VOICE BASED EMAIL SYSTEM FOR VISUALLY IMPAIRED

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2023-24)

Submitted by

Aviral
2200290140044
Divanshu Aggarwal
2200290140056

Submitted in partial fulfillment of the
Requirements for the Degree of
MASTER OF COMPUTER APPLICATION
Under the Supervision of
Dr. VIPIN KUMAR

Submitted to
Department Of Computer Applications
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

ABBREVATIONS	MEANING
WS API	Web Speech API
DTMF	Dual Tone Multi Frequency
ASR	Automatic Speech Recognition
STT	Speech To Text
TTS	Text To Speech
GUI	Graphical User Interface
PCA	Principle Component Analysis
HTML	Hypertext Markup Language
VRU	Voice Reaction Unit
ADC	Analog To Digital Converter
API	Application Program Interface
gTTS	Google Text To Speech
SMTP	Simple Mail Transfer Protocol
IMAP	Internet Message Access Protocol
TCP	Transmission Control Protocol

Candidate's Declaration

Certified that **Aviral (2200290140044) and Divanshu Aggarwal(2200290140056)** has/ have carried out the project work having **"VOICE BASED EMAIL SYSTEM FOR BLIND"** (**Mini Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project 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/Institution.

Date:01/01/2024 **Aviral (2200290140044)**
Divanshu Aggarwal (2200290140056)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.
Date:01/01/2024

Dr. Vipin Kumar
Associate Professor
Master of Computer Application
KIET Group of Institutions Ghaziabad
(U.P), India Dated:01/01/2024

Dr. Arun Kumar Tripathi
Head
Master of Computer Application
KIET Group of Institutions Ghaziabad
(U.P), India Dated:01/01/2024

ACKNOWLEDGEMENT

There were lots of individual who were certainly very helpful, kind, cooperative and generous along the development of this project. I express my appreciation to those concerned. This project and report would have been not possible if it was not for the kind support, tremendous support and continuous supervision of my learned project guide. The deep sense of sincere appreciation that I owe to my learned guide and supervisor **"Dr. Vipin Kumar"** cannot be expressed in words. His impossible help, constant support, invaluable supervision, kind nature are among the very few consent that he bestowed upon me time to time from the beginning of this project.

ABSTRACT

Internet is one of the basic luxury for daily living. Every person is using the facts and information on internet. On the other hand, blind people face difficulty in accessing the text resources. The development in computer based handy systems has opened up numerous oppurtunities for the visually disabled across. Audio response based virtual environment, the screen readers are helps blind people a lot to use internet applications.

This project introduces the Voice email system structural design that can be used by a blind person to access E-Mails easily. The involvement of research is helping blind individual to send and receive voice based mails messages in their inhabitant language with the help of a computer.

Chapter 1:
INTRODUCTION

We have seen that the introduction of Internet has revolutionized many fields. Internet has made life of people so easy that people today have access to any information they want easily. Communication is one of the main fields highly changed by Internet.

E-mails are the most dependable way of communication over Internet, for sending and receiving some important information. But there is a certain norm for humans to access the Internet and the norm is you must be able to see. But there are also differently abled people in our society who are not gifted with what you have. There are some visually impaired people or blind people who can’t see things and thus can’t see the computer screen or keyboard.

A survey has shown that there are more than 253 million visually impaired people around the globe. That is, around 253 million people are unaware of how to use Internet or E-mail. The only way by which a visually challenged person can send an E-mail is, they have to speak the entire content of the mail to another person(not visually challenged) and then that third person will compose the mail and send on the behalf of the visually challenged person. But this is not a right way to deal with the problem. It is very unlikely that every time a visually impaired person can find someone for help.

1.1: Project Objectives

This project proposes a React and Node JS based application, designed specifically for visually impaired people. This application provide a voice based mailing service where they could read and send mail on their own, without any guidance through their g-mail accounts. Here, the users have to use certain keywords which will perform certain actions for e.g. Read, Send, Compose Mail etc. The V-MAIL system can be used by a blind person to access mails easily and adeptly. Hence dependence of visually challenged on other individual for their activities associated to mail can be condensed.

The application will be React and node JS based application for visually challenged persons using Web Speech API voice response, thus sanctioning everyone to control their mail accounts using their voice only and to be able to read, send, and perform all the other useful tasks. The system will ask the user with voice commands to perform certain action and the user will respond to it. **The main advantage of this system is that use of keyboard is completely eliminated , the user will have to respond through voice only.**

1.2 Web Speech API

The Web Speech API makes web apps able to handle voice data. There are two components to this API:

Speech recognition is accessed via the SpeechRecognition interface, which provides the ability to recognize voice context from an audio input (normally via the device's default speech recognition service) and respond appropriately. Generally you'll use the interface's constructor to create a new SpeechRecognition object, which has a number of event handlers available for detecting when speech is input through the device's microphone. The SpeechGrammar interface represents a container for a particular set of grammar that your app should recognize. Grammar is defined using JSpeech Grammar Format.

Speech synthesis is accessed via the SpeechSynthesis interface, a text-to-speech component that allows programs to read out their text content (normally via the device's default speech synthesizer.) Different voice types are represented by SpeechSynthesisVoice objects, and different parts of text that you want to be spoken are represented by SpeechSynthesisUtterance objects. You can get these spoken by passing them to the SpeechSynthesis.speak() method.

Web Speech API Interfaces

SpeechRecognition

The controller interface for the recognition service; this also handles the SpeechRecognitionEvent sent from the recognition service.

SpeechRecognitionAlternative

Represents a single word that has been recognized by the speech recognition service.

SpeechRecognitionErrorEvent

Represents error messages from the recognition service.

1.3:Speech Recognition

Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT). It incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields.Some speech recognition systems require "training" (also called "enrollment") where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called "speaker independent" systems. Systems that use training are called "speaker dependent".

Speech recognition applications include voice user interfaces such as voice dialing (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), domotic appliance control, search (e.g. find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g. a radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed Direct Voice Input).

The term *voice recognition* or *speaker identification* refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice or it can be used to authenticate or verify the identity of a speaker as part of a security process.

From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data Speech recognition works using algorithms through acoustic and language modeling. Acoustic modeling represents the relationship between linguistic units of speech and audio signals;

language modeling matches sounds with word sequences to help distinguish between words that sound similar. Often, hidden Markov models are used as well to recognize temporal patterns in speech to improve accuracy within the system. The most frequent applications of speech recognition within the enterprise include call routing, speech-to-text processing, voice dialing and voice search.

While convenient, speech recognition technology still has a few issues to work through, as it is continuously developed. The pros of speech recognition software are it is easy to use and readily available. Speech recognition software is now frequently installed in computers and mobile devices, allowing for easy access. The downside of speech recognition includes its inability to capture words due to variations of pronunciation, its lack of support for most languages outside of English and its inability to sort through background noise. These factors can lead to inaccuracies.

Speech recognition performance is measured by accuracy and speed. Accuracy is measured with word error rate. WER works at the word level and identifies inaccuracies in transcription, although it cannot identify how the error occurred. Speed is measured with the real-time factor. A variety of factors can affect computer speech recognition performance, including pronunciation, accent, pitch, volume and background noise.It is important to note the terms *speech recognition* and *voice recognition* are sometimes used interchangeably.

However, the two terms mean different things. Speech recognition is used to identify words in spoken language. Voice recognition is a biometric technology used to identify a particular individual's voice or for speaker identification.

1.4:Speech Recognition In React JS

The improvement and accesibility alone in the field of speech recognition are worth considerable. It allows the physically and the elderly and visually challenged people to collaborate with state of the art products and services quickly and naturally no graphical user interface is needed.

If you want to use speech recognition or simply convert speech to text in as react componanet it is very easy to use. Let’s see how:-

- Working of speech recognition.
- Available in Web speech API .
- How to use and how to use speech recognition package using in react.

SpeechRecognition is a library that acts as a wrapper for many popular speech APIs and is thus very flexible to use. One of these is the Google Web Speech API which supports a default API key that is hard coded into the SpeechRecognition library.

The elasticity and easy to use features of the SpeechRecognition package in python make it a very good choice for developers who are working on any python project. It does not guarantee to support every feature that is wrapped with this API. You will have to dispense some time searching for the easily available options to find out if SpeechRecognition is going work in your particular case.

1.4.1:Required Installations

Using the SpeechRecognition API in React involves several steps. The SpeechRecognition API allows you to capture audio input from a user's microphone and convert it into text. Here's a basic step-by-step guide on how to use SpeechRecognition in a React application:

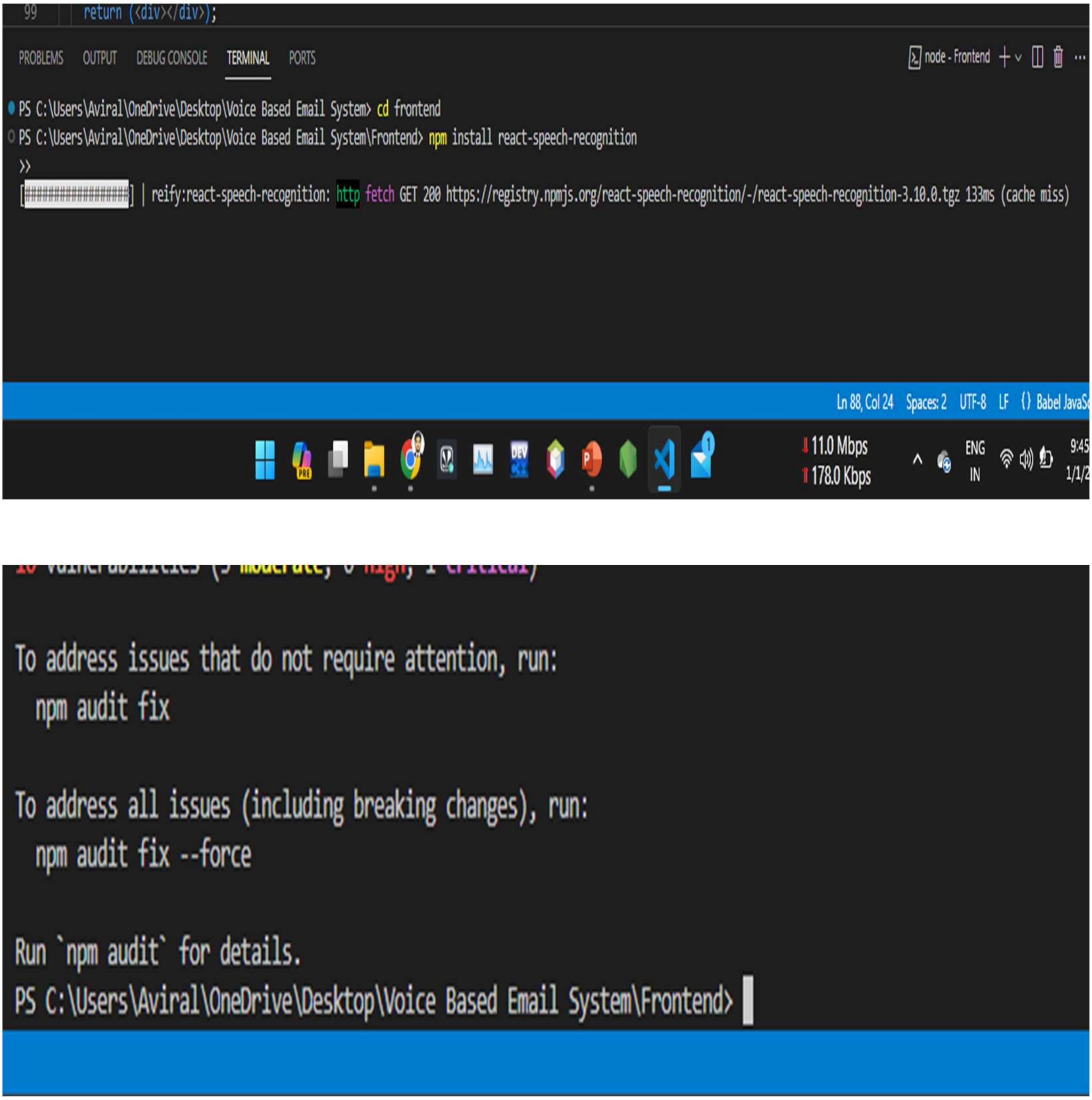
Create a new React app:

If you haven't already, use create-react-app or your preferred method to set up a new React application.

```
npx create-react-app speech-recognition-app
cd speech-recognition-app
```

Install react-speech-recognition package:

Install a package like react-speech-recognition that simplifies the usage of the SpeechRecognition API in a React app.

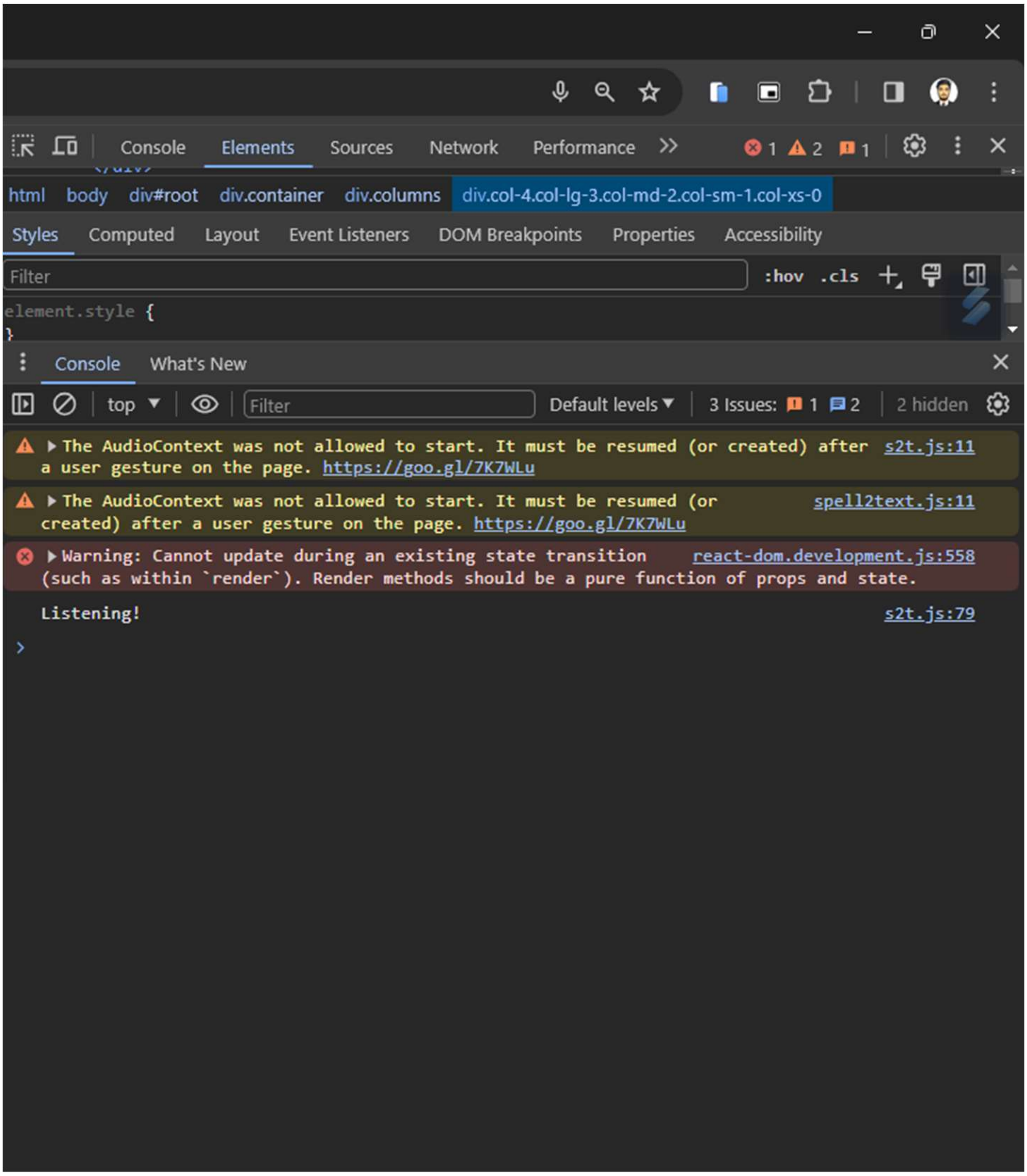


```
return (<div></div>);
```

```
PS C:\Users\Aviral\OneDrive\Desktop\Voice Based Email System> cd frontend
PS C:\Users\Aviral\OneDrive\Desktop\Voice Based Email System\Frontend> npm install react-speech-recognition
>>
[###############] | reify:react-speech-recognition: http fetch GET 200 https://registry.npmjs.org/react-speech-recognition/-/react-speech-recognition-3.10.0.tgz 133ms (cache miss)
```

```
10 vulnerabilities (3 moderate, 6 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS C:\Users\Aviral\OneDrive\Desktop\Voice Based Email System\Frontend>
```

