

Project Report

Name: *Avani Jain(avani.jain.cs.2016@miet.ac.in)*

Title: *Intelligent customer help desk with smart document understanding.*

Category: *Artificial Intelligence*

Table of Contents

1. INTRODUCTION

1.1 Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing Problem

2.2 Proposed Solution

3. THEORETICAL ANALYSIS

3.1 Block Diagram

3.2 Hardware/Software Designing

4. EXPERIMENTAL INVESTIGATIONS

5. FLOWCHART

6. RESULT

7. ADVANTAGES AND DISADVANTAGES

8. APPLICATIONS

9. CONCLUSION

10. FUTURE SCOPE

11. BIBLIOGRAPHY

APPENDIX

A. Source Code

1. INTRODUCTION

1.1 Overview:

The project is about building a chatbot that answers customer's queries by using smart document understanding as its base. The application affects multiple watson AI services such as Discovery, Assistant, Cloud Function and Node Red. Throughout the project, Watson will be able to integrate services and to build interactive information retrieval systems with Discovery and assistant.sistant. The requirements of the project are depicted below:

- **Overall Project Requirements:** IBM Cloud, Python, IBM Watson.
- **Technical Requirements:** AI, Watson AI, Python, ML.
- **Functional Requirements:** IBM Cloud
- **Software Requirements:** Watson Assistant, Watson Discovery.
- **Project Deliverables:** SmartInternz Internship 2020.
- **Project Team:** Avani Jain
- **Project Duration:** 29 Days.

1.2 Purpose:

The earlier chatbot used for Customer Care was able to answer simple questions like locations, timings, directions, and other such things. If a question is asked that falls outside of the scope of pre- specified query set, the solution is to tell the customer about the validity of the query asked or make an offer to talk to the real person.

But in this, the alternative is if the customer queries about the working of the device then the application matches the query into the Watson discovery that has been pre-loaded with the document like user guide. Now instead of talking to a real person most of the queries can be answered by the chatbot.

The most important feature of this project is it will use Watson Discovery service for Smart Document Understanding that is used to train the bot and its basic functioning is it improves answer by the questions.

The following is our scope of work-

- First, we need to create a customer care dialog skill in the Watson Assistant.
- Then using the smart document understanding for building an enhanced collection in the Watson Discovery.
- Then we will create an IBM Cloud Function that is used for posting queries to the

Watson Discovery.

- Lastly, we would build a web application by integrating all these services and then deploying the same on the IBM Cloud.

2. LITERATURE SURVEY

2.1 Existing problem:

The meaning of a chatbot is a service that allows a user to give input or a query and it will provide a relevant output and if the input is not from the pre-stored data in the chatbot then it returns the output like "I am unable to understand", "Try again", etc. And then it makes an offer to the user to talk to a real person. But in today's world no one has enough time for these question and answers and the company employees are not that much idle to respond to each and every query so to resolve this we would try to create a virtual agent within the chatbot that would be clarifying each and every doubt of the customer without making an offer to talk to a real agent. The main benefit is it would increase the speed and will reduce wastage of time.

2.2 Proposed solution:

To solve the above problem we will integrate a virtual agent in our chatbot so as to make him understand the asked queries by the users or customers. The agent would be trained by loading a data that would contain all answers or keywords to answer the queries of the users and for this purpose I used Watson discovery along with Assistant and Discovering on Node-RED.

3. THEORETICAL ANALYSIS

3.1 Block/Flow Diagram:

1. The document is glossed on Watson Discovery SDU.
2. The interaction of the user is done with the backend server using the app user interface. And on the frontend the chatbot engages with the user.
3. The dialog between the user and the backend server is done through Watson Assistant dialog skill.
4. When the user asks a query related to the product, then that is passed to the pre-defined IBM cloud function.
5. The cloud function is then used to analyze the query by asking the same in the Watson Discovery Service and provide the results.

