Project report

Smartinternz-Jatin mundra

|  |  |  |
| --- | --- | --- |
| Internship Title | : | Intelligent Customer Help Desk with Smart Document Understanding - SB14928 |
| Project ID | : | SPS\_PRO\_99 |
| Project Title | : | Intelligent Customer Help Desk with Smart Document Understanding |

1. INTRODUCTION
   1. Overview

This project is based on a solving major technical issues related to customer care helpdesks with the use of an AI implemented chat-bot. This chat-bot is structured and made possible with the help of various IBM cloud services in order to give the most relevant answers that can address the customer queries clearly.

In this project, a chat-bot is built that addresses the queries related to an MI phone.

* 1. Purpose

The typical customer care chat-bot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. When a question falls outside of the scope of the pre-determined question set, the option is typically to tell the customer the question isn’t valid or offer to speak to a real person.  
In this project, there will be another option. If the customer question is about the operation of a device, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded with the device’s owner’s manual. So now, instead of “Would you like to speak to a customer representative?” we can return relevant sections of the owner’s manual to help solve our customers’ problems.

1. LITERATURE SURVEY
   1. Existing problem

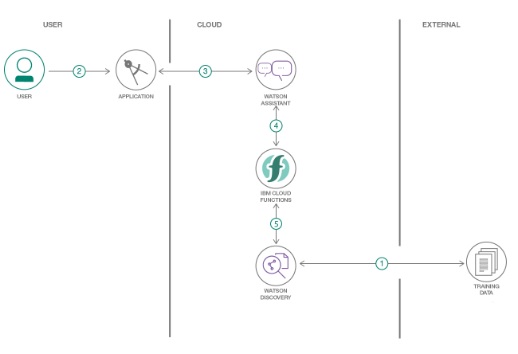
It all starts when you reach any customer care website to get an answer to your device related queries. After a bit of searching, you find a chat-bot that prompts you to enter a query. After a while of sharing your technical issues with the bot, when the query lands beyond the scope of the backend data of the chat-bot which mostly is a technical one, it eventually transfers the conversation to a customer care executive to whom you’ll have to describe your query all over again.

* 1. Proposed solution

This AI based chat-bot is integrated with Smart document understanding feature of IBM Watson Discovery which chooses the most relevant answer with a bit of machine learning.

Now that we have the relevant answer, to have the chat-bot use it in conversations IBM Watson Assistant comes into picture where the conversation data is trained. Therefore, with a few services in addition to this, a chat-bot is made to address the user’s queries with a better relevance.

1. THEORETICAL ANALYSIS
   1. Block diagram



* 1. Hardware/Software designing

As mentioned, we’ve used various IBM cloud services. They’re as follows:

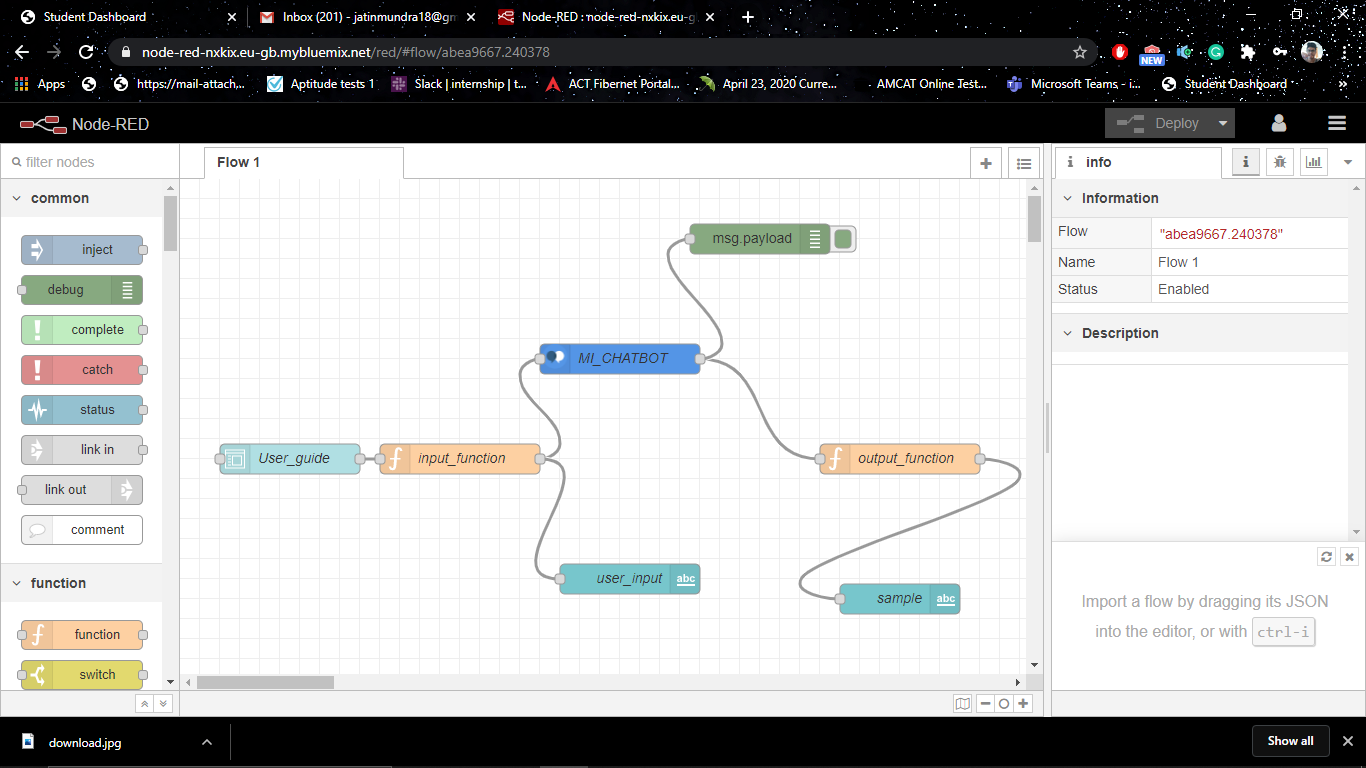
* IBM Watson Assistant
  + This service is used to create and train a vital organ of the chat-bot that is to detect the type of user inputs like greetings, gratitude, queries etc., and answer appropriately.
  + We create different types of intents and entities in order to detect the user input and then set it up with the appropriate answer in the Dialog node.
  + In order to answer a technical query with the trained data on Watson discovery api, webhooks are used which are linked with the dialog node.
  + Finally, we train the conversational skills of the bot by giving it several experimental inputs and observing its outputs. We can train the bot to let it know whether a certain type of query refers to a certain intent/entity. This improves the conversational skills of our chat-bot.
* IBM Watson Discovery
  + We use this service to train the answerable data with help of its Smart Document Understanding feature.
  + The data (for ex: user manual) is uploaded to the discovery dashboard. We now highlight the various aspects of the file such as title, subtitle, headers etc., and set a breaking point as to which set of data could be relevant.
  + We can check the understanding of the bot by asking it some questions.
  + The Watson discovery can now be integrated with Watson Assistant with the help of **IBM cloud function**: **Action**.

* Node-Red Application
  + The final step of our project is to integrate each and everything used to interact with the user and present it on a Web-based dashboard.
  + We use a form node to prompt the user for a query and interact with assistant to get the answer and present it to the user.
  + The Node-Red flow is given in the flowchart section below.

1. EXPERIMENTAL INVESTIGATIONS

While giving the experimental inputs, it was found that the primary bot needed to be modified in order to provide a descriptive result with respect to the user’s need. Typically, for the technical query, we use the JSON text “<?passages[0].passage\_text?>” to print the first paragraph of the answer suggested by Watson discovery. But, this isn’t always relevant to the user’s query. Therefore, after a few additions to the dialog node and using “<?results[0].text?>”, if in an event where user doesn’t understand the answer given by the bot, he/she can ask for an elaborated description via keywords such as ‘explain’, ‘brief’ or ‘m0re’. This will provide them with a detailed information to their query.