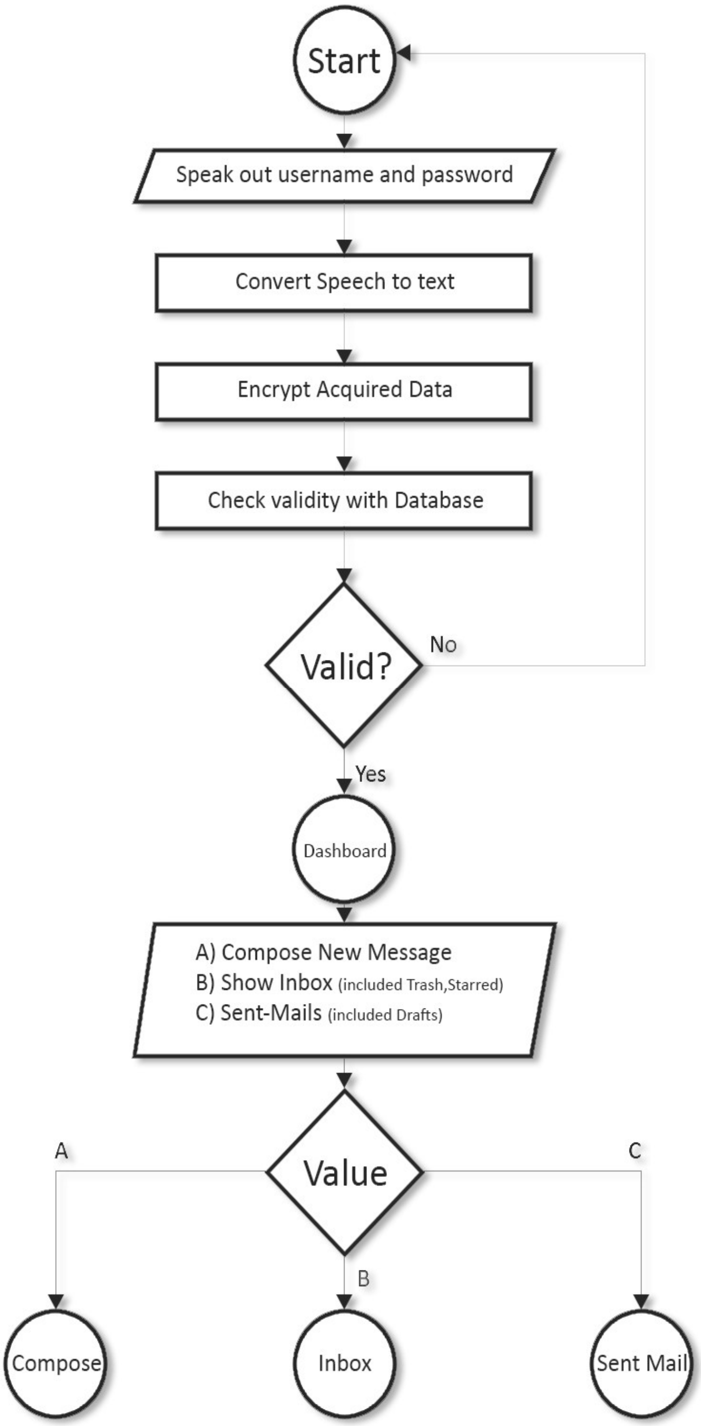
Figure-I 1.4.1-: Installation of the React app



```
element.style {
}
```

```
▶The AudioContext was not allowed to start. It must be resumed (or created) after  s2t.js:11
a user gesture on the page. https://goo.gl/7K7WLu
▶The AudioContext was not allowed to start. It must be resumed (or        spell2text.js:11
created) after a user gesture on the page. https://goo.gl/7K7WLu
▶Warning: Cannot update during an existing state transition     react-dom.development.js:558
(such as within `render`). Render methods should be a pure function of props and state.
Listening!                                                                         s2t.js:79
>
```

Figure-II 1.4.1: Listening your voice


Start
Speak out username and password
Convert Speech to text
Encrypt Acquired Data
Check validity with Database
Valid?
No
Yes
Dashboard
A) Compose New Message
B) Show Inbox (included Trash,Starred)
C) Sent-Mails (included Drafts)
Value
A
B
C
Compose
Inbox
Sent Mail

1.4.2 SYSTEM BLOCK DIAGRAM

1.5: Speech to text Converter

The Speech-to-Text (STT) conversion in a React application typically involves using the Web Speech API or a third-party library. Here's a theoretical overview of how Speech-to-Text conversion works in React:

- **Web Speech API or Third-Party Library:**

Web Speech API:

React applications can leverage the Web Speech API, a browser-based API that allows developers to integrate speech recognition features. The API provides interfaces for both speech recognition and speech synthesis.

Third-Party Libraries:

Alternatively, developers can use third-party libraries like react-speech-recognition or annyang that provide a React-friendly interface for speech recognition.

- **Initialization:**

Web Speech API:

To use the Web Speech API for speech recognition, you need to create an instance of the SpeechRecognition object, configure its settings, and attach event listeners to handle recognition events.

Third-Party Libraries:

Third-party libraries typically abstract away the complexities of the Web Speech API and provide React hooks or components to simplify integration. Initialization involves importing the necessary functions or components and incorporating them into your React components.

- **User Permission:**

Web Speech API:

The browser will ask the user for permission to access the microphone.

Third-Party Libraries:

The library may handle user permission internally, or you might need to handle it explicitly based on the library's documentation.

- **Start and Stop Recognition:**

Web Speech API:

You can start and stop recognition using methods like start() and stop() on the SpeechRecognition object.

Third-Party Libraries:

Third-party libraries often provide hooks or functions like startListening() and stopListening() to initiate or stop speech recognition.

1.5.1:Speech to text Converter

Event Handling:

Web Speech API:

The Web Speech API triggers events such as result, end, and error. You need to attach event listeners to handle these events and extract the recognized text.

Third-Party Libraries:

Libraries often provide hooks like useSpeechRecognition() that return values like transcript containing the recognized text. You can use this data to update your React component.

Integration with React Components:

Web Speech API:

You'll integrate the Web Speech API functionality into your React components by handling events and updating the component state accordingly.

Third-Party Libraries:

Integration with third-party libraries is typically done through React components or hooks. You'll use these components or hooks in your JSX, and they will handle the underlying speech recognition logic.

Error Handling:

Web Speech API:

You need to handle errors that may occur during the speech recognition process by listening for the error event.

Third-Party Libraries:

Third-party libraries may provide error handling mechanisms or expect you to handle errors based on their documentation.

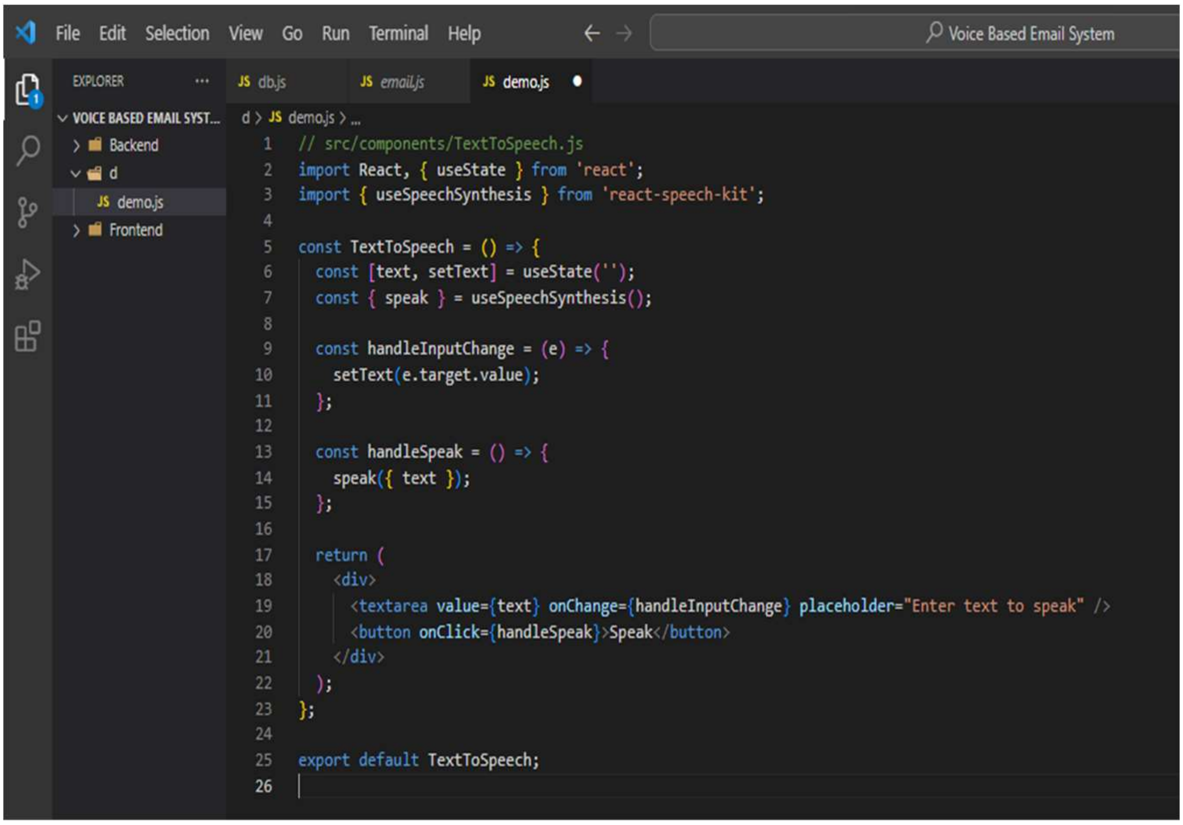
1.6 TEXT TO SPEECH IN REACT

Implementing Text-to-Speech (TTS) functionality in a React application can be achieved using the Web Speech API or a third-party library. Below is a guide on how to implement Text-to-Speech using both approaches:

Using the Web Speech API:

Create a React Component:

Create a React component where you want to incorporate the Text-to-Speech functionality.

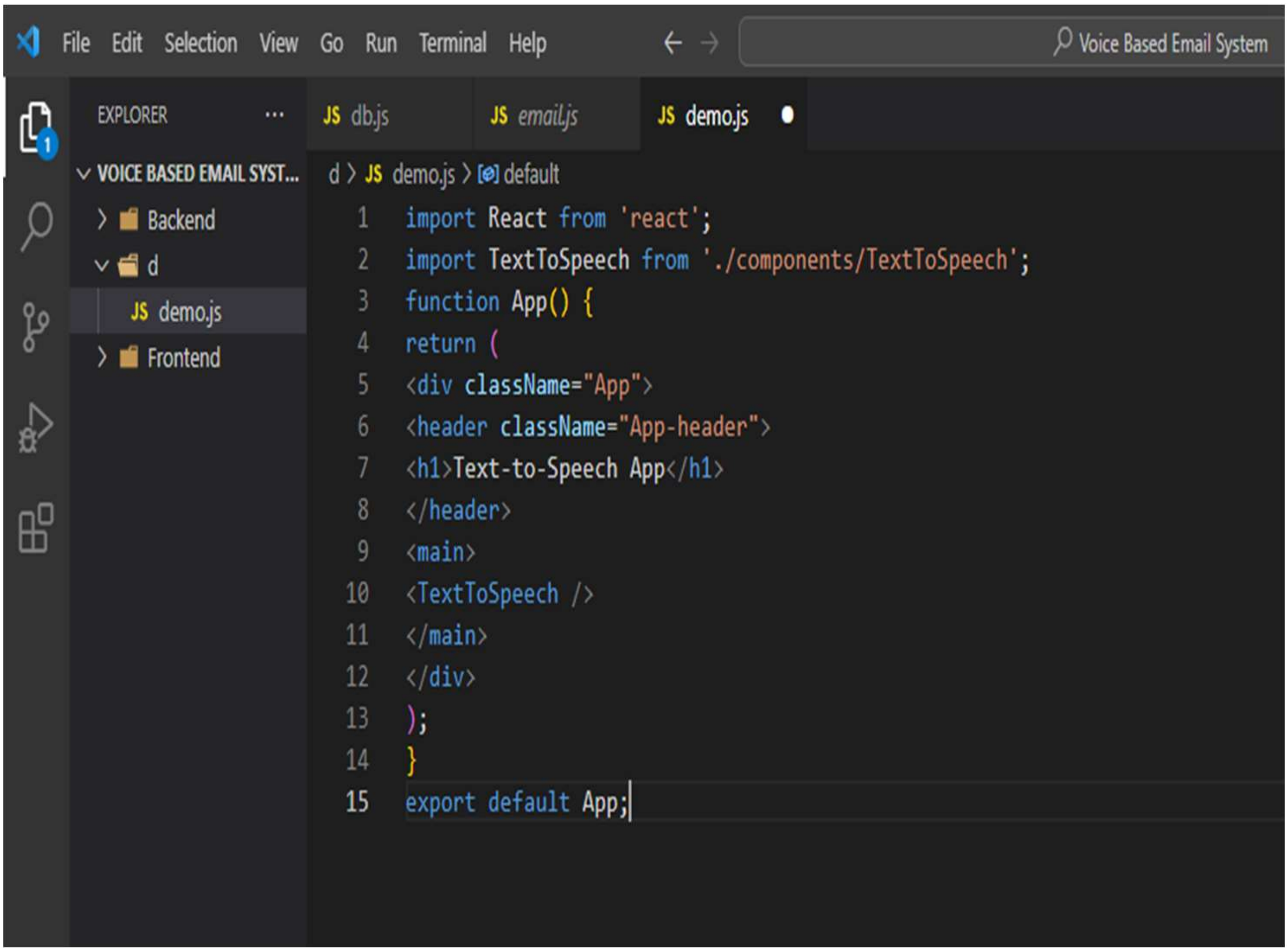


```
// src/components/TextToSpeech.js
import React, { useState } from 'react';
import { useSpeechSynthesis } from 'react-speech-kit';

const TextToSpeech = () => {
  const [text, setText] = useState('');
  const { speak } = useSpeechSynthesis();

  const handleInputChange = (e) => {
    setText(e.target.value);
  };

  const handleSpeak = () => {
    speak({ text });
  };

  return (
    <div>
      <textarea value={text} onChange={handleInputChange} placeholder="Enter text to speak" />
      <button onClick={handleSpeak}>Speak</button>
    </div>
  );
};

export default TextToSpeech;
```

Figure-I 1.6: Text To Speech Example

Integrate the Component:

Integrate the TextToSpeech component into your main App component.



```
import React from 'react';
import TextToSpeech from './components/TextToSpeech';
function App() {
return (
<div className="App">
<header className="App-header">
<h1>Text-to-Speech App</h1>
</header>
<main>
<TextToSpeech />
</main>
</div>
);
}
export default App;
```

Figure-II 1.6: Integration of Component

Chapter 2
LITRATURE REVIEW

2.1 “Voice Based System in Desktop and Mobile Devices for Blind People”. In International Journal of Emerging Technology and Advanced Engineering (IJETAE), 2014

This paper deals with “Voice Based System in Desktop and Mobile Devices for Blind People”. Voice mail architecture helps blind people to access e-mail and other multimedia functions of operating system (songs, text).Also in mobile application SMS can be read by system itself. Now a days the advancement made in computer technology opened platforms for visually impaired people across the world. It has been observed that nearly about 60% of total blind population across the world is present in INDIA. In this paper, we describe the voice mail architecture used by blind people to access E-mail and multimedia functions of operating system easily and efficiently. This architecture will also reduce cognitive load taken by blind to remember and type characters using keyboard. There is bulk of information available on technological advances for visually impaired people. This includes development of text to Braille systems, screen magnifiers and screen readers. Recently, attempts have been made in order to develop tools and technologies to help Blind people to access internet technologies. Among the early attempts, voice input and input for surfing was adopted for the Blind people. In IBM’s Home page the web page is an easy-to-use interface and converts the text-to-speech having different gender voices for reading texts and links. However, the disadvantage of this is that the developer has to design a complex new interface for the complex graphical web pages to be browsed and for the screen reader to recognize.

