

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in

Information Technology

by

DANIEL N

(37120018)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

AUGUST 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **Daniel N(Reg. No. 37120018)** who underwent the professional training in **“Processing an Email by using the input as voice via an API”** under our supervision from May 2020 to June 2020.

Internal Guide

Dr.V.MARIA ANU.,M.E.,Ph.D.,

Head of the Department

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

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Daniel N(Reg. No.37120018)** hereby declare that the Professional Training Report on “**Processing an Email by using the input as voice via an API**” done by me under the guidance of **Dr.V.Maria Anu.,M.E.,Ph.D**,at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information And Technology.

DATE:

PLACE:

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

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

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

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. V.Maria Anu, M.E.,Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

TRAINING CERTIFICATE:



CERTIFICATE OF COMPLETION

Presented to

Daniel N

For successfully completing a free online course
Building Games using JavaScript

Provided by
Great Learning Academy
(On January 2021)

To verify this certificate visit verify.greatlearning.in/YBGLNJTX

ABSTRACT

This project is for sending an electronic mail via an Application Programming Interface by using the input as voice or text. It contains a user interface where the user is asked to enter the receiver name. The input can be given as voice or text by the user. Then the receiver name will be searched from the database. Next, the receiver Mail ID will be taken from database. Later the user need to give the content or message as voice or text. At last, the interface will ask for confirmation. If the user said 'yes', The mail will be sent. This project will be much useful for the visually challenged people.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF SYMBOLS AND ABBREVIATIONS	viii
	LIST OF TABLES	xi
1	INTRODUCTION	
	1.1 OUTLINE OF THE PROJECT	10
2	AIM & SCOPE	
	2.1 AIM	11
	2.2 PROBLEM STATEMENT	11
	2.3 LITERATURE REVIEW	11
	2.4 SCOPE	13
3	ALGORITHMS AND METHODS	
	3.1 GENERAL	14
	3.2 OVERVIEW	14
	3.3 PROJECT DESCRIPTION	14
	3.4 SUMMARY	15
4	SYSTEM IMPLEMENTATION	
	4.1 REQUIREMENTS	16
	4.2 ROLE OF HTML	16
	4.3 RESULTS AND DISCUSSION	17
5	CONCLUSION AND FUTURE WORK	
	5.1 CONCLUSION	23
	5.2 FUTURE WORK	23
	REFERENCES	23
	APPENDIX	24

LIST OF FIGURES

Figure No.	Figure Name	Page No.
4.1	UI page	19
4.2	Recipient name - voice	19
4.3	Subject - voice	20
4.4	Body - voice	20
4.5	Attaching File	21
4.6	Giving text input	21
4.7	Success intimation	22
4.8	Failure intimation	22

LIST OF SYMBOLS AND ABBREVIATIONS

ABBREVIATION		EXPANSION
HTML	-	Hyper Text Markup Language
API	-	Application Programming Interface
JS	-	JavaScript
Email	-	Electronic mail
UI	-	User Interface

LIST OF TABLES

Table : Email Details

SI No	Name	Datatype
1	Name	Varchar(20)
2	Email ID	Varchar(40)

CHAPTER 1

INTRODUCTION

1.1 OUTLINE OF THE PROJECT:

This project is to create a user interface that will be used to the user to send an electronic mail to their friends and colleagues. This project contains an application interface in which the user need to enter the recipient mail address, subject and content of the mail. This interface is developed using the HTML and JS (JavaScript). The main objective of this project is to send electronic mail by using the input as voice or text. It is much useful for the visually challenged people.

This application Programming interface contains a button for voice recognition. If the user clicks it, the interface will start recognizing for the recipient mail name, then the name will be searched in the database. If the name of the recipient is available in the database, then the recipient's mail address is directly fetched from the database. After giving the voice input for the subject and content, the interface will ask for the confirmation.

If the user confirms it, then the mail will be sent immediately. If the user rejects it, that mail will be ignored. The user's mail address and password will only be available in backend code. It won't be available in the interface. For sending email first template for the email should be created in Email-js or node mailer.

CHAPTER 2

AIM & SCOPE

2.1 AIM :

The main objective of this project is to send an electronic mail with recognizing voice input. This user interface is developed for the user to send their recipient mail address with voice input, but the recipient will receive it as text. It is especially designed for the visually challenged people. Additional feature is, there is no need to tell the recipient mail address, if the user said their recipient name that is more than enough. Their recipient mail address will be taken from the database.

2.2 PROBLEM STATEMENT:

For sending email there are number of applications available in the internet. But we don't have any system which is sending email by recognizing our voice input. So our job is to design a user interface which is sending electronic mail using the voice input or text. Where the user need to give recipient mail address, subject and body or content of the mail. If the user wants to give input as voice then he just want to press the start button. Then interface will start asking for input. If the user wants to give the input as text. He can directly start giving the input in text fields.

2.3 LITERATURE REVIEW:

Hyper Text Markup Language :

HTML stands for Hypertext Markup Language. It allows the user to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications. HTML is not a programming language, meaning it doesn't have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word.

When working with HTML, we use simple code structures (tags and attributes) to mark up a website page. For example, we can create a paragraph by placing the enclosed text within a starting <p> and closing </p> tag. Overall, HTML

is a markup language that is really straightforward and easy to learn even for complete beginners in website building.

Hypertext means a text that contains references (links) to other texts that viewers can access immediately. He published the first version of HTML in 1991, consisting of 18 HTML tags. Since then, each new version of the HTML language came with new tags and attributes (tag modifiers) to the markup. According to Mozilla Developer Network's HTML Element Reference, currently, there are 140 HTML tags, although some of them are already obsolete (not supported by modern browsers).