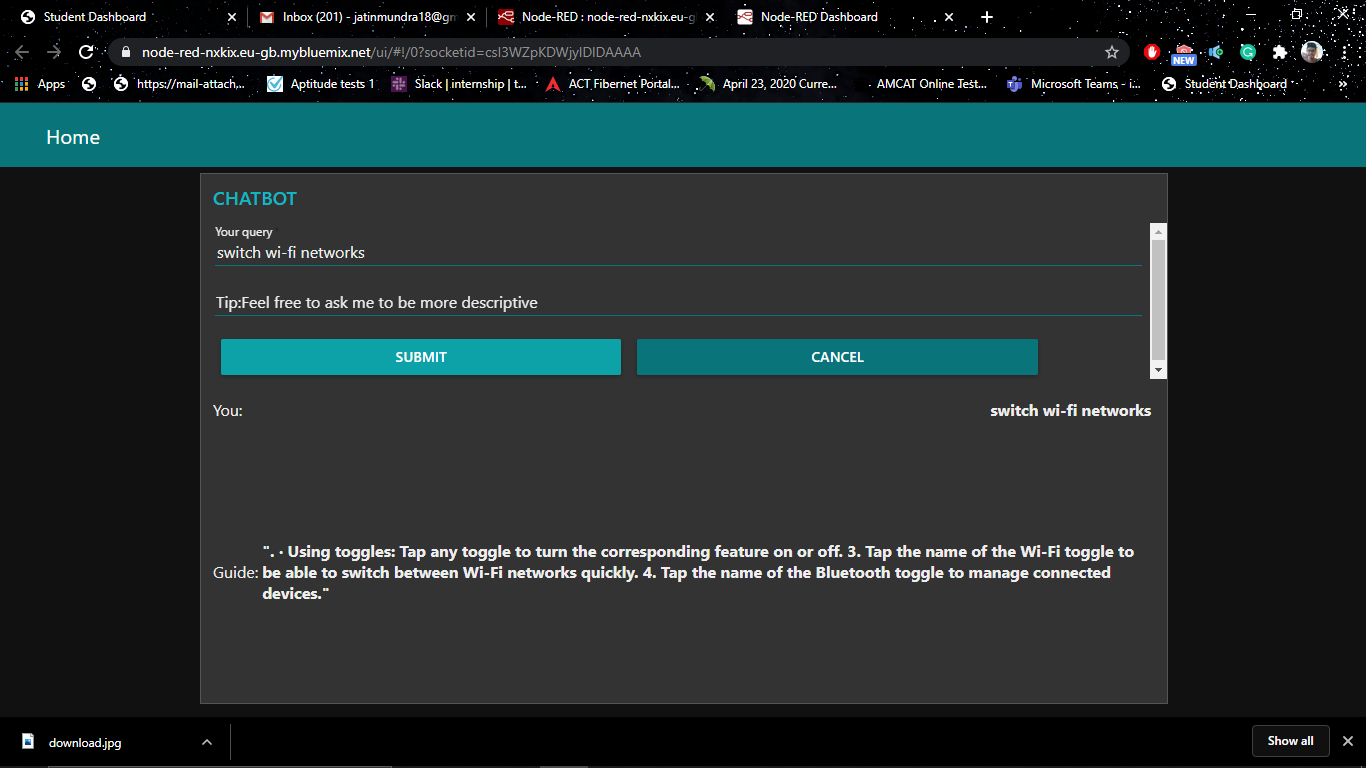
1. FLOWCHART (Node-RED)



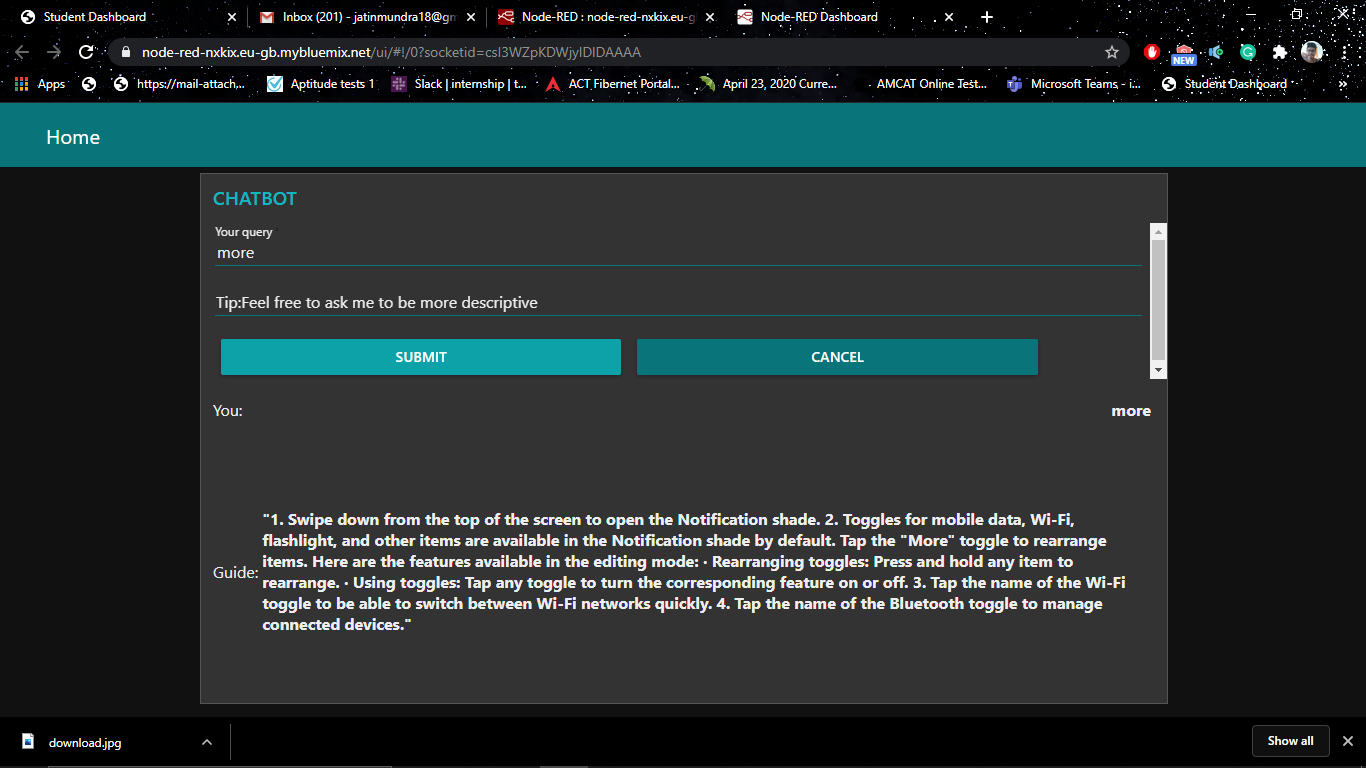
1. RESULT

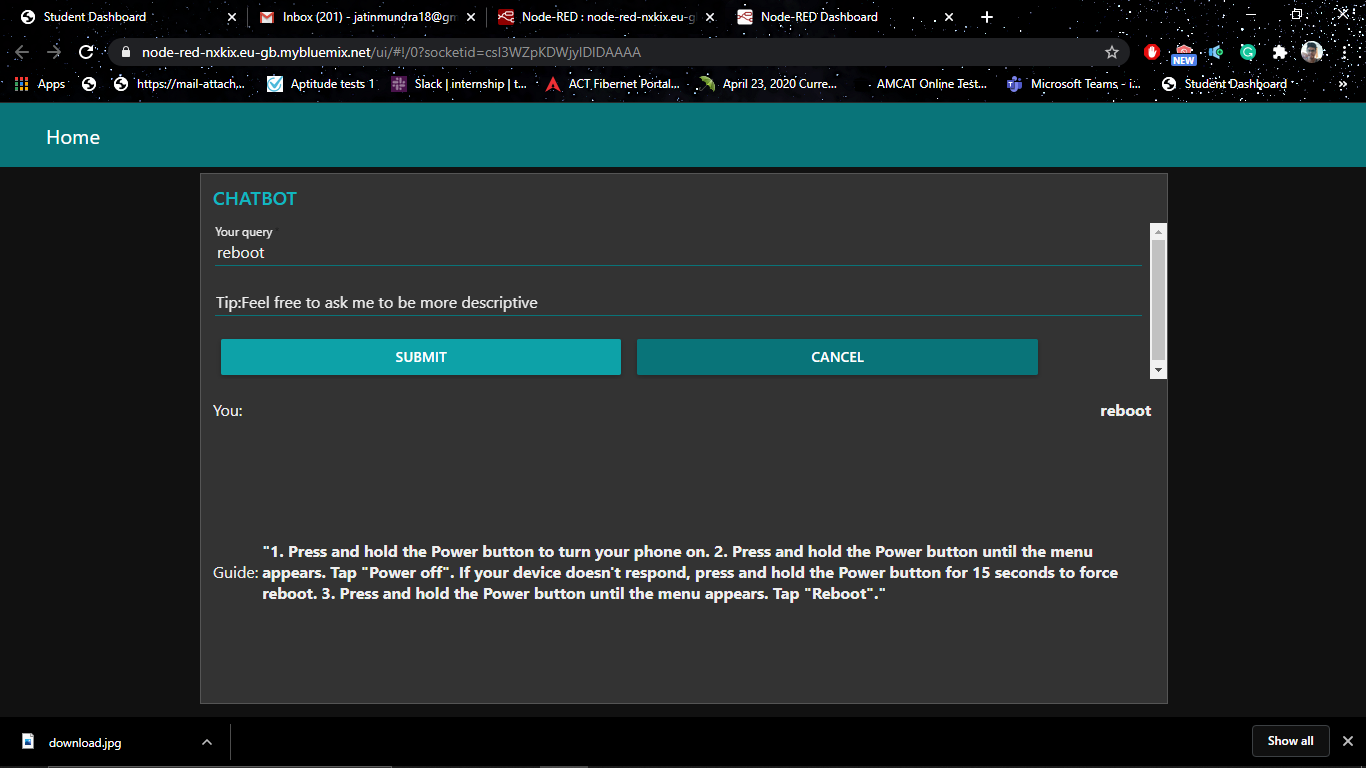
The following queries were answered by the bot when prompted by the user.

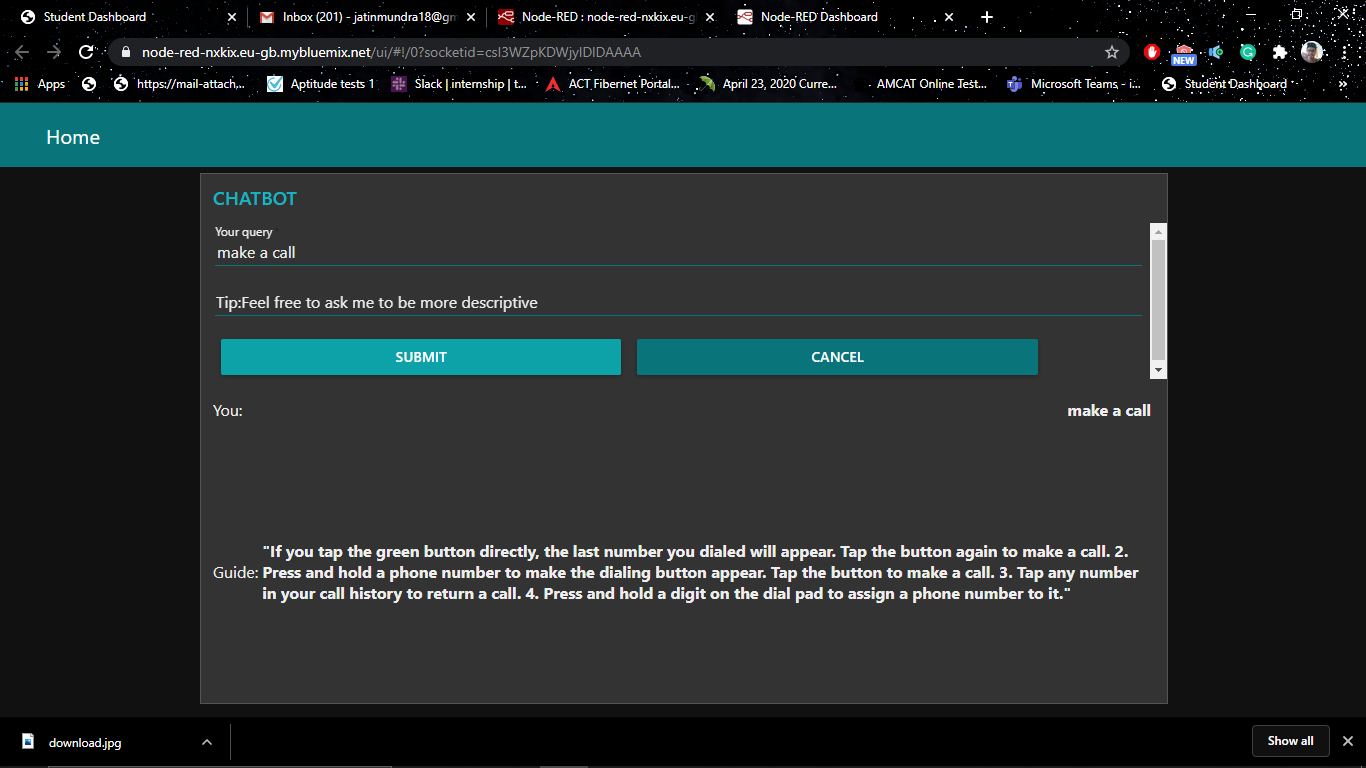
1. To toggle Wi-Fi networks.



After using the ‘more’ keyword:



1. To reboot device. 
2. Make a call.



1. ADVANTAGES AND DISADVANTAGES
   1. Advantages

* An AI powered chat-bot can provide much relevant answers when compared with an ordinary chat-bot.
* This can save a lot of company time, human work and money.
* With more training, it will also be able to detect complex queries and provide appropriate solutions.
* A user doesn’t have to go through pages and pages of the device’s user manual to address his/her query. They can just type it in this chat box and they’ll eventually have their result.
* Spelling errors/typos will be automatically ignored and the correct word will be considered.
  1. Disadvantages
* It isn’t guaranteed that a right answer might come up every single time.
* It’ll require maintenance as new data will be needed to be fed in.

1. APPLICATIONS

An AI based chat-bot can be used in various domains where a company tries to reduce its human work allotment for customer care. A few applications include:

* Any university’s website: To guide the potential candidate by providing the information regarding fees, placements etc.
* Any technology oriented company’s helpdesk to provide the relevant technical support to the user’s queries.
* Any mobile application to guide a user through the same.

1. CONCLUSION

To conclude this project, I would like to thank Smartbridge for giving me this wonderful opportunity to work in an AI based project which indeed has expanded my skill set by working with various IBM services to build up this chat-bot.

When it comes to the conclusion of this project, we all can safely say that an Artificial Intelligence backed up chat-bot is a lot helpful than a conventional one.

1. FUTURE SCOPE

With a few more integrations with various IBM services and training the data to bring out the most relevant answers, we can get this bot to have almost Human-like conversations. Thus, making the user feel as if he/she were speaking with an actual person on the other end. This will increase customer satisfaction as we indeed will be providing a cutting edge service to their queries.

IBM cloud functions like trigger can be used to fetch the answers directed by the chat-bot in a swift manner.

We can use Watson search skill to browse through various resources if in any case that the answer goes out of the scope of the bot.

1. BIBLIOGRAPHY

YouTube Link to the video: <https://youtu.be/s89Q-9NxlqQ>

GitHub repository: [https://github.com/SmartPracticeschool/llSPS-INT-1227-Intelligent-Customer-Help-Desk-with-Smart-Document-Understanding](https://github.com/SmartPracticeschool/llSPS-INT-1227-Intelligent-Customer-Help-Desk-with-Smart-Document-Understanding%20)

Link to NodeRED UI: <https://node-red-nxkix.eu-gb.mybluemix.net/ui>

Link to IBM cloud: <https://cloud.ibm.com/login>

All the source codes for NodeRED flow, Watson assistant, cloud function have been uploaded to the github repository.