VOICE BASED EMAIL FOR VISUALLY IMPAIRED PEOPLE

A Project Work Synopsis

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

(SPECIALIZATION IN CLOUD COMPUTING)

Submitted by:

SALONI PAWAR (20BCS4188) **SHUBHI KATIYAR**(20BCS4176)

ROHIT KUMAR RB (20BCS4155) **AYUSH CHAUHAN**(20BCS4143)

Under the Supervision of:

MR. Hariharan Udhayakumar

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,
PUNJAB

FEB 2022

TABLE OF CONTENTS

1. **INTRODUCTION**

Will cover the overview of our project.

2. **LITERATURE REVIEW**

Includes the literature available for our project.The findings of the researchers will be highlighted which will become the basis of current implementation.

3. **BACKGROUND OF PROPOSED METHOD**

Will provide introduction to the concepts which are necessary to understand the proposed system.

4. **METHODOLOGY**

Will cover the technical details of the proposed approach.

5. **EXPERIMENTAL SETUP**

Will provide information about the subject system and tools used for evaluation of the proposed method.

6. **RESULTS AND DISCUSSION**

The result of the proposed technique will be discussed here.

7. **CONCLUSION AND FUTURE SCOPE**

The major finding of the work will be presented here. Also directions for extending the current study will be discussed.

8. **REFERENCES**

1. INTRODUCTION

Before introducing our project to you , I would like to ask a very simple question: How do you communicate with someone who can't see ? Of Course by talking to them , Right ? You will not send messages or emails to them because you think they will not be able to read them .

But what if they could read out emails you send to them ? Wouldn't it be really amazing ? That's what our project will work upon . It will be a web-based application(Chrome extension - to be specific) that will work totally on voice commands .

Users don't need to type anything nor do they have to click anywhere . Just give commands with your voice and our product will do that for you . For example , a voice prompt will ask you the address of the recipient and then you have to say the address . That address will directly get written under "TO" tab in gmail .

We are using a chrome extension because gmail already has their database so there's no need to create one for our product . As a very famous quote DRY (Don't repeat yourself) exists , So we are following it by not creating what's already has been created .

2. LITERATURE REVIEW

We did a lot of research for our project and below are all the links from where we studied .

(Everything we learned is from these sites and there's a lot of data that needs to be included which cannot be written here since it's a lot . So , we are providing all the links for a better and clear understanding .)

1. All about chrome extension : -
https://developer.chrome.com/docs/extensions/mv3/overview/

2. What are API's :-
https://blogs.mulesoft.com/learn-apis/api-led-connectivity/what-are-apis-how-do-apis-work/

3. How to convert voice to text or text to voice :-
https://cloud.google.com/text-to-speech/docs/basics

4. How to access user's gmail :-
https://developers.google.com/gmail/api

3. BACKGROUND OF PROPOSED METHOD

Chrome Extensions :-

Extensions are built on web technologies such as HTML, JavaScript, and CSS. They run in a separate, sandboxed execution environment and interact with the Chrome browser. Extensions let you "extend" the browser by using APIs to modify browser behavior and access web content. Extensions operate by means of an end-user UI and a developer API:

Text -to-Speech :-

Text-to-speech (TTS) is a type of assistive technology that reads digital text aloud. It's sometimes called "read aloud" technology.TTS works with nearly every personal digital device, including computers, smartphones and tablets. All kinds of text files can be read aloud, including Word and Pages documents. Even online web pages can be read aloud.

Speech-to-Text :-

Speech to text conversion is the process of converting spoken words into written texts. This process is also often called *speech recognition*. Although these terms are almost synonymous, *Speech recognition* is sometimes used to describe the wider process of extracting meaning from speech, i.e. *speech understanding*.

API's :-

API stands for "Application Programming Interface." An API is a software intermediary that allows two applications to talk to each other. In other words, an API is the messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.

When developers create code, they don't often start from scratch thanks to the reusability of APIs. APIs enable developers to make repetitive yet complex processes highly reusable with a little bit of code. Through API reuse, developers can reduce repetitive yet complex processes and dramatically speed up their application development processes.

JavaScript :-

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved.

HTML :-

HTML stands for Hyper Text Markup Language . It is the standard markup language for creating Web pages. It describes the structure of a Web page .HTML elements tell the browser how to display the content.

CSS :-

CSS stands for Cascading Style Sheets. It describes how HTML elements are to be displayed on screen, paper, or in other media .It saves a lot of work. It can control the layout of multiple web pages all at once . External stylesheets are stored in CSS files

4. METHODOLOGY

We have divided our project into 2 modules .

1. Designing part
2. Technical part - a) to convert Text-to-Speech & Speech-to-Text
   b) to access user’s gmail account

We have assigned 2-2 members for each module .
In the 1st module, the overall structure of our product will be created. For that HTML and CSS will be used and will be included in manifest.json file which will create the chrome extension .
In the 2nd module , it’s technical part will be created with the help of JavaScript and it will include all the api’s that is required to build the product .

5. EXPERIMENTAL SETUP

We will use VS Code to create our HTML and Script files . And one GitHub account , in which we’ll upload our files in a repository so that all the members can access those files .

VS Code :-

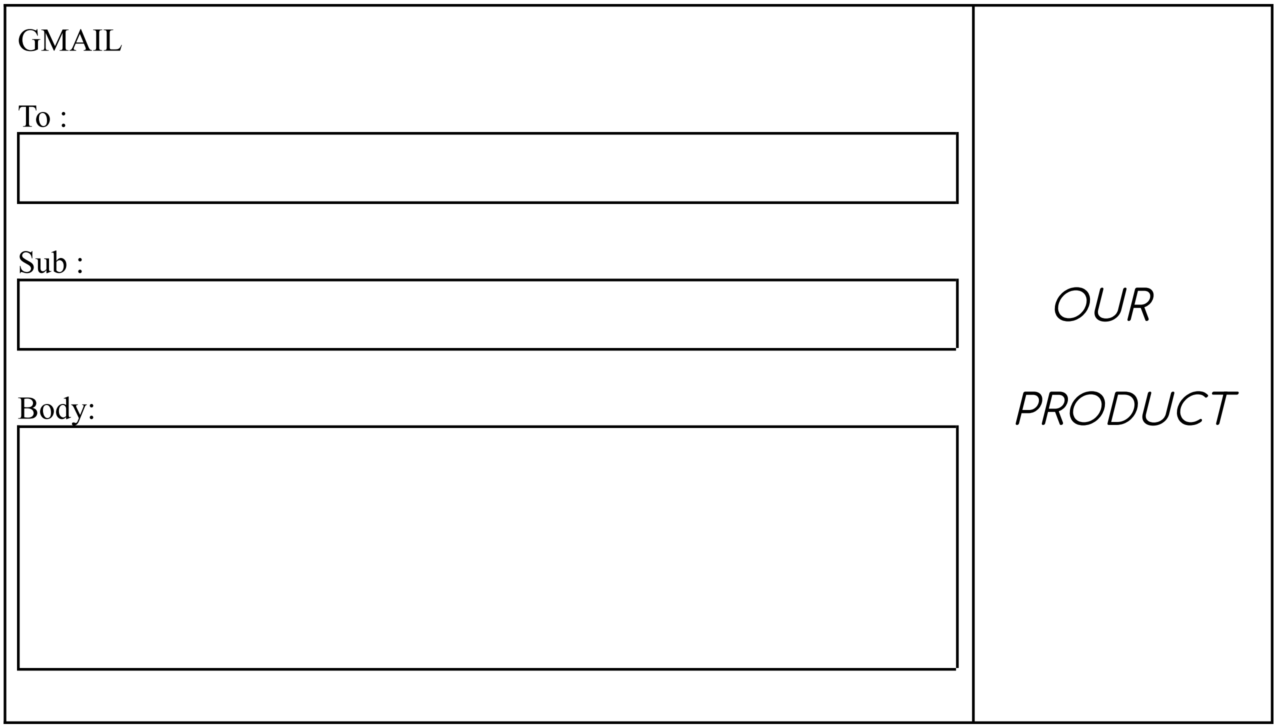
Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

GitHub :-

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.A repository is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets -- anything your project needs.

6. RESULTS AND DISCUSSION

The end result of our project will be a product - a chrome extension , which users have to download from googles extensions . It will be a small screen like a side window . like this ↓



7. CONCLUSION AND FUTURE SCOPE

For now , our product will work only for emails via a chrome extension but in future we would like to expand it more and more .
For example, we would like it to work with Siri and Google Assistant as well . like a user can command Siri to open our product directly via voice commands , and our product will then start working. So , everything from starting to the end will depend entirely on voice commands .

8. REFERENCES

[1] Marc Rochkind ,"Programming Chrome Apps: Develop Cross-Platform Apps for Chrome" O′Reilly; 1st edition (9 January 2015).

[2] Prateek Mehta ," Creating Google Chrome Extensions" Apress; 1st edition (1 January 2020).