Simple browsing solution, which divides a web page into two dimensions. This greatly simplifies a web page’s structure and makes it easier to browse. Another web browser generated a tree structure from the HTML document through analyzing links. As it attempted to structure the pages that are linked together to enhance navigability, it did not prove very efficient for surfing. After, it did not handle needs regarding navigability and usability of current page itself. Another browser developed for the visually handicapped people was eGuideDog which had an integrated TTS engine. This system applies some advanced text extraction algorithm to represent the page in a user-friendly manner. However, still it did not meet the required standards of commercial use. Considering Indian scenario, ShrutiDrishti and WebBrowser for Blind are the two web browser framework that are used by Blind people to access the internet including the emails. Both the systems are integrated with Indian language ASR and TTS systems. But the available systems are not portable for small devices like mobile phones.

2.2“Voice Based Search Engine and Web page Reader”. In International Journal of Computational Engineering Research (IJCER)

This paper aims to develop a search engine which supports Man-Machine interaction purely in the form of voice. A novel Voice based Search Engine and Web-page Reader which allows the users to command and control the web browser through their voice, is introduced. The existing Search Engines get request from the user in the form of text and respond by retrieving the relevant documents from the server and displays in the form of text .Even though the existing web browsers are capable of playing audios and videos, the user has to request by typing some text in the search text box and then the user can play the interested audio/video with the help of Graphical User Interfaces (GUI). The proposed Voice based Search Engine aspires to serve the users especially the blind in browsing the Internet. The user can speak with the computer and the computer will respond to the user in the form of voice. The computer will assist the user in reading the documents as well. Voice-enabled interface with addition support for gesture based input and output approaches are for the “Social Robot Maggie” converting it into an aloud reader . This voice recognition and synthesis can be affected by number of reasons such as the voice pitch, its speed, its volume etc. It is based on the Loquendo ETTS (Emotional Text-To-Speech) software. Robot also expresses its mood through gesture that is based on gestionary. Speech recognition accuracy can be improved by removal of noise. In A Bayesian scheme is applied in a wavelet domain to separate the speech and noise components in a proposed iterative speech enhancement algorithm. This proposed method is developed in the wavelet domain to exploit the selected features in the time frequency space representation. It involves two stages: a noise estimate stage and a signal separation stage. In the Principle Component Analysis (PCA) based HMM for the visual modality of audio-visual recordings is used. PCA (Principle Component Analysis) and PDF (Probabilistic Density Analysis). Presents an approach to speech recognition using fuzzy modelling and decision making that ignores noise instead of its detection and removal. In the speech spectrogram is converted into a fuzzy linguistic description and this description is used instead of precise acoustic features.In Voice recognition technique combined with facial feature interaction to assist virtual artist with upper limb disabilities to create visual cut in a digital medium, preserve the individuality and authenticity of the art work. Techniques to recover phenomena such as Sentence Boundaries, Filler words and Disfluencies referred to as structural Metadata are discussed in and describe the approach that automatically adds information about the location of sentence boundaries and speech disfluencies in order to enrich speech recognition output. Clarissa a voice enabled procedure browser that is deployed on the international space station (ISS). The main components of the Clarissa system are speech recognition module a classifier for executing the open microphone accepts/reject decision, a semantic analysis and a dialog manager. Mainly focuses on expressions.

To build a prosody model for each expressive state, an end pitch and a delta pitch for each syllable are predicted from a set of features gathered from the text. The expression-

tagged units are then pooled with the neutral data, In a TTS system, such paralinguistic events efficiently provide clues as to the state of a transaction, and Markup specifying these events is a convenient way for a developer to achieve these types of events in the audio coming from the TTS engine.

Main features of are smooth and natural sounding speech can be synthesized, the voice characteristics can be changed, it is “trainable. Limitations of the basic system is that synthesized speech is “buzz” since it is based on a vocoding technique, it has been overcome by high quality vocoder and hidden semi-Markov model based acoustic modelling. Speech synthesis consists of three categories: Concatenation Synthesis, Articulation Synthesis, and Formant Synthesis.

Mainly focuses on formant synthesis, array of phoneme of syllable with formants frequency is given as input, frequency of given input is processed, on collaborated with Thai-Tonal-Accent Rules convert given formants frequency format to wave format, so that audio output via soundcard.

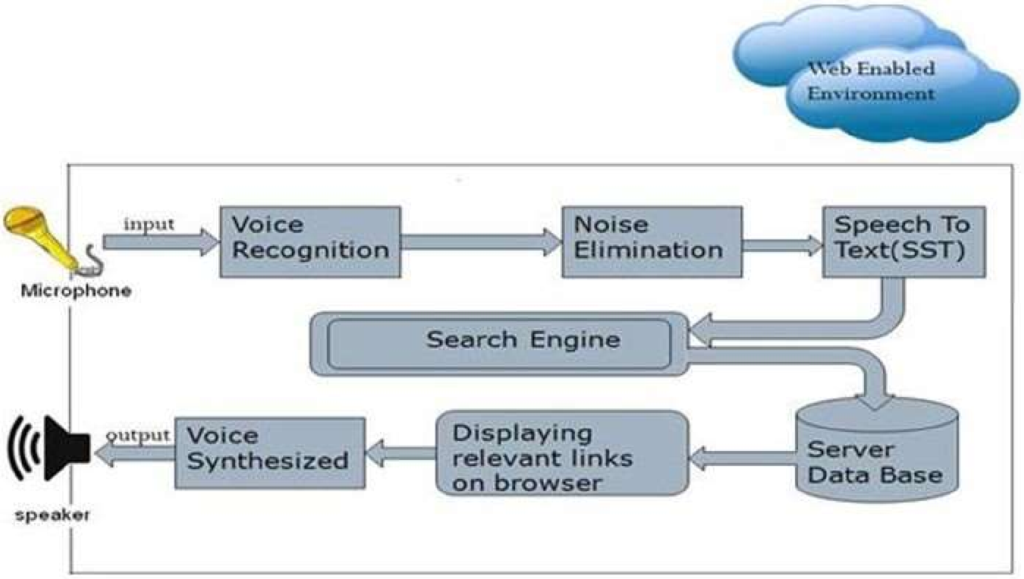


Figure-I 2.2: Voice Recognition Flow Diagram

2.3 “Voice Based Services for Blind People”. In International Journal of Advance Research, Ideas and Innovations in Technology(IJARIIT)

The advancement in computer based accessible systems has opened up many avenues for the visually impaired across a wide majority of the globe. Audio feedback based virtual environment like, the screen readers have helped blind people to access internet applications immensely. However, a large section of visually impaired people in different countries, in particular, the Indian sub-continent could not benefit much from such systems. This was primarily due to the difference in the technology required for Indian languages compared to those corresponding to other popular languages of the world. In this paper, we describe the voicemail system architecture that can be used by a blind person to access e-mails easily and efficiently. The contribution made by this research has enabled the blind people to send and receive voice-based e-mail messages in their native language with the help of a mobile device. Our proposed system GUI has been evaluated against the GUI of a traditional mail server. We found that our proposed architecture performs much better than that of the existing GUIS. In this project, we use voice to text and text to voice technique access for blind people.

The navigation system uses TTS (Text-to-Speech) for blindness in order to provide a navigation service through voice. Suggested system, as an independent program, is fairly cheap and it is possible to install onto Smartphone held by blind people.This allows blind people to easy access the program. An increasing number of studies have used technology to help blind people to integrate more fully into a global world. We present software to use mobile devices by blind users. The software considers a system of instant messenger to favor interaction of blind users with any other user connected to the network. Nowadays the advancement made in computer technology opened platforms for visually impaired people across the world. It has been observed that nearly about 60% of the total blind population across the world is present in INDIA. In this paper, we describe the voice mail architecture used by blind people to access E-mail and multimedia functions of the operating system easily and efficiently.This architecture will also reduce cognitive load taken by the blind to remember and type characters using the keyboard. It also helps handicapped and illiterate people. In previous work, blind people does not send email using the system. The multitude of email types along with the ability setting enables their use in nomadic daily contexts. But these emails are not useful in all types of people such as blind people they can't send the email. Audio based email are only preferable for blind peoples. They can easily respond to the audio instructions. In this system is very rare. So there is less chance to available this audio based email to the blind people. We describe the voicemail system architecture that can be used by a blind person to access e-mails easily and efficiently. The contribution made by this research has enabled the blind people to send and receive voice-based e-mail messages in their native language with the help of a computer or a mobile device. Our proposed system GUI has been evaluated against the GUI of a traditional mail server. We found that our proposed architecture performs much better than that of the existing GUIS.

It involves the development of the following modules:

SPEECH_ TO_ TEXT Converter :

The system acquires speech at run time through a microphone and processes the sampled speech to recognize the uttered text. The recognized text can be stored in a file. We are developing this on Android platform using Eclipse workbench. Our speech to-text system directly acquires and converts speech to text. It can supplement other larger systems, giving users a different choice for data entry. A speech-to-text system can also improve system accessibility by providing data entry options for blind, deaf, or physically handicapped users. Speech recognition system can be divided into several blocks: feature extraction, acoustic models database which is built based on the training data, dictionary, language model and the speech recognition algorithm. Analog speech signal must first be sampled at time and amplitude axes, or digitized. Samples of the speech signal are analyzed in even intervals. This period is usually 20 ms because the signal in this interval is considered stationary. Speech feature extraction involves the formation of equally spaced discrete vectors of speech characteristics. Feature vectors from training database are used to estimate the parameters of acoustic models. The acoustic model describes properties of the basic elements that can be recognized. The basic element can be a phoneme for continuous speech or word for isolated

words recognition.

TEXT_ TO_ SPEECH Converter:

Converting text to voice output using speech synthesis techniques. Although initially used by the blind to listen to written material, it is now used extensively to convey financial data, e-mail messages, and other information

via telephone for everyone. Text-to-speech is also used on handheld devices such as portable GPS units to announce street names when giving directions. Our Text-to-Speech Converter‖ accepts a string of 50 characters of text (alphabets and/or numbers) as input. In this, we have interfaced the keyboard with the controller and defined all the alphabets as well as digits keys on it. The speech processor has an unlimited dictionary and can speak out almost any text provided at the input most of the times. Hence, it has an accuracy of above 90%. It is a microcontroller based hardware coded in Embedded C language. Further research is to be done to optimize various methods of inputting the text i.e. Reading the text using optical sensor and converting it to speech so that almost all sorts of physical challenges faced by the people while communicating are overcome.

WORD RECOGNITION :Voice recognition software (also known as speech to text software)allows an individual to use their voice instead of typing on a keyboard. Voice recognition may be used to dictate text into the computer or to give commands to the computer. Voice recognition software allows for a quick method of writing onto a computer. It is also useful for people with disabilities who find it difficult to use the keyboard. This software can also assist those who have difficulty with transferring ideas onto paper as it helps take the focus out of the mechanics of writing. Word recognition is measured as a matter of speed, such that a word with a high level of recognition is read faster than a novel one. This manner of testing suggests that comprehension

of the meaning of the words being read is not required, but rather the ability to recognize them in a way that allows proper pronunciation. Therefore, context is unimportant, and word recognition is often assessed with words presented in isolation in formats such as flash cards Nevertheless, ease in word recognition, as in fluency, enables proficiency that fosters comprehension of the

text being read.

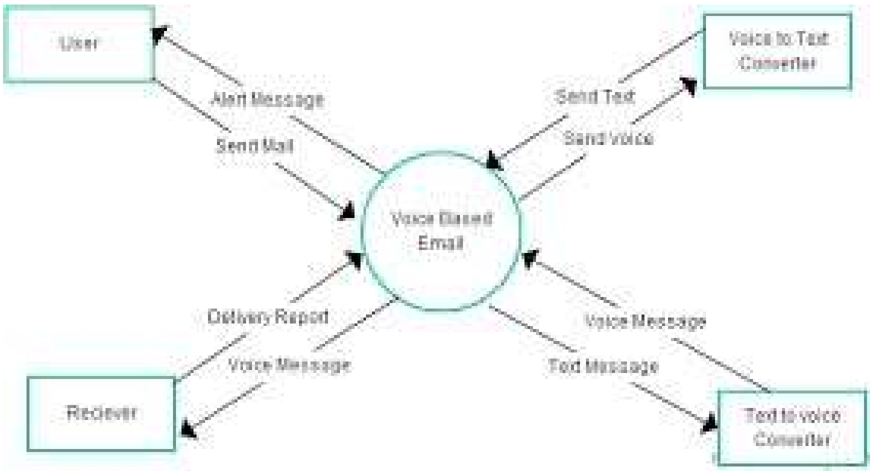


Figure-II 2.3: System Data Flow Diagram

2.4 "Voice based e-mail System for Blinds". In International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)

Internet plays a vital role in today's world of communication. Today the world is running on the basis of internet. No work can be done without use of internet. Electronic mail i.e. email is the most important part in day to day life. But some of the people in today's world don't know how to make use of internet, some are blind or some are illiterate. So it goes very difficult to them when to live in this world of internet. Nowadays there are various technologies available in this world like screen readers, ASR, TTS, STT, etc. but these are not that much efficient for them. Around 39 million people are blind and 246 people have low vision and also 82 of people living with blindness are 50 aged and above. We have to make some internet facilities to them so they can use internet. Therefore we came up with our project as voice based email system for blinds which will help a lot to visually impaired peoples and also illiterate peoples for sending their mails. The users of this system don’t need to remember any basic information about keyboard shortcuts as well as location of the keys. Simple mouse click operations are needed for functions making system easy to use for user of any age group. Our system provides location of where user is prompting through voice so that user doesn’t have to worry about remembering which mouse click operation

The visually challenged people find it very difficult to utilize this technology because of the fact that using them requires visual perception. However not all people can use the internet. This is because in order to access the internet you would need to know what is written on the screen. If that is not visible it is of no use. This makes internet a completely useless technology for the visually impaired and illiterate people.

In this system mainly three types of technologies are used namely:

STT (Speech-to-text):

here whatever we speak is converted to text. Their will a small icon ofmic on whose clicking the user had to speak and his/her speech will be converted to text format, which the naked people would see and read also.

TTS (text-to-speech) this, method is full opposite of STT. In this method, which converts the text format of the emails to synthesized speech? A text-to-speech (TTS) system converts language text into speech, alternative systems render symbolic linguistic representations. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database.

IVR (Interactive voice response): IVR is an advanced technology describes the interaction between the user and the system in the way of responding by using keyboard for the respective voice message. IVR allows user to interact with an email host system via a system keyboard, after that users can easily service their own enquiries by listening to the IVR dialogue. IVR systems generally respond with pre-recorded audio voice to further assist users on how to proceed.

The audio that would be pre-recorded and the system need to have large volumes.

