A REPORT

ON

Developing an Interactive Voice Application for use in Government Hospitals

BY

A. IMMANUEL BLESSY	2017A7PS0255H
ANSHUMAN SRIVASTAVA	2017A8PS0241P
ARJUN RAMESH KAUSHIK	2017A8PS0563H
C. ESWAR BABU	2017A3PS0297P
C.SRI KRISHNA	2017AAPS0461H
K.AJITH KRISHNA	2017A7PS0104H
KAMALESH S	2017B3A30560P
M.RAJINEESH REDDY	2017AAPS1618H
P. YASESHWINI REDDY	2017A8PS0825H
SAI KRISHNA NEERAJ B	2017B4A80784P
SAI VAMSI ADAPA	2017A7PS0023H

AT

TAMIL NADU e-GOVERNANCE AGENCY, CHENNAI

A Practice School – I station of

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
JUNE-2019

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN) Practice School Division

Station : Tamil Nadu e-Governance Agency (TNeGA) Centre : Chennai

Duration : 8 weeks Date of Start : 28th MAY 2019

Date of Submission : 14th JULY 2019

Title of the Project : Developing an Interactive Voice Application for use in General Hospitals.

Students involved:		
ID No.	Name	Discipline
2017A7PS0255H	A. Immanuel Blessy	Computer Science
2017A8PS0241P	Anshuman Srivastava	Electronics and Instrumentation
2017A8PS0563H	Arjun Ramesh Kaushik	Electronics and Instrumentation
2017A3PS0297P	C. Eswar Babu	Electrical and Electronics
2017AAPS0461H	C.Sri Krishna	Electronics and communication
2017A7PS0104H	K.Ajith Krishna	Computer Science
2017B3A30560P	Kamalesh S	Msc. Economics + Electrical and Electronics
2017AAPS1618H	M.Rajineesh Reddy	Electronics and communication
2017A8PS0825H	P. Yaseshwini Reddy	Electronics and Instrumentation
2017B4A80784P	Sai Krishna Neeraj B	Msc. Mathematics + Electronics and Instrumentation
2017A7PS0023H	Sai Vamsi Adapa	Computer Science
Experts involved:		
Name	Designation	
Mr. Santosh K Misra, IAS	CEO/Commissioner of TNeGA	
Dr. Bruno Mascarenhas	Consultant, Neurosurgeon	Consultant in Directorate of Medical and Rural Health Services

Name of the PS Faculty:

Mrs. Bharathi R

Key Words:

- Dialog Flow
- FireBase
- Intents
- Entities
- Natural Language Processing

Project Areas:

- Cloud Functions
- Natural Language Processing
- Firebase database.
- Development of Interactive Bots
- Web development

Signature(s) of Student(s) **Signature of PS Faculty**

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

We would like to thank Mr. S Nagarajan, IAS, Director of Tamil Nadu eGovernance Agency for providing us with the opportunity to work on the crucial problem of providing ease of access to information about hospital services and also for his valuable suggestions for the project. We would also like to thank Dr. J.Mariano Anto Bruno Mascarenhas for helping us to familiarize ourselves with the problem and the data available and also for introducing to us the existing project. We would like to thank Mrs. Bharathi R, PS In-charge, for introducing us to the people in charge at the Practice School station. We also thank Sunny Jose, Subject Matter Expert, for his advice and expertise. We also thank Mr Faheem, for his constant encouragement.

ABSTRACT

With the current estimate of India’s population being 133.92 crores, any process, especially in the medical sphere involving a human can be a protracted process. This calls for decreased human interaction and increased machine involvement, hence the project. This project is aimed at making information retrieval a tad bit easier and all in all ameliorate the process.

The project involves the development of an application which would guide the user based on his/her requirement. The interactive mobile application receives voice input from the user which would be recognized through voice recognition using Google’s existing API, Dialogflow, integrated with Facebook Messenger and Google Assistant. With the given input, the data is transferred to the cloud function which would then process the data and return an appropriate response. This response would then enable the user to conveniently make decisions, locate and avail the services of the public service center. The application would also have different facets to it, one interface for the user, one for the public service center, and one for the developers to update and optimize the application when need be. Thus all the arms of the main application would be working in tandem, and would enable a neat transition to make the application versatile.

The app developed by our team pertains to hospitals. The app has been programmed to pilot the user to the doctor after acquiring the necessary input. DialogFlow is being used to interact with the user for inputs like the symptoms of the patient and personal details. This information is then processed to book an appointment. The app is also programmed to give voice outputs informing about the doctor's visitation hours and leisure hours, and the directions to his/her room as well. Javascript along with Firebase, a data management application, is used to integrate the necessary data to guide the user after predicting which speciality doctor has to be consulted.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

Practice School Division

Response Option Sheet

Station : Tamil Nadu e-Governance Agency **Centre : Chennai**

Names and ID numbers:

A. IMMANUEL BLESSY	**2017A7PS0255H**
ANSHUMAN SRIVASTAVA	**2017A8PS0241P**
ARJUN RAMESH KAUSHIK	**2017A8PS0563H**
C. ESWAR BABU	**2017A3PS0297P**
C.SRI KRISHNA	**2017AAPS0461H**
K.AJITH KRISHNA	**2017A7PS0104H**
KAMALESH S	**2017B3A30560P**
M.RAJINEESH REDDY	**2017AAPS1618H**
P. YASESHWINI REDDY	**2017A8PS0825H**
SAI KRISHNA NEERAJ B	**2017B4A80784P**
SAI VAMSI ADAPA	**2017A7PS0023H**

Title of the Project :

Developing an Interactive Voice Application for use in General Hospitals.

Code No .	**Response Options**	**Course No. (s) & Name**
1.	A new course can be designed out of this project	
2.	The Project can help modification of the course content of Some of the existing Courses	CS F212(DBMS)
3.	The Project can be used directly in some of the existing Compulsory Discipline Courses (CDC)/Discipline Courses Other than Compulsory (DCOC)/ Emerging Area (EA) etc. courses	
4.	The Project can be used in Preparatory courses like Analysis and Application Oriented Courses (AAOC)/Engineering Science (ES)/Technical Art (TA) and Core Courses	
5.	This Project cannot come under any of the above mentioned options as it relates to the professional work of the host organisation	

Signature of Student

Signature of Faculty

1.INTRODUCTION

Tamil Nadu Health Systems Project is an initiative of the Government of Tamil Nadu, in partnership with the World Bank, to create a health system in Tamil Nadu that is highly accessible, equitable and effective. It is also responsible for providing equitable access to information about the services provided by government hospitals.

Citing the dearth of a proper guideline system to direct people to the right doctor at the right place at the right time, our team has taken the initiative to design an interactive voice application to do so based on the input consisting of symptoms/problems of the patient. With the guidance system intact and decreased human interchange, the app advances the process with efficiency. The application would have vast implications and uses in the industry. It would bring about a sense of accountability in doctors and medical professionals working in the government hospital as the absence of the doctor/medical professional is notified through the system. This app also effectively circumvents briberies existing inside the management system of a hospital. Thus we would develop an Interactive Voice Application for use in Government Hospitals, to direct the patients to correct rooms at the right timing for services.