Due to a quick rise in popularity, HTML is now considered an official web standard. The HTML specifications are maintained and developed by the World Wide Web Consortium (W3C). You can check out the latest state of the language anytime on W3C's website. The biggest upgrade of the language was the introduction of HTML5 in 2014. It added several new semantic tags to the markup, that reveal the meaning of their own content, such as <article>, <header>, and <footer>.

JavaScript :

JavaScript ("JS" for short) is a full-fledged dynamic programming language that can add interactivity to a website. It was invented by Brendan Eich (co-founder of the Mozilla project), the Mozilla Foundation, and the Mozilla Corporation. JavaScript is versatile and beginner-friendly. With more experience, you'll be able to create games, animated 2D and 3D graphics, comprehensive database-driven apps, and much more.

JavaScript itself is relatively compact, yet very flexible. Developers have written a variety of tools on top of the core JavaScript language, unlocking a vast amount of functionality with minimum effort. These include: Browser Application Programming Interfaces (APIs) built into web browsers, providing functionality such as dynamically creating HTML and setting CSS styles; collecting and manipulating a video stream from a user's webcam, or generating 3D graphics and audio samples.

Third-party APIs that allow developers to incorporate functionality in sites from other content providers, such as Twitter or Facebook. Third-party frameworks and libraries that you can apply to HTML to accelerate the work of building sites and applications. It's outside the scope of this article—as a light introduction to JavaScript—to present the details of how the core JavaScript language is different from the tools listed above. You can learn more in MDN's JavaScript learning area, as well as in other parts of MDN.

The section below introduces some aspects of the core language, and offers an opportunity to play with a few browser API features too. JavaScript is one of the most popular modern web technologies. As your JavaScript skills grow, your websites will enter a new dimension of power and creativity. However, getting comfortable with JavaScript is more challenging than getting comfortable with HTML and CSS. You may have to start small, and progress gradually.

2.4 SCOPE:

Web designing is the planning and creation of user interface. It deals with the process of developing a website. By using images, HTML5, CSS3, colors, fonts and other interactive elements the scope of the project will be increased. As this application is having the user's mail address and password is in backend, there won't be any loss of data from the database. By using the latest developing languages, the scope as well as the security of the project can be increased. Thus our project has been secured by having the user's mail and password only available in javascript file.

CHAPTER 3

ALGORITHMS AND METHODS

3.1 GENERAL :

Our project contains a single user interface which is designed using html and css. Initially user should submit the username and password of their email address. They have to submit the valid mail login credentials. Our interface has start button for giving input as voice, user needs to press or click it. After clicking it, the synthesizer will start asking for the inputs. The user needs to give the recipient name, subject and body of the email. If the recipient name is not found in the database, then prompt message will display as your recipient name is not found. If the user wants to give input as text ,he or she can give directly in the textboxes. Finally, they need to click send button, If the mail sent successfully, prompt will display success.

3.2 OVERVIEW :

There will be single page or interface for user. First he need to fill the username and password. And It contains a button for starting voice recognition. For giving the subject, body and recipient mail address there will be text fields available. For confirm sending there will “send” button. After clicking the send button mail will be sent. After denoting any alert or message a label available above the send button. If the user wants to clear the entries which he or she filled, there will be button available called “reset” button, which is available near the send button.

3.3 PROJECT DESCRIPTION :

HTML is used for front end developing. Javascript is used for back end developing. When the user enters the username, password in background it is being validated by the program. When the unregistered user wants to send an email our interface won't allow. It will show error or error will be prompted. While giving voice input, if the recipient mail address is not available in database then the user need to fill their mail address as text. By mistake, the user gives wrong input as voice - no

worries, the synthesizer will again repeat what the user said, At last our interface will ask for confirmation, if the user confirms with 'yes' then mail will be sent otherwise our mail will be ignored. For each and every step, there will be confirmation message displays.

3.4 SUMMARY :

A user interface page is created for the user's convenient. Start button is for recognizing and synthesizing voice input. Reset button is for clearing the input fields. Send button is only for validating the email address and sending the electronic mail to the recipient. In this page, there won't be any redirection to another web page or other websites. Recipient mail address is stored in separate mail database.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 REQUIREMENTS :

I. HARDWARE REQUIREMENTS

- ❖ Any operating system with minimum 2GB of RAM.
- ❖ Keyboard
- ❖ Mouse
- ❖ Monitor

II. SOFTWARE REQUIREMENTS

- ❖ Any one browser installed in the OS
- ❖ Visual Studio code (Or Other code editors)

BROWSER REQUIREMENTS :

HTML is supported in all modern browsers. In addition, all browsers, old and new, automatically handle unrecognized elements as inline elements. Because of this, you can "teach" older browsers to handle "unknown" HTML elements. Few supported browsers are GOOGLE CHROME, MOZILA FIREFOX, INTERNET EXPLORER, MICROSOFT EDGE, etc.