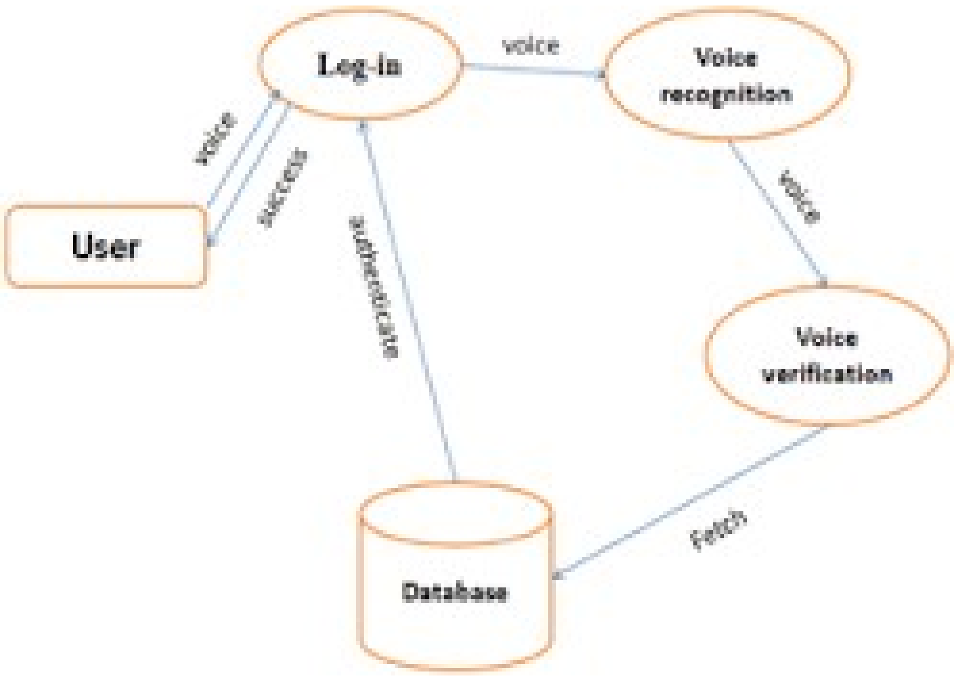


Figure-III 2.4: Proposed System Architecture

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Proposed model

The planned system is relies on a very fresh plan and obscurity just like the accessible mail systems. The foremost necessary facet that erstwhile unbroken in brain whereas developing the planned system's accessibility.

The present systems don't give this much convenience. So the systems present have a tendency to area unit developing is totally dissent from this system. In contrast to present system which emphasize more on user easiness of naive users, this system focus more on user easiness of all kind of folks including naive folks visually disabled people as well as uneducated people.

The entire structure is based on React speech API. When using this system the computer will prompt the client to perform precise operations to gain relevant services and if the client needs to way in the relevant services then they need to perform that particular operation. One of the most important recompense of this system is that user will not need to use the keyboard. All operations will be based on voice proceedings.

3.2 Existing System

There are a total number of 4.1 billion email accounts created until 2014 and an there will be estimated 5.2 billion accounts by end of 2018.[4] this makes emails the most used form of communication. The most common mail services that we use in our day to day life cannot be used by visually challenged people. This is because they do not provide any facility so that the person in front can hear out the content of the screen. As they cannot visualize what is already present on screen they cannot make out where to click in order to perform the required operations.[3] For a visually challenged person using a computer for the first time is not that convenient as it is for a normal user even though it is user friendly.

Although there are many screen readers available then also these people face some minor difficulties. Screen readers read out whatever content is there on the screen and to perform those actions the person will have to use keyboard shortcuts as mouse location cannot be traced by the screen readers. This means two things; one that the user cannot make use of mouse pointer as it is completely inconvenient if the pointer location cannot be traced and second that user should be well versed with the keyboard as to where each and every key is located. A user is new to computer can therefore not use this service as they are not aware of the key locations.

Another drawback that sets in is that screen readers read out the content in sequential manner and therefore user can make out the contents of the screen only if they are in basic HTML format. Thus the new advanced web pages which do not follow this paradigm in order to make the website more user-friendly only create extra hassles for these people. All these are some drawbacks of the current system which we will overcome in the system we are developing.

3.3 Design

A. Phase-1:

The tasks that can be performed using the program developed will be prompted using the voice prompt. In background Web speech API is used for text to speech conversion.

User will be asked to provide input for the following tasks written below.

The input is expected in the form of speech by the user which will be converted to text by the API in React and accordingly tasks will be performed.

- **Login to their Gmail account.**
- **Send e-mail through Gmail.**
- **Read e-mail through Gmail.**

B. Phase-2:

In phase-2 of our program the user will give speech input to the system. This speech input will be handled by window.SpeechRecognition.

module.It is a React which is used to handle the voice requests and it converts speech into text. Now after receiving input from the user speech to text converter will save the response in respective variables used in the script and based on their value it will further enter into respective modules.

C. Phase-3:

In this phase our program will handle the requests by the user. Based on the speech input given by the user it will launch the modules.

Login to G-mail account:- This module will handle the request by user to login in their g-mail account.This module will make the connection with the user's gmail account based on the credentials provided through voice input. This module's script designed as such it will prompt user to enter their g- mail username and password and then it will use selenium web-driver to automate the task for the user and as a result connection will be made.

- **Send E-mail through G-mail:-**

This module will handle the request by user to send email through their g-mail account. The React and node script for this module will prompt the user to enter their credentials and then it will make connection with their account. After the connection has been done it will further prompt the user to enter the receiver's account e- mail id and it will then allow the user to speak their message and it will repeat it for them and by saying ok it will send the mail.

- **Read E-mail through G-mail:-**

This module will handle the request by user to read email through their g-mail account. The script for this module will prompt the user to enter their credentials and then it will make connection with their account. After the connection has been done it will start fetching the unread mails for the user and will speak it for them.

3.3.1 SMTP
(Simple Mail Transfer Protocol)

Email is rising because the one among the foremost valuable service in net nowadays. Most of the web systems use SMTP as a technique to transmit mail from one client to different.

SMTP may be a thrust set of rules and is employed to send the mail whereas POP (post workplace protocol) or IMAP (internet message access protocol) square measure accustomed retrieve those mails at the receiver's aspect.

SMTP is Associated with the application layer protocol of OSI model of network. The user who desires to launch the mail open a TCP (Transmission Control Protocol) connection to the SMTP server and then sends the mail to the other connection. The SMTP server is mostly on listening mode. No sooner the server listens for a TCP connection from any user, the SMTP procedure initiate a connection usually on port number 25.

When the successful establishment of TCP connection has been done, the client can send the mail.

The two processes that is sender process and the receiver process carry out a simple request response dialogue, outlined by the SMTP protocol within which the client process transmits the mail address of the mastermind and the recipient for a message. Once the server method accept these mail addresses, the consumer method broadcast the e-mail instant message. The message should include a message header and message text ("body") formatted in accord with RFC 822.

The following example illustrates a message in the RFC 822 message format:

```
From:  aviralupadhyay98@example.com
To: hello.01aviral@example.com
Subject: An RFC 822 formatted message

This is a simple text body of the message.
The blank line separates the header and body of the message.
```

The **SMTP** model is of two types :-

1.End-to- end method

2.Store-and- forward method

The SMTP model chains both end-to-end no intermediate message transfer agents and store- and-forward mail delivery methods. The end-to-end method of SMTP is used between organization, and the store-and forward method is chosen for sending mails within organizations which have TCP/IP and SMTP-based networks

End-To-End

In this method , a SMTP client will speak to the destination host's SMTP server directly to transport the mail. It will keep the mail item from being transmitted until it has been successfully copied to the recipient's SMTP.

Store-and-Forward

In this method a mail can be sent through a number of intermediary hosts, before reaching to the final destination.

A successful transmission from a hosts signify only that the mails has been sent to the next host, and then the mail will be sent to next host.

3.3.2 SENDING EMAIL IN REACT USING GMAIL-SEND

Create a Backend Server:

Set up a backend server using a server-side technology such as Node.js with Express.

Use 'gmail-send' in the Backend:

Install and use 'gmail-send' in your backend code to send emails. Configure it with your Gmail credentials.

Example:

```
    const express = require('express’);
    const gmailSend = require('gmail-send’);
    const bodyParser = require('body-parser’);
```

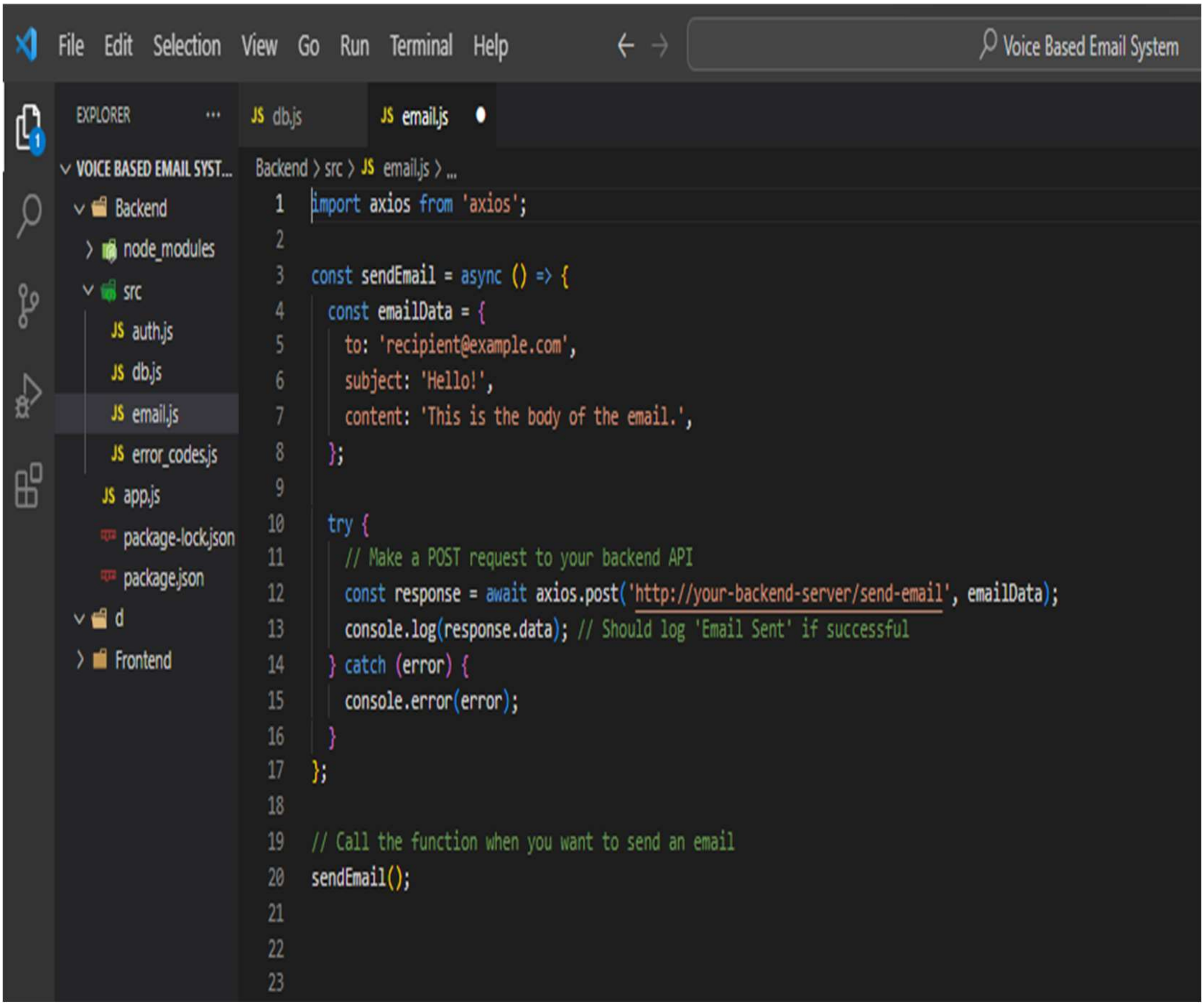
Use **'your-email@gmail.com'** and **'your-password'** with your Gmail credentials. Here we note that storing credentials directly in the code is not recommended for production.So we used environment variables or a more secure configuration method.

Create an API Endpoint:

Expose an API endpoint on your server (e.g., /send-email) that the React application can call to trigger the email sending process.

Make an API Request from React:

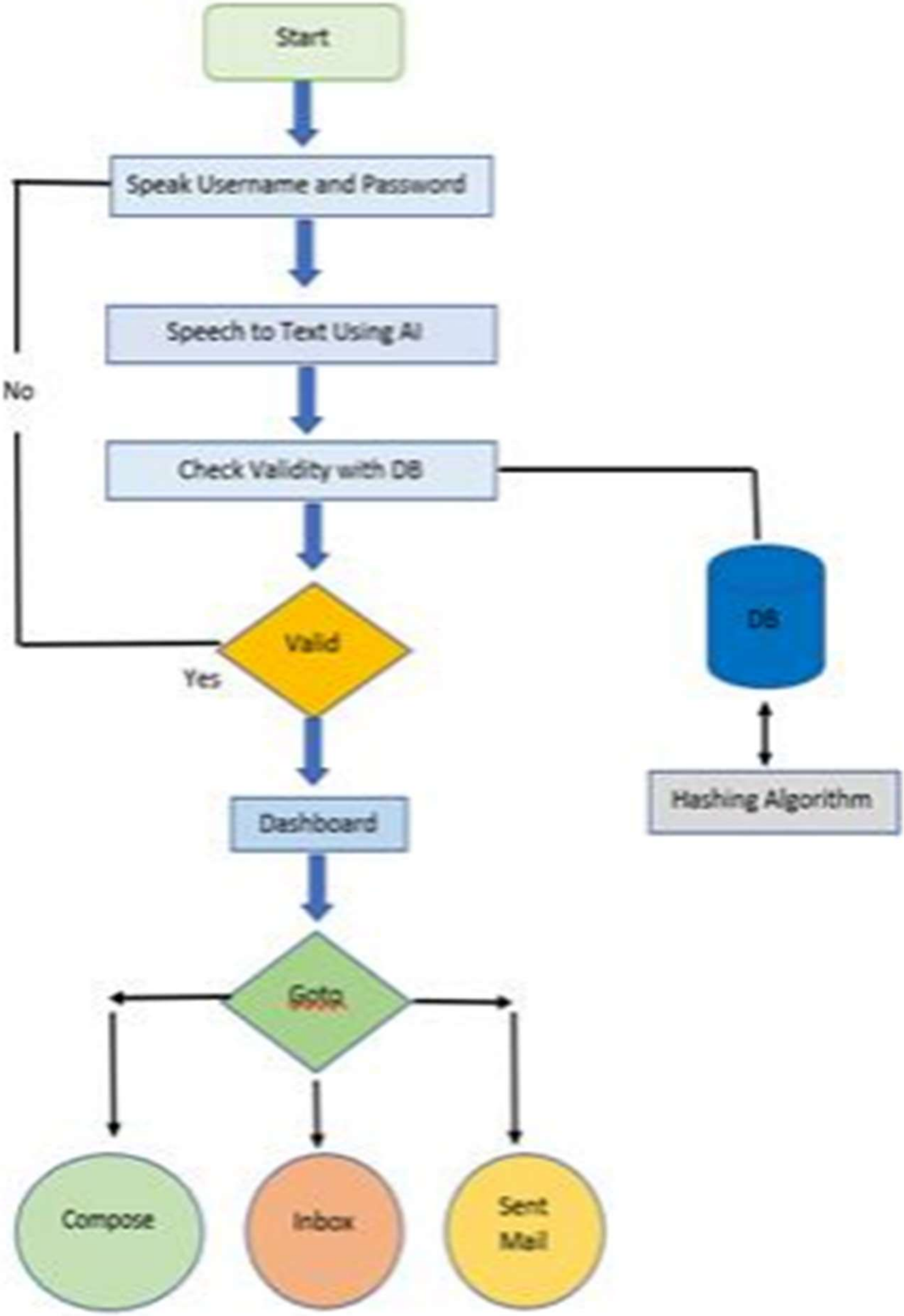
In your React application, use a library like axios or the built-in fetch to make an HTTP request to the API endpoint when you want to send an email.