2. FINDING SUITABLE VOICE RECOGNITION PLATFORM

Parameter Platforms	Accuracy	Language Support	Operating System	Speed and Accuracy (Trial 1) Medical Conditions used in trial Excessive thirst and hunger, Frequent urination, Weight loss or gain, Fatigue, irritability, Blurred vision.	Speed and Accuracy (Trial 2) Medical Conditions used in trial Nausea, indigestion, heartburn, abdominal pain, shortness of breath, cold sweat, Fatigue	Speed and Accuracy (Trial 3) Medical Conditions used in trial Joint pain, tenderness, swelling redness, warmth, Joint stiffness.
Google Voice	99%	English,Arabic, Chinese (Traditional), Danish, Dutch, French, German, Hindi, Indonesian, Italian, Japanese, Korean, Norwegian, Polish, Portuguese (Brazil), Russian, Spanish, Swedish, Thai, Turkish, Vietnamese	Android, Chrome OS, iOS and KaiOS	Instant	Instant	Instant
Alexa	0%	English, French, German, Japanese, Italian Spanish Portuguese Punjabi Marathi	Fire OS 5.0 or later, iOS 8.0 or later; Android 4.4 or later	Failure	Failure	Failure

Siri	80%	English, French, Arabic Chinese (Cantonese, Mandarin, Mandarin (Taiwan)) Danish Dutch Finnish German Hebrew Italian Japanese Korean Malay Norwegian Portuguese Russian Spanish Swedish Thai Turkish	iOS 5 onward, macOS Sierra onward, tvOS (all versions), watchOS (all versions), audioOS (all versions)	Successful and fast	Mostly Successful and fast. However difficulty in some words.	Successful and fast
Web Speech API	90%	Afrikaans, አማርኛ, Azərbaycanca, বাংলা, Bahasa, Indonesia, Bahasa, Melayu, Català, Čeština, Dansk, Deutsch, English, Español, Euskara, Filipino, Français, Basa, Jawa, Galego, ગુજરાતી, Hrvatski, , IsiZulu, , Íslenska, , Italiano, ಕನ್ನಡ, ភាសាខ្មែរ, Latviešu, Lietuvių, മലയാളം, मराठी, Magyar, ລາວ, , Nederlands, , नेपाली, भाषा, Norsk, bokmål, Polski, Português, Română, සිංහල, Slovenščina, Basa, Sunda, Slovenčina, Suomi, Svenska, Kiswahili, ქართული, Հայերեն, தமிழ், తెలుగు, Tiếng, Việt, Türkçe, اُردو, Ελληνικά, български, Русский, Српски, Українська, 한국어, 中文, ภาษา ไทย, 日本語, हिन्दी	Android, Chrome OS, and iOS	Instant	Instant	Instant
Cortana	70%	English French Chinese (Mandarin) German Spanish Italian Japanese Brazilian Portuguese	Windows, iOS, Android, Xbox OS	Mostly fast. However difficulty in many words.	Mostly fast. However difficulty in many words.	Successful and fast

Table 2.1

Voice recognition or Speech recognition is the recognition and conversion of speech into text by computers. Since our bot interacts with the user mainly through voice, having an efficient, accurate and lucid voice recognition platform is vital. The chosen platform should also be easily integrable into our cloud function.

As given in the above table, we gathered data on various popularly used voice recognition platforms and conducted elementary voice recognition tests on them. Some of these tests' results have been tabulated above. Based on these results, Google Voice is the obvious choice. But the Google Cloud Speech API is a paid service. This is a hindrance as our application is designed to benefit the user at minimal cost. Dialogflow platform side steps these issues, hence the choice.

This platform not only allows us to integrate Google Assistant with our bot, but also has the capability to integrate with Faceboook Messenger. It also provides easy integration into the Cloud Function to facilitate the easy exchange of data between the two mediums to help provide quick and useful responses.

3.DIALOG FLOW

Dialogflow lets us build conversational interfaces on top of our products and services by providing a powerful natural language understanding (NLU) engine to process and understand natural language input. Traditional computer interfaces require structured and predictable input to function properly, which makes the use of these interfaces unnatural and sometimes difficult. If users can't easily figure out this structured input, they'll have a hard time figuring out what to do.

A traditional computer interface would tend to force users to input a well-known, standard request at the detriment of the user experience, because it's just easier. However, Dialogflow lets us easily achieve a conversational user experience by handling the natural language understanding (NLU) for us. When we use Dialogflow, we can create agents that can understand the vast and varied nuances of human language and translate that to standard and structured meaning that your apps and services can understand.

3.1 Agents

Agents are best described as Natural Language Understanding (NLU) modules. These modules can be included in our app, website, product, or service and translate text or spoken user requests into actionable data. This translation occurs when a user's utterance matches an intent within our agent. An agent helps us process user input into structured data that we can use to return an appropriate response. We define all of these things inside one or many intents, which define how to map user input to a corresponding response.

3.2 Intents

In Dialogflow, the basic flow of conversation involves the user giving input, Dialogflow agent parsing that input and the agent returning a response to the user. To define how conversations work, we create intents in our agent that maps user input to responses. In each intent, we define examples of user utterances that can trigger the intent, what to extract from the utterance, and how to respond.

Generally, an intent represents one dialog turn within the conversation. For example, you could create an agent that recognizes and responds to users' input about their favorite color. If a user said something like "My favorite color is purple", your agent would match that input to its corresponding intent and return the response you defined within that intent. Your agent's response usually prompts users for another utterance, which your agent will attempt to match to another intent, and the conversation continues.

Let's take a look at an intent to get a better idea of how it works. A basic intent is comprised of these components:

1. **Training phrases:**

   Defines example phrases of what users can say. Dialogflow uses these training phrases and naturally expands them to many more similar phrases to build a language model that accurately matches user input. Through further training and machine learning, Dialogflow builds a more robust and varied language model to better match user input.

2. **Action and Parameters:**

   To improve an intent's language model, you can also annotate your training phrases with entities, or categories of data that you want Dialogflow to match. This lets you tell Dialogflow that you want a particular type of input and to not just match the literal input of the user. Dialogflow extracts matched entities as parameters from the training phrases. You can then process these parameters in logic called fulfillment to further customize a response to the user.

3. **Responses:**

   Defines a text, speech, or visual response to the user, which usually prompts users in a way that lets them know what to say next or that the conversation is ending. To send responses, you can use Dialogflow's built-in response handler or call fulfillment to process the extracted data and return a response back to Dialogflow.

3.3 Entities

Entities are Dialogflow's mechanism for identifying and extracting useful data from natural language inputs. While intents allow your agent to understand the motivation behind a particular user input, entities are used to pick out specific pieces of information that your users mention — anything from street addresses to product names or amounts with units. Any important data you want to get from a user's request will have a corresponding entity.

Entity type: Defines the type of information you want to extract from user input.

Entity entry: For each entity type, there are many entity entries. Each entity entry provides a set of words or phrases that are considered equivalent.

Dialogflow is equipped with numerous system entities, which allow agents to extract information about a wide range of concepts without any additional configuration. If you need to extract information about concepts beyond those covered by Dialogflow's system entities, you can define your own developer entity types. It is also possible to define entity types that apply only to a specific conversation.These are called session entity types.

3.4 Contexts

Contexts represent the current state of a user's request and allow your agent to carry information from one intent to another. You can use combinations of input and output contexts to control the conversational path the user takes through your dialog. There are two types of context that let you activate and deactivate contexts and can control the flow of your conversation:

1. Input contexts:

   When applied to an intent, an input context tells Dialogflow to match the intent only if the user utterance is a close match and if the context is active.

2. Output contexts:

   When applied to an intent, an output context tells Dialogflow to activate a context if it's not already active or to maintain the context after the intent is matched.