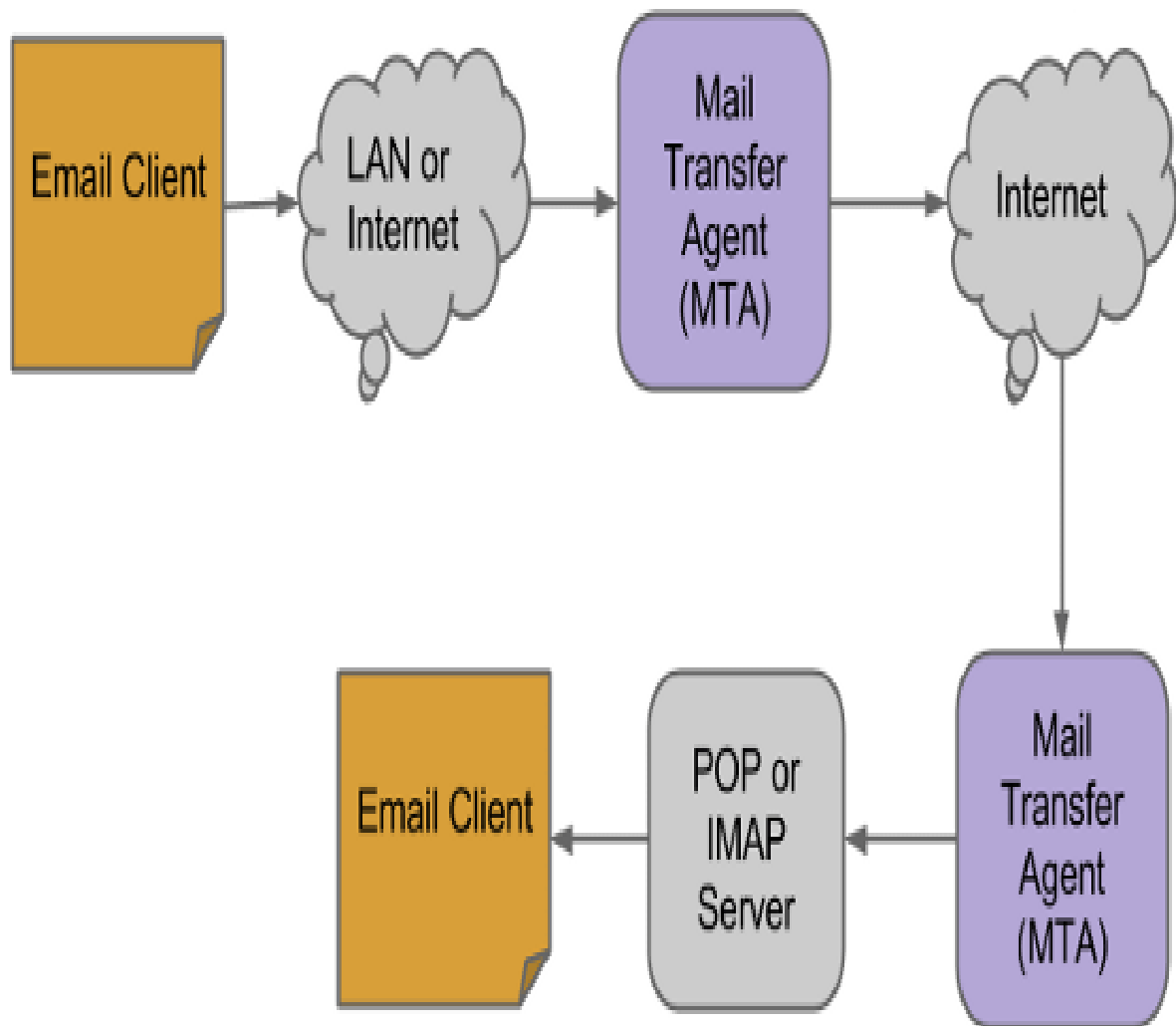
4.2 ROLE OF HTML :

Block-level elements take up the full available space and always start a new line in the document. Headings and paragraphs are a great example of block tags. Inline elements only take up as much space as they need and don't start a new line on the page. They usually serve to format the inner contents of block-level elements. Links and emphasized strings are good examples of inline tags.

4.3 RESULTS AND DISCUSSION :

The result of this project is the user can be securely logged in the javascript file of the user interface. The user can send electronic mail by using voice input or text input. This project is for the user who wants to give the input as voice which majorly useful for the people who is visually challenged.

ARCHITECTURE DIAGRAM:



FIGURES

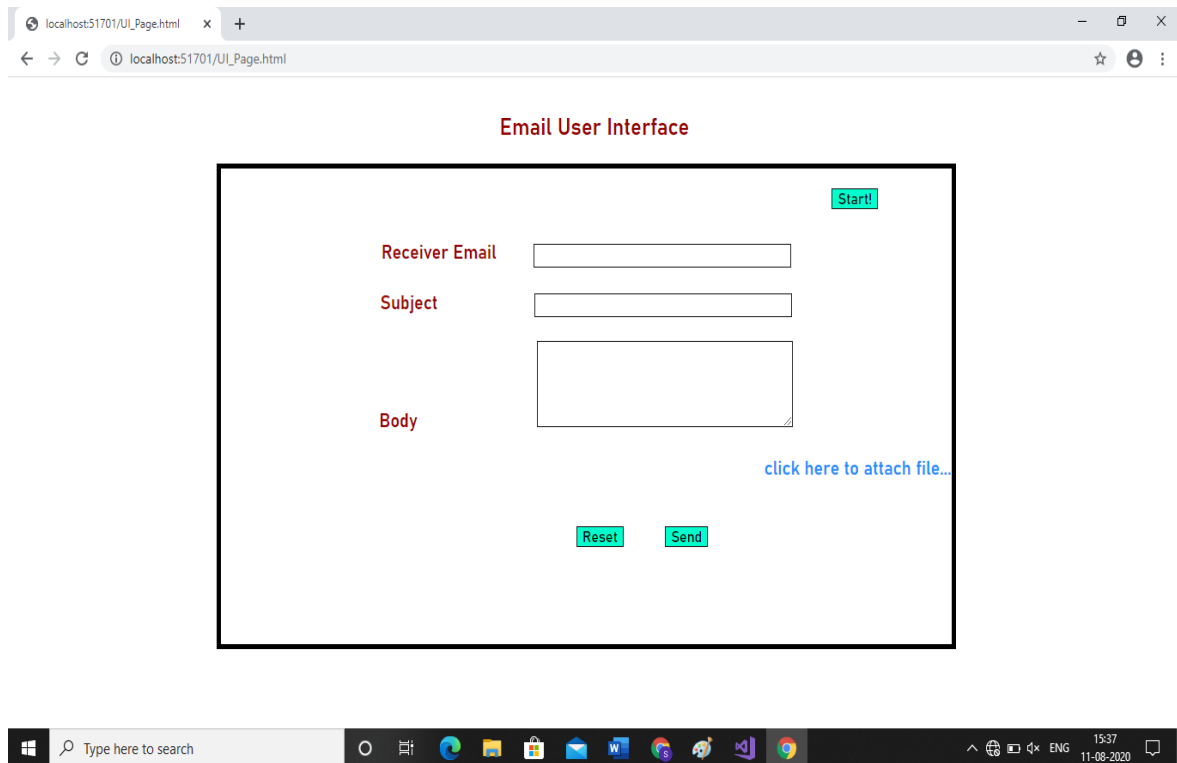


Fig 4.1 UI page

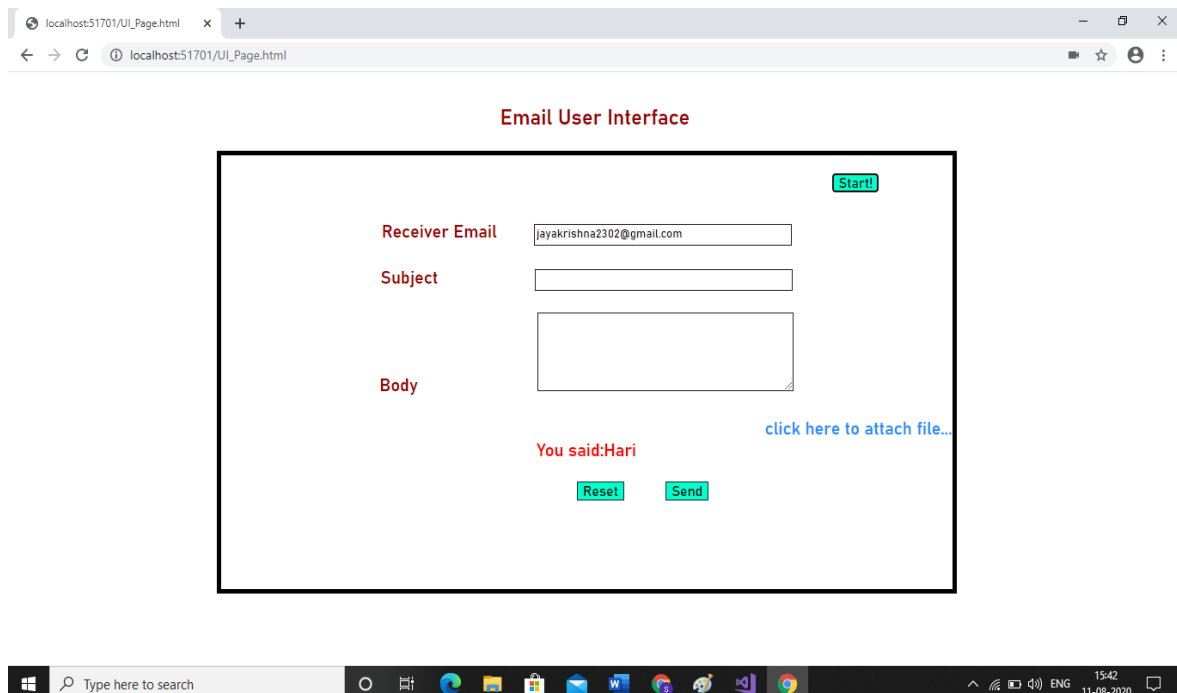


Fig 4.2 Recipient name - voice

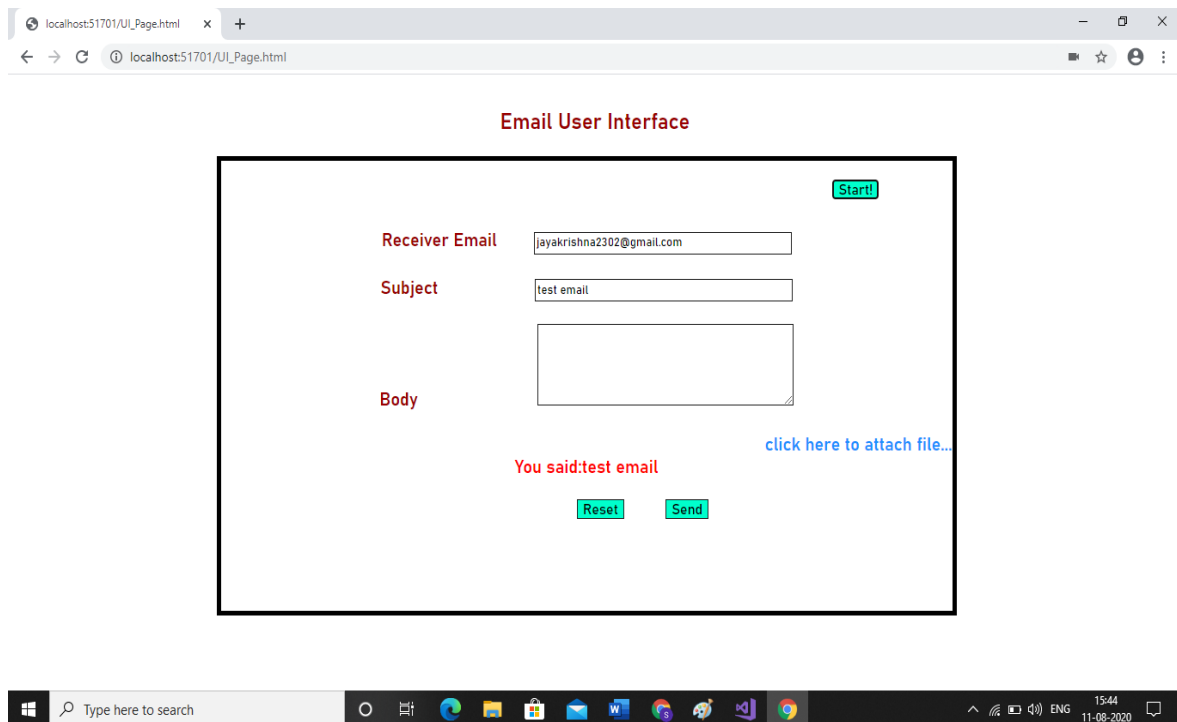


Fig 4.3 Subject - voice

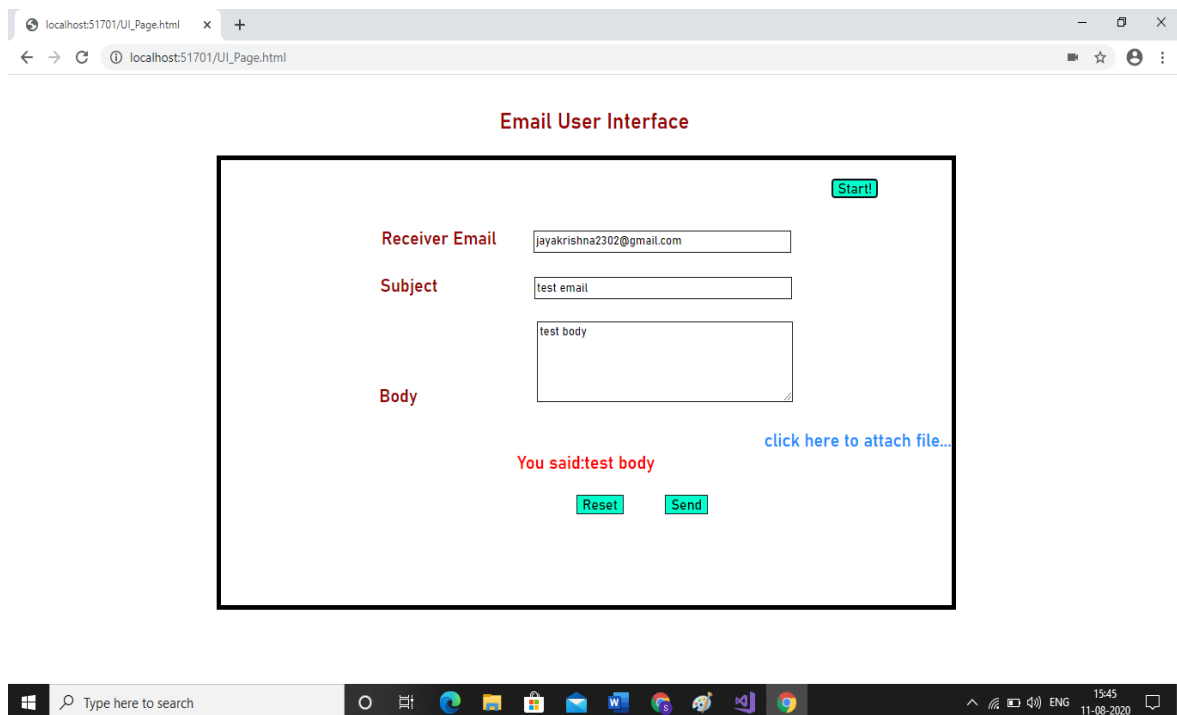


Fig 4.4 Body - voice

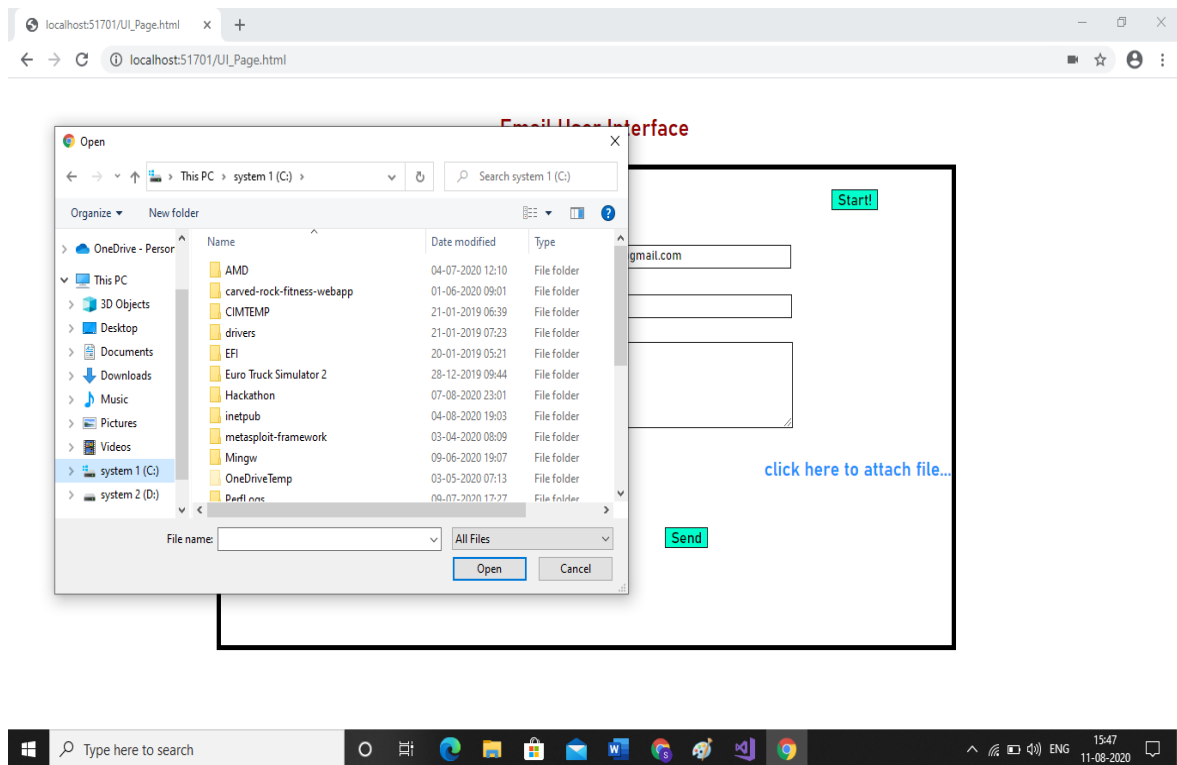


Fig 4.5 Attaching file

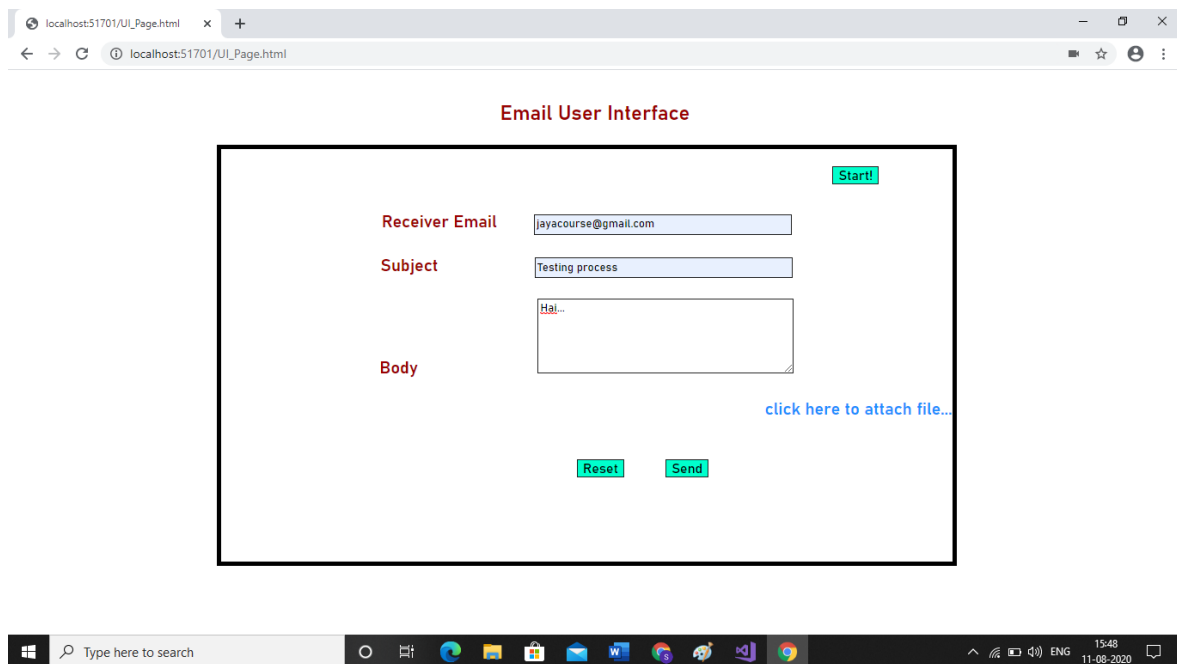


Fig 4.6 Giving text input

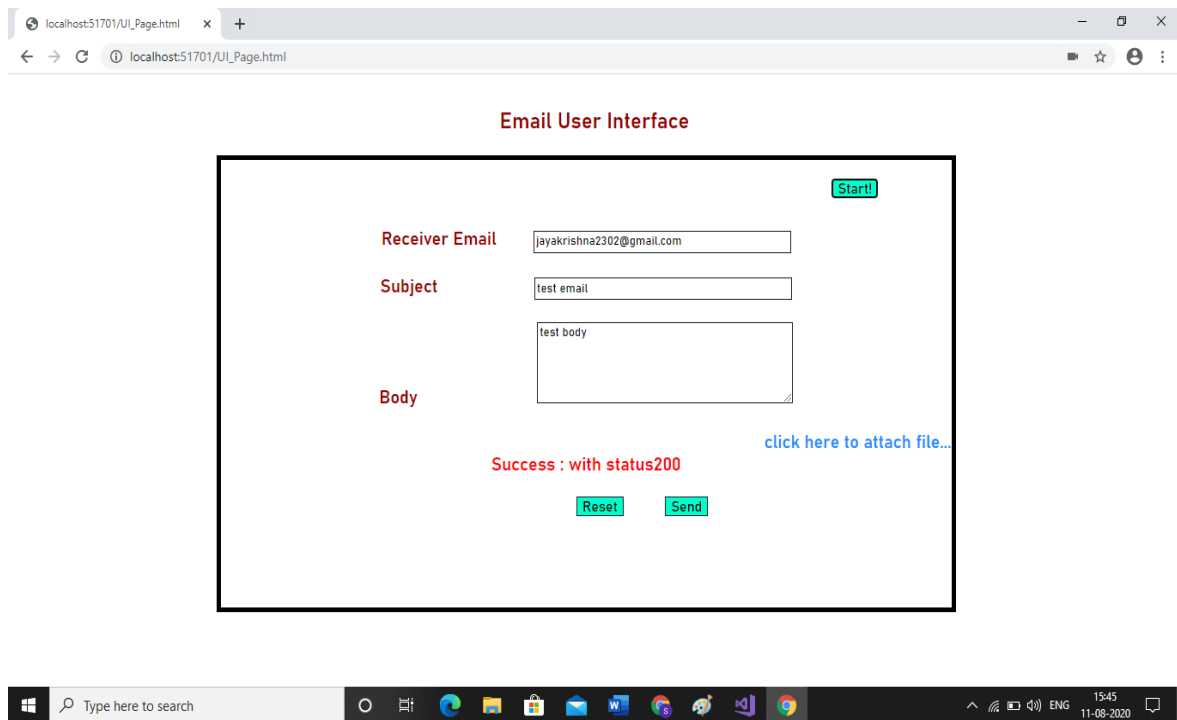


Fig 4.7 Success intimation

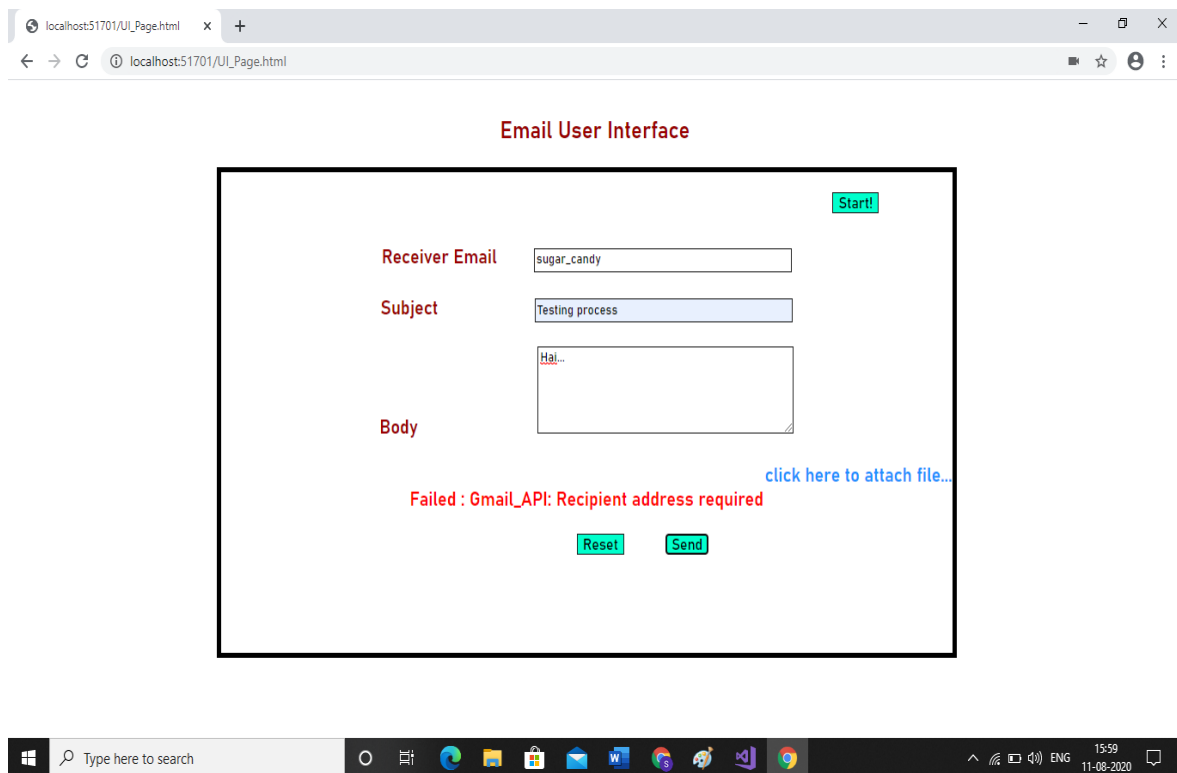


Fig 4.8 Failure intimation

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION :

Our Application Programming Interface is very simple. It is easily understood by the users who has little knowledge on email application and English. This project will be useful for the virtually challenged users and also to the user who wants to give input as voice.

5.2 FUTURE WORK :

This project can be further implemented by using CSS language for a better look and a better presentation. The latest web developing languages namely JQUERY, HTML5, CSS3, React and NODEJS and many advance level languages can be used in this project further to obtain a better appearance of the web application.

REFERENCES :

- HTML & CSS: Responsive Web Design with HTML5 and CSS–Ben Frain
- Javascript: The Principles of Object-Oriented JavaScript - Nicholas C. Zakas
- HTML - <https://www.w3schools.com/html/default.asp>
- Javascript - <https://online-learning.harvard.edu/course/introduction-web-programming-using-javascript-4?delta=0>
- SQL queries - <https://www.codecademy.com/learn/learn-sql>

APPENDICES

SAMPLE SOURCE CODE :

"Index.html" :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<title>Webpage</title>
```

```
<script      type="text/javascript"      src="https://cdn.jsdelivr.net/npm/emailjs-  
com@2.3.2/dist/email.min.js"></script>
```

```
<script type="text/javascript">
```

```
    (function () {
```

```
emailjs.init("user_flhjvrn1pMvDuPhGEUlj6");
```

```
    })();
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<div style="font-size: x-large; font-family: 'BahnschriftSemiBold'; color: #990000;  
text-align: center;">
```

```
<br />
```

```
    Email User Interface<br />
```

```
<div style="border: 5px groove #000000; font-family: 'BahnschriftSemiBold'; font-  
size: 20px; height: 480px; width:63%; margin-left: 235px; margin-top: 21px;">
```

[illegible]