The block diagram is below:

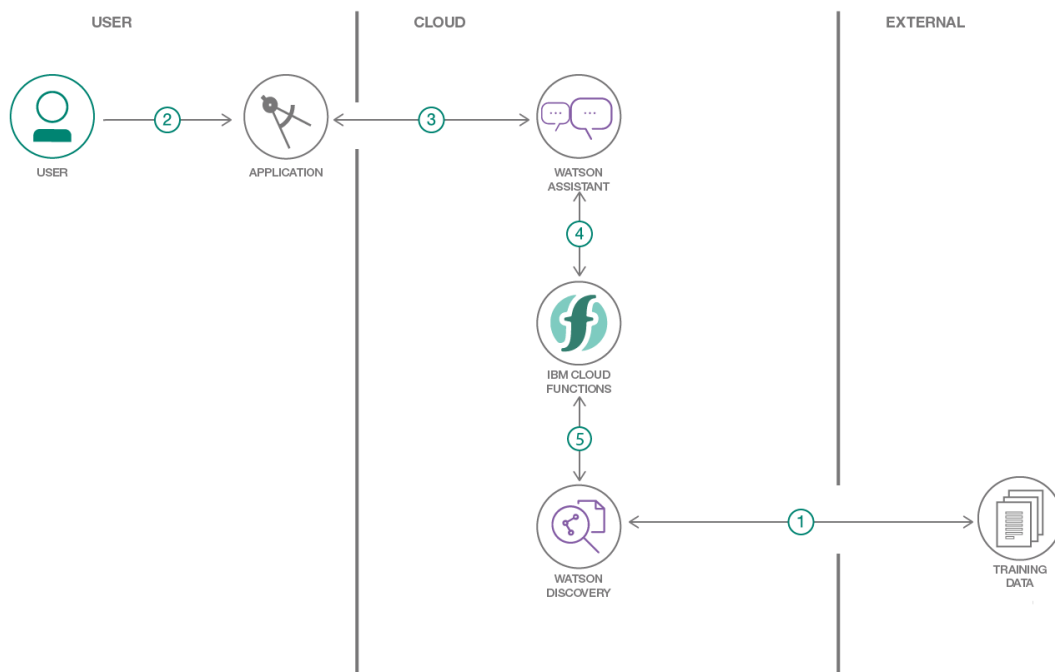


Figure: Block Diagram for a chatbot.

3.2 Hardware/Software designing:

1. Creating the necessary IBM Cloud Services.
2. Then configuring Watson Discovery.
3. Creating the IBM cloud function.
4. Then configuring the Watson Assistant.
5. Creating flow on Node-RED and configuring its node.
6. Deploying the Node-Red app and running it.

4.EXPERIMENTAL INVESTIGATIONS

1.Creating the following services on the IBM Cloud:

- Watson Discovery
- Watson Assistant
- IBM Cloud Function
- Node-Red

2. Configuring the Watson Discovery with the following:

- Login to IBM Cloud -->Catalog-->Services(right navigation bar-->choose discovery-->open it-->click on create.
- In the resources list open your service to configure it-->click on launch discovery-->upload your data-->give it a name-->click create-->on the next tab upload your data or the ecobee user guide from your system.
- After creation another tab opens-->click on build your won query at the bottom right.
- On the next page-->click on search for document-->enter your query(like how to turn on heater)-->then click run.
- You will observe that the result is a paragraph, now we need to modify to get required output.
- Click close-->Switch the folder icon--> Click your loaded data-->click configure-->then start configuring your data-->Choose the tabs from right like text,subtitle,footer and configure your document accordingly.
- After all this click on submit and then apply changes to collection-->upload the same document.
- Then again click on the folder icon-->manage field tab-->Toggle the buttons so that subtitle and text are green-->split the document by subtitle.
- Click apply changes to collection-->again upload same docuemnt.
- You can see the change in the number of documents on the page that comes up.
- Now again click build your query-->On the next page-->click on search for document-->enter your query(like how to turn on heater)-->then click run.
- You can observe that the result are a bit more relevant.

3. Create Watson Assistant:

- Now again goto catalog-->services-->IBM Assistant.
- On the next page click create, leave all the entries as it is.

- Click on launch watson assistant-->duplicate the tab-->go to catalog-->search node red app-->create a node red instance by clicking on create button.
- on the next page-->click deploy the app-->beside the api key-->click new-->press OK on the pop up box-->make the size to 256 MB by dragging the circle ahead-->select region as london-->click create.
- Now you have created an instance of the node red app.