By default, output contexts expire after either five requests or 20 minutes after its corresponding intent is matched. If the same output context is included in another intent, the context resets the counter and clock to five requests and 20 minutes. Contexts in follow-up intents have a default lifespan of two requests. Follow-up intents provide a simple way to shape a conversation without having to create and manage contexts manually. These special intents are nested under their parent intent and are designed to handle preset replies from the user

3.5 Events

Events allow you to invoke intents based on something that has happened instead of what a user communicates. Dialogflow supports events from several platforms (like Google Assistant, Slack, and more) based on actions users take on those platforms.
Platform events are events triggered by actions users take on platforms Dialogflow interacts with, like the Google Assistant, Slack, and Facebook Messenger. These events give you a way to respond to the user's action in your Dialogflow agent and in fulfillment.
Custom events are events you create to signify some communication that can't be captured easily through text or voice. Custom events can indicate that a user has clicked a button, provided authorization, or that a certain amount of time has passed. These events can be triggered through Dialogflow fulfillment or the detect intent API. These events can be handled in your Dialogflow agent or in fulfillment.

3.6 Fulfillment

Fulfillment is code that's deployed as a webhook that lets your Dialogflow agent call business logic on an intent-by-intent basis. During a conversation, fulfillment allows you to use the information extracted by Dialogflow's natural language processing to generate dynamic responses or trigger actions on your back-end. Fulfillment is custom logic that you implement as a webhook, which services requests, processes them, and returns responses. You can enable fulfillment for any of your agent's intents.

To use fulfillment, you need to set up a webhook. A webhook is a web server endpoint that you create and host. Here, we connect to the Cloud Function.

3.7 Training and Analytics

Dialogflow provides features that can help you build and refine your agents using real world data. As a developer, you can leverage existing sources of conversation data you might have access to, as well as usage and performance data that pertains to your running agents:

1. Use training to add existing data to training phrases.
2. Use training to add additional data to training phrases.
3. Use analytics to assess performance of agent.

Since Dialogflow's natural language processing is based on machine learning, you can add training data that the agent learns from and uses to improve its performance. Dialogflow's training feature provides an interface for incorporating both external and internal customer interaction logs into an agent's training phrases. You can use this feature to build a new Dialogflow agent using logs of existing customer interactions and to improve the performance of a live Dialogflow agent using its own logs. The analytics page gives you insight into how well your agent is performing, so you can work to further improve the user experience you’re providing. We show two types of data related to the agent and the conversations it’s been a part of:

- Usage data: Number of sessions and queries per session.
- NLU data: Most frequently used intents and exit percentages.

3.8 Integrations

Once we build our agent on Dialogflow, we can use Dialogflow's integration tools to make our agent available on multiple platforms. Dialogflow's one-click integrations help us manage the integration of our agent with the Google Assistant via Actions on Google and a number of popular messaging platforms, such as Slack, Facebook Messenger, and Twitter. Dialogflow also allows an agent to be easily exported to, or imported from, other natural language understanding platforms such as Amazon Alexa and Microsoft Cortana.

Actions on Google

Actions on Google is a platform where developers can create software to extend the functionality of the Google Assistant. Integrating your Dialogflow agent with Actions on Google enables your agent to reach users on over 500 million devices that support the Google Assistant, like smart speakers, phones, cars, TVs, headphones, watches, and more.

Integration with messenger

The process of integrating our bot with Messenger, involved the following steps Setting up Facebook, by creating an app in Facebook for developers and create a Facebook page in which our bot operates.

Then we generated a page access token and used it to set up the bot from Dialogflow’s side and secured it with a password.
Configuring our agent’s webhook, in the Facebook Developer’s console.
And finally making our app public for testing.

Integration with google assistant

As Dialogflow and Google Assistant are both Google services, integration is easy and straightforward.

To integrate we simply specify the invocation that we would like to use for the application.(in this case : Dr.Hospihelp.)Then we specify requirements of the application, we also are required to provide a privacy policy stating personally identifiable information of the user collected by us, and then the regions where we would like the application to work.

We have completed the process, requested by Google to integrate our bot with the Google Assistant.

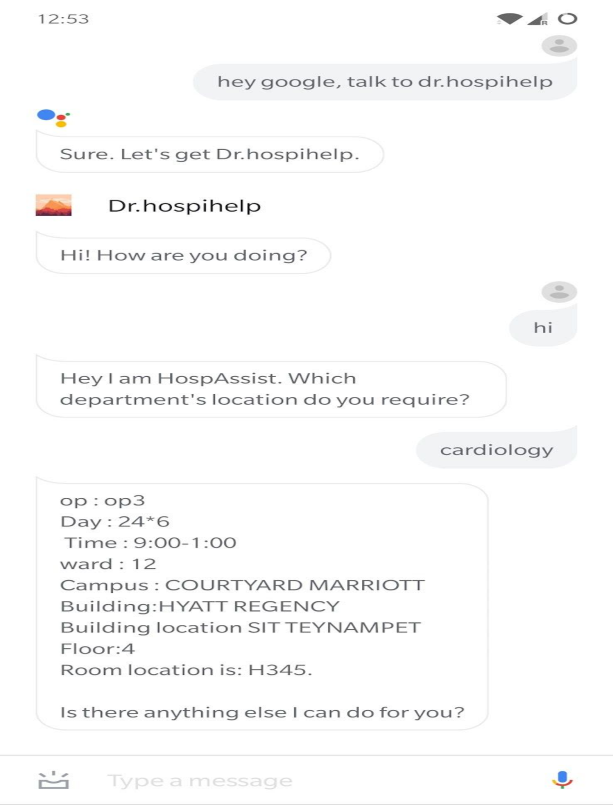
After we sent the application for review, the response from google after trials and corrections, was positive, and our app was made public by google.

Using app on google assistant:

Step 1: Open Google Assistant
Step 2: Type or say Invocation (e.g : “Talk to Dr.Hospihelp”)*
Step 3: Enter the name of the department present in the hospital that you would like to go to. (e.g : Urology Department)
Step 4: Note Data given by the Application.**

**The data provided by the bot are the op no, ward no, Building and floor no, room no and the working timings of the specified medical department.

*The user can interact with our bot either through text or through voice.



hey google, talk to dr.hospihelp

Sure. Let's get Dr.hospihelp.

Dr.hospihelp

Hi! How are you doing?

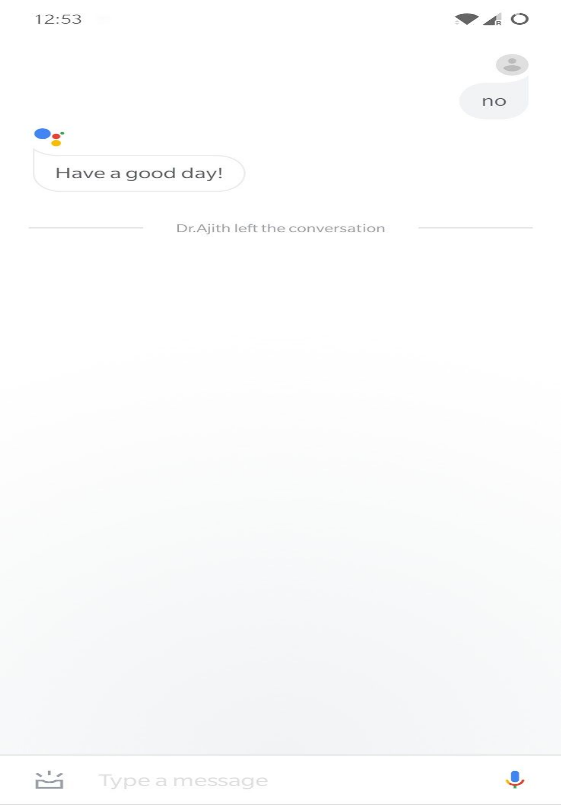
hi

Hey I am HospAssist. Which department's location do you require?

cardiology

op : op3
Day : 24*6
Time : 9:00-1:00
ward : 12
Campus : COURTYARD MARRIOTT
Building:HYATT REGENCY
Building location SIT TEYNAMPET
Floor:4
Room location is: H345.