```

    var SpeechGrammarList = SpeechGrammarList ||
webkitSpeechGrammarList;

    var grammar = "#JSGF V1.0;"

    var txt_receiver = document.getElementById("txt_receiver");

    var txt_subject = document.getElementById("txt_subject");

    var txt_body = document.getElementById("txt_body");

    var message = document.getElementById("message");

    var btn_send = document.getElementById("btn_send");

    var speechRecognitionList = new SpeechGrammarList();

speechRecognitionList.addFromString(grammar, 1);

    //creation of first recognition:

    var recognition1 = new SpeechRecognition();

    recognition1.grammars = speechRecognitionList;

    recognition1.lang = 'en-US';

    recognition1.interimResults = false;

    //creation of second recognition:

    var recognition2 = new SpeechRecognition();

    recognition2.grammars = speechRecognitionList;

    recognition2.lang = 'en-US';

    recognition2.interimResults = false;

    //creation of third recognition:

    var recognition3 = new SpeechRecognition();

    recognition3.grammars = speechRecognitionList;

```

```

recognition3.lang = 'en-US';

recognition3.interimResults = false;

//creation of four recognition:

var recognition4 = new SpeechRecognition();

recognition4.grammars = speechRecognitionList;

recognition4.lang = 'en-US';

recognition4.interimResults = false;

var namelist = {
"dani": "danieljackson2888@gmail.com",
    "selva": "jayacourse@gmail.com",
    "hari" : "jayakrishna2302@gmail.com"
}

//functioning of first recognition

recognition1.onstart = function () {

    var toSpeak = new SpeechSynthesisUtterance("say your receiver");

toSpeak.lang = 'en-US';

window.speechSynthesis.speak(toSpeak);

}

recognition1.onresult = function (event) {

    var last = event.results.length - 1;

    var command = event.results[last][0].transcript;

    if (command.toLowerCase() in namelist) {

txt_receiver.value = namelist[command.toLowerCase()];

```

```

message.textContent = 'You said:' + command;

    }

    else {

message.textContent = 'You said:' + command + 'But,not found';

    }

    var toSpeak = new SpeechSynthesisUtterance(message.textContent);

window.speechSynthesis.speak(toSpeak);

};

recognition1.onspeechend = function () {

    recognition1.stop();

};

recognition1.onerror = function (event) {

message.textContent = 'Error occurred in recognition: ' + event.error;

    }

    //functioning of second recognition

    recognition2.onstart = function () {

        var toSpeak = new SpeechSynthesisUtterance("Say your subject");

toSpeak.lang = 'en-US';

window.speechSynthesis.speak(toSpeak);

    }

    recognition2.onresult = function (event) {

        var last = event.results.length - 1;

        var command = event.results[last][0].transcript;

```

```

message.textContent = 'You said:' + command;

txt_subject.value = command;

    var toSpeak = new SpeechSynthesisUtterance(message.textContent);

window.speechSynthesis.speak(toSpeak);

};

recognition2.onspeechend = function () {

    recognition2.stop();

};

recognition2.onerror = function (event) {

message.textContent = 'Error occurred in recognition: ' + event.error;

    }//functioning of third recognition

    recognition3.onstart = function () {

        var toSpeak = new SpeechSynthesisUtterance("Say the content");

toSpeak.lang = 'en-US';

window.speechSynthesis.speak(toSpeak);

    }

    recognition3.onresult = function (event) {

        var last = event.results.length - 1;

        var command = event.results[last][0].transcript;

message.textContent = 'You said:' + command;

txt_body.value = command;

        var toSpeak = new SpeechSynthesisUtterance(message.textContent);

window.speechSynthesis.speak(toSpeak);

```

```

};

recognition3.onspeechend = function () {

    recognition3.stop();

};

recognition3.onerror = function (event) {

message.textContent = 'Error occurred in recognition: ' + event.error;

} //functioning of four recognition

recognition4.onstart = function () {

    var toSpeak = new SpeechSynthesisUtterance("Could you please
confirm");

toSpeak.lang = 'en-US';

window.speechSynthesis.speak(toSpeak);

}

recognition4.onresult = function (event) {

    var last = event.results.length - 1;

    var command = event.results[last][0].transcript;

    if (command.toLowerCase() === 'yes' || 's') {

message.textContent = "";

btn_send.click();

    }

    else if (command.toLowerCase() == 'no') {

message.textContent = "Mail is being ignored";

    }

```

```

        else {

message.textContent = "say yes or no";

        }

        var toSpeak = new SpeechSynthesisUtterance(message.textContent);

toSpeak.lang = 'en-US';

window.speechSynthesis.speak(toSpeak);

    };

    recognition4.onspeechend = function () {

        recognition4.stop();

    };

    recognition4.onerror = function (event) {

message.textContent = 'Error occurred in recognition: ' + event.error;

    }

    function Recording() {

        recognition1.start();

setTimeout(() => { recognition2.start(); }, 12000);

setTimeout(() => { recognition3.start(); }, 30000);

setTimeout(() => { recognition4.start(); }, 60000);

    }

```

“ProcessEmail.js”:

“use strict”

```
function Reset() {
```

```

txt_receiver.value = "";

txt_body.value = "";

txt_subject.value = "";

message.textContent = "";

    }

    function sendEmail() {

        var forms = { to_mail: txt_receiver.value, subject: txt_subject.value, body:
txt_body.value };

        emailjs.send("gmail", "chat", forms).then(function (response) {

            message.textContent = "Success : with status" + response.status;

            var toSpeakProcess = new
            SpeechSynthesisUtterance(message.textContent);

            window.speechSynthesis.speak(toSpeakProcess);

            }, function (error) {

                message.textContent = "Failed : " + error.text;

                var toSpeakProcess = new
                SpeechSynthesisUtterance(message.textContent);

                window.speechSynthesis.speak(toSpeakProcess);});

    }

```