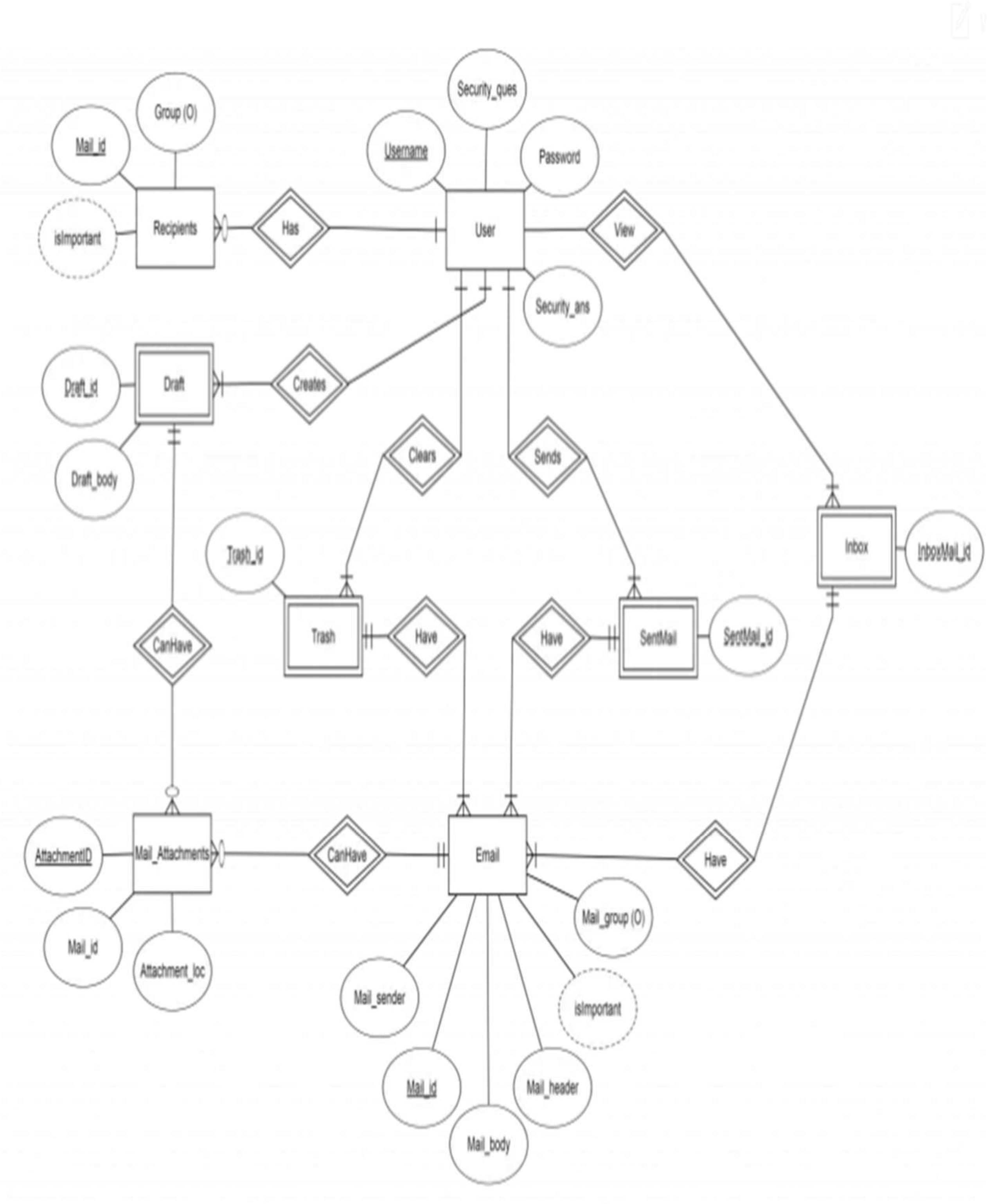
```javascript
import axios from 'axios';

const sendEmail = async () => {
  const emailData = {
    to: 'recipient@example.com',
    subject: 'Hello!',
    content: 'This is the body of the email.',
  };

  try {
    // Make a POST request to your backend API
    const response = await axios.post('http://your-backend-server/send-email', emailData);
    console.log(response.data); // Should log 'Email Sent' if successful
  } catch (error) {
    console.error(error);
  }
};

// Call the function when you want to send an email
sendEmail();
```