4. Creating IBM Cloud function:

- Go to catalog and search function--> click on action in right bar-->click create.
- Enter action name-->click create-->enter the code.
- Go to parameter-->add to parameter-->they are mentioned as below:
 1. url
 - 2.apikey
 - 3.collection_id
 - 4.environment_id
 - 5.configuration_id
- After adding every parameter, click save.
- Copy the required parameter values from the other page.
- Go to code-->click save-->invoke.
- Go to endpoints-->check the enable as web action checkbox-->click save.
- Copy the url below it.
- Now go to assistant-->click launch assistant-->now following steps are for creating an assistant and its skill:
 - Go to the first icon-->click create assistant-->give a name to it like customer helpdesk-->click create.
 - Go to second icon-->click create skill-->select dialog skill-->create-->switch to use sample skill tab-->click on it-->click create an intent-->name it as product information or something like this.
 - Click create-->you will be prompted to provide some example to depict how the input should look like--> Enter examples like how to turn on heater, how to turn it off, how to set the time,etc.
 - Click add example after adding each example.
 - Now click on first icon --> click try it-->give a query-->result will show product information.
 - Now go to second icon-->dialog-->add a node below the what can I do node.
 - Enter the details for the node--> name it as ask about product-->and

select product information.

- Go to option in the left pane-->select webhooks-->paste the url you copied in the previous step and add .json to it.
- Again go to dialog tab-->customize the node you just created-->toggle the switch of webhooks-->click apply.
- Fill in the required values for parameters and the message it should respond with.
- Click try it--> enter your query-->you will see that the results are quite accurate.
- Now its time to deploy it with node red app.

5.Creating the node-red dashboard:

- From the initial page-->goto services-->select cloud foundry apps and click it to open it.
- On the next page that appears-->click visit app url-->keep on clicking next and finish the setup.
- Now click on go to your flow editor.
- Start creating the flow node red editor--> click the three lines -->manage palette-->install tab-->install node red dashboard.
- Then all the required nodes appear to you, select them and create a node red web dashboard.

Now all your services have been created. Copy the node-red app url and add ui to it and paste it in a new tab. Your bot appears, then you need to provide input to it and observe the output. Your chatbot is ready.

5. FLOWCHART

The flow of the chatbot is made by using the node red editor and it is mentioned as below:

```
[{"id":"8e260e36.af87b","type":"tab","label":"Flow
1","disabled":false,"info":""},{ "id":"e37a3244.eacb5","type":"ui_form","z":"8e260e3
6.af87b","name":"","label":"","group":"fd994811.737c88","order":1,"width":0,"height
":0,"options":[{"label":"Enter your
input","value":"text","type":"text","required":true,"rows":null}], "formValue":{"tex
t":""},"payload":"","submit":"submit","cancel":"cancel","topic":"","x":130,"y":200,
"wires":[["aceca082.e809a"]]}, {"id":"aceca082.e809a","type":"function","z":"8e260e3
6.af87b","name":"","func":"msg.payload=msg.payload.text;\nreturn
msg;","outputs":1,"noerr":0,"x":290,"y":140,"wires":[["aa762517.282f58","ba7f0471.1
c7df8"]]}, {"id":"fc0b5dbc.ba276","type":"function","z":"8e260e36.af87b","name":"","
func":"msg.payload=msg.payload.output.text[0];\nreturn
msg;","outputs":1,"noerr":0,"x":630,"y":140,"wires":[["93d455bf.8f39b8"]]}, {"id":"a
a762517.282f58","type":"watson-conversation-v1","z":"8e260e36.af87b","name":"Custom
er Care
","workspaceid":"1046b103-e6ea-472e-90d2-38c60692c15c","multiuser":false,"context":
true,"empty-payload":false,"service-endpoint":"https://api.eu-gb.assistant.watson.c
loud.ibm.com/instances/b8becd33-36fc-42bf-85d0-174e766675ec","timeout":"","optout-l
earning":false,"x":480,"y":80,"wires":[["9f8482ce.bcb88","fc0b5dbc.ba276"]]}, {"id":
"ba7f0471.1c7df8","type":"ui_text","z":"8e260e36.af87b","group":"fd994811.737c88","
order":2,"width":0,"height":0,"name":"","label":"You
:","format":"{{msg.payload}}","layout":"row-left","x":410,"y":240,"wires":[]}, {"id"
:"9f8482ce.bcb88","type":"debug","z":"8e260e36.af87b","name":"","active":true,"tosi
debar":true,"console":false,"tostatus":false,"complete":false,"x":830,"y":40,"wir
es":[]}, {"id":"93d455bf.8f39b8","type":"ui_text","z":"8e260e36.af87b","group":"fd99
4811.737c88","order":3,"width":0,"height":0,"name":"","label":"Bot","format":"{{msg
.payload}}","layout":"col-center","x":770,"y":240,"wires":[]}, {"id":"fd994811.737c8
8","type":"ui_group","z":"","name":"Chatbot","tab":"97d3d5de.249e58","order":1,"dis
p":true,"width":11,"collapse":false}, {"id":"97d3d5de.249e58","type":"ui_tab","z":"","
name":"Customer Care
Helpdesk","icon":"dashboard","disabled":false,"hidden":false}]}
```