Is there anything else I can do for you?



no

Have a good day!

Dr.Ajith left the conversation

4.CLOUD FUNCTION

Cloud Function gives developers access to Firebase and Google Cloud events, along with scalable computing power to run code in response to those events. While it's expected that Firebase applications will use Cloud Functions in unique ways to meet their unique requirements, typical use cases might fall into these areas:

- Notify users when something interesting happens.
- Perform Realtime Database sanitization and maintenance.
- Execute intensive tasks in the cloud instead of in your app.
- Integrate with third-party services and APIs.

Here, we use the cloud function to execute intensive tasks in the cloud instead of the app. Developers can take advantage of Cloud Functions to offload to the Google cloud resource-intensive work (heavy CPU or networking) which wouldn't be practical to run on a user's device.

In our application we use the cloud function as follows:

1. A function triggers when an intent is recognized.
2. It takes the parameter (symptom) defined, as input.
3. Function has authorisation to access the database.
4. It maps the symptoms to departments and gives output.

Webhook request :

- When an intent with fulfillment enabled is matched, Dialogflow will make an HTTP POST request to your webhook with a JSON object containing information about the matched intent.
- After receiving a request, the webhook can perform any required tasks. For example, the webhook might use information from the request to look up a product in a database or place an order.
- Finally, your webhook should respond back with instructions for what Dialog flow should do next.

Webhook response

Once requested, your webhook should provide a response. In the response, you can specify the following things:

- The response that Dialogflow returns to the user.
- Updates to the contexts attached to the conversation.
- A followup event name that would cause another intent to be invoked.
- An arbitrary payload that can be sent to the original Dialog flow caller.

5.FIREBASE

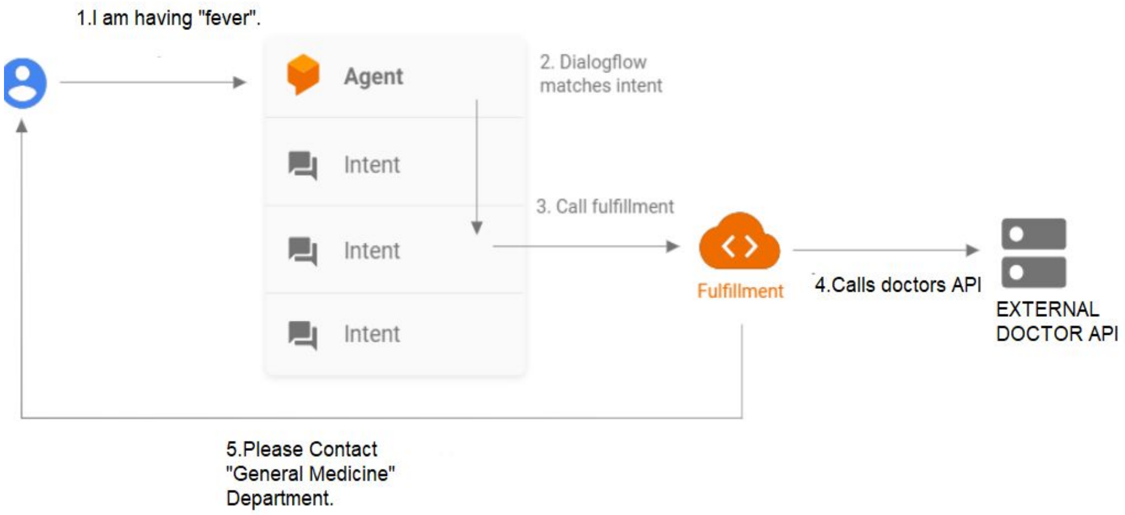
The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real time. Firebase provides a realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored in Firebase's cloud.

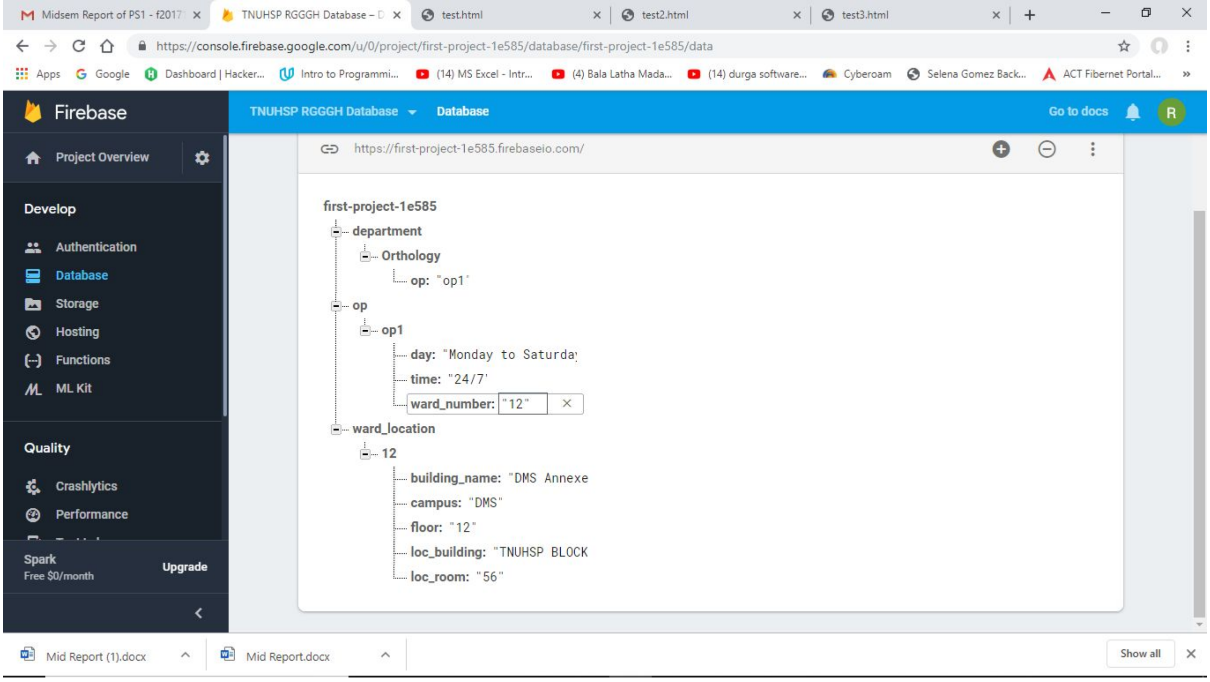
We particularly chose to use the Firebase Database for our project because of the ease with which it can integrate and interact with web and voice-based applications, and the integration process involved is also very simple.

From fulfillment inline editor, we permit the cloud function to use the database by replacing https with ws. For example, if database URL is "https://firstassistant-esmape.firebaseio.com/" is replaced with "'ws://firstassistant-esmape.firebaseio.com/'".

Cloud Functions in Firebase lets you run backend code in response to events triggered by Firebase features, and also, HTTPS requests. The code you write is stored in Google's cloud and runs in a managed environment, and there's no need to manage and scale your own servers.

Shown below is the working of the app engine. The external doctor API is Firebase.





```
first-project-1e585
  department
    Orthology
      op: "op1"
  op
    op1
      day: "Monday to Saturday
      time: "24/7"
      ward_number: "12"
  ward_location
    12
      building_name: "DMS Annexe
      campus: "DMS"
      floor: "12"
      loc_building: "TNUHSP BLOCK
      loc_room: "56"
```

The above picture displays how information is stored inside the database.

Our database involves three different levels of mapping:

1) Mapping the Ward number with its respective location
2) Mapping the OP with its Ward numbers
3) Mapping the Department with its OP’s

The modification of data becomes easier with three levels of mapping. The information is stored in a key and value type of format (Key: Value) called the JSON format. Such a format has been used to make it easier to access information.

6.CREATION OF HTML WEBFORMS

Every website consists of two parts - One, the front end, the part with which the user can interact with the website. Two, the back end, which consists of a server,

an application, and a database.

The HTML webforms we have designed take the data to perform three levels of data mapping -

1) **Ward Number & Location**

   Example: Ward XYZ

   The fields would be:

   - Campus
   - Building Name & location
   - Floor number of the Building
   - Ward location in the Floor

2) **OP or Ward with Ward Number and Time**

   The fields would be:

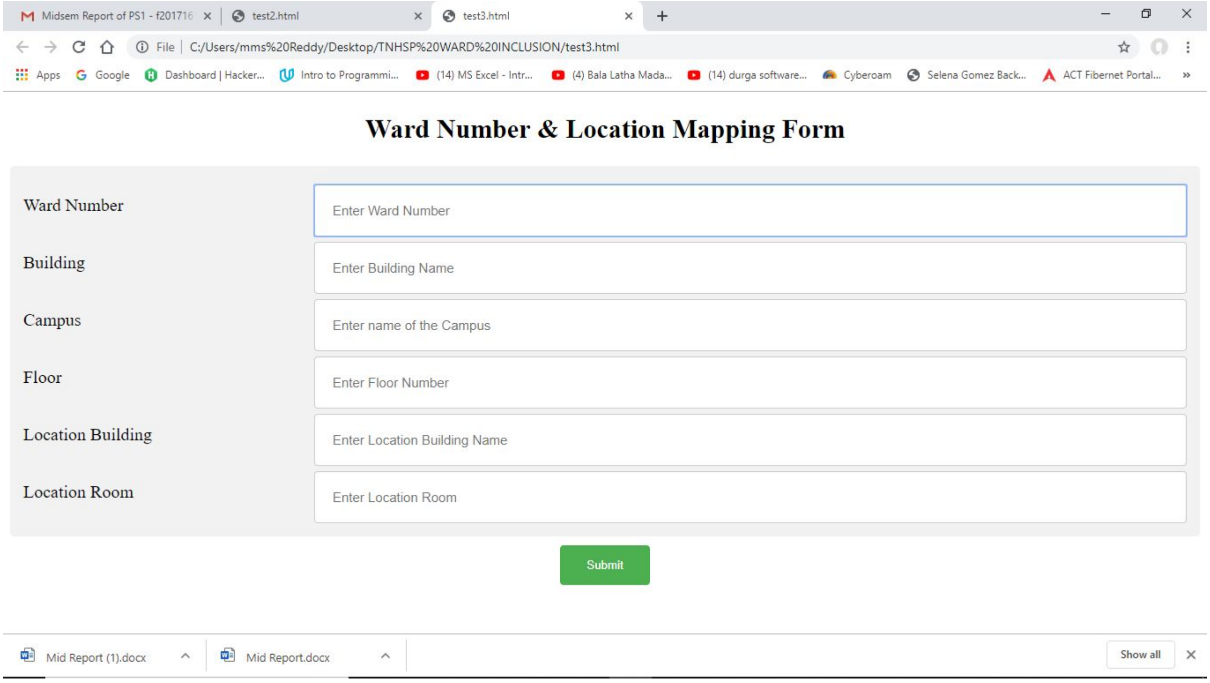
   - Ward Number,
   - Operating days,
   - Time,
   - OP

3) **Department with all its OP and Wards**

   The fields would be:

   - Department
   - OP

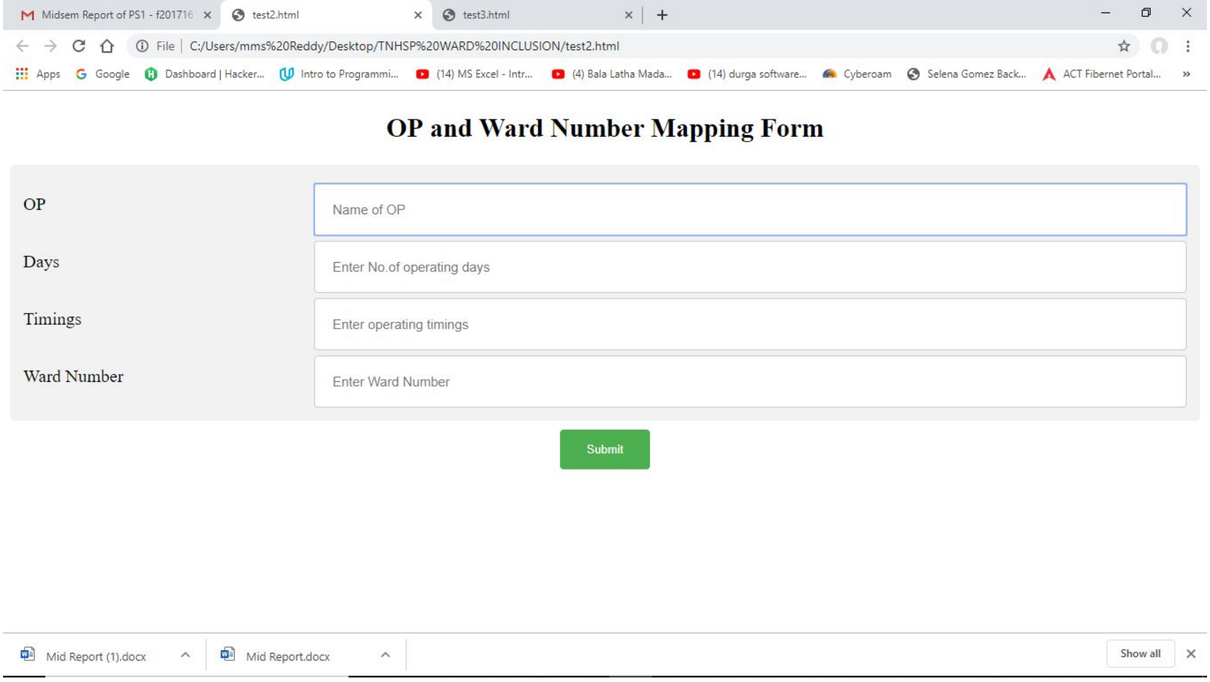
LEVEL 1



Ward Number & Location Mapping Form

Ward Number	Enter Ward Number
Building	Enter Building Name
Campus	Enter name of the Campus
Floor	Enter Floor Number
Location Building	Enter Location Building Name
Location Room	Enter Location Room

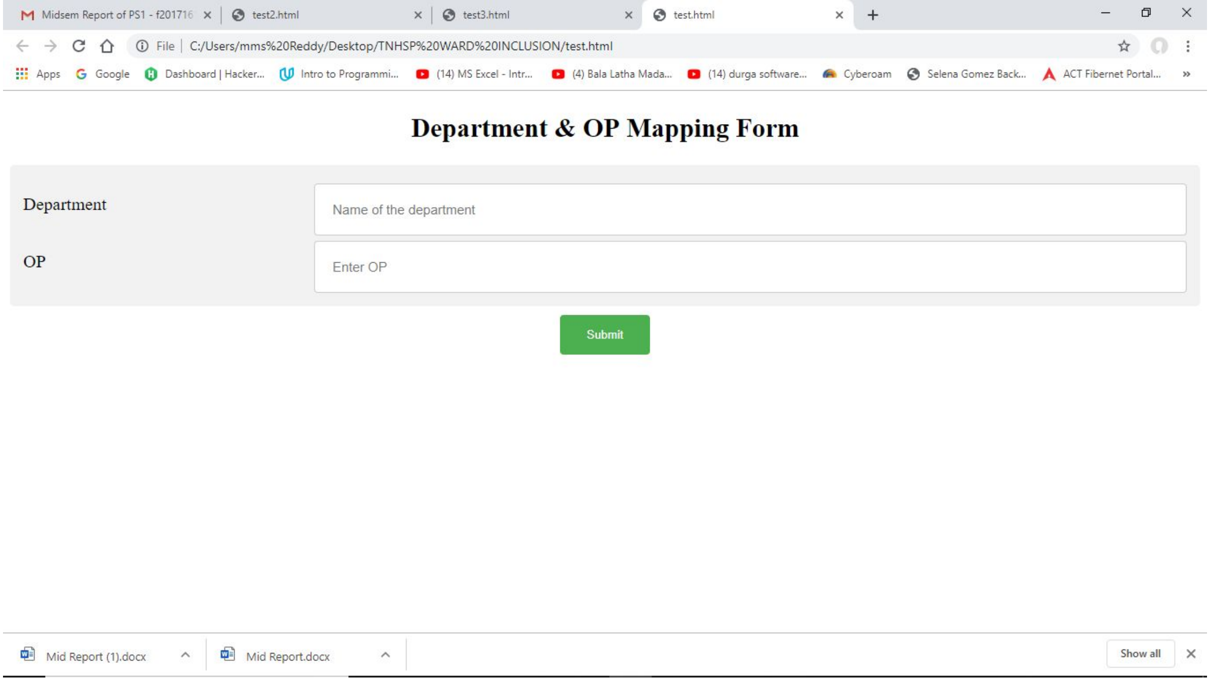
LEVEL 2



OP and Ward Number Mapping Form

OP	Name of OP
Days	Enter No.of operating days
Timings	Enter operating timings
Ward Number	Enter Ward Number

LEVEL3



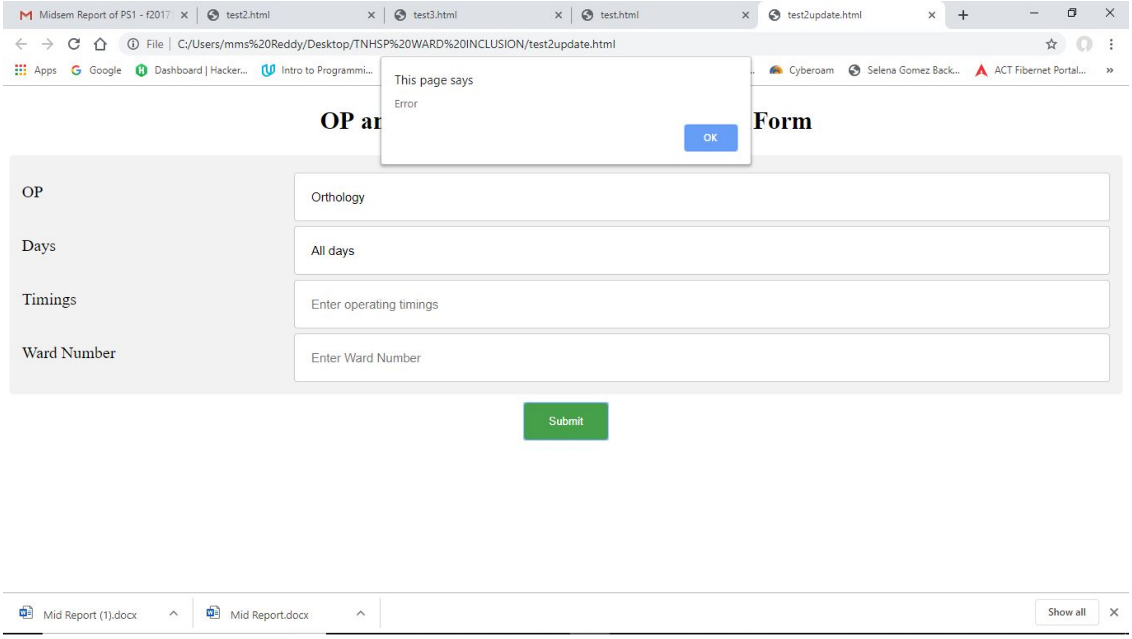
Department & OP Mapping Form

Department	Name of the department
OP	Enter OP

6.1 Forms used to update data

The forms have been programmed such that they can be submitted only when the admin fills up all the fields, not doing so will prompt an error message.

The below picture is testament to this.



This page says

Error

OP ar   Form

OP	Orthology
Days	All days
Timings	Enter operating timings
Ward Number	Enter Ward Number

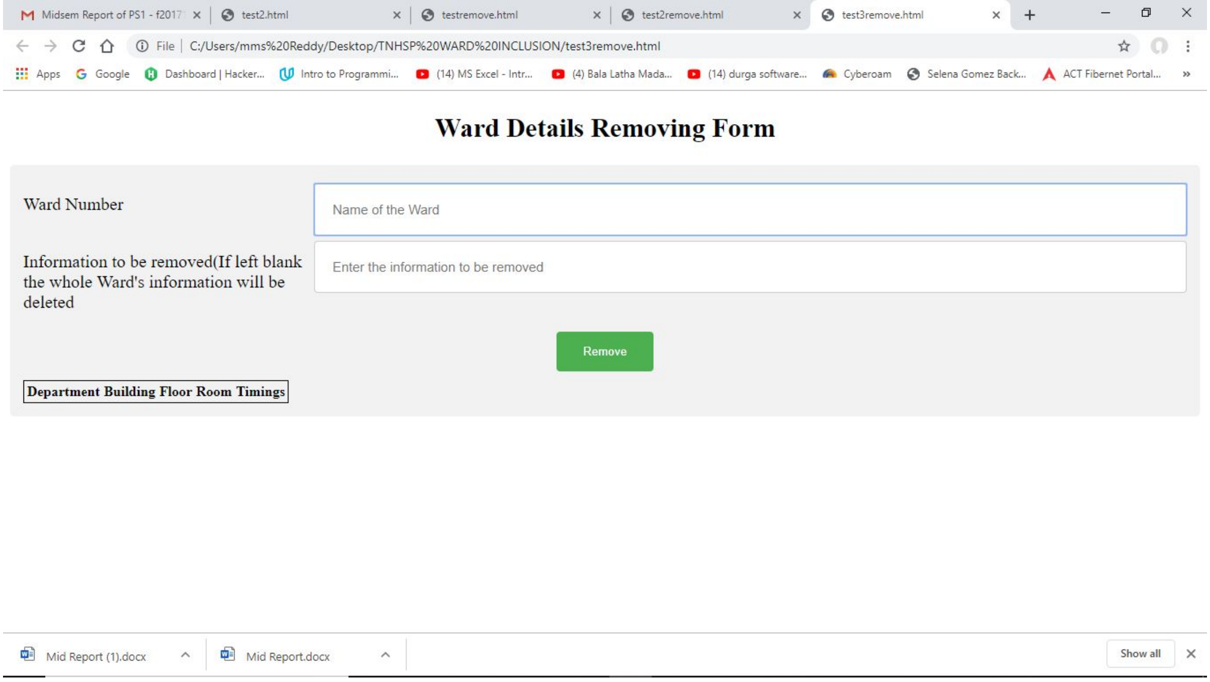
6.2 Forms used for removing data

There are three levels of removing data -

- Removing a particular field of its location or the entire ward
- Removing a particular field of its operating details or the entire OP
- Removing the entire department

The picture below shows the working -

LEVEL 1



Ward Details Removing Form

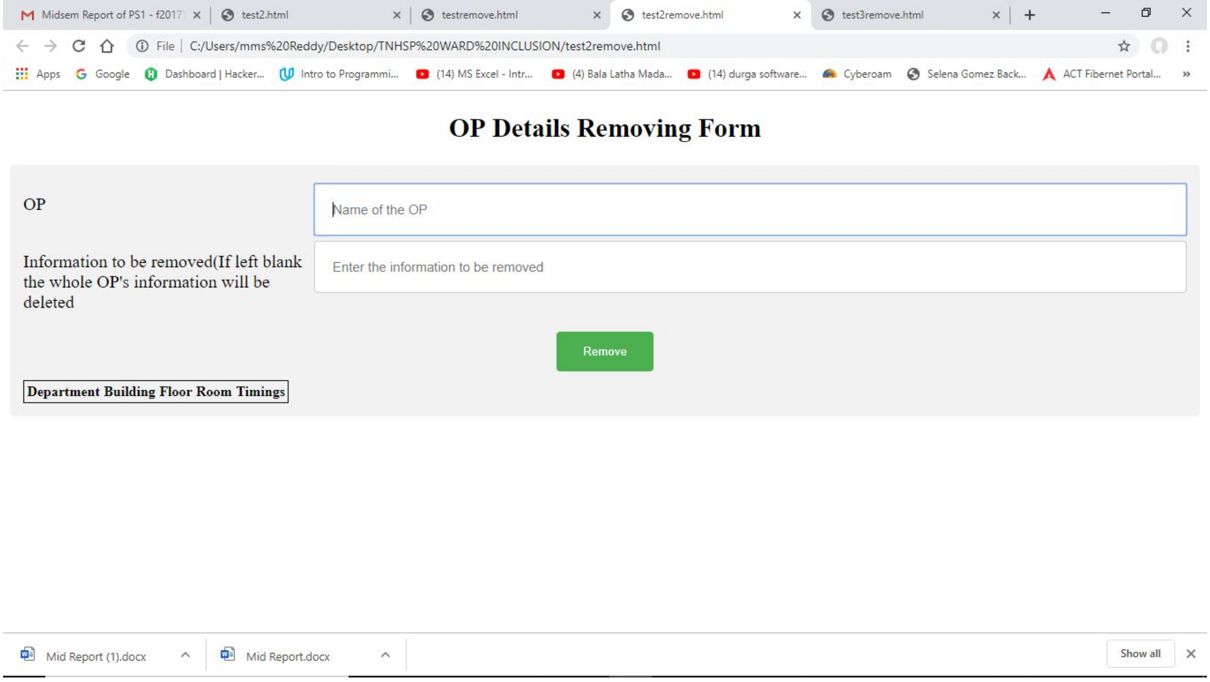
Ward Number: Name of the Ward

Information to be removed(If left blank the whole Ward's information will be deleted: Enter the information to be removed

Remove

Department Building Floor Room Timings

LEVEL 2



OP Details Removing Form

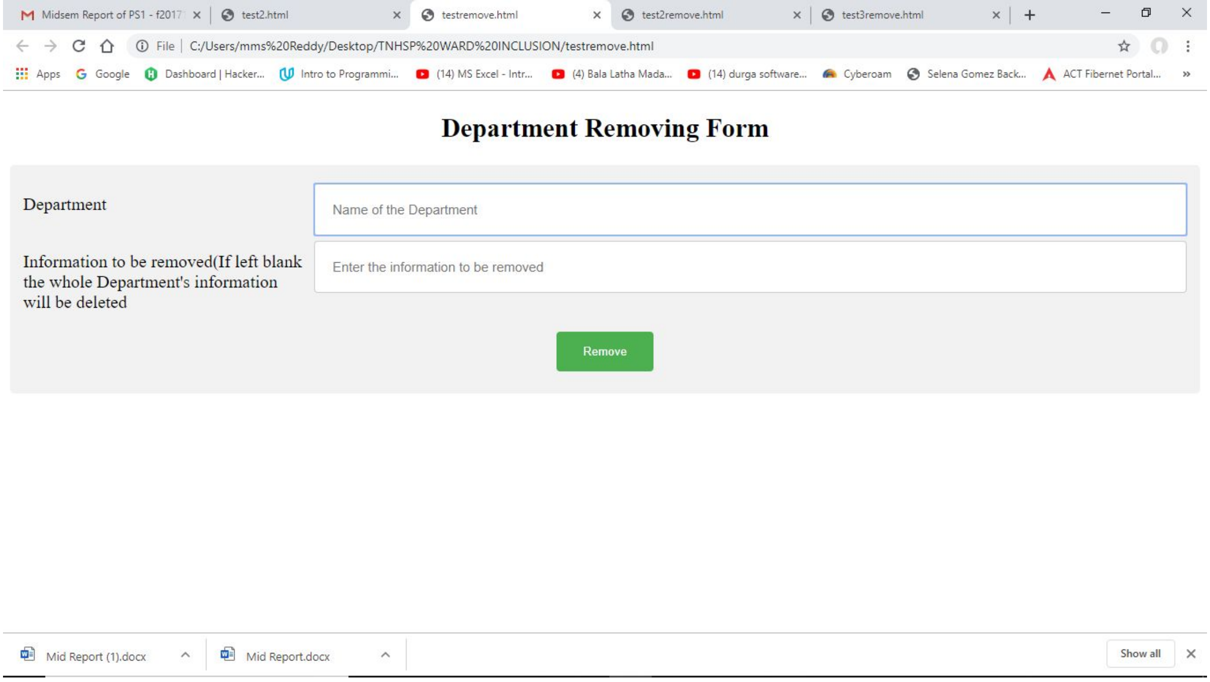
OP: Name of the OP

Information to be removed(If left blank the whole OP's information will be deleted: Enter the information to be removed

Remove

Department Building Floor Room Timings

LEVEL 3



Department Removing Form

Department: Name of the Department

Information to be removed(If left blank the whole Department's information will be deleted: Enter the information to be removed

Remove

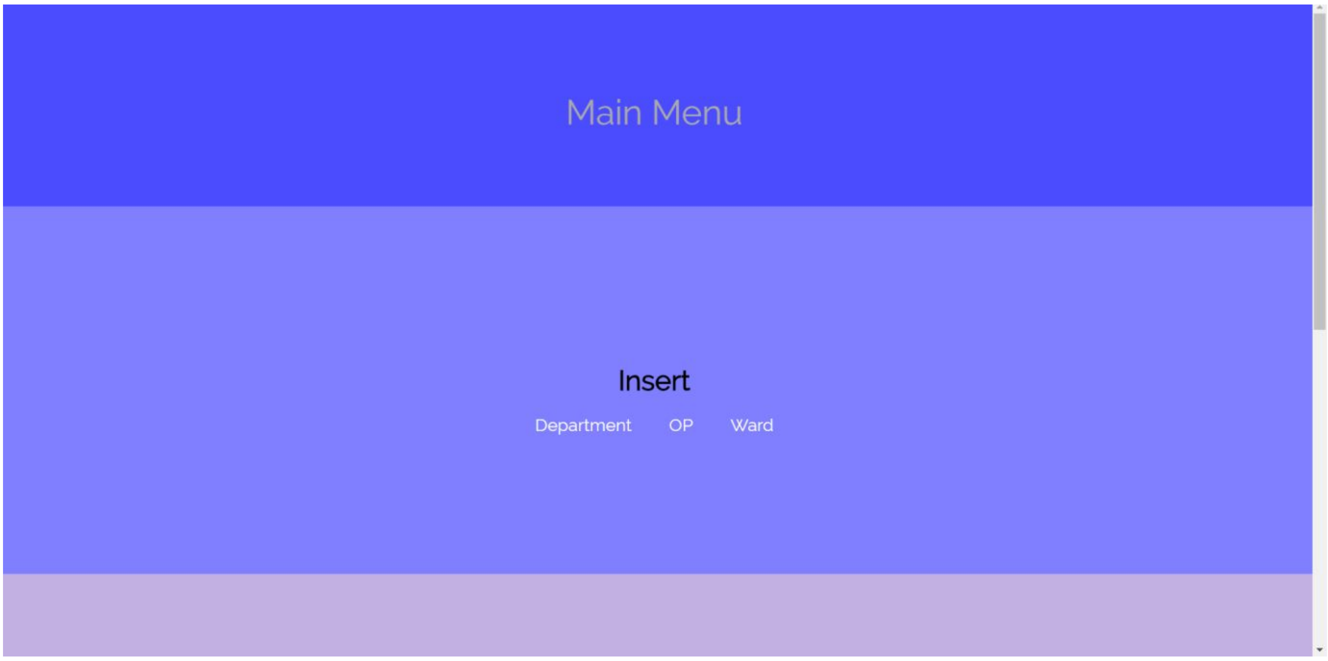
6.3 Softwares used for Front End Development

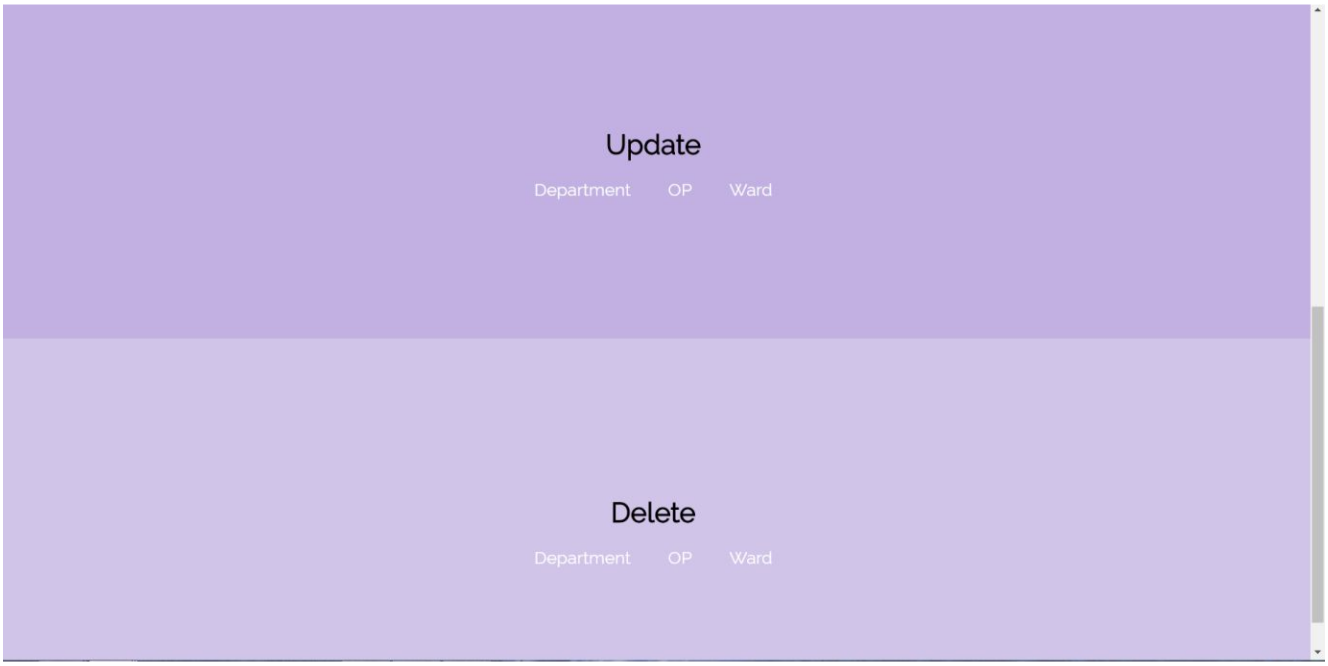
- HTML to display the contents of the website.
- HTML forms to receive and store inputs.
- CSS to style the webpage.

7.Creation of admin dashboard

The admin dashboard provides an easy interface for the admin for insertion, and deletion of information in the database. As shown in the images below it provides separate options for insertion and deletion.

HTML was used to give the dashboard its basic structure and links to the forms, and CSS was used to style the website.





8.LINKING DATABASE WITH WEBSITE

This consists of the backend of the designed website. The softwares being used are:

- ***JavaScript:***

  The functions of each button in the website is defined using JavaScript. It is also used to assign a particular function to a particular button. JavaScript finds its usage in establishing a connection between the Firebase Database and the website as well.

- ***Firebase Database:***

  A unique Database with a unique ID is used to store the information.

The detailed description of the procedure to link up the website to the database is as follows:

- First, a form has to be submitted by defining new variables and assigning them with the values of the inputs as provided by the user. All these operations can be executed at once by defining them inside a function.
- Second, we need to point our website to the firebase database in which the data needs to be stored. This is done by inserting a snippet of code into our HTML file or JavaScript file consisting of the unique id assigned to the database.
- Third, to save the data onto the firebase database we have to define the JSON format in which the variables should go into the database.
- Last, to read the database values and display them on the website, a table along with Jquery can be included in the website.

9.CONCLUSION

Post development plans/ideas -

- In case of any problem with the application, it has been built such that we can interact, debug and update the application from any part of the world using internet.
- The compatibility for new languages can be added to the application.
- The functionality of the application can be expanded to different use cases such as laboratory timings and scheduling appointments etc.
- It can also be used to update the user with patient details, both medical and logistical in the app, this will ease the nerves of anxious relatives unable to come in contact with the patient. This functionality will not come with additional workload for the doctors, as the status of the patient is automatically updated from a digital checklist which will be implemented as a substitute for the normal checklist, the hospital uses.

10. SOURCES

1. https://dialogflow.com/docs/
2. https://dev.to/apcelent/how-to-create-a-facebook-messenger-bot-with-python-flask-50j2
3. https://support.microsoft.com/en-us/instantanswers/557b5e0e-0eb0-44db-87d6-5e5db6f9c5b0/cortana-s-regions-and-languages
4. https://www.google.com/intl/en/chrome/demos/speech.html
5. https://w3c.github.io/speech-api/speechapi.html
6. https://support.apple.com/en-us/HT204389
7. https://developer.amazon.com/docs/custom-skills/develop-skills-in-multiple-languages.html
8. https://dialogflow.com/docs/integrations