Figure-I 3.3.2: Email Send

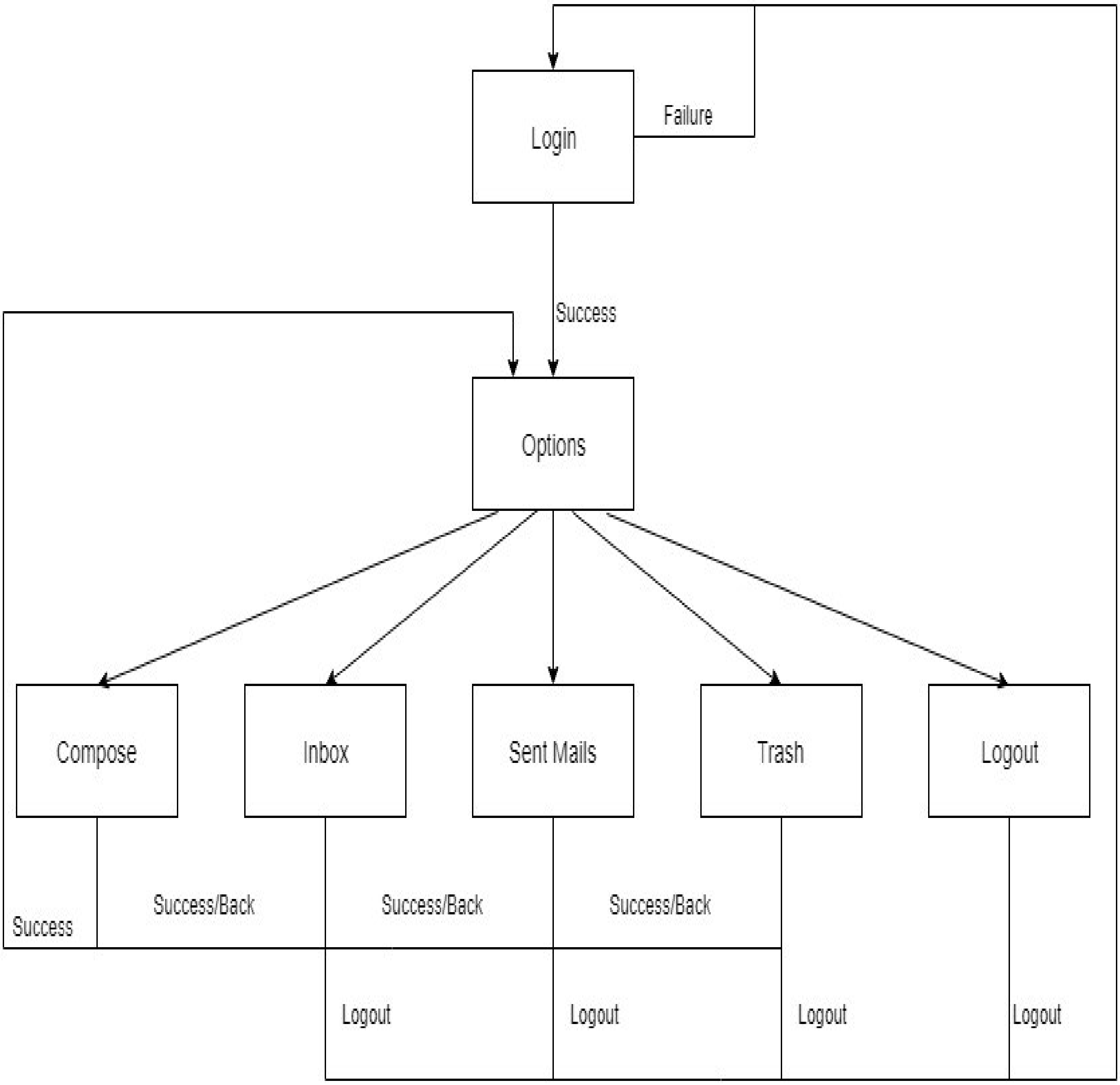
Chapter 4



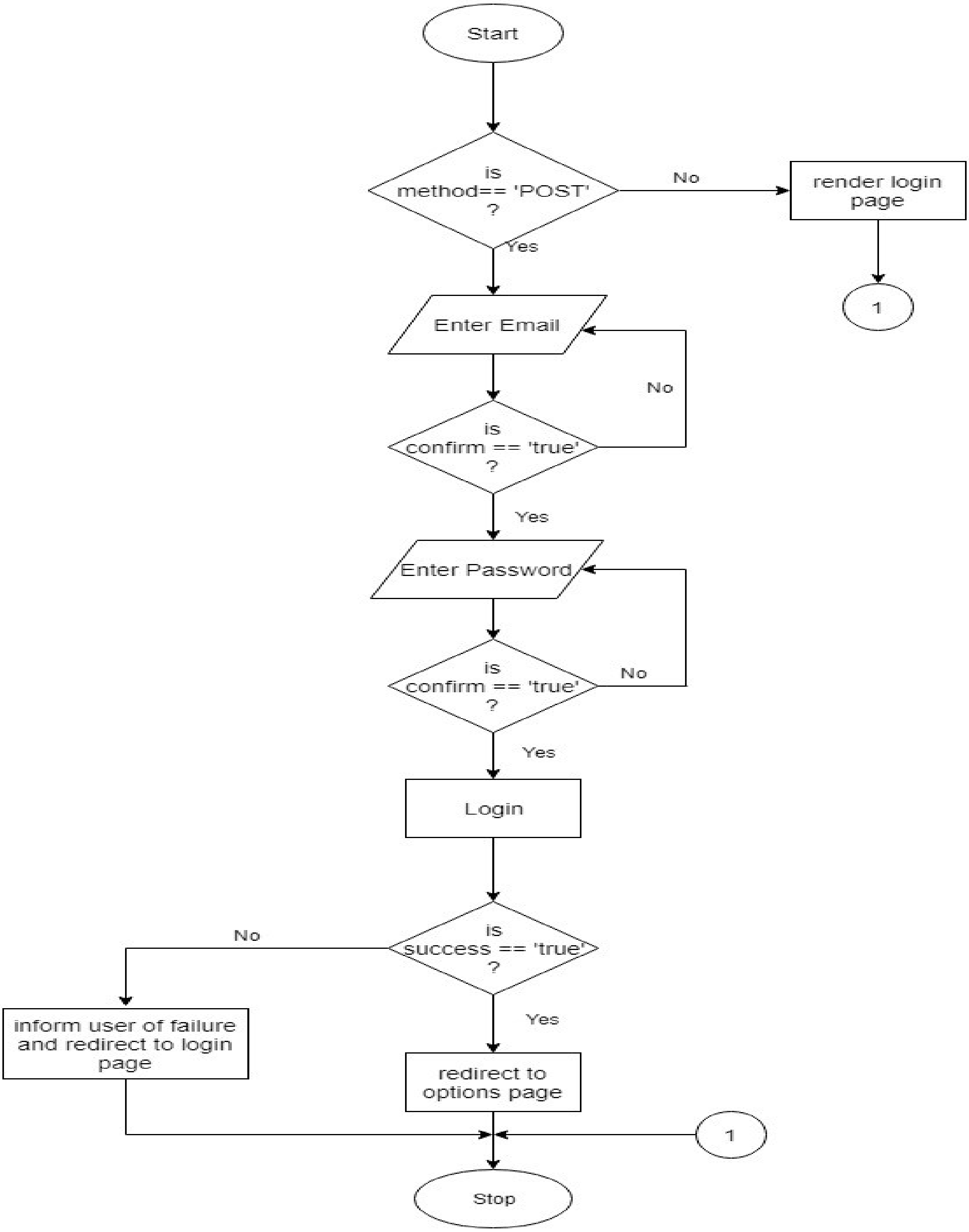
4.1 FLOW CHART OF PROPOSED MODEL



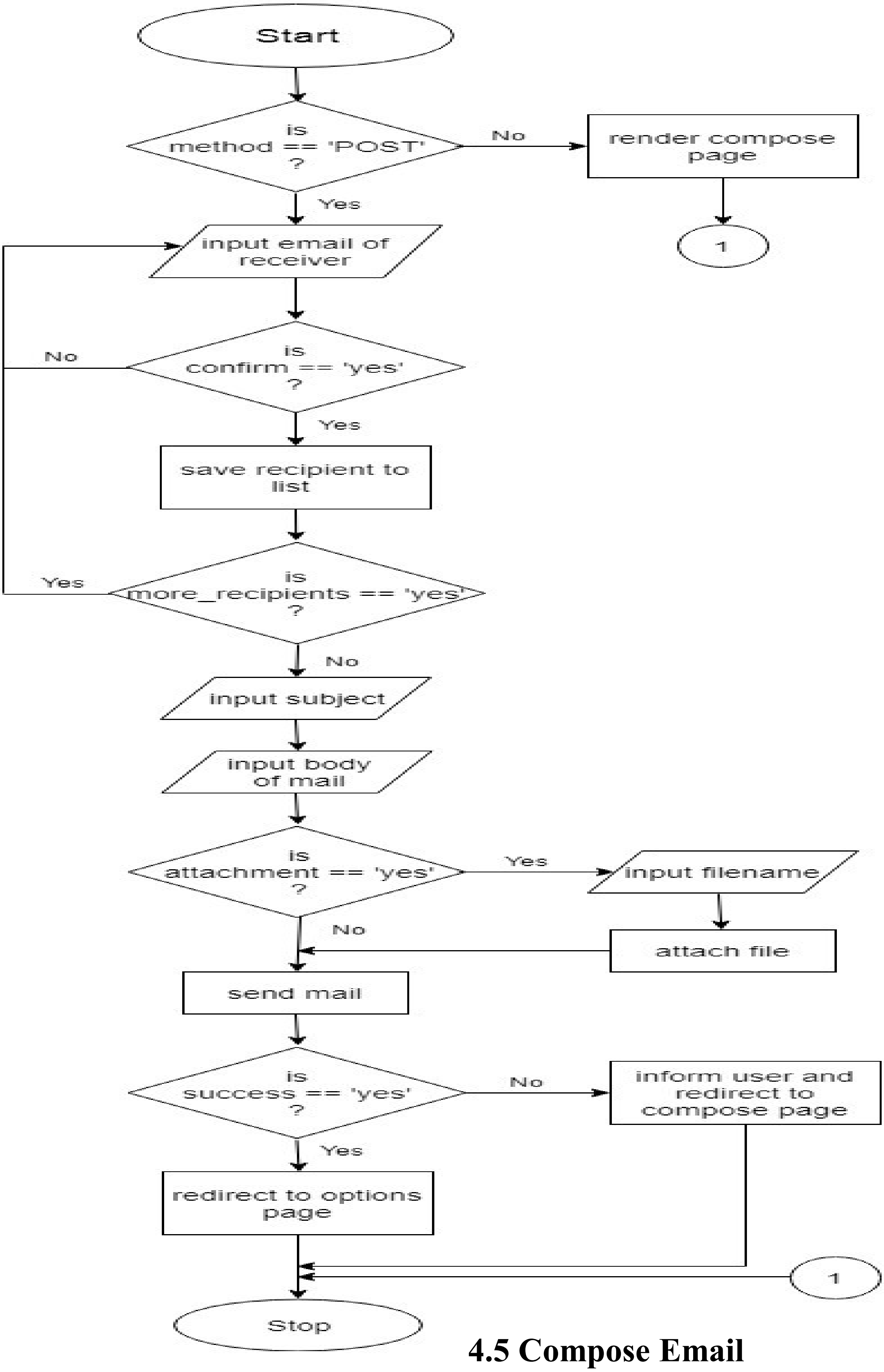
4.2 ER diagram of system



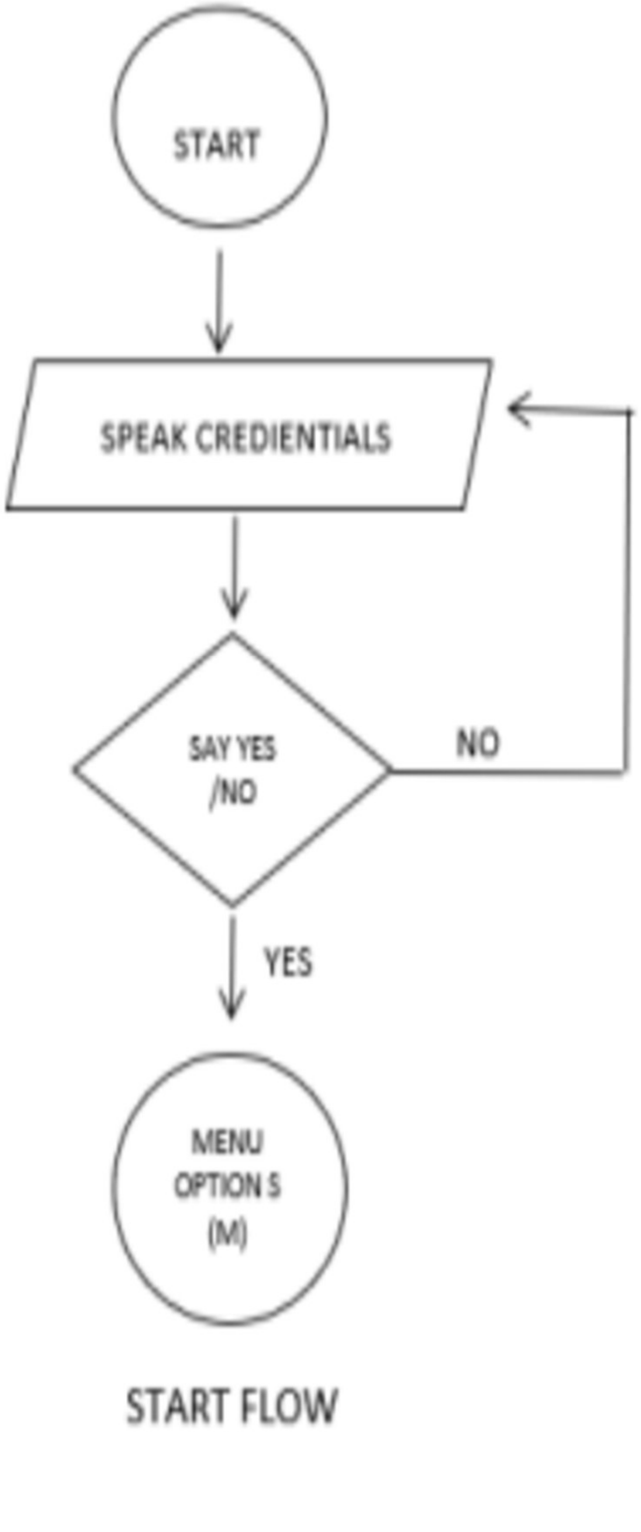
4.3 Block Diagram:



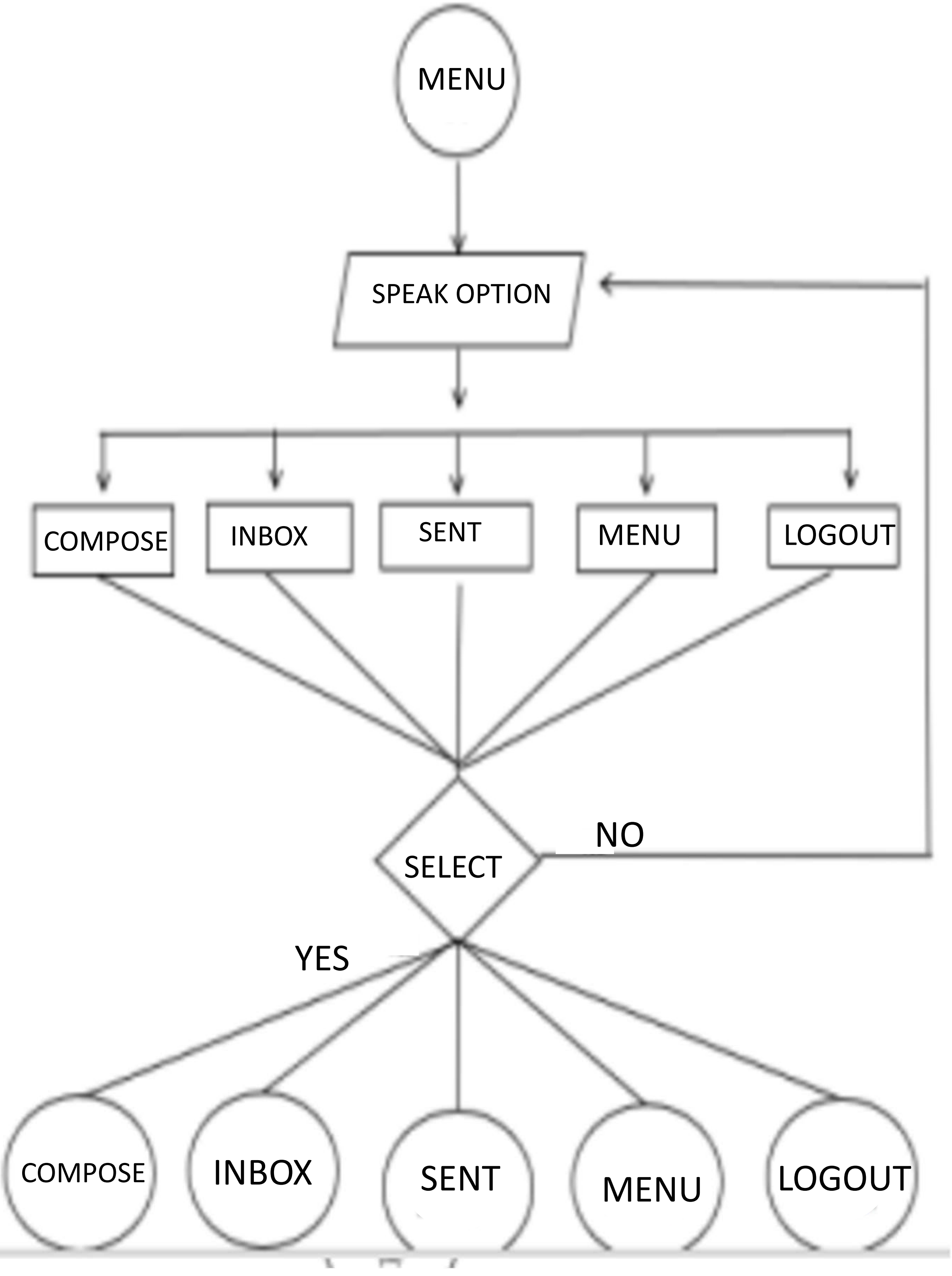
4.4 Login Diagram



4.5 Compose Email



4.6 Start Flow



4.7 MENU OPTIONS

CHAPTER 5: PERFORMANCE ANALYSIS

5.1 Software Requirements

Tools Used:

- React js.
- Node js/Express js.
- Postgres SQL.
- Google Speech-to-text and text-to-speech Converters of React.
- Web speech api in React.

5.2 Hardware /Functional Requirements:

- Windows Desktop

5.2:Non functional requirements

Performance:

Response Time: In addition to providing timely responses to user commands, minimizing delays in reading or composing emails, the system should also optimize its response time to accommodate the unique needs of visually impaired users. This includes ensuring that voice commands are recognized and executed promptly, without undue latency, to enhance the overall user experience.

Scalability:

As the system scales to accommodate an increasing number of users and emails, it must do so while maintaining consistent performance levels. Scalability testing should assess the system's ability to efficiently allocate resources and handle concurrent user interactions, ensuring that performance remains stable even under high load conditions.

Reliability and Availability:

Building on the foundation of reliability, the system should demonstrate high availability, allowing users uninterrupted access to their emails whenever they need them. This entails implementing robust failover mechanisms and redundant infrastructure to minimize downtime and service interruptions, thereby enhancing user trust and satisfaction.

Accessibility:

Adhering to accessibility standards such as WCAG is paramount to ensuring that the system is usable by individuals with varying degrees of visual impairment. Beyond mere compliance, the system should prioritize inclusive design principles, incorporating features such as alternative text for images, keyboard navigation support, and adjustable font sizes to accommodate diverse user needs effectively.

Security

Data Security:

Robust data encryption mechanisms should safeguard sensitive user information and emails against unauthorized access or interception. By implementing industry-standard encryption protocols, the system can mitigate the risk of data breaches and uphold user privacy.

Authentication:

Secure authentication methods, such as multi-factor authentication, should be employed to verify user identities and prevent unauthorized access to user accounts. By bolstering authentication mechanisms, the system can fortify its defenses against unauthorized access attempts.

Audit Trails:

Maintaining comprehensive logs of user interactions and system activities enables robust security auditing, facilitating the detection and investigation of potential security breaches or unauthorized activities. By retaining detailed audit trails, the system enhances accountability and transparency, fostering user trust and confidence.

Scalability:

Designing the system with scalability in mind involves more than just accommodating a growing user base and increasing email volumes. It requires adopting flexible architectural patterns and scalable infrastructure solutions that can adapt to evolving demands seamlessly. By prioritizing scalability from the outset, the system can future-proof itself against scalability challenges, ensuring sustained performance and reliability as it grows.

Usability:

Usability testing with visually impaired users is crucial to evaluating the system's ease of use and effectiveness in meeting their needs. By soliciting feedback from end-users and incorporating their insights into the design process, the system can iteratively improve its usability and accessibility, ultimately delivering a more intuitive and user-friendly experience for all users.

Chapter 6 : Testing

6.1 Introduction

The purpose of testing is not just to identify errors but to thoroughly evaluate the robustness and reliability of a software system. Testing encompasses a systematic process of scrutinizing every aspect of a work product, aiming to uncover any potential faults or weaknesses that may compromise its functionality or performance.

At its core, testing serves as a critical quality assurance mechanism, providing assurance that the software system meets its specified requirements and aligns with user expectations. By subjecting the software to rigorous examination, testers aim to ensure that it operates as intended and does not exhibit any undesirable behavior or failures under normal usage conditions.

Testing involves exercising the software across various components, sub-assemblies, and the final product, systematically evaluating its behavior and performance at each level of granularity. This process allows for the detection of defects or inconsistencies that may arise due to errors in design, implementation, or integration.

Furthermore, testing encompasses a diverse range of methodologies and techniques, each tailored to address specific testing requirements and objectives. From unit tests that focus on individual software components to system tests that evaluate the software as a whole, each test type plays a crucial role in validating different aspects of the software's functionality, performance, and reliability.

In essence, testing serves as a fundamental pillar of software development, providing a structured approach to ensuring the quality and integrity of software systems. By identifying and addressing issues early in the development lifecycle, testing helps mitigate risks, enhance user satisfaction, and ultimately contribute to the successful delivery of high-quality software products.

6.2 Types of Testing

6.2.1 Unit Testing

Unit testing represents a critical phase in the software development lifecycle, honing in on the smallest functional units of software design, typically modules or individual functions. Our approach to unit testing is characterized by a white box orientation, wherein the internal structure and logic of the software are thoroughly examined to ensure robustness and correctness.

Moreover, in certain modules, our testing methodology adopts a parallel execution strategy, wherein multiple steps or tests are conducted simultaneously. This parallelization of testing procedures enhances efficiency and expedites the overall verification process, enabling us to identify and address potential issues swiftly.

By combining white box analysis with parallel testing methodologies, we strive to achieve comprehensive test coverage while optimizing resource utilization and minimizing time-to-market. This rigorous approach not only enhances the reliability and quality of our software but also underscores our commitment to delivering exceptional user experiences.

Unit testing for a voice-based email system tailored for the visually impaired is essential for ensuring its reliability, accessibility, and usability. Given the unique nature of this system, unit testing focuses on verifying the functionality of individual components or modules that contribute to its overall operation. Here's how unit testing might be approached for such a system:

1. **Voice Input Module Testing**: This module handles voice commands from users. Unit tests would verify that the system accurately interprets spoken commands and translates them into actionable tasks within the email system. Tests would cover various scenarios, including different accents, speech patterns, and command structures.

2. **Email Access Module Testing**: This module is responsible for retrieving, composing, and managing emails. Unit tests would validate functionalities such as fetching emails from servers, parsing email content, and composing new messages through voice commands. Tests would ensure that emails are read aloud clearly and that users can navigate through their inbox effectively.

3. **Navigation and Interaction Module Testing**: This module enables users to navigate through the email interface using voice commands. Unit tests would verify that users can efficiently move between different sections of the email interface, such as inbox, drafts, sent items, etc. Tests would also cover interactions with email elements like subject lines, sender information, and message bodies.

4. **Accessibility Module Testing**: This module ensures that the email system complies with accessibility standards and provides a seamless experience for visually impaired users. Unit tests would assess factors such as screen reader compatibility, contrast ratios for text-to-speech conversion, and adherence to voice interface guidelines.

5. **Error Handling Module Testing**: This module handles errors and exceptions that may occur during voice interactions or email operations. Unit tests would validate the system's ability to detect and gracefully recover from errors, such as network connectivity issues, invalid commands, or unexpected responses from email servers.

6. **Integration Testing with External Services**: Unit tests may also include integration testing with external services, such as email servers and speech recognition APIs, to ensure seamless communication and interoperability.

7. **Performance and Scalability Testing**: While primarily focused on unit-level functionality, unit tests may also include performance and scalability checks to ensure that the system can handle a large volume of voice commands and email operations efficiently.

By conducting thorough unit testing across these modules, developers can identify and address issues early in the development process, ultimately delivering a robust and user-friendly voice-based email system for the visually impaired.

6.2.2 Integration Testing

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface.

Integration testing for a voice-based email system designed for the visually impaired involves validating the interaction and compatibility between various system components and external services. Here's how integration testing might be conducted for such a system:

1. **Voice Recognition Integration Testing:** Ensure that the voice recognition software seamlessly integrates with the email system, accurately transcribing spoken commands into actionable tasks within the email interface. Test various scenarios to verify the system's ability to interpret different accents, speech patterns, and command structures effectively.

2. **Email Server Integration Testing:** Validate the system's ability to interact with email servers for tasks such as retrieving, sending, and managing emails. Test integration with different types of email servers (e.g., IMAP, POP3) to ensure compatibility and reliability.

3. **Accessibility Tool Integration Testing:** Verify that accessibility tools, such as screen readers, are compatible with the voice-based email system. Test the system's ability to provide clear and understandable audio feedback to visually impaired users through screen reader integration.

4. **User Interface Integration Testing**: Ensure seamless integration between the voice-based interface and other user interface elements, such as buttons, menus, and navigation controls. Test the consistency and coherence of the user experience across different interaction modalities, including voice commands and traditional input methods.

5. **Email Content Parsing and Rendering Testing**: Validate the system's ability to parse and render email content correctly, including text, attachments, and multimedia elements. Test integration with third-party libraries or services for tasks such as text-to-speech conversion and multimedia playback.

6. **Error Handling and Recovery Testing**: Verify that the system can detect and handle errors gracefully during voice interactions and email operations. Test integration with error logging and reporting mechanisms to ensure timely identification and resolution of issues.

7. **Security Integration Testing**: Validate the integration of security features, such as user authentication and data encryption, to protect sensitive information within the email system. Test integration with security protocols and frameworks to ensure compliance with industry standards and best practices.

8. **Compatibility Testing with Assistive Technologies**: Ensure compatibility with assistive technologies commonly used by visually impaired individuals, such as braille displays and voice assistants. Test integration with these technologies to ensure a seamless and accessible user experience.

By conducting comprehensive integration testing across these areas, developers can ensure that the voice-based email system for the visually impaired functions smoothly, reliably, and inclusively, meeting the needs and expectations of its users.

6.2.3 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

System testing for a voice-based email system tailored for the visually impaired involves evaluating the system as a whole to ensure that it meets its functional and non-functional requirements. Here's how system testing might be conducted for such a system:

1. **Functional Testing:** Verify that the voice-based email system performs its intended functions correctly, including tasks such as:

- Composing, sending, receiving, and managing emails through voice commands.
- Navigating through different sections of the email interface, such as inbox, drafts, sent items, etc., using voice commands.
- Reading email content aloud in a clear and understandable manner.
- Interacting with email elements, such as subject lines, sender information, and message bodies, through voice commands.

2. **Usability Testing:** Evaluate the user-friendliness and accessibility of the voice-based email system for visually impaired users, including:

- Ease of learning and using the voice commands and navigation controls.
- Clarity and comprehensibility of the audio feedback provided by the system.
- Consistency and intuitiveness of the user interface design.
- Accessibility features, such as screen reader compatibility and voice assistance options.
- Efficiency of performing common email tasks through voice commands compared to traditional input methods.

3. **Performance Testing**: Assess the performance of the voice-based email system under various load conditions, including:

- Response time for processing voice commands and executing email operations.
- Scalability of the system to handle multiple concurrent users and high volumes of email traffic.
- Resource utilization, such as CPU, memory, and network bandwidth, under normal and peak load conditions.
- Stability and reliability of the system under prolonged usage and stress testing scenarios.

4. **Compatibility Testing**: Ensure that the voice-based email system is compatible with a wide range of devices, operating systems, browsers, and assistive technologies commonly used by visually impaired individuals. Test compatibility with different hardware configurations and software environments to ensure a consistent user experience across platforms.

5. **Security Testing**: Validate the security features and protocols implemented in the voice-based email system to protect user data and privacy. Test for vulnerabilities such as unauthorized access, data breaches, and malware attacks, and ensure compliance with relevant security standards and regulations.

By conducting thorough system testing across these areas, developers can ensure that the voice-based email system for the visually impaired meets the highest standards of functionality, usability, performance, compatibility, security, and accessibility, delivering a seamless and empowering user experience.

CHAPTER 7: CONCLUSION

7.1 Conclusions

The project we've envisioned is a comprehensive system designed to empower visually impaired individuals in efficiently accessing email services. By addressing inherent challenges faced by the blind community in email usage, our system represents a significant advancement in accessibility technology. One of the key innovations lies in the elimination of reliance on keyboard shortcuts alongside screen readers, thereby alleviating the cognitive burden associated with memorizing complex commands.

Moreover, our solution transcends the limitations of traditional keyboard-based input methods, ensuring that even users unfamiliar with keyboard layouts can effortlessly navigate and interact with the email interface. Voice input functionality further enhances accessibility, allowing users to seamlessly provide information through spoken commands when required.

Notably, India hosts a substantial portion of the global blind population, accounting for approximately 70%. Recognizing this demographic reality, our project aims to revolutionize email accessibility for visually impaired individuals within the Indian context. The architecture we've developed facilitates streamlined access to email and multimedia functions within the operating system, simplifying navigation and interaction for users with visual impairments.

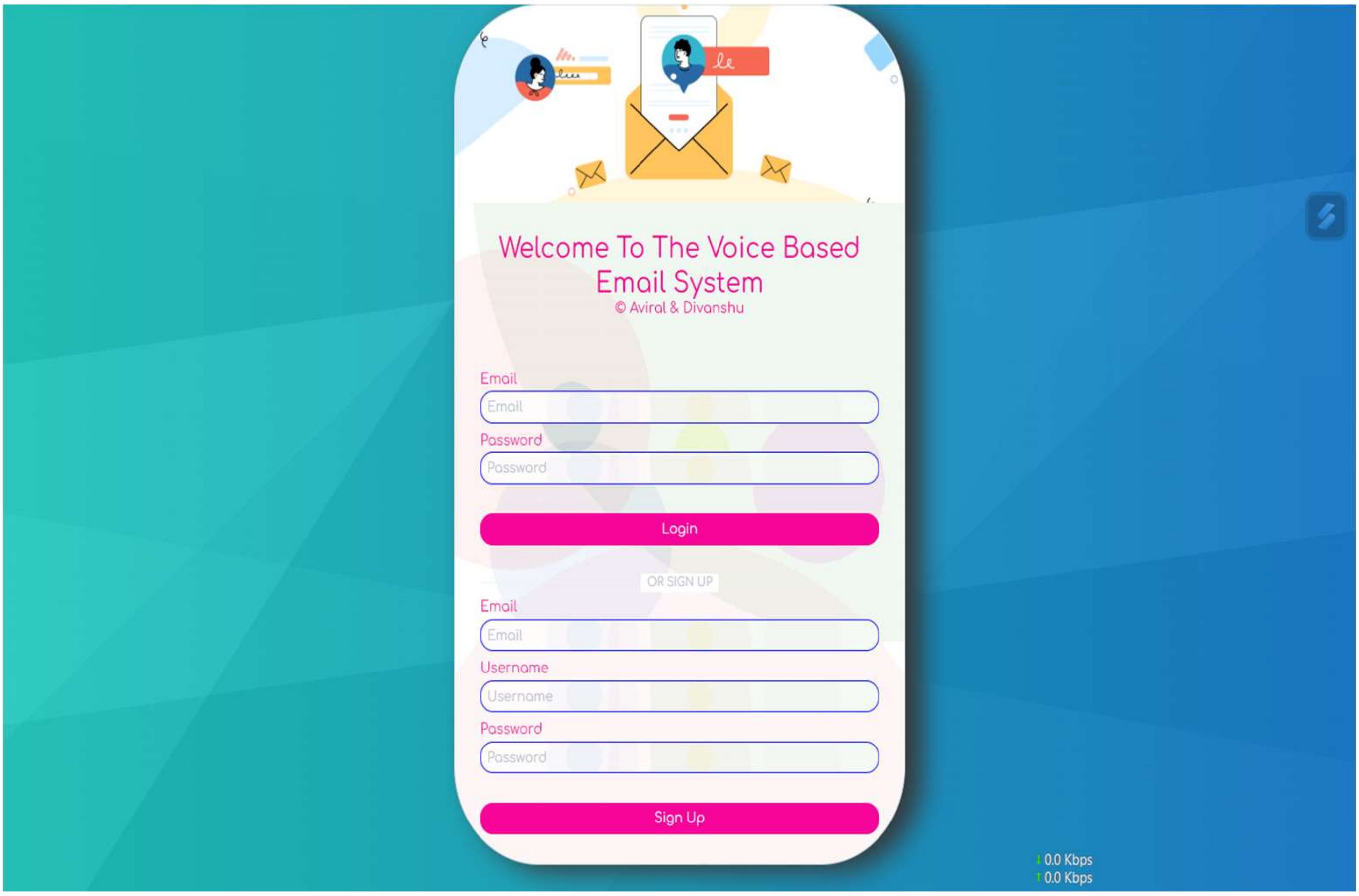
By reducing the cognitive load associated with keyboard input, our architecture not only enhances efficiency but also promotes inclusivity by catering to the needs of handicapped and illiterate individuals. Through these advancements, we strive to foster a more equitable and accessible digital landscape, where all members of society can participate and thrive.

Furthermore, our project emphasizes the importance of user-centered design, prioritizing the needs and preferences of visually impaired individuals throughout the development process. By incorporating feedback from end-users and accessibility experts, we've ensured that our solution addresses real-world challenges effectively and intuitively.

In addition to facilitating email access, our architecture lays the foundation for broader digital inclusion initiatives, extending its benefits beyond email usage. With its user-friendly interface and intuitive voice commands, our system empowers individuals with diverse abilities to navigate the digital realm with confidence and independence.

Looking ahead, we envision our project as a catalyst for positive societal change, inspiring further innovation and investment in accessibility technology. By championing the rights and capabilities of visually impaired individuals, we strive to create a more inclusive and equitable world where every individual can participate fully in the digital age.

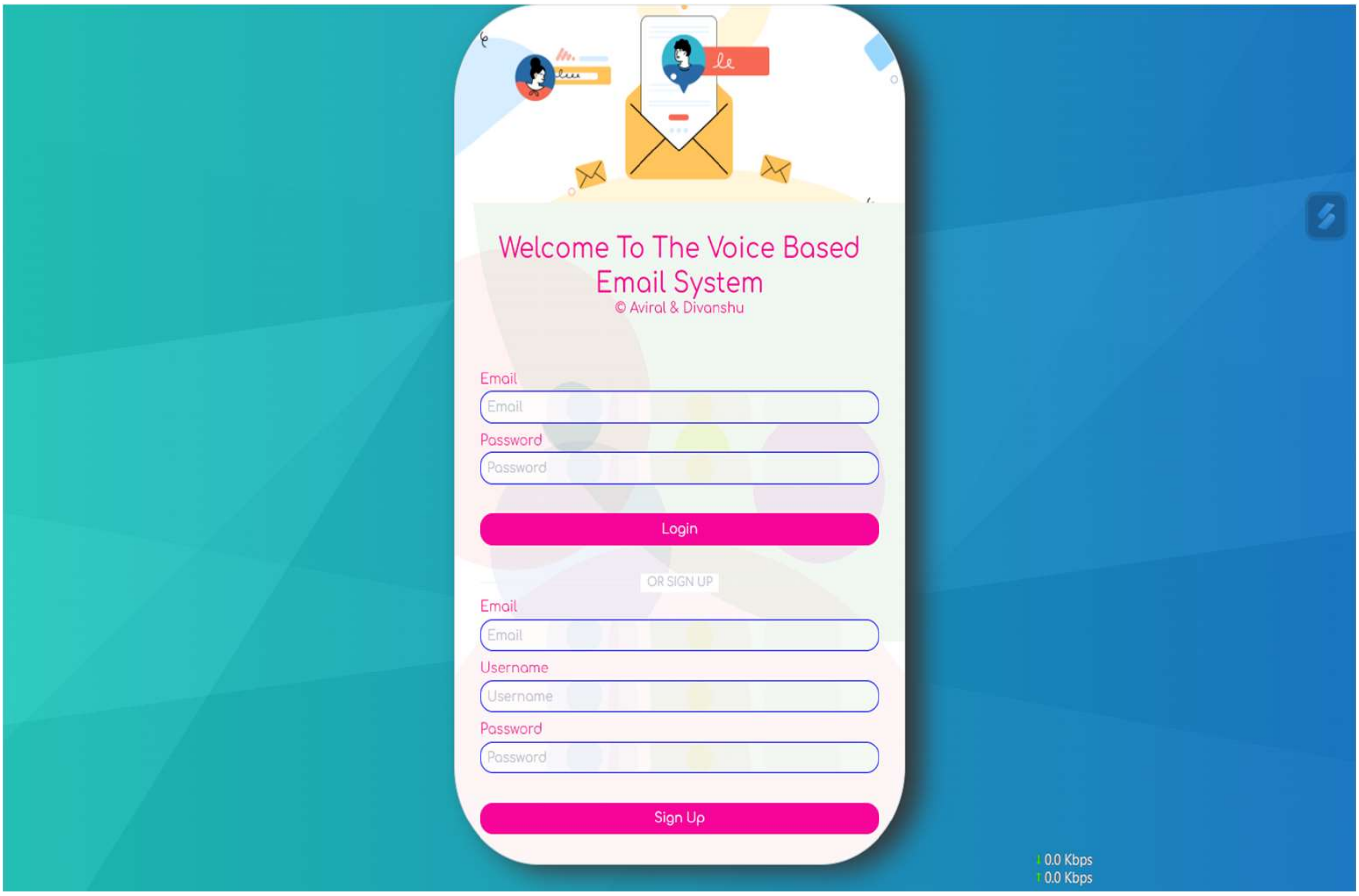
7.2 Project Output



Welcome To The Voice Based Email System

© Aviral & Divanshu

Figure-I 7.2 : WELCOME PAGE

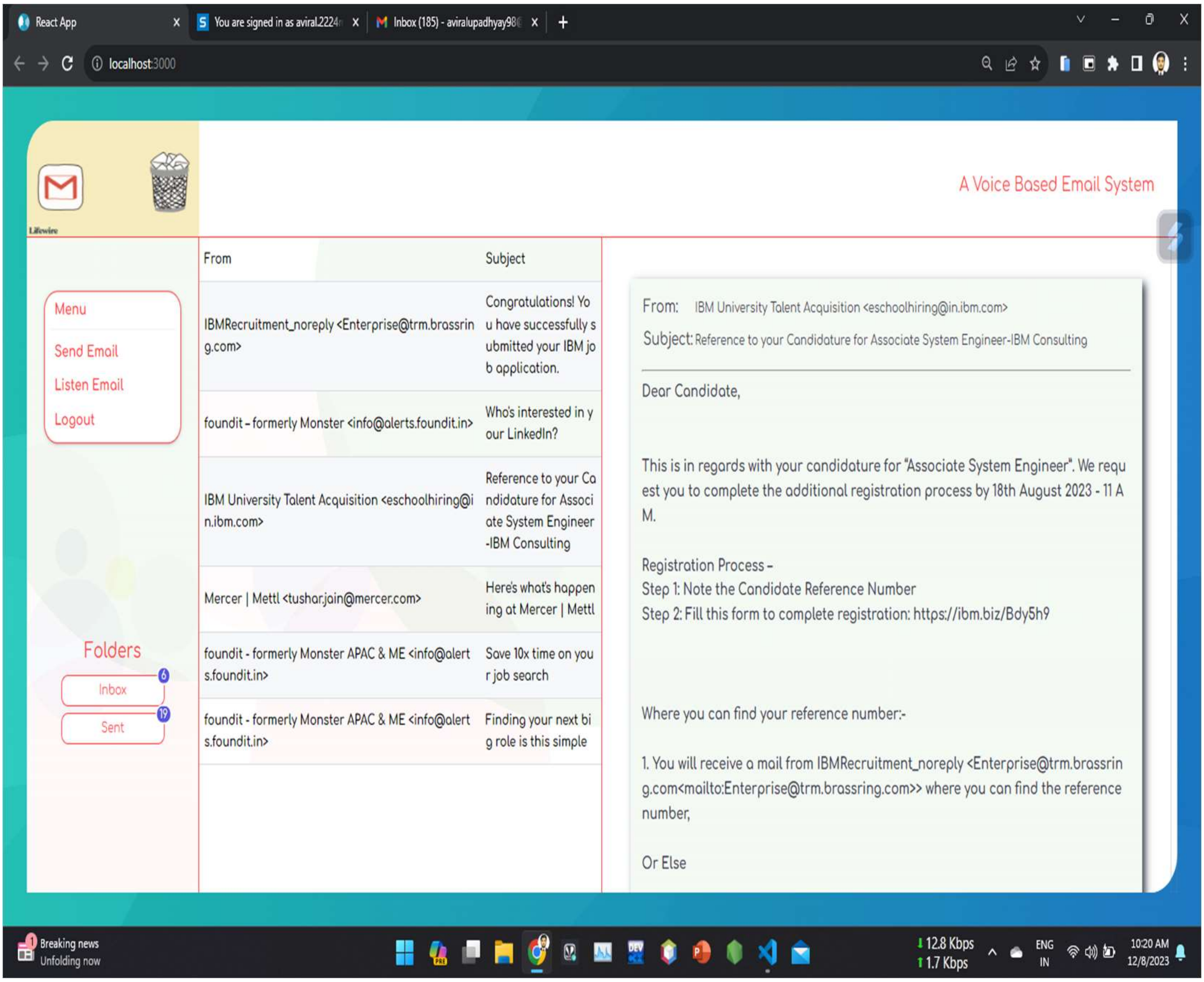


Welcome To The Voice Based Email System
© Aviral & Divanshu

Email
Email
Password
Password

Email
Email
Username
Username
Password
Password

Figure-II 7.2 LOG IN / SIGN UP



From	Subject
IBMRecruitment_noreply <Enterprise@trm.brassring.com>	Congratulations! You have successfully submitted your IBM job application.
foundit - formerly Monster <info@alerts.foundit.in>	Who's interested in your LinkedIn?
IBM University Talent Acquisition <eschoolhiring@in.ibm.com>	Reference to your Candidature for Associate System Engineer -IBM Consulting
Mercer | Mettl <tushar.jain@mercer.com>	Here's what's happening at Mercer | Mettl
foundit - formerly Monster APAC & ME <info@alerts.foundit.in>	Save 10x time on your job search
foundit - formerly Monster APAC & ME <info@alerts.foundit.in>	Finding your next big role is this simple

From: IBM University Talent Acquisition <eschoolhiring@in.ibm.com>

Subject: Reference to your Candidature for Associate System Engineer-IBM Consulting

Dear Candidate,

This is in regards with your candidature for "Associate System Engineer". We request you to complete the additional registration process by 18th August 2023 - 11 A M.

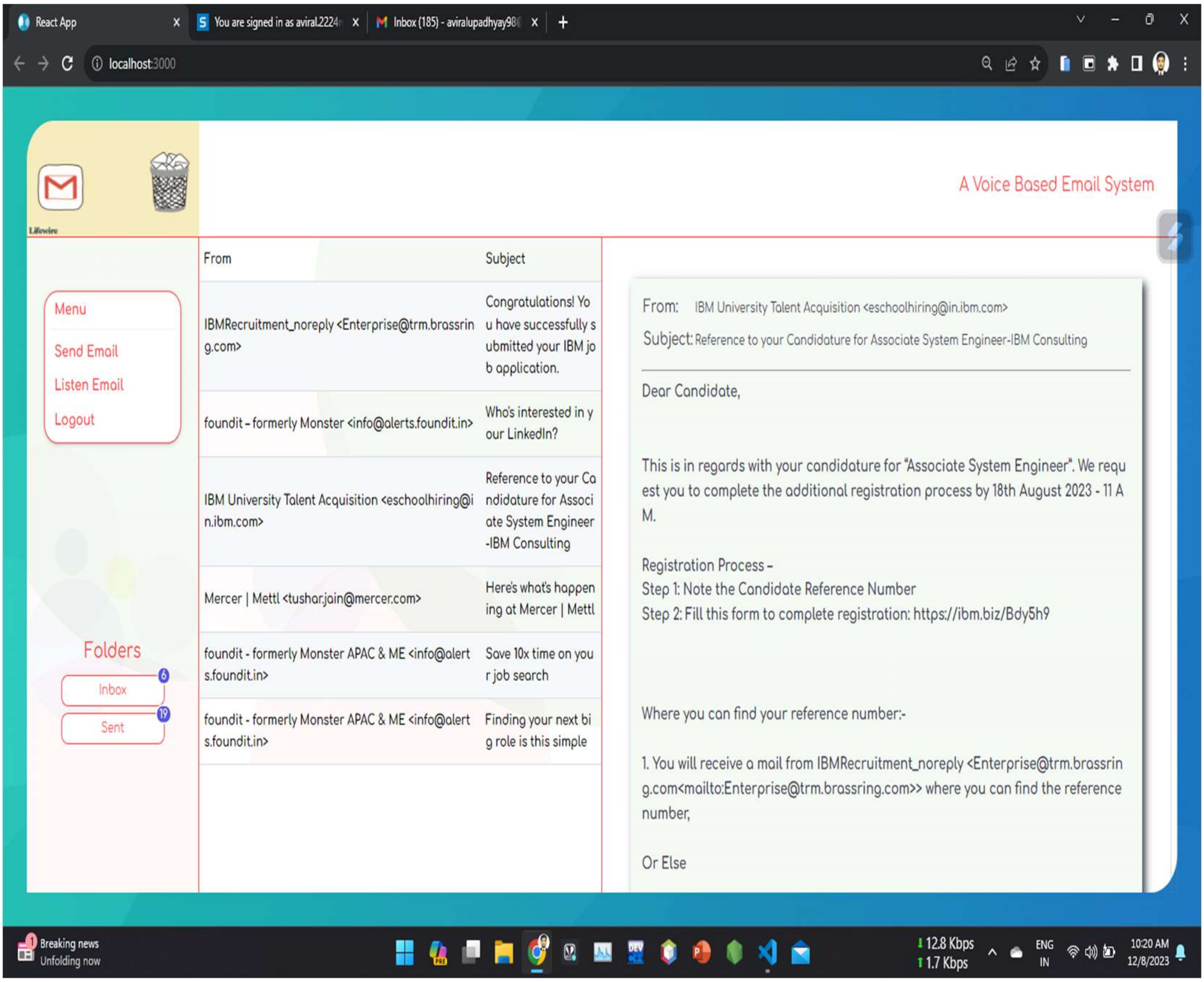
Registration Process -
Step 1: Note the Candidate Reference Number
Step 2: Fill this form to complete registration: https://ibm.biz/Bdy5h9

Where you can find your reference number:-

1. You will receive a mail from IBMRecruitment_noreply <Enterprise@trm.brassring.com<mailto:Enterprise@trm.brassring.com>> where you can find the reference number,

Or Else

Figure-III 7.2: SENT



From	Subject
IBMRecruitment_noreply <Enterprise@trm.brassring.com>	Congratulations! You have successfully submitted your IBM job application.
foundit - formerly Monster <info@alerts.foundit.in>	Who's interested in your LinkedIn?
IBM University Talent Acquisition <eschoolhiring@in.ibm.com>	Reference to your Candidature for Associate System Engineer -IBM Consulting
Mercer | Mettl <tushar.jain@mercer.com>	Here's what's happening at Mercer | Mettl
foundit - formerly Monster APAC & ME <info@alerts.foundit.in>	Save 10x time on your job search
foundit - formerly Monster APAC & ME <info@alerts.foundit.in>	Finding your next big role is this simple

From: IBM University Talent Acquisition <eschoolhiring@in.ibm.com>

Subject: Reference to your Candidature for Associate System Engineer-IBM Consulting

Dear Candidate,

This is in regards with your candidature for "Associate System Engineer". We request you to complete the additional registration process by 18th August 2023 - 11 A M.

Registration Process -
Step 1: Note the Candidate Reference Number
Step 2: Fill this form to complete registration: https://ibm.biz/Bdy5h9

Where you can find your reference number:-

1. You will receive a mail from IBMRecruitment_noreply <Enterprise@trm.brassring.com<mailto:Enterprise@trm.brassring.com>> where you can find the reference number,

Or Else

Figure-IV 7.2: INBOX

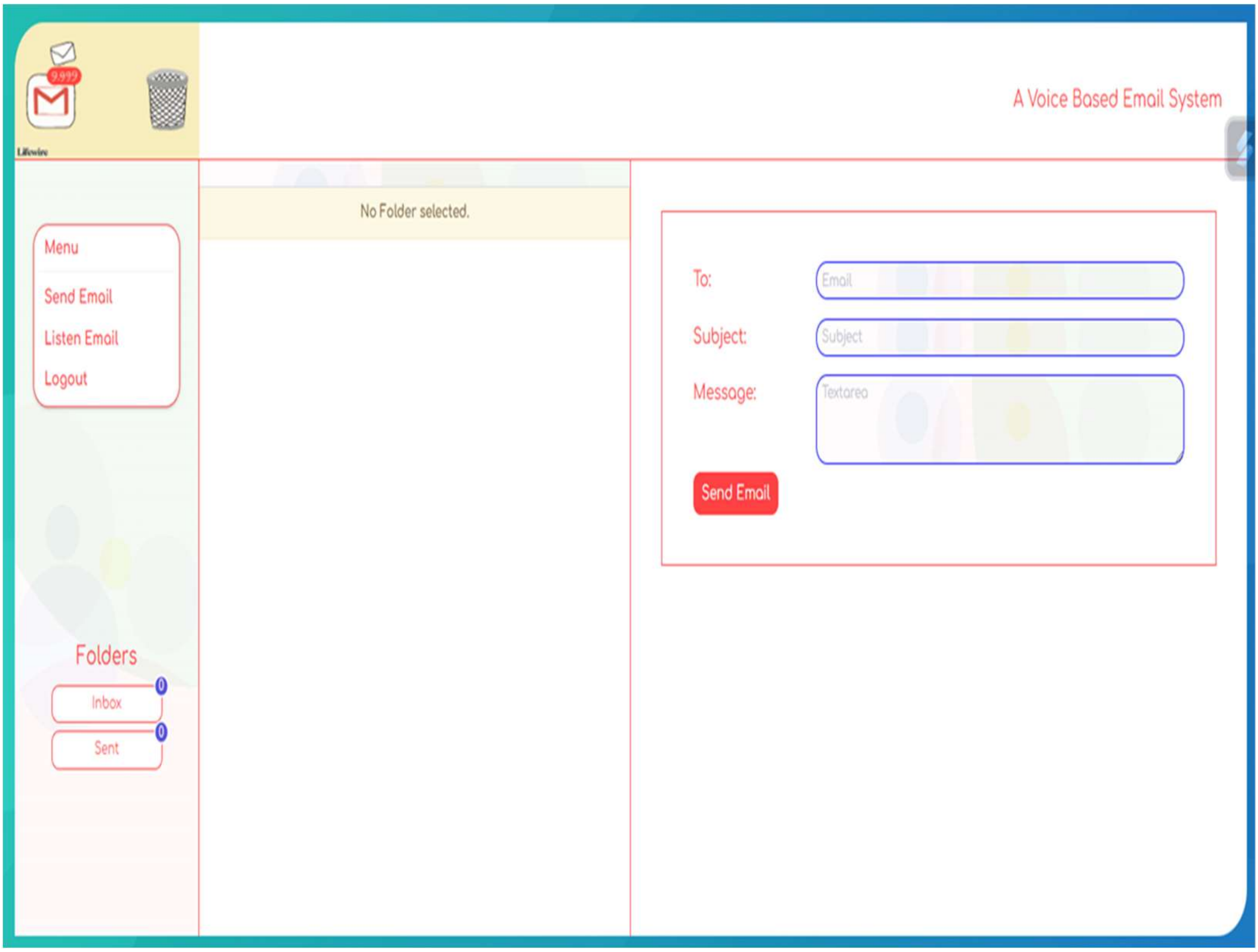


Figure-V 7.2: MENU

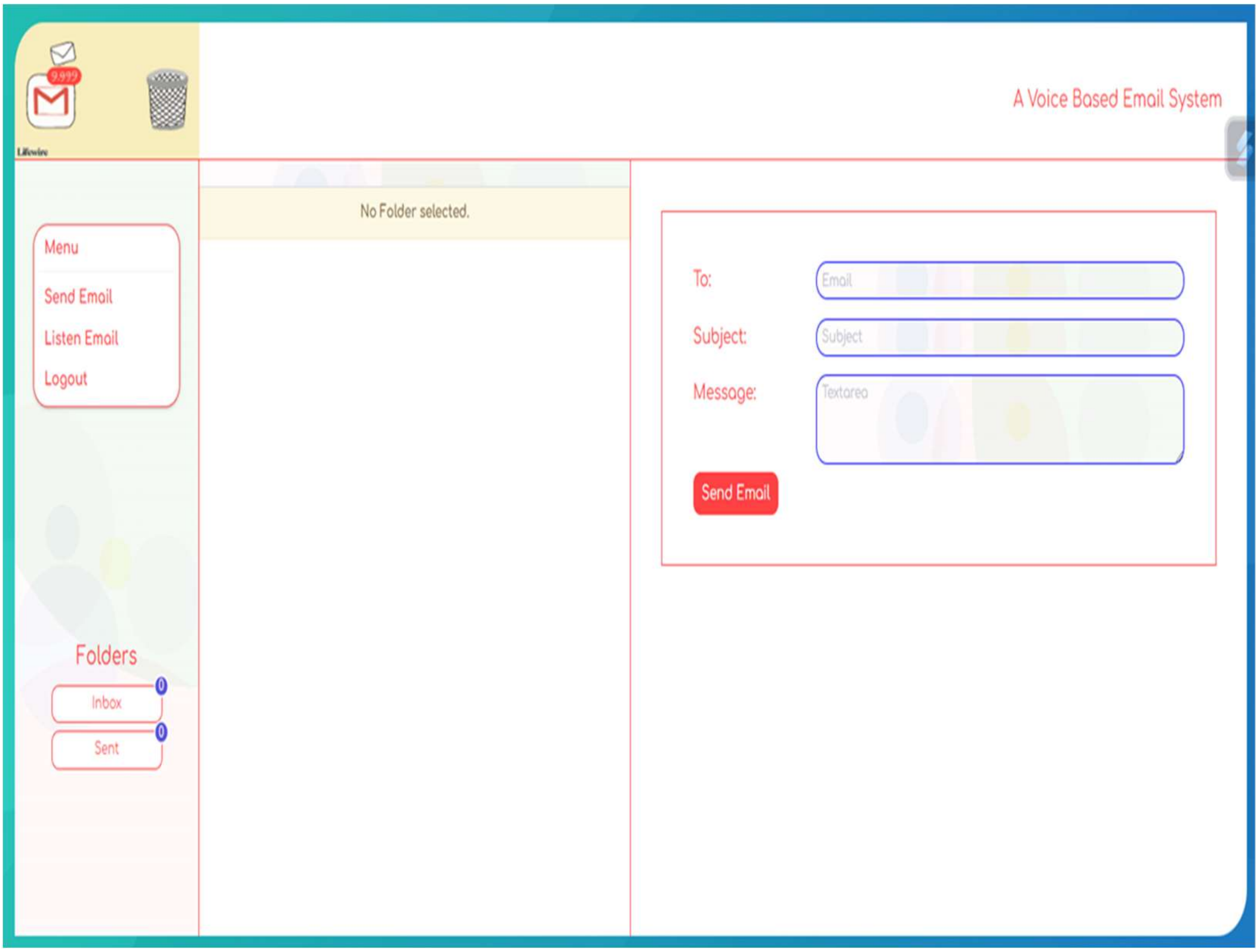


Figure-VI 7.2 COMPOSE EMAIL

7.3 Application

This project is being proposed with the noble intention of enhancing the overall welfare and advancement of society. The primary objective of this endeavor is to extend assistance to individuals who are visually impaired, enabling them to actively participate in the ever-evolving landscape of digital India by harnessing the power of the internet. Additionally, its aim encompasses the facilitation of a more convenient and manageable lifestyle for this demographic. Furthermore, the realization of success in this venture holds the potential to serve as a catalyst, inspiring developers to channel their efforts towards creating even more beneficial solutions tailored to the needs of visually impaired individuals, as well as those who are illiterate, ensuring that they too are afforded equitable opportunities within society.

This visionary project has been conceptualized with the overarching goal of fostering societal progress and inclusivity. Through targeted initiatives, it endeavors to bridge the digital divide for visually impaired individuals, empowering them to actively engage in the digital revolution sweeping across India. By leveraging the vast resources of the internet, this project seeks to unlock new avenues of accessibility and independence for this marginalized community.

Moreover, the multifaceted benefits of this project extend beyond immediate assistance, aiming to cultivate a culture of innovation and compassion within the developer community. The successful implementation of this endeavor promises to serve as a beacon of inspiration, encouraging technologists and innovators to dedicate their skills and resources towards addressing the unique challenges faced by visually impaired and illiterate individuals.

In essence, this project represents a pivotal step towards creating a more inclusive and equitable society, where every individual, regardless of their abilities, can actively contribute to and benefit from the advancements of the digital age.

7.4. **Future Scope**

The incorporation of voice functionality could potentially be expanded to encompass a broader array of features, including the ability to interact with image attachments and utilize additional formatting options such as indentation and various fonts, akin to those commonly found in traditional email platforms.

Expanding the voice capabilities to encompass interactions with image attachments would offer users a seamless and inclusive experience, enabling them to access and engage with visual content effortlessly. Furthermore, integrating features such as indentation and diverse font selections would enhance the versatility and expressiveness of communication within the email platform, catering to individual preferences and communication styles.

By embracing these enhancements, the email interface can evolve into a more dynamic and accommodating platform, accommodating diverse user needs and preferences. This expanded functionality not only enhances accessibility for all users but also enriches the overall user experience, fostering greater engagement and satisfaction with the email service.

In essence, the integration of voice technology alongside enhanced image attachment handling and formatting options represents a significant step towards modernizing and enriching the email experience, aligning it with contemporary communication standards and user expectations.