during our work.

Rajat Deol

Tanu Sharma

Himanshu Tomar

List of Figures

TABLE OF CONTENTS

Chapter -1- Introduction

Text-to-speech (TTS) is a type of assistive technology that reads digital text aloud. It's sometimes called "read aloud" technology. With a click of a button or the touch of a finger, TTS can take words on a computer or other digital device and convert them into audio. TTS is very helpful for kids and adults who struggle with reading. But it can also help with writing and editing, and even with focusing.

TTS works with nearly every personal digital device, including computers, smartphones, and tablets. All kinds of text files can be read aloud, including Word and Pages documents. Even online web pages can be read aloud. The voice in TTS is computer-generated and reading speed can usually be sped up or slowed down. Many TTS tools highlight words as they are read aloud. This allows the user to see text and hear it at the same time.

Some TTS tools can also read text aloud from images. For example, a user could take a photo of a street sign on their phone and have the words on the sign turned into audio. You might be wondering what the connection is between TTS and audiobooks.

TTS is a tool that reads text aloud. An audiobook is a recording of a book read by a human voice (or created by TTS). Sometimes, people say TTS or audiobooks to mean the same thing. There are many different TTS tools: Built-in text-to-speech: Many devices have built-in TTS tools. This includes desktop and laptop computers, smartphones, digital tablets, and Chromebooks.

Web-based tools: Some websites have TTS tools on-site. Text-to-speech apps: Users can download TTS apps on smartphones and digital tablets. There are also TTS tools that can be added to web browsers, like Chrome. Text-to-speech software programs: Many literacies software programs for desktop and laptop computers have TTS.

1.1 - Technologies Used

Hardware and Software used in this Project

Hardware:

- Windows 10
- 4 RAM
- i5 10^{th} generation
- 512 GB SSD.

Software:

- Python.
- Django.
- Tesseract OCR.

Python:

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s. Python consistently ranks as one of the most popular programming languages. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including metaprogramming and metaobjects [magic methods]). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It uses dynamic name resolution (late binding), which binds method and variable names during program execution.

Its design offers some support for functional programming in the Lisp tradition. It has `filter`, `map` and `reduce` functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (`itertools` and `functools`) that implement functional tools borrowed from Haskell and Standard ML.

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

We have used this language because there are some certain packages already present in python language which needs to be use in our project so it can perform well in effective and efficient manner.

Django:

Django is an MVT web framework that is used to build web applications. The huge Django web-framework comes with so many “batteries included” that developers often get amazed as to how everything manages to work together.

The principle behind adding so many batteries is to have common web functionalities in the framework itself instead of adding latter as a separate library.
One of the main reasons behind the popularity of Django framework is the huge Django community. The community is so huge that a separate website was devoted to it where developers from all corners developed third-party packages including authentication, authorization, full-fledged Django powered CMS systems, e-commerce add-ons and so on.
There is a high probability that what you are trying to develop is already developed by somebody and you just need to pull that into your project.
Django is designed in such a way that encourages developers to develop websites fast, clean and with practical design. Django's practical approach to getting things done is where it stands out from the crowd.

If you're planning to build a highly customizable app, such as social media website, Django is one of the best frameworks to consider. Django strength lies in its interaction between users or its ability to share different types of media. One of the great advantages of Django is its ability to utilize large community-based support which gives you highly customizable third-party ready to use plugins in your applications.

Below are the reasons to choose Django for web development –

Python:

Python is arguably one of the easiest programming languages to learn because of its simple language constructs, flow structure and easy syntax. It is versatile and runs websites, desktop applications and mobile applications embedded in many devices and is used in other applications as a popular scripting language.

Batteries Included:

Django comes with common libraries which are essential to build common functionalities like URL routing, authentication, an object-relational mapper (ORM), a templating system and db-schema migrations.

Built-in admin:

Django has an in-built administration interface which lets you handle your models, user/ group permissions and to manage users. With model interface in place, there is no need for a separate database administration program for all but advanced database functions.

Scalable:

Django is based on MVC design pattern. It means that all the entities like db (database), back-end and front-end code are individual entity. Django allows us to separate code from the static media including pictures, files, CSS, and JavaScript that make up your site.
Django supports a full list of third-party libraries for web servers, caching, performance management, clustering, and balancing. One of the advantages Django provides is the support for major email and messaging applications and services like Rest and OAuth.

Huge package support:

Because of its large community support and huge developers' network, there is a high possibility that whatever you intend to do might have been done before. Large international community of developers contribute to the community by releasing their projects as open-source packages.
One such repository of these projects is Django Package site. Currently, Django packages list over 3400 plus reusable Django apps, sites, and tools to use in our Django projects.
Actively developed:

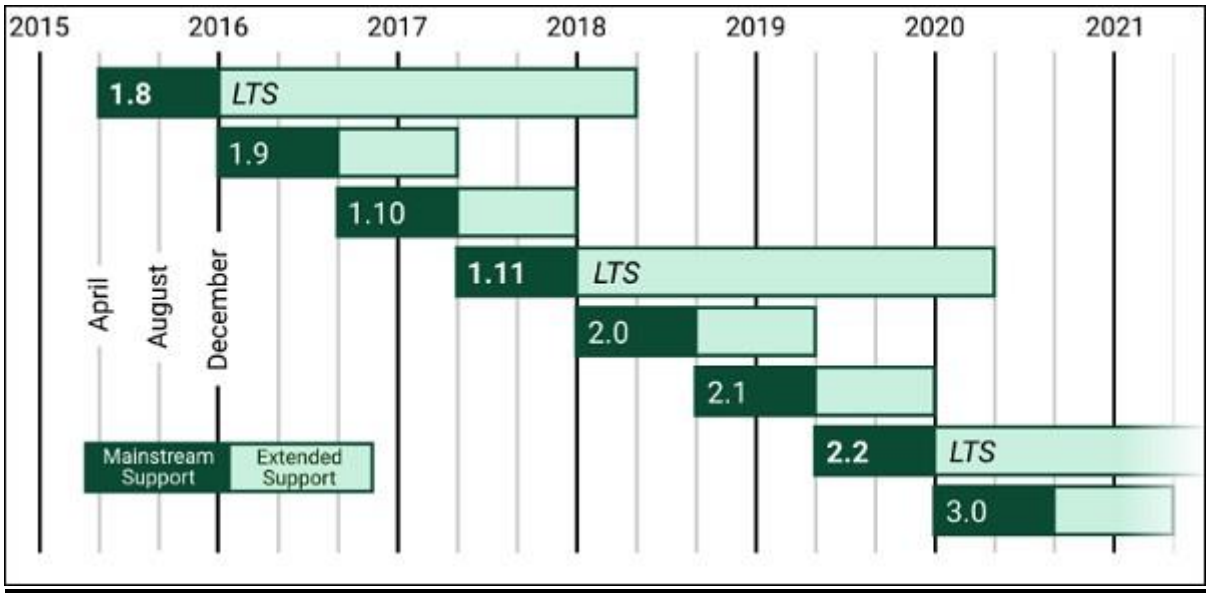
One of the biggest risks associated with open-source project is its sustainability. We cannot be sure if it lasts long.
There is no such risk with Django as it is 12 years old. Its consistent releases, newer/better versions and active community is growing every-day with a large core team of voluntary contributors who maintains and improve the code base every-day.

Stable releases:

Open-source software projects like Django are, in many cases, actively developed and more secure than competing proprietary software as many developers are developing and testing it every day. However, the drawback of an open-source software project is the absence of a stable codebase to commercially viable development.

In Django, we have Long Term Support (LTS) versions of the software and a defined release process as shown in the below image –

LTS Versions of Django:



First Class Documentation:

From the very first release, Django developers made sure that there must be proper comprehensive documents available, and the tutorials are easily understood.

Setting the Development Environment

To create a virtual environment type the below command in the terminal –
python3 -m venv <name>

To activate it type the below command in the terminal.
source ./venv/bin/activate

Installing Django

We can install Django using the pip command. To install this type the below command in the terminal.

pip install django

Starting the Project

- To initiate a project of Django on Your PC, open Terminal and Enter the following command

django-admin startproject projectName

- A New Folder with the name projectName will be created. To enter in the project using the terminal enter command

cd projectName

- Now let's un the server and see everything is working fine or not. To run the server type the below command in the terminal.

python manage.py runserver

HTML:

It is used for giving eye catching look to the website. And providing easy to use GUI.

CSS:

CSS is cascading style sheet which is used to give designer look to HTML using the external file.

Java script:

Java script is used for client-side scripting which can help in using validation on the website and many more other functions.

Tesseract OCR:

Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images.

Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file.

1.2 Feasibility Study

Feasibility Study is a study to evaluate feasibility of proposed project or system. Feasibility study is one of stage among important four stages of Software Project Management Process.

As name suggests feasibility study is the feasibility analysis or it is a measure of the software product in terms of how much beneficial product development will be for the organization in a practical point of view.

Feasibility study is carried out based on many purposes to analyze whether software product will be right in terms of development, implantation, contribution of project to the organization etc.

Technical feasibility
The objective of the technical feasibility step is to confirm that the product will perform and to verify that there are no production barriers. Product: The product of this activity is a working model.

In Technical Feasibility current resources both hardware software along with required technology are analyzed/assessed to develop project. This technical feasibility study gives report whether there exists correct required resources and technologies which will be used for project development. Along with this, feasibility study also analyzes technical skills and capabilities of technical team, existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology etc.

Operational Feasibility
The operational feasibility to help users find items that they deem of interest to them. They can be seen as an application of data mining process. In this paper, a new recommender system based on multi-features is introduced. Demographic and psychographic features are used to asses' similarities between users.

In Operational Feasibility degree of providing service to requirements is analyzed along with how much easy product will be to operate and maintenance after deployment. Along with these other operational scopes are determining usability of product, determining suggested solution by software development team is acceptable or not etc.

Economic Feasibility

In Economic Feasibility study cost and benefit of the project is analyzed. Means under this feasibility study a detail analysis is carried out what will be cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on. After that it is analyzed whether project will be beneficial in terms of finance for organization or not.

Behavioral Feasibility

It evaluates and estimates the user attitude or behavior towards the development of new system. It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee's job status on new ways of conducting business.

Chapter – 2 Literature Review

In this literature review, researcher has focused on the effectiveness of Text to speech technologies such as computer, smartphone, and tablets as assistive technology for blind students in the primary schools.

The major objective of the scholar is to increase the academic performance of visual impairment children by use of Text to speech technologies. For this objective, scholar has discussed various driving forces such as technologies and its application. In this literature review, the effectiveness and practical application of the text speech technology has been addressed.

With help of this literature review, the detail knowledge and information about the text to speech technology has been collected by the scholar because they used various articles of the different authors who have given the statement about the assistive technology.

The major objective of this literature review is to conduct study on the topic of text to speech technology and its importance for the blind children who have visual impairment.

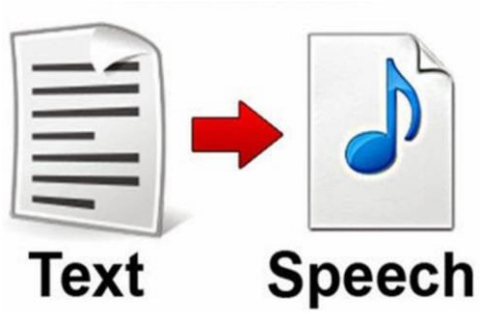


Figure 1: Basic Structure of our Project.

Chapter - 3 Requirement Engineering

3.1 Software Requirement Specification

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

3.2 E-R Diagram

ER Diagram:

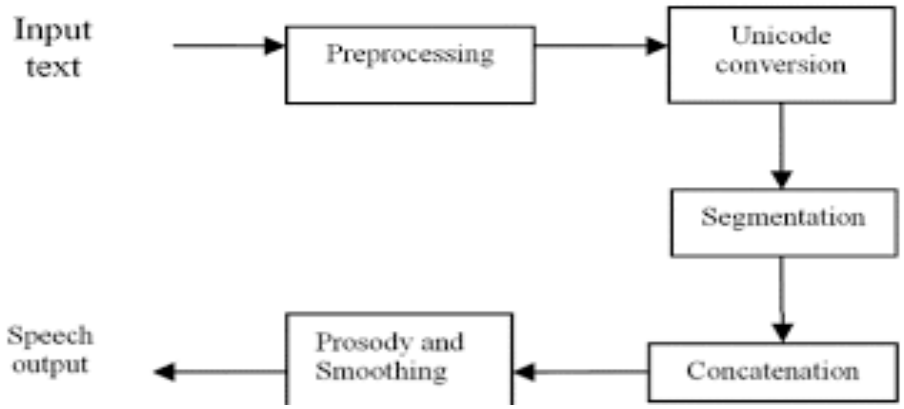


Figure 2: This E R Diagram Shows Detailed Structure and Relations of entities of this project.

3.3 Flowchart of Text-to-Speech

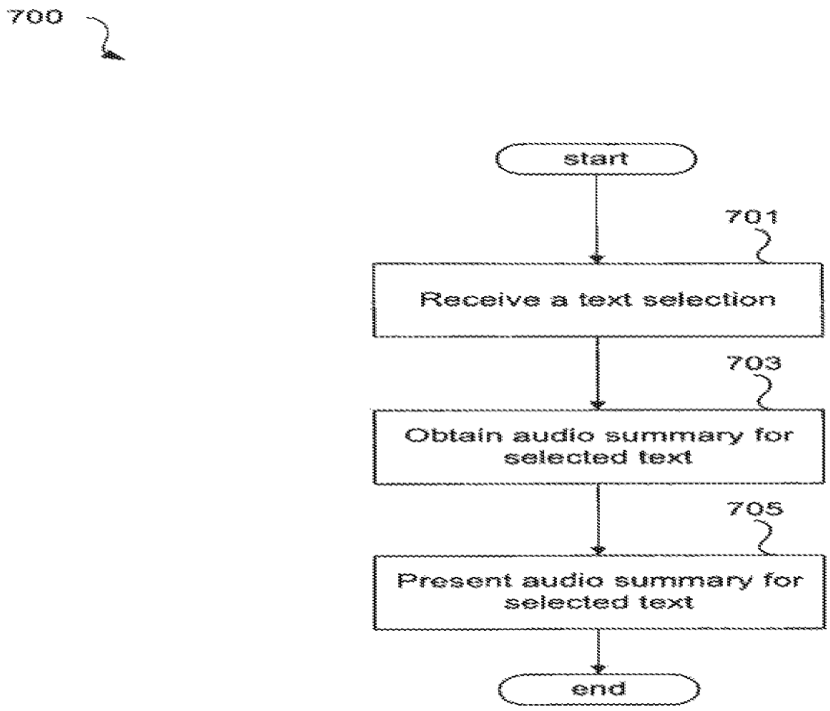


Figure 3: This diagram shows graphical representation of the project methodology.

3.4 Block Diagram for image to text then speech conversion

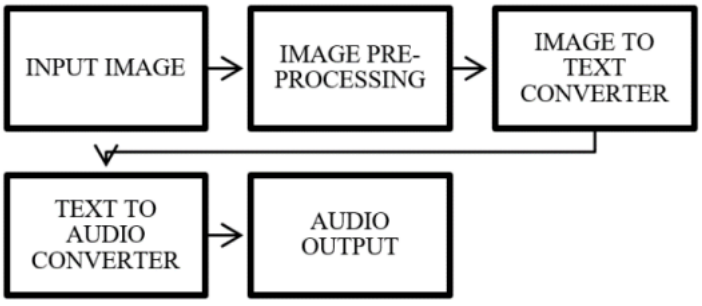


Figure 4: This diagram shows the basic structure of the Project.

Chapter – 4 Designing and Implementation

4.1 System Design of Voco Buddy

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client's requirements into a logically working system. Normally, design is performed in the following two steps:

1. Primary Design phase:
In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

2. Secondary Design phase:
In this secondary phase the detailed design of every block is performed.

4.2 General Task involved in the design process are following:

A. Design various blocks for overall system process.
B. Design smaller, compact, and workable modules in each block.
C. Design various database structures.
D. Specify details of programs to achieve desired functionality.
E. Design the form of inputs, and outputs of the system.
F. Perform documentation of the design.
G. System reviews

4.3 Detailed Design of Implementation:

This phase of the system development lifecycle refines hardware and software Specifications, establishes programming plans, trains users, and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

- **Technical Design:** This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

- **Test Specifications and Planning:** This activity prepares detailed test specifications for individual modules and programs, job streams, subsystem, and for the system.

Chapter – 5 Testing

5.1 Programming and Testing

This activity encompasses actual development, writing, and testing of program units or modules.

- **User Training**

This activity encompasses writing user procedure manuals, materials, conducting training programs and testing procedures.

- **Acceptance Test**

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

- **Installation Phase**

In this phase the new computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

- **System Installation**

This process of starting the actual use of a system and training user personnel in its operation.

- **Review Phase**

This phase evaluates the successes and failures during a systems development project and to measure the results of a new computerized Tran system in terms of benefits and savings projected at the start of the project.

- **Development Recap**

A review of a project immediately after completion to find success and potential problems in future work.

- **Post-implementation Review**

A review, conducted after a new system has been a operation for some time, to evaluate actual system performance against original expectations and projections for cost- benefit improvements. Also identifies maintenance projects to enhance or improve a system.

5.2 The Steps in the Software Testing

The steps involved during Unit testing are as follows:

a) Preparation of the test cases.
b) Preparation of the possible test data with all the validation checks.
c) Complete code review of the module.
d) Actual testing done manually.
e) Modifications done for the errors found during testing.
f) Prepared the test result scripts.

The unit testing done included the testing of the following items:

1. Functionality of the entire module/forms.
2. Validations for user input.
3. Checking of the coding standards to be maintained during coding.
4. Testing the module with all the possible test data.
5. Testing of the functionality involving all type of calculations etc.
6. Commenting standard in the source file.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While system integration, we integrated the modules one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

- The steps involved during System testing are as follows:
- Integration of all the modules in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modification done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.

5.3 The System Testing done included the testing of following items:

- Functionality of the entire system ad a whole.
- User interface of the system.
- Testing the dependent module together with all the possible test data.
- Verification and validation testing.
- Testing the reports with all its functionality.

5.4 Existing System of a Learning Management System

In the system the exams are done only manually but in process system we have to computerize the exams using this application

- Local security of data.
- More manpower.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculation.
- No direct role for the higher officials.

Chapter – 6 Use Case and Benefits of Text-to-Speech

6.1 Use Case:

Banking and Finance

Since text to speech software has begun to penetrate the financial services industry, it's safe to say that the integration of the technology has paid off (no pun intended). Beyond the ability to check your finances and the stock market on the go, using nothing more than voice commands, text to speech can be used to enhance security and improve the customer experience by making it more accessible, dynamic, and personalized.

Fintech, which is short for financial technology, is a growing field that is changing the way financial services are offered. With text to speech software, "customers can pre-define a list of favorites that allows them to transfer money into these accounts by name of individual or entity rather than by inputting 9–16-digit account numbers." By cutting out the need to remember various passwords, the banking experience becomes more enjoyable and less burdensome.

HSBC was among the first banks to launch voice recognition services as part of its banking experience. Mobile banking customers were now able to access their accounts without providing passwords or other data. "This marked a leap toward a new direction in biometric authentication to the financial services sector," writes Codete.

Travel and Tourism

The travel and tourism industries have perennially struggled to fluently communicate with visitors from a variety of linguistic backgrounds. Well, by harnessing text to speech software, companies in the hospitality industry can make it easier for people to get around and offer tours in numerous languages, all at the same time.
One of the great benefits of text to speech is how it can be used to help travellers get from point A to point B. The technology can lend itself to "PA systems sharing real-time information, travel announcements in airports, train stations, and other transportation hubs, and self-service ticketing options in public areas that offer instructions in your spoken language of choice."

Text to speech software can also enable the creation of self-guided audio tours powered by synthetic voices. By inputting the transcription of the tour into your synthetic voice engine, text to speech software can allow the tour of a museum, monument, or other points of interest to be spoken aloud to your audience in the language of their preference.

Mapping and navigation software is another field that makes use of text to speech. "Apps like Google Maps and Apple Maps are designed to automatically read turn-by-turn directions aloud using text-to-speech technology," writes Business Insider.

The opportunities with text to speech in the travel industry don't stop at vacations and trips for leisure. Business travel and corporate conferences can also stand to gain a lot by adopting text to speech solutions, making travel more accessible for many, regardless of the language they speak.

Telecommunications

When a customer or client gives your company a call, the last thing you want is for them to be met with the sound of dead air. This is why it's important to assemble a strong interactive voice response (IVR) system. Text to speech can be used to offer customized messaging that the caller can engage with, and it can generate words from a customer's records that are read back to them in a friendly, professional voice.

Even addressing a customer by name can go a long way toward gaining their trust. "Everybody loves hearing his or her own name," writes customer experience expert IST Networks. "Addressing someone by name creates a trigger in that person's brain that says, 'I'm being spoken to on an individual level.'"

Conversely, speech to text software has proven an incredibly useful tool in telephony. By converting speech recorded from a call into accurate text, you can save transcripts of any spoken interaction, which are later searchable by date and keyword. These transcripts can be used for employee training and establishing best practices.

Speech to text software, also known as automated speech recognition (ASR), can also be used "to create transcripts of conversations for use with speech analytics applications, to allow companies greater insight into the customer experience."

The more convenient and meaningful you can make the customer experience, the better, and both text to speech and speech to text technology represent a step toward achieving this.

Automotive Manufacturing

Another professional domain that has been revolutionized by text to speech software is the automotive industry. When you think about it, TTS and automobiles are a perfect match. Driving is an activity that necessarily requires an individual's full attention, but there are regularly instances where a driver must double-check directions, or briefly look away from the road to perform an action like changing the radio station or adjusting the car temperature. Integrating text to speech software into a vehicle's system is an optimal way to make driving safer and more convenient with voice-enabled, hands-free controls. "Car manufacturers are increasingly demanding embedded speech solutions in their GPS and navigation systems, as well as their telematics systems," ReadSpeaker reports. When

car navigation systems are combined with media and phone controls, drivers will have the opportunity to stay connected without the temptation to look down at their phone.

Given the personalized nature of text to speech software, a driver can receive custom audio directions that guide them to work, back home, and anywhere else the driver needs to go. Text to speech software can also read text messages—or even an entire email—back to a driver, and vehicles equipped with speech to text capabilities can allow drivers to dictate the message they'd like to send in return, all without taking their hands from the wheel.

The presence of voice recognition technology coming built-in with new automobiles is only going to accelerate with the rise of the 'Internet of Things.' When a vehicle is 'connected,' so to speak, it can even enable "outbound communications between dealerships and customers for items like appointment confirmations, scheduled service reminders, and promotion and sales updates."
Text to speech software for automobiles make all of these functions possible, and ultimately, allow drivers to get around easier while keeping their eyes exactly where they're meant to be on the road.

E-learning
Text to speech software presents the opportunity to bring static content, like ebooks, PDFs, and other training documents, to life. This technology is highly beneficial when you need to convert long passages of text into playable audio. Instead of hiring a voice actor to read hours upon hours of technical materials, your text to speech voice can automatically render your words into speech.

Opting for text to speech as a producer of eLearning also means that you're allowing your learners to partake in bimodal learning. When educational or training content is presented in both audio and visual formats, there is shown to be a higher rate of learner retention.
Text to speech can also serve as an aid for students with learning disabilities. "Studies who have been diagnosed with dyslexia did benefit from the use of TTS software," reports Reading Rockets, noting that they "saw improvements in motivation to read, improved comprehension, and improved fluency."
Other features of text to speech software for learning, including word prediction features and phonetic spell checking, can help "younger students who still struggle with reading or pronouncing new words," as well as "students who commute and have limited time to read."

While the technology is valuable for young learners, text to speech software is also well worth integrating into your corporate compliance training or microlearning apps, for ongoing training outside of the traditional classroom. When you use text to speech, you can easily translate your corporate training into a new language, and instantly generate a spoken delivery to go along with it using text to speech software.

6.2 Benefits:

Accessibility

Text to speech can be used as an assistive technology that helps people with visual impairment, medical conditions that have impacted their voice, and learning disabilities. Read all about the ways that text to speech is helping to build a more accessible world.

Scalability

Text to speech is scalable. Once you have a custom text to speech voice, you can input immense amounts of data and have them instantly converted into audio recordings. Text to speech is ideal when you are working with large passages of text, because in many cases, hiring a voice actor to read each word may be a grueling undertaking.

Flexibility

Text to speech software allows you to listen to text content on the go. This can reduce screen time and enable learning and consumption while one is doing other things, so they don't have to be staring at the page or screen.

Affordability

When you're going to be customizing your words on the regular, it will be more inexpensive in the long run to create a tailored synthetic voice for your brand then hire a voice actor anew each time to record your new content.

Heightened web presence

When you offer an audio version of your content in addition to a text version, it will be more accessible to a wider audience, who will choose whether to read or listen to content based on preferences. The more options available, the more of a web presence you can build out.

Chapter - 7 Snapshots and Source code

7.1 Snapshots:

Home Page:

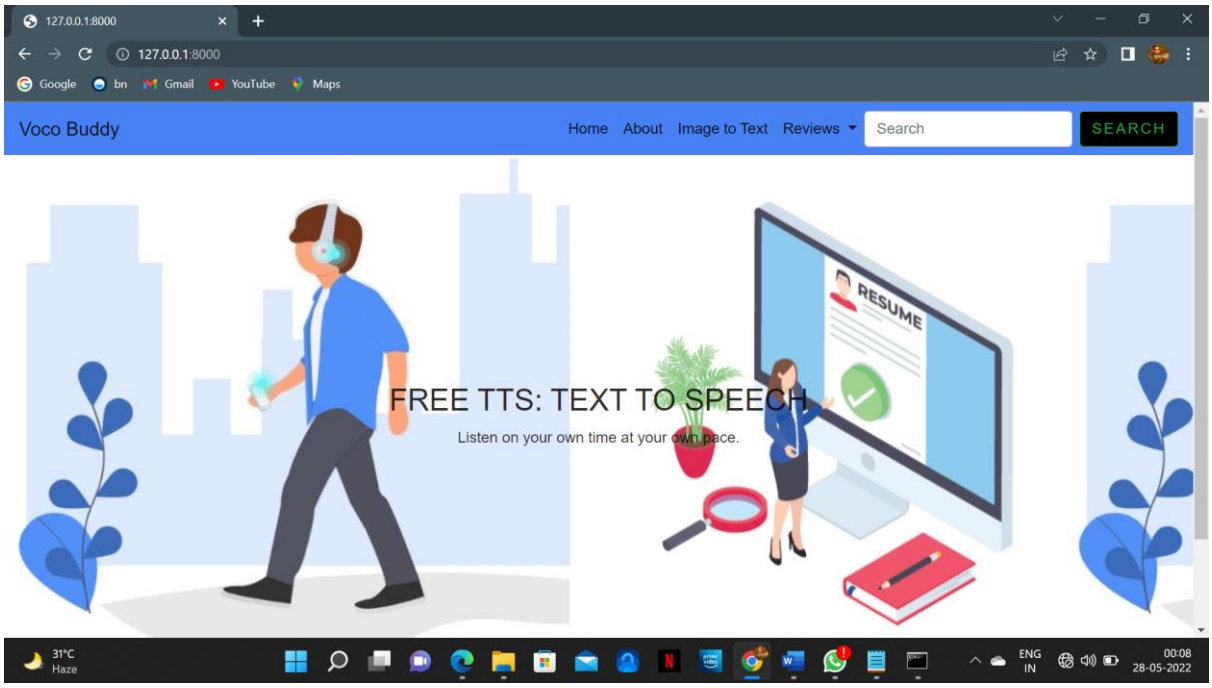


Figure 5: This is our Home Page, or you can say landing page of our Web Application.

About US Page:

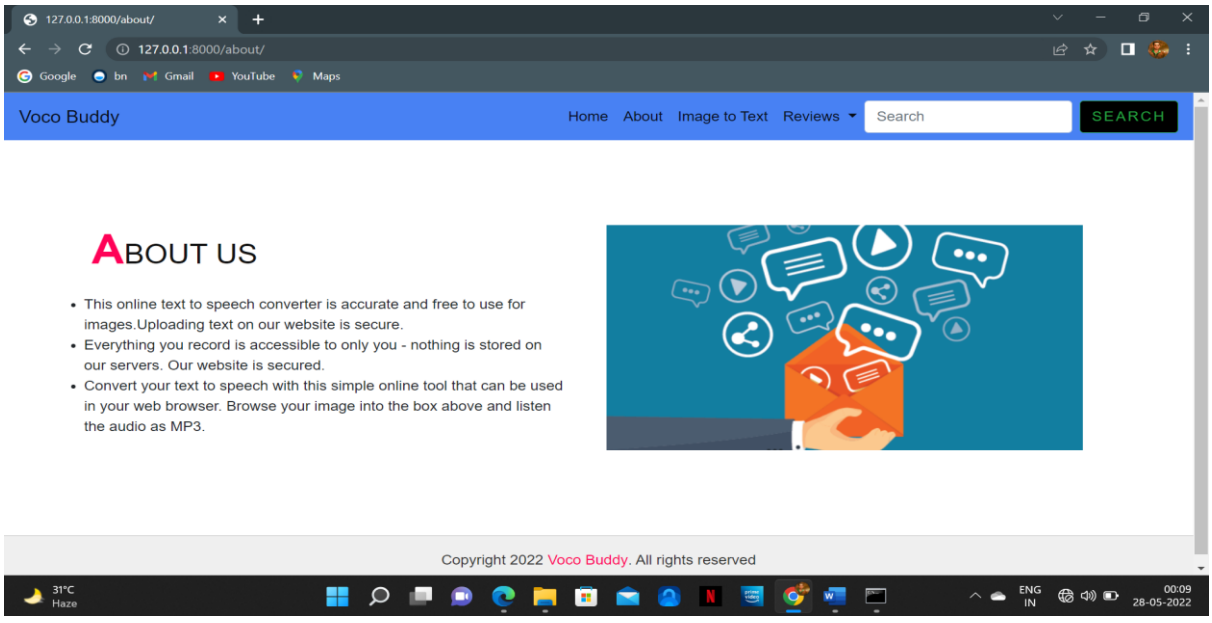


Figure 6: This page shows the detailed knowledge about our web application.

Contact us Page:

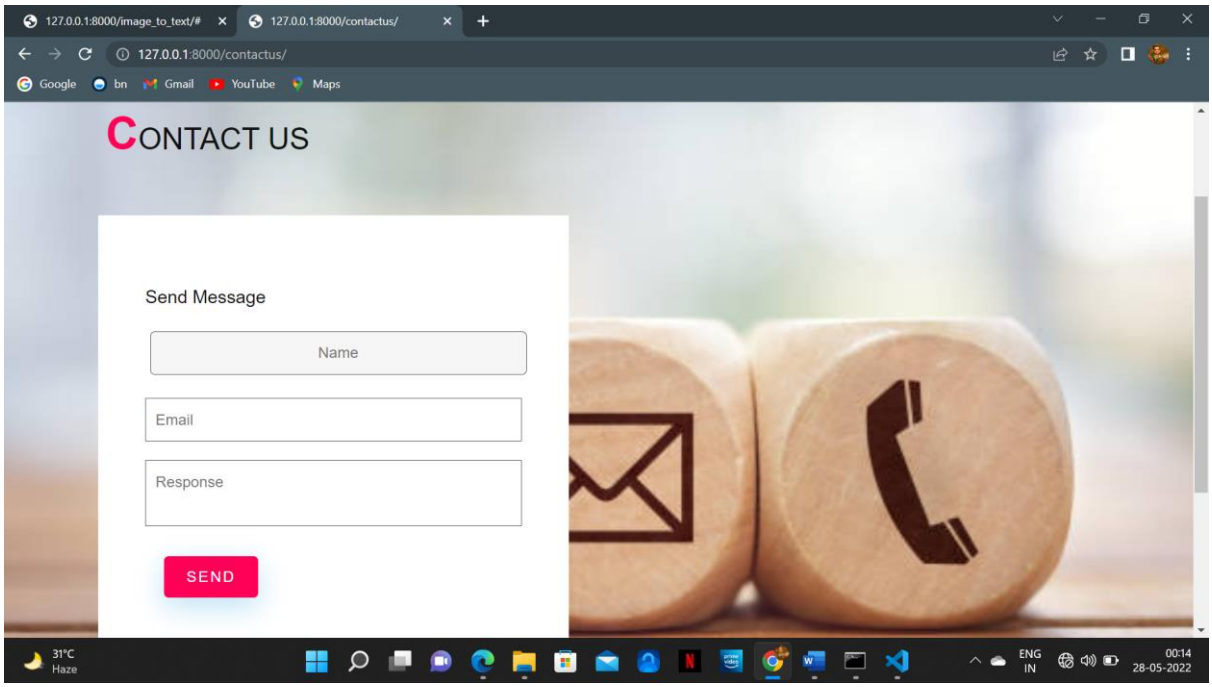


Figure 7: This page helps user to connect with us.

Testimonial Page:

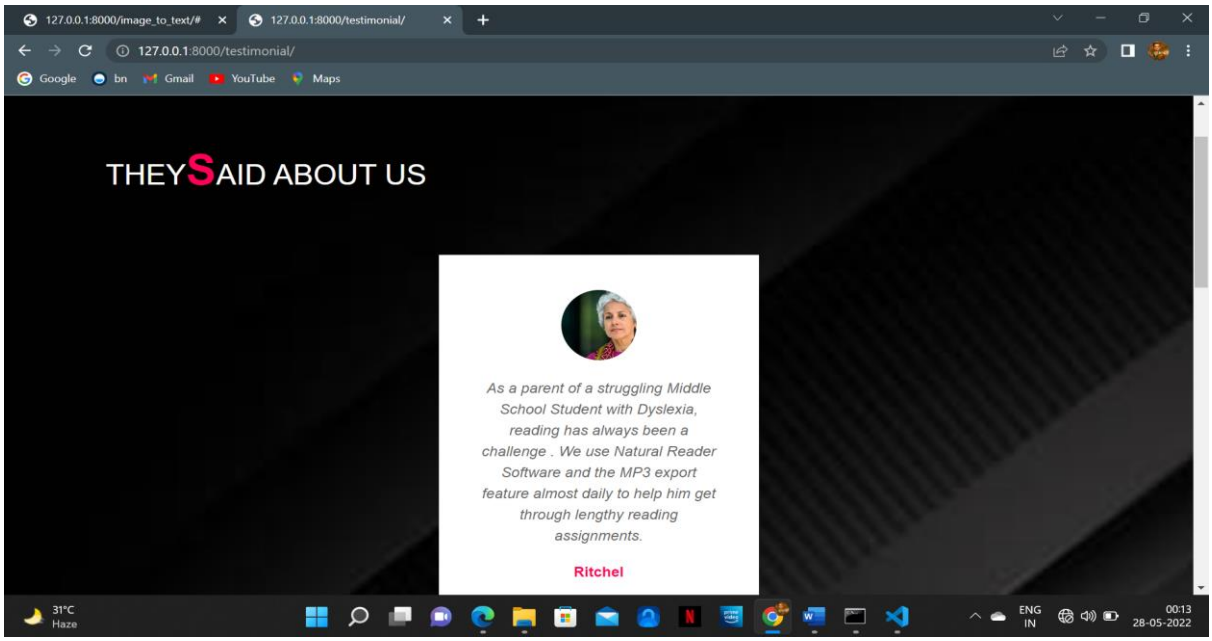


Figure 8: This page shows customer or users reviews after using our web application.

Image-to-text-then-Speech Page:

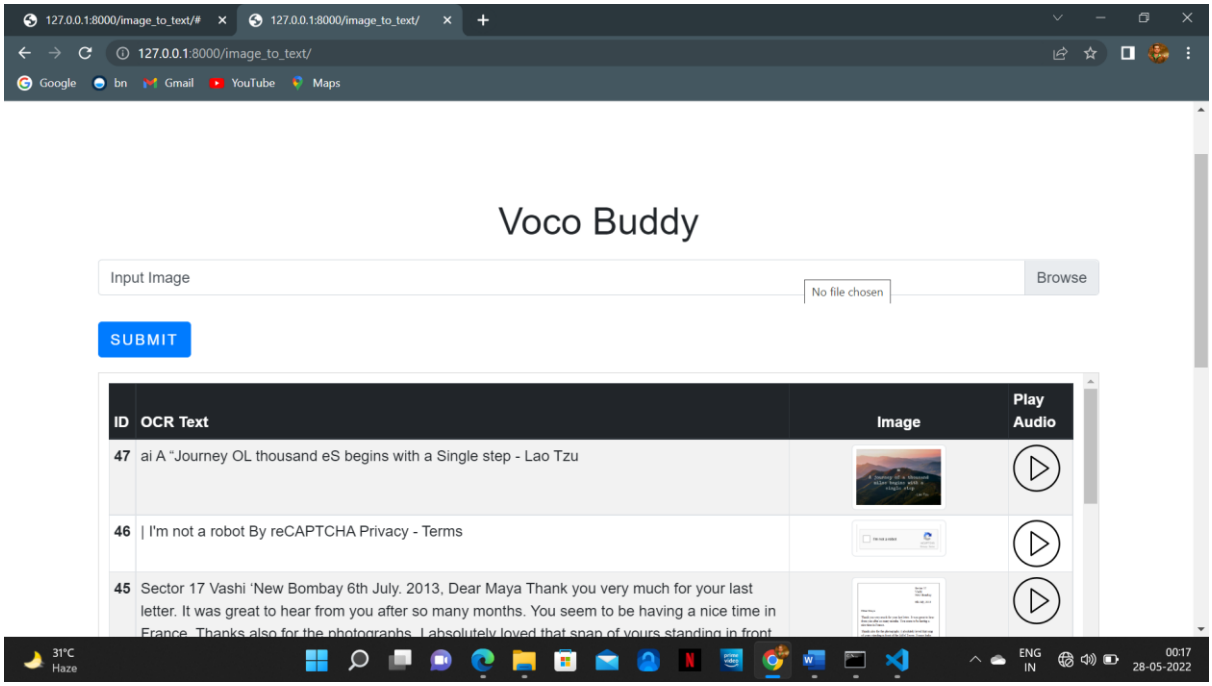


Figure 9: This page helps you to listen text or image text you want.

7.2 Source Code:

Home Page:

```
<!doctype html>
{% load static %}

<html lang="en">
<head>
   <!-- Required meta tags -->
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">



   <link href="{% static 'css/style1.css' %}" rel="stylesheet" type="text/css"/>
   <link href="{% static 'css/Stylevoco.css' %}" rel="stylesheet" type="text/css"/>
   <!-- Bootstrap CSS -->
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css"
        integrity="sha384-
GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706t
WS" crossorigin="anonymous">



   <!-- <title>MainPage Apps</title> -->
   <style>
body
body, html {
   height: 100%;
   margin: 0;
   font-family: Arial, Helvetica, sans-serif;
  }

  * {
   box-sizing: border-box;
  }

  .bg-image {
   /* The image used */
   background-image: url("navbar\bg.jpg");

   /* Add the blur effect */
```

```
    filter: blur(8px);
    -webkit-filter: blur(8px);

    /* Full height */
    height: 100%;

    /* Center and scale the image nicely */
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
  }

  /* Position text in the middle of the page/image */
  .bg-text {
    background-color: rgb(0,0,0); /* Fallback color */
    background-color: rgba(0,0,0, 0.4); /* Black w/opacity/see-through */
    color: white;
    font-weight: bold;
    border: 3px solid #f1f1f1;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    z-index: 2;
    width: 80%;
    padding: 20px;
    text-align: center;
  }
</style>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light"style="background-color:
#4881F3;">
    <a class="navbar-brand" href="{% url 'mainpage' %}">Voco Buddy</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>


    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item active">
```

```
        <a class="nav-link" href="{% url 'mainpage' %}"> Home <span class="sr-
only">(current)</span></a>
      </li>
      <li class="nav-item active">
        <a class="nav-link" href="{% url 'about' %}">About</a>
      </li>
      <li class="nav-item active">
        <a class="nav-link" href="{% url 'image_to_text' %}">Image to Text</a>
      </li>
      <li class="nav-item dropdown active">
        <a class="nav-link dropdown-toggle" href="{% url 'contactus' %}"
id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true"
aria-expanded="false">
          Reviews
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdown">
          <a class="dropdown-item" href="{% url 'testimonial' %}">Testimonial</a>
          <a class="dropdown-item" href="{% url 'contactus' %}">Contact Us</a>

        </div>
      </li>

    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-
label="Search">
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit"
style="background-color: black;">Search</button>
    </form>
  </div>
 </nav>


<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
      integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
      crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js"
      integrity="sha384-wHAiFfRlMF
y6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut"
      crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js"
```

```
      integrity="sha384-
B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k"
      crossorigin="anonymous"></script>
{% block extrascript %}{% endblock %}

{% block extendme%}
<center>
  <section class="banner" id="banner">
    <div class="content">
      <h2>Free TTS: Text To Speech</h2>
      <p>Listen on your own time at your own pace.
      </p>

        <!-- <a href="#menu" class="btn">Our Menu</a> -->
    </div>
</section>
<footer>
  {% include "uploader/footer.html" %}

</footer>
</center>

{% endblock %}
</body>
</html>
```

About us Page:

```
{% block extendme %}
<section class="about" id="about">
  <div class="row">
     <div class="col50">
        <h2 class="titleText"><span>A</span>bout Us</h2>
        <p><ul>
           <li>This online text to speech converter is accurate and free to use for
images.Uploading text on our website is secure.</li>
           <li>Everything you record is accessible to only you - nothing is stored on our
servers. Our website is secured.</li>
           <li>Convert your text to speech with this simple online tool that can be used in
your web browser. Browse your image into the box above and listen the audio as
MP3.</li>

        </ul></p>

     </div>
```

```
    <div class="col50">
       <div class="imgBx">
          <img src="/media/OCR_image/image/VOCO 3.png">
       </div>
    </div>
  </div>
</section>
<footer>
  {% include "uploader/footer.html" %}

</footer>

{% endblock %}
```

Contact us Page:

```
{% extends "uploader/mainpage.html" %}
{% load static %}

{% block title %} Contact Us {% endblock %}

{% block extrahead %} {% endblock %}

{% block extrastyle %}
{% endblock %}

{% block extrascript %}
<script>

</script>

{% endblock %}

{% block extendme%}
<section class="contact" id="contact">
    <div class="title">
       <h2 class="titleText"><span>C</span>ontact Us</h2>
    <p></p>
    </div>
    <div class="contactForm">
    <h3> Send Message</h3>

    <form>
    <div class="inputBox">
```

```
        <input type="text" placeholder="Name" name="T1">
      </div>
      <div class="inputBox">
        <input type="email" placeholder="Email" name="T2">
      </div>
      <div class="inputBox">
        <textarea placeholder="Response" name="T3"></textarea>
      </div>
      <div class="inputBox">
        <input type="submit" value="Send" onclick="myFunction()">
      </div>
      </form>

      </div>
</section>
<script>
  function myFunction() {
    alert("Your Details has been Successfully Submitted!");
  }
  </script>
<footer>
  {% include "uploader/footer.html" %}

</footer>
<!-- <div class="container">
  <div class="jumbotron">
    <center>
      <h1>Contact US</h1>
      <address>
        Developed By <a href="mailto:webmaster@example.com">PM Venkat</a>.<br>
        Chennai<br>
        Tamilnadu.
        </address>
    </center>
</div> -->
{% endblock %}
```

Testimonial Page:

```
{% extends "uploader/mainpage.html" %}
{% load static %}


{% block title %} Testimonial {% endblock %}
```

```
{% block extrahead %} {% endblock %}

{% block extrastyle %}
{% endblock %}

{% block extrascript %}
<script>

</script>

{% endblock %}


{% block extendme%}
<section class="testimonial" id="testimonial">
  <div class="title white">
     <h2 class="titleText">They<span>S</span>aid About Us</h2>
  <p></p>
  </div>
  <div class="content">
     <div class="box">
        <div class="imgBx">
           <img src="/media/OCR_image/image/menu1.png">
        </div>
        <div class="text">
           <p>As a parent of a struggling Middle School Student with Dyslexia, reading has always been a challenge . We use Natural Reader Software and the MP3 export feature almost daily to help him get through lengthy reading assignments.
           </p>
           <h3>Ritchel</h3>
        </div>

        <div class="box">
           <div class="imgBx">
              <img src="/media/OCR_image/image/menu2.png">
           </div>
           <div class="text">
              <p>I no longer strain my eyes trying to read tiny fonts in e-mails or web pages or spend time recording my own voice for teaching purposes. I have a “bilingual“ Voco Buddy and it has become a very useful tool.
              </p>
              <h3>Marry</h3>
           </div>
```

```
            <div class="box">
               <div class="imgBx">
                  <img src="/media/OCR_image/image/menu3.png">
               </div>
               <div class="text">
                  <p>This Software helps me pronounce words that i cannot and it helps me
avoid making grammatical errors when writing my case briefs and my essays. I enjoyed
the fact that i can convert my online textbooks into mp3's and listen to them on my
smart phone or while driving in my car.
                  </p>
                  <h3>Elizabeth</h3>
               </div>
      </div>
   </div>

</section>

<footer>
   {% include "uploader/footer.html" %}
</footer>
{% endblock %}
```

Image-to-text Page:

```
{% extends "uploader/mainpage.html" %}
{% load widget_tweaks %}
{% block title %} Image to Text< {% endblock %}

{% block extrahead %} {% endblock %}

{% block extrastyle %}

{% endblock %}

{% block extrascript %}
<script src="https://cdn.datatables.net/1.10.11/js/jquery.dataTables.min.js"
type="text/javascript"></script>
<script src="https://cdn.datatables.net/1.10.11/js/dataTables.bootstrap.min.js"
type="text/javascript"></script>
<script>
   $('.custom-file-input').on('change', function() {
      let fileName = $(this).val().split('\\').pop();
      $(this).next('.custom-file-label').addClass("selected").html(fileName);
   });
```

```
    $(document).ready(function () {
        $('#dtBasicExample').DataTable({
         "order": [[0, "desc"]]
      });
    });

    function show_img_details(img_path){
        $('#img_detail').attr("src", img_path);
        $('#img_detail_modal').modal('toggle');
    }

    let text_to_speech = new SpeechSynthesisUtterance();
    let synth = window.speechSynthesis;
    var playButton = function(image_text) {
        // SpeechSynthesisUtterance Configuration
        text_to_speech.lang = 'en';
        text_to_speech.voice = synth.getVoices()[0];
        text_to_speech.text = image_text;
        if (image_text){
            console.log(image_text,"text_to_speech")
            synth.speak(text_to_speech)
        }

    }

</script>

{% endblock %}


{% block extendme%}
<section>

    <div class="mt-5 text-center">
        <h1>Voco Buddy</h1>
    </div>

    <div class="" id="image_form">
        <form action="#image_form" method="POST" enctype="multipart/form-data">
            {% csrf_token %}
            <fieldset class="form-group">
                <div class="custom-file my-2">
                    {{image_form.image|add_class:"custom-file-input"}}
```

```
            <label class="custom-file-label">{{ image_form.image.label }}</label>
            {% for error in field.errors %}
            <span class="badge badge-danger">{{ error }}</span>
            {% endfor %}
          </div>
          <button type="submit" class="btn btn-primary">Submit</button>
        </fieldset>
      </form>
    </div>

    <div id="collapse1">
      <table id="dtBasicExample" class="table table-striped table-bordered table-sm ">
        <thead class="thead-dark">
        <tr>
          <th scope="col" >ID</th>
          <th scope="col" class="d-none d-md-table-cell">OCR Text</th>
          <th scope="col" class="text-center">Image</th>
          <th scope="col" >Play Audio</th>
        </tr>
        </thead>
        <tbody>
        {% for image in image_list %}
        <tr>
          <th scope="row">{{image.id}}</th>
          {% with image.ocrtext_set.last as ocrtext%}
          <td class="d-none d-md-table-cell">{{ ocrtext.text|truncatechars:350}}</td>
          {% endwith%}
          <td class="text-center">
            <div  id="clickable_img"
onclick="show_img_details('/media/{{image.image}}')">
              <img class="img-fluid img-thumbnail " src="/media/{{image.image}}"
alt="{{image.image}}" width="100" />
            </div>
          </td>
          {% with image.ocrtext_set.last as ocrtext%}

          <td onclick="playButton(`{{ocrtext.text}}`)" class="d-none d-md-table-cell
play_button">
            <div  id="play_button">
              <img src="/media/OCR_image/input/play-button.svg"/ width="50">
            </div>
          </td>
          {% endwith%}
        </tr>
```

```
        {% endfor %}
        </tbody>
      </table>
    </div>


    <div id="img_detail_modal" class="modal mt-5" tabindex="-1" role="dialog">
      <div class="modal-dialog" role="document">
        <div class="modal-content">
          <div class="modal-header">
            <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
              <span aria-hidden="true">&times;</span>
            </button>
          </div>
          <div class="modal-body">
            <img id="img_detail" class="img-fluid" src="" alt="" />
          </div>
        </div>
      </div>
    </div>
</section>

<footer>
    {% include "uploader/footer.html" %}
</footer>
{% endblock %}
```

CSS Logic :

```
@import
url('https://fonts.googleapis.com/css?family=Poppins:200,300,400,500,600,700,800,900
&display=swap');
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
    scroll-behavior: smooth;
}
p
{
    font-weight: 300;
    color: #111;
}
```

```
.banner
{
   position: relative;
   width:100%;
   min-height: 100vh;
   display: flex;
   justify-content: center;
   align-items: center;
   background: url(image/VOCO\ 1.png);
   background-size: cover;
}
.banner .content
{
   max-width: 900px;
   text-align: center;
}
.banner .content h2
{
   font-size: 5em;
   color: rgb(11, 8, 8);
}
.banner .content p
{
   font-size: 1em;
   color: rgb(15, 13, 13);
}
.btn
{
   font-size: 1em;
   color: rgb(9, 8, 8);
   background: #ff0157;
   display: inline-block;
   padding: 10px 30px;
   margin-top: 20px;
   text-transform: uppercase;
   text-decoration: none;
   letter-spacing: 2px;
   transition: 0.5s;
}
.btn:hover



   letter-spacing: 6px;
```

```
}
header
{
   position: fixed;
   top: 0;
   left: 0;
   width: 100%;
   padding: 40px 100px;
   z-index: 10000;
   display: flex;
   justify-content: space-between;
   align-items: center;
   transition: 0.5s;
}
header.sticky
{
   background: #fff;
   padding: 10px 100px;
   box-sizing: 0 5px 20px rgba(0,0,0,0.05);
}
header .logo
{
   color: #fff;
   font-weight: 700;
   font-size: 2em;
   text-decoration: none;
}
header.sticky .logo
{
   color: #111;
}
header .logo span
{
   color: #ff0157;
}
header .navigation
{
   position: relative;
   display: flex;
}
header .navigation li
{
   list-style: none;
   margin-left: 30px;
```

```
}
header .navigation li a
{
   text-decoration: none;
   color: #fff;
   font-weight:300;
}
header.sticky .navigation li a
{
   color: #111;
}
header .navigation li a:hover
{
   color: #ff0157;
}
section
{
   padding: 100px;
}
.row
{
   position: relative;
   width: 100%;
   display: flex;
   justify-content: space-between;
}
.row .col50
{
   position: relative;
   width: 48%;
}
.titleText
{
   color: #111;
   font-size: 2em;
   font-weight: 300;
}
.titleText span
{
   color: #ff0157;
   font-weight: 700;
   font-size: 1.5em;
}
.row .col50 .imgBx
```

```
{
    position: relative;
    width: 100%;
   /* min-height: 300px;*/
   height: 100%;

}
.row .col50 .imgBx img
{
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    object-fit: cover;
}
title
{
    width: 100%;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
.menu .content
{
    display: flex;
    justify-content: center;
    flex-direction: row-reverse;
    flex-wrap: wrap;
    margin-top: 40px;
}
.menu .content .box
{
    width: 340px;
    margin: 20px;
    border: 15px solid #fff;
    box-shadow: 0 5px 35px rgba(0,0,0,0.08);
}
.menu .content .box .imgBx
{
    position: relative;
    width: 100%;
    height: 300px;
```

```
}
.menu .content .box .imgBx img
{
   position: absolute;
   top: 0;
   left: 0;
   width: 100%;
   height: 100%;
   object-fit: cover;
}
.menu .content .box .text
{
   padding: 15px 0 5px;
}
.menu .content .box .text h3
{
   font-weight: 400;
   color: #111;
}
.expert .content
{
   display: flex;
   justify-content: center;
   flex-wrap: wrap;
   flex-direction: row;
   margin-top: 40px;
}
.expert .content .box
{
   width: 250px;
   margin: 15px;
}
.expert .content .box .imgBx
{
   position: relative;
   width: 100%;
   height: 300px;
}
.expert .content .box .imgBx img
{
   position: absolute;
   top: 0;
   left: 0;
   width: 100%;
```

```
    height: 100%;
    object-fit: cover;
}
.expert .content .box .text
{
    padding: 15px 0 5px;
}
.expert .content .box .text h3
{
    font-weight: 400;
    color: #111;
}
.testimonial
{
    background: url(image/test.jpg);
    background-size: cover;
}
.white .titleText,
.white p
{
    color: #fff;
}
.testimonial .content
{
    display: flex;
    justify-content: center;
    flex-wrap: wrap;
    flex-direction: row;
    margin-top: 40px
.testimonial .content .box
{
    width: 340px;
    margin: 20px;
    padding: 40px;
    background: #fff;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
.testimonial .content .box .imgBx
{
    position: relative;
    width: 80px;
```

```
    height: 80px;
    margin-bottom: 20px;
    border-radius: 50%;
    overflow: hidden;
}
.testimonial .content .box .imgBx img
{
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    object-fit: cover;
}
.testimonial .content .box .text
{
    text-align: center;
}
.testimonial .content .box .text p
{
    color: #666;
    font-style: italic;
}
.testimonial .content .box .text h3
{
    margin-top: 20px;
    color: #111;
    font-size: 1em;
    color: #ff0157;
    font-weight: 600;
}
.contact
{
    background: url(image/contact.png);
    background-size: cover;
}
.contactForm
{
    padding: 75px 50px;
    background: #fff;
    box-shadow: 0 15px 50px rgba(0,0,0,0.01);
    max-width: 500px;
    margin-top: 50px;
}
```

```
.contactForm h3
{
   color: #111;
   font-size: 1.2em;
   margin-bottom: 20px;
   font-weight: 500;
}
.contactForm .inputBox
{
   position: relative;
   width: 100%;
   margin-bottom: 20px;
}
.contactForm .inputBox input,
.contactForm .inputBox textarea
{
   width: 100%;
   border: 1px solid #555;
   padding: 10px;
   color: #111;
   outline: none;
   font-size: 16px;
   font-weight: 300;
   resize: none;
}
.contactForm .inputBox input[type="submit"]
{
   font-size: 1em;
   color: #fff;
   background: #ff0157;
   display: inline-block;
   text-transform: uppercase;
   text-decoration: none;
   letter-spacing: 2px;
   transition: 0.5s;
   max-width: 100px;
   font-weight: 500;
   border: none;
   cursor: pointer;
}

.copyrightText
{
   padding: 8px 40px;
```

```
    border-top: 1px solid rgba(0,0,0,0.1);
    text-align: center;
}

.copyrightText p
{
    color: #333;
}

.copyrightText a
{
    color: #ff0157;
    font-weight: 500;
    text-decoration: none;
}

@media(max-width: 991px)
{
    header,
    header.sticky
    {
        padding: 10px 20px;
    }
    header .navigation
    {
        display: none;
    }
    header .navigation.active
    {
            width: 100%;
            height: calc(100% - 68px);
            position: fixed;
            top: 68px;
            left:0;
            display: flex;
            justify-content: center;
            align-items: center;
            flex-direction: column;
            background: #fff;
    }
    header .navigation li
    {
            margin-left: 0;
    }
```

```
header .navigation li a
{
        color: #111;
        font-size: 1.6em;
}
.menuToggle
{
        position: relative;
        width: 40px;
        height: 40px;
        background: url(image/3img.jpg);
        background-size: 30px;
        background-repeat: no-repeat;
        background-position: center;
        cursor: pointer;
}
.menuToggle.active
{
        background: url(image/4img.jpg);
        background-size: 25px;
        background-repeat: no-repeat;
        background-position: center;
}
header.sticky .menuToggle
{
        filter: invert(1);
}
section
{
        padding: 20px;
}
.banner .content h2
{
        font-size: 3em;
        color: #fff;
}
.row
{
        flex-direction: column;
}
.row .col50
{
        position: relative;
        width: 100%;
```

```
  }
  .row .col50 .imgBx
  {
        height: 300px;
        margin-top: 20px;
  }
  .menu .content
  {
        margin-top: 20px;
  }
  .menu .content .box
  {
        margin: 10px;
  }
  .menu .content .box .imgBx
 {
        height: 260px;
 }
  .title
  {
        text-align: center;
  }
  .titleText
  {
        font-size: 1.8em;
        line-height: 1.5em;
        margin-bottom: 15px;
        font-weight: 300;
  }
  .testimonials .content .box
  {
        margin: 10px;
        padding: 20px;
  }
  .contactForm
  {
        padding: 35px 40px;
        margin-top: 20px;
        margin-bottom: 30px;
  }
}
@media (max-width: 480px)
{
.banner .content h2
```

```
{
        font-size: 2.2em;
        color: #fff;
}
}
@import url('https://fonts.googleapis.com/css?family=Poppins');

/* BASIC */

html {
  background-color: #56baed;
}

body {
  font-family: "Poppins", sans-serif;
  height: 100vh;
}

a {
  color: #92badd;
  display:inline-block;
  text-decoration: none;
  font-weight: 400;
}

h2 {
  text-align: center;
  font-size: 16px;
  font-weight: 600;
  text-transform: uppercase;
  display:inline-block;
  margin: 40px 8px 10px 8px;
  color: #cccccc;
}



/* STRUCTURE */

.wrapper {
  display: flex;
  align-items: center;
  flex-direction: column;
  justify-content: center;
```

```
  width: 100%;
  min-height: 100%;
  padding: 20px;
}

#formContent {
  -webkit-border-radius: 10px 10px 10px 10px;
  border-radius: 10px 10px 10px 10px;
  background: #fff;
  padding: 30px;
  width: 90%;
  max-width: 450px;
  position: relative;
  padding: 0px;
  -webkit-box-shadow: 0 30px 60px 0 rgba(0,0,0,0.3);
  box-shadow: 0 30px 60px 0 rgba(0,0,0,0.3);
  text-align: center;
}

#formFooter {
  background-color: #f6f6f6;
  border-top: 1px solid #dce8f1;
  padding: 25px;
  text-align: center;
  -webkit-border-radius: 0 0 10px 10px;
  border-radius: 0 0 10px 10px;
}



/* TABS */

h2.inactive {
  color: #cccccc;
}

h2.active {
  color: #0d0d0d;
  border-bottom: 2px solid #5fbae9;
}



/* FORM TYPOGRAPHY*/
```

```
input[type=button], input[type=submit], input[type=reset]  {
  background-color: #56baed;
  border: none;
  color: white;
  padding: 15px 80px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  text-transform: uppercase;
  font-size: 13px;
  -webkit-box-shadow: 0 10px 30px 0 rgba(95,186,233,0.4);
  box-shadow: 0 10px 30px 0 rgba(95,186,233,0.4);
  -webkit-border-radius: 5px 5px 5px 5px;
  border-radius: 5px 5px 5px 5px;
  margin: 5px 20px 40px 20px;
  -webkit-transition: all 0.3s ease-in-out;
  -moz-transition: all 0.3s ease-in-out;
  -ms-transition: all 0.3s ease-in-out;
  -o-transition: all 0.3s ease-in-out;
  transition: all 0.3s ease-in-out;
}

input[type=button]:hover, input[type=submit]:hover, input[type=reset]:hover  {
  background-color: #39ace7;
}

input[type=button]:active, input[type=submit]:active, input[type=reset]:active  {
  -moz-transform: scale(0.95);
  -webkit-transform: scale(0.95);
  -o-transform: scale(0.95);
  -ms-transform: scale(0.95);
  transform: scale(0.95);
}

input[type=text] {
  background-color: #f6f6f6;
  border: none;
  color: #0d0d0d;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
```

```
  margin: 5px;
  width: 85%;
  border: 2px solid #f6f6f6;
  -webkit-transition: all 0.5s ease-in-out;
  -moz-transition: all 0.5s ease-in-out;
  -ms-transition: all 0.5s ease-in-out;
  -o-transition: all 0.5s ease-in-out;
  transition: all 0.5s ease-in-out;
  -webkit-border-radius: 5px 5px 5px 5px;
  border-radius: 5px 5px 5px 5px;
}

input[type=text]:focus {
  background-color: #fff;
  border-bottom: 2px solid #5fbae9;
}

input[type=text]:placeholder {
  color: #cccccc;
}



/* ANIMATIONS */

/* Simple CSS3 Fade-in-down Animation */
.fadeInDown {
  -webkit-animation-name: fadeInDown;
  animation-name: fadeInDown;
  -webkit-animation-duration: 1s;
  animation-duration: 1s;
  -webkit-animation-fill-mode: both;
  animation-fill-mode: both;
}

@-webkit-keyframes fadeInDown {
  0% {
    opacity: 0;
    -webkit-transform: translate3d(0, -100%, 0);
    transform: translate3d(0, -100%, 0);
  }
  100% {
    opacity: 1;
    -webkit-transform: none;
```

```
    transform: none;
  }
}

@keyframes fadeInDown {
  0% {
    opacity: 0;
    -webkit-transform: translate3d(0, -100%, 0);
    transform: translate3d(0, -100%, 0);
  }
  100% {
    opacity: 1;
    -webkit-transform: none;
    transform: none;
  }
}

/* Simple CSS3 Fade-in Animation */
@-webkit-keyframes fadeIn { from { opacity:0; } to { opacity:1; } }
@-moz-keyframes fadeIn { from { opacity:0; } to { opacity:1; } }
@keyframes fadeIn { from { opacity:0; } to { opacity:1; } }

.fadeIn {
  opacity:0;
  -webkit-animation:fadeIn ease-in 1;
  -moz-animation:fadeIn ease-in 1;
  animation:fadeIn ease-in 1;

  -webkit-animation-fill-mode:forwards;
  -moz-animation-fill-mode:forwards;
  animation-fill-mode:forwards;

  -webkit-animation-duration:1s;
  -moz-animation-duration:1s;
  animation-duration:1s;
}

.fadeIn.first {
  -webkit-animation-delay: 0.4s;
  -moz-animation-delay: 0.4s;
  animation-delay: 0.4s;
}

.fadeIn.second {
```

```
  -webkit-animation-delay: 0.6s;
  -moz-animation-delay: 0.6s;
  animation-delay: 0.6s;
}

.fadeIn.third {
  -webkit-animation-delay: 0.8s;
  -moz-animation-delay: 0.8s;
  animation-delay: 0.8s;
}

.fadeIn.fourth {
  -webkit-animation-delay: 1s;
  -moz-animation-delay: 1s;
  animation-delay: 1s;
}

/* Simple CSS3 Fade-in Animation */
.underlineHover:after {
  display: block;
  left: 0;
  bottom: -10px;
  width: 0;
  height: 2px;
  background-color: #56baed;
  content: "";
  transition: width 0.2s;
}

.underlineHover:hover {
  color: #0d0d0d;
}

.underlineHover:hover:after{
  width: 100%;
}



/* OTHERS */

*:focus {
    outline: none;
}
```

```
#icon {
  width:60%;
}

* {
  box-sizing: border-box;
}
```

Backend Logic:

```
# Create your views here.

class MainPageView(TemplateView):
    template_name = 'uploader/mainpage.html'

class ContactPageView(TemplateView):
    template_name = 'uploader/contactus.html'

class DocumentationPageView(TemplateView):
    template_name = 'uploader/documentation.html'


class TestimonialPageView(TemplateView):
    template_name = 'uploader/testimonial.html'

class AboutPageView(TemplateView):
    template_name = 'uploader/about.html'


def home(request):
    data = dict()

    image_form = ImageFileForm(request.POST or None, request.FILES or None)
    if image_form.is_valid():
        image = image_form.save()
        image.execute_and_save_ocr()
        redirect('image_to_text')

    image_list = ImageFile.objects.all().order_by('-id')

    data['image_form'] = image_form
    data['image_list'] = image_list
    return render(request, "uploader/index.html", data)
```

```
def image_to_text(request):
    data = dict()

    image_form = ImageFileForm(request.POST or None, request.FILES or None)
    if image_form.is_valid():
        image = image_form.save()
        image.execute_and_save_ocr()
        redirect('image_to_text')

    image_list = ImageFile.objects.all().order_by('-id')

    data['image_form'] = image_form
    data['image_list'] = image_list
    return render(request, "uploader/image_to_text.html", data)
from django.db import models
from core import utils
import hashlib
from PIL import Image
import pytesseract


class ImageFileManager(models.Manager):

    def search(self, query):
        return self.get_queryset().filter(models.Q(internal_reference__icontains=query) |
                                          models.Q(name__icontains=query) |
                                          models.Q(description__icontains=query)
                                          )


class ImageFile(models.Model):

    name = models.CharField("Name", max_length=100)
    internal_reference = models.CharField("Internal Reference", max_length=100,
editable=False)
    description = models.TextField("Description", blank=True, null=True)
    image = models.ImageField(upload_to="OCR_image/input/", verbose_name="Input
Image")
    create_at = models.DateTimeField("Create at", auto_now_add=True)
    updated_at = models.DateTimeField("Update at", auto_now=True)

    def __str__(self):
        return "{0:03d} - {1}".format(self.id, self.image)
```

```
    def execute_and_save_ocr(self):
        import time
        start_time = time.time()

        img = Image.open(self.image)
        txt = pytesseract.image_to_string(img, lang='eng')
        execution_time = time.time() - start_time
        ocr_txt = OCRText(image = self, text = txt, lang = "EN", execution_time =
execution_time)
        ocr_txt.save()

        print("The image {0} was opened.".format(self.image))
        print('OCR: \n{0}\n'.format(txt))
        print('Execution Time: {0}'.format(ocr_txt.execution_time))

        return ocr_txt

    """
    def get_absolute_url(self):
        from django.urls import reverse
        return reverse('course_details', args=[], kwargs={'slug': self.slug})
    """

    def save(self, *args, **kwargs):

        if not self.internal_reference:
            random_value = utils.random_value_generator(size=20)
            while ImageFile.objects.filter(internal_reference=random_value).exists():
                random_value = utils.random_value_generator(size=20)
            hash_value = hashlib.md5(bytes(str(self.id) + str(random_value), 'utf-8'))
            self.internal_reference = hash_value.hexdigest()
        super(ImageFile, self).save(*args, **kwargs)

    class Meta:
        verbose_name = "ImageFile"
        verbose_name_plural = "ImageFiles"
        ordering = ['id']

    objects = ImageFileManager()
class OCRText(models.Model):
    text = models.TextField("OCR text", blank=True)
    lang = models.TextField("Language", default="EN")
    execution_time = models.IntegerField("Execution Time", editable=False, null=True);
```

```
    image = models.ForeignKey('ImageFile', on_delete=models.CASCADE)
    create_at = models.DateTimeField("Create at", auto_now_add=True)
    updated_at = models.DateTimeField("Update at", auto_now=True)

    def __str__(self):
        return "{0:03d} - {1}".format(self.id, self.image.internal_reference)

    class Meta:
        verbose_name = "OCRText"
        verbose_name_plural = "OCRTexts"
        ordering = ['id']
from django.urls import path
from uploader.views import home,image_to_text,MainPageView, ContactPageView,
DocumentationPageView,TestimonialPageView,AboutPageView

urlpatterns = [
    # path('', home, name='home'),
    path('image_to_text/', image_to_text, name='image_to_text'),
    path('', MainPageView.as_view(), name='mainpage'),
    path('contactus/', ContactPageView.as_view(), name='contactus'),
    path('testimonial/', TestimonialPageView.as_view(), name='testimonial'),
    path('documentation/', DocumentationPageView.as_view(), name='documentation'),
    path('about/', AboutPageView.as_view(), name='about'),
]

import os
import sys

if __name__ == '__main__':
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'OCR_web.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
from django.shortcuts import render, redirect
from uploader.forms import ImageFileForm
from uploader.models import ImageFile

from django.views.generic import TemplateView
```

```
# Create your views here.

class MainPageView(TemplateView):
    template_name = 'uploader/mainpage.html'


class ContactPageView(TemplateView):
    template_name = 'uploader/contactus.html'

class DocumentationPageView(TemplateView):
    template_name = 'uploader/documentation.html'
class TestimonialPageView(TemplateView):
    template_name = 'uploader/testimonial.html'

class AboutPageView(TemplateView):
    template_name = 'uploader/about.html'


def home(request):
    data = dict()

    image_form = ImageFileForm(request.POST or None, request.FILES or None)
    if image_form.is_valid():
        image = image_form.save()
        image.execute_and_save_ocr()
        redirect('image_to_text')

    image_list = ImageFile.objects.all().order_by('-id')

    data['image_form'] = image_form
    data['image_list'] = image_list
    return render(request, "uploader/index.html", data)

def image_to_text(request):
    data = dict()

    image_form = ImageFileForm(request.POST or None, request.FILES or None)
    if image_form.is_valid():
        image = image_form.save()
        image.execute_and_save_ocr()
        redirect('image_to_text')
    image_list = ImageFile.objects.all().order_by('-id')
```

```
    data['image_form'] = image_form
    data['image_list'] = image_list
    return render(request, "uploader/image_to_text.html", data)


from django import forms
from uploader.models import ImageFile


class ImageFileForm(forms.ModelForm):
    class Meta:
        model = ImageFile
        fields = ('image', )

from django.contrib import admin
from uploader.models import ImageFile

admin.site.register(ImageFile)


    def execute_and_save_ocr(self):
        import time
        start_time = time.time()

        img = Image.open(self.image)
        txt = pytesseract.image_to_string(img, lang='eng')
        execution_time = time.time() - start_time
        ocr_txt = OCRText(image = self, text = txt, lang = "EN", execution_time =
execution_time)
        ocr_txt.save()

        print("The image {0} was opened.".format(self.image))
        print('OCR: \n{0}\n'.format(txt))
        print('Execution Time: {0}'.format(ocr_txt.execution_time))

        return ocr_txt

    """
    def get_absolute_url(self):
        from django.urls import reverse
        return reverse('course_details', args=[], kwargs={'slug': self.slug})
    """
```

```
    def save(self, *args, **kwargs):

        if not self.internal_reference:
            random_value = utils.random_value_generator(size=20)
            while ImageFile.objects.filter(internal_reference=random_value).exists():
                random_value = utils.random_value_generator(size=20)
            hash_value = hashlib.md5(bytes(str(self.id) + str(random_value), 'utf-8'))
            self.internal_reference = hash_value.hexdigest()
        super(ImageFile, self).save(*args, **kwargs)

    class Meta:
        verbose_name = "ImageFile"
        verbose_name_plural = "ImageFiles"
        ordering = ['id']

    objects = ImageFileManager()
class OCRText(models.Model):
    text = models.TextField("OCR text", blank=True)
    lang = models.TextField("Language", default="EN")
    execution_time = models.IntegerField("Execution Time", editable=False, null=True);
    image = models.ForeignKey('ImageFile', on_delete=models.CASCADE)
    create_at = models.DateTimeField("Create at", auto_now_add=True)
    updated_at = models.DateTimeField("Update at", auto_now=True)

    def __str__(self):
        return "{0:03d} - {1}".format(self.id, self.image.internal_reference)

    class Meta:
        verbose_name = "OCRText"
        verbose_name_plural = "OCRTexts"
        ordering = ['id']
from django.urls import path
from uploader.views import home,image_to_text,MainPageView, ContactPageView,
DocumentationPageView,TestimonialPageView,AboutPageView

urlpatterns = [
    # path('', home, name='home'),
    path('image_to_text/', image_to_text, name='image_to_text'),
    path('', MainPageView.as_view(), name='mainpage'),
    path('contactus/', ContactPageView.as_view(), name='contactus'),
    path('testimonial/', TestimonialPageView.as_view(), name='testimonial'),
    path('documentation/', DocumentationPageView.as_view(), name='documentation'),
    path('about/', AboutPageView.as_view(), name='about'),
]
```

```
import os
import sys

if __name__ == '__main__':
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'OCR_web.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
from django.shortcuts import render, redirect
from uploader.forms import ImageFileForm
from uploader.models import ImageFile

from django.views.generic import TemplateView


# Create your views here.

class MainPageView(TemplateView):
    template_name = 'uploader/mainpage.html'


class ContactPageView(TemplateView):
    template_name = 'uploader/contactus.html'

class DocumentationPageView(TemplateView):
    template_name = 'uploader/documentation.html'
class TestimonialPageView(TemplateView):
    template_name = 'uploader/testimonial.html'

class AboutPageView(TemplateView):
    template_name = 'uploader/about.html'


def home(request):
    data = dict()

    image_form = ImageFileForm(request.POST or None, request.FILES or None)
```

```
    if image_form.is_valid():
        image = image_form.save()
        image.execute_and_save_ocr()
        redirect('image_to_text')

    image_list = ImageFile.objects.all().order_by('-id')

    data['image_form'] = image_form
    data['image_list'] = image_list
    return render(request, "uploader/index.html", data)

def image_to_text(request):
    data = dict()

    image_form = ImageFileForm(request.POST or None, request.FILES or None)
    if image_form.is_valid():
        image = image_form.save()
        image.execute_and_save_ocr()
        redirect('image_to_text')
    image_list = ImageFile.objects.all().order_by('-id')
```

Chapter - 8 Results

Observed outcome of project:

• Text is extracted from the image and converted to audio.
• It recognizes both capital as well as small letters.
• It recognizes numbers as well.
• Range of reading distance was 38-42cm.
• Character font size should be minimum 12pt.
• Maximum tilt of the text line is 4-5 degree from the vertical

Chapter – 9 Future Scope

Text to speech converters is a relatively new industry, but it is exploding with growth at the moment. It is not only just about assisted reading, but it is becoming something bigger and more important as the days go by.

Current Text to Speech Converters

Before we get into the near future, I should lay out how the current climate is.
They are offering great features like adjusting the voice and using a totally different natural voice to convert as well.
Do not get it twisted and think that is all that is going on right now.
The crazy thing is that we are evolving to even more complex solutions.

Machine Learning

Do you remember in the science fiction movies when robots could speak exactly like us and in the same tone? Well, that is looking like more and more of a possibility in the near future because of machine learning.
Mixing text to speech converters and a machine learning how to speak and mimicking humans is going to lead to some next-level technology. Just think about a robot being able to have a conversation with you someday. If that does not freak you out, then you are crazy!

Taking Over Education

Another wild reality is that it could possibly overthrow some forms of education. The technology is already being used to help people with disabilities and not able to read, but it could advance even more.

It could be a possibility that text to speech converters end up taking over the education system because it would be cheaper for a school to pay a converter than a full-time teacher employee! Even though that would be a long way away before it would happen, it is still a crazy thought to ponder.

No More Typing

The last point that I will leave you with is the fact that typing could be a skill of the past if we continue down this road. Text to speech converters is big, but what if speech to text takes over?

It would be a thing of the past to type out your message or paper, because we could just use our voice. It does make sense in some regards because we can probably speak much faster than typing in most cases. But there are certain drawbacks that could hinder the expansion of this idea.

The only thing that we do right now is wait and see how the text to speech world is going to change. Not only are the converters getting better with their methods, but the entire industry has the ability to make a lasting impact very soon.

Chapter – 10 References

1. Archana A, Shinde D. Text pre-processing and text seg-mentation for OCR. International Journal of Computer Science Engineering and Technology. 2012:810–12.
2. Mithe R, Indalkar S, Divekar N. Optical character recog-nition. International Journal of Recent Technology and Engineering. 2013 Mar; 2(1).
3. Smith R. An overview of the Tesseract OCR engine, USA: Google Inc; 2007.
4. Shah H, Shah A. Optical character recognition of Gujarati numerical. International Conference on Signals, Systems and Automation. 2009; 49–53.
5. Monk S. Raspberry pi cook.
6. Text localization and extraction in images using mathemat-ical morphology and OCR Techniques; 2013.
7. Kumar GS, Krishna MNVLM. Low-cost speech recognition system running on Raspberry Pi to support Automation applications. International Journal of Engineering Trends and Technology. 2015; 21(5).
8. Bhargava A, Nath KV, Sachdeva P, Samel M. Reading assis-tant for visually Impaired. International Journal of current Engineering and Technology. 2015 Apr; 5(2).