The various nodes used were:

- Form
- text
- debug
- function

Each node was configured by writing the function within it.

The final node red dashboard looked like this as shown below:

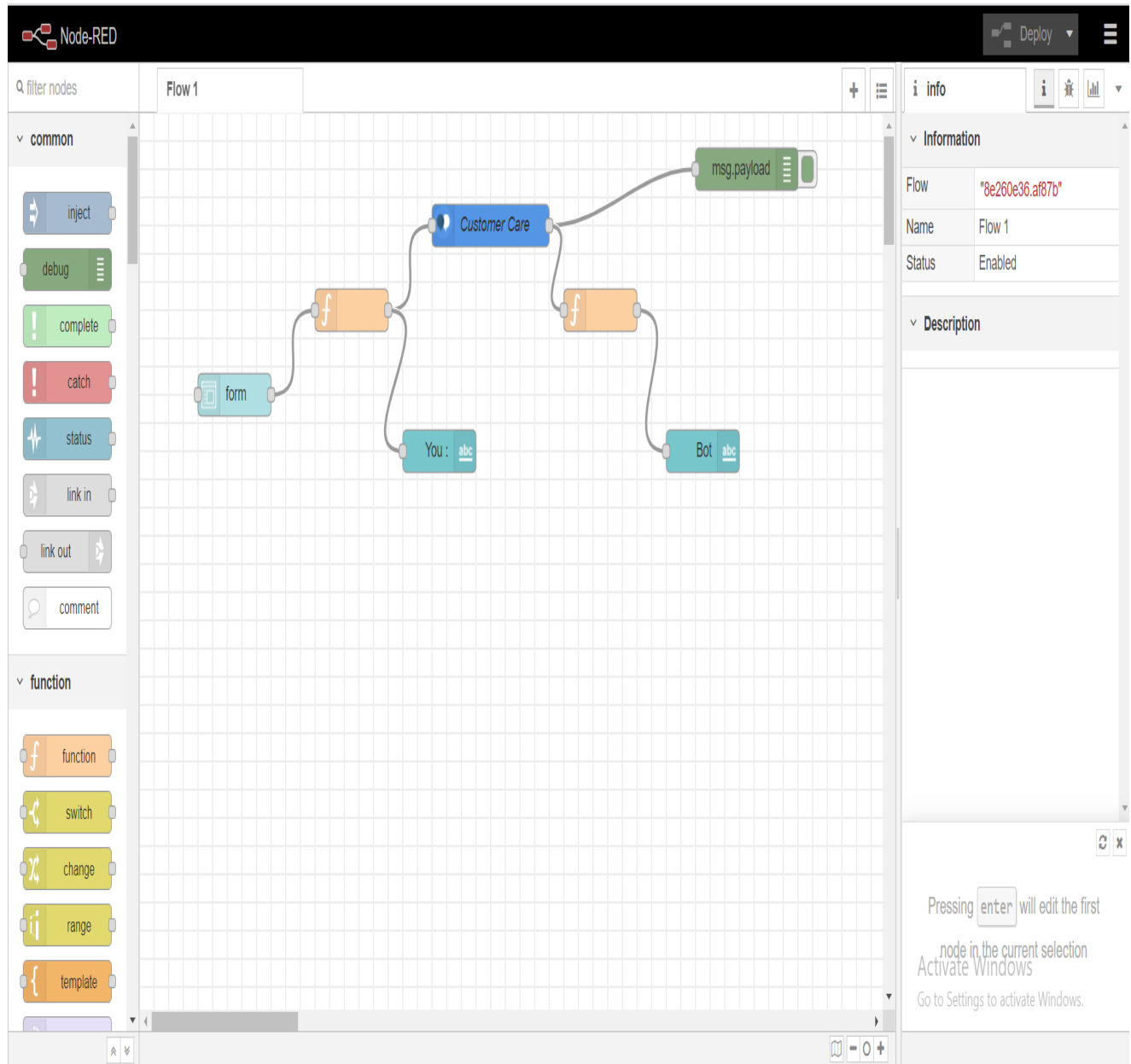


Fig: Node-red Dashboard

6. RESULT

The resulting product is a chatbot that takes user queries and gives the required response.

Here are some examples:

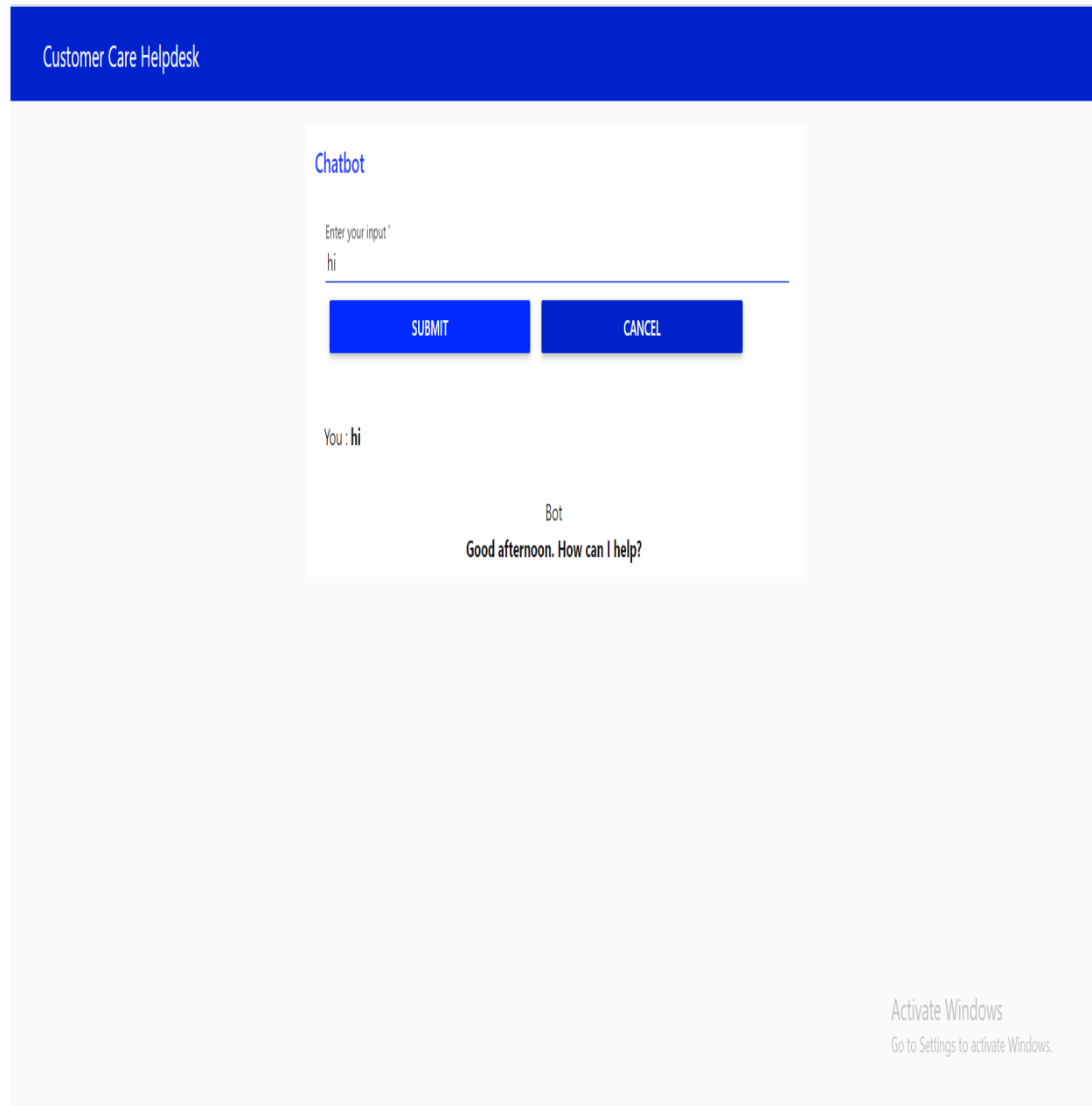


Fig: Output when user enter hi.

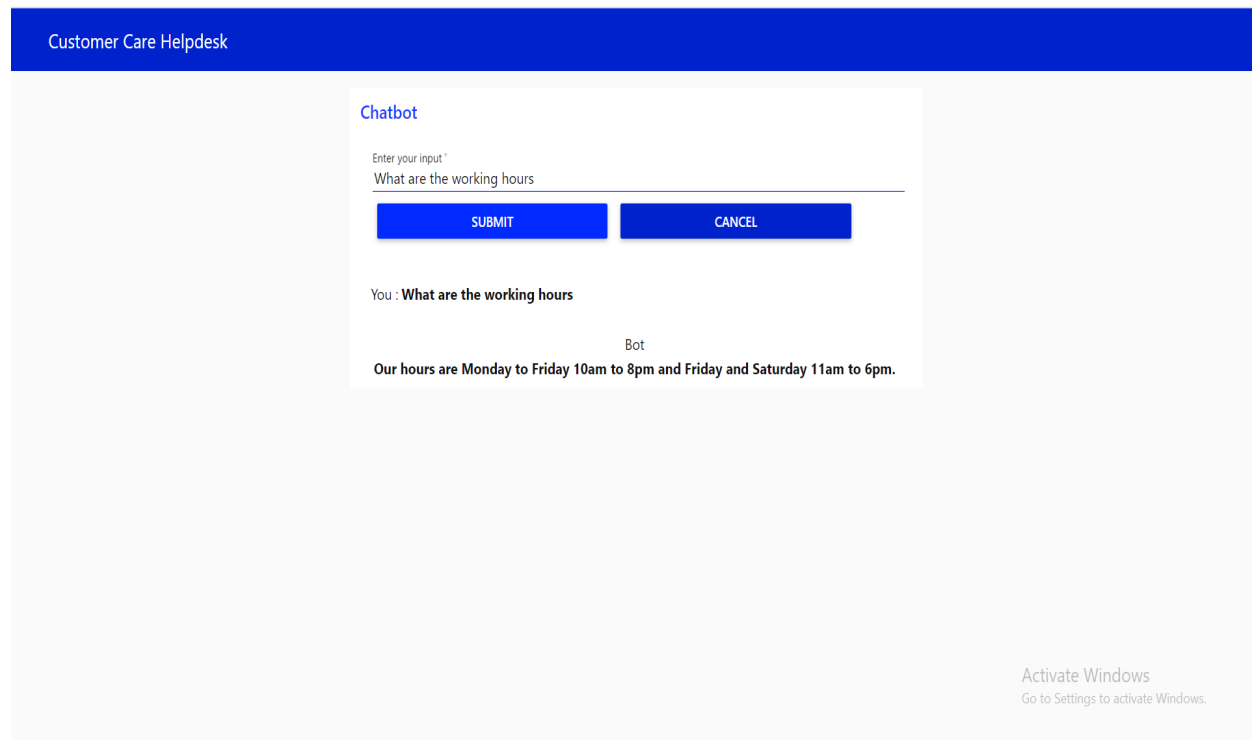


Fig: When user enter what are the working hours.

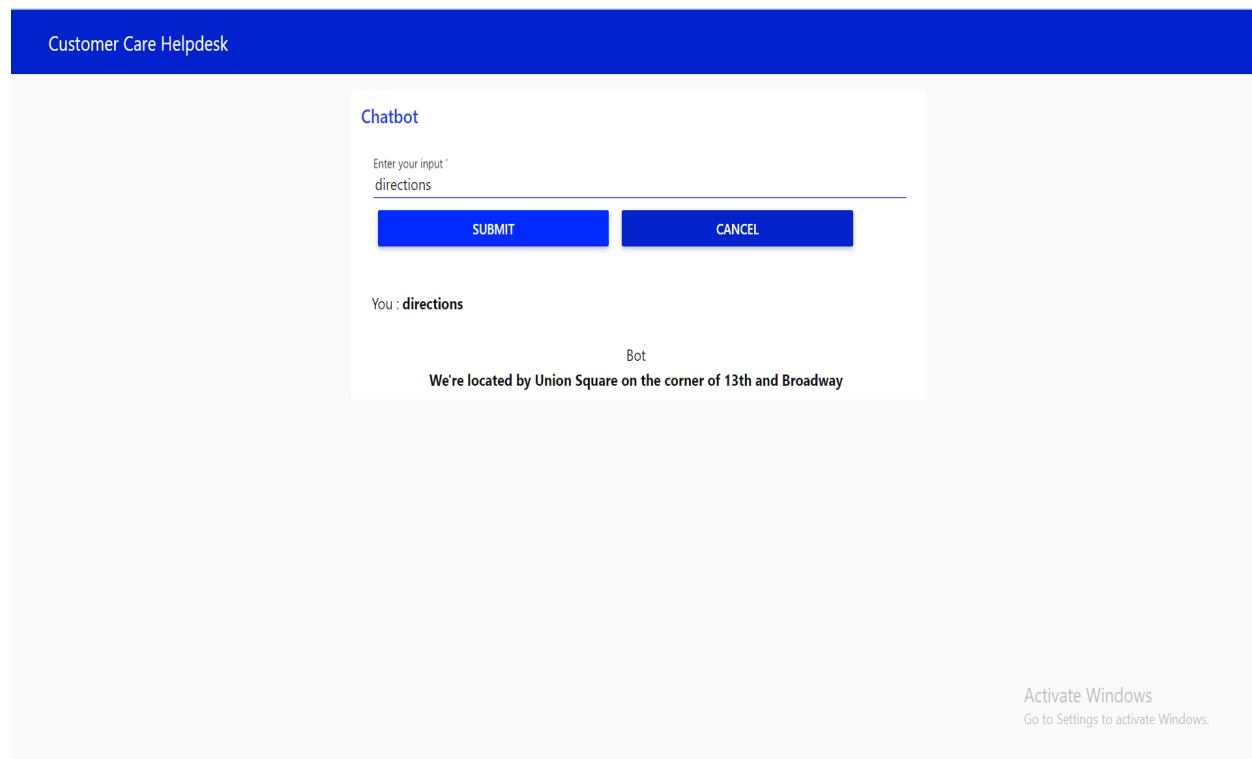


Fig: When input is directions.

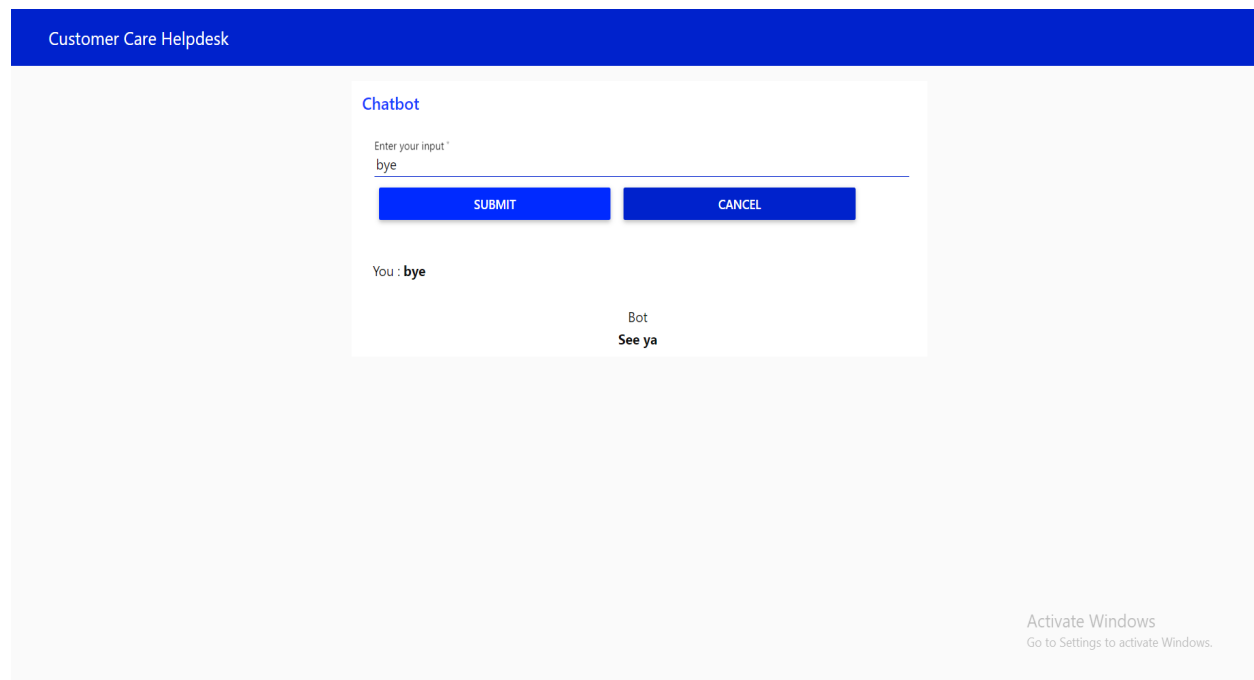


Fig: When user enter bye.

These are some of the input and output scenarios depicting the working of the chatbot.

7. ADVANTAGES AND DISADVANTAGES

Following are the advantages:

- These chatbots can be used by organizations to solve the common queries asked by the user or the customer.
- It will reduce the cost.
- Using this there will be less need for the hiring employees for the customer care department.
- The main benefit is the user queries would be answered without offering him to talk to a real person.
- It will also reduce wastage of time.

Following are the disadvantages:

- They can create confusion at most of the times.
- Sometimes the user wants to talk to a real person because they are not getting their queries resolved.
- It also gives error sometimes by answering the same to different queries having same keywords.
- Most of the times does not provide relevant output.

8. APPLICATIONS

Following are its applications:

- ☐ They can be used by any organizations, schools, colleges, help centers to clarify the customer doubts.
- ☐ They can also integrate within the social sites like facebook, instagram, etc.
- ☐ Any startup can deploy the chatbot on its business site so that their customer's queries are resolved and their time is also not wasted by attending each and every customer.

9.CONCLUSION

With all these guidelines and following the procedure depicted, we were able to build a chatbot that can resolve user's query to a greater extent. We also came to know about the various services and their creation like watson assistant, discovery, node-red app, and cloud function. And finally built our own chatbot perfectly.

10. FUTURE SCOPE

We can modify this chatbot by making it effortless to use and integrating text-to-speech and vice-versa feature in it so that the user or the customer can use it without having the effort of typing.

This will save customer's time and the process would become much easy.

11.BIBLIOGRAPHY

APPENDIX

SOURCE CODE

■ Cloud Function(cloudfunction.js)

```
/**
 *
 * @param {object} params
 * @param {string} params.iam_apikey
 * @param {string} params.url
 * @param {string} params.username
 * @param {string} params.password
 * @param {string} params.environment_id
 * @param {string} params.collection_id
 * @param {string} params.configuration_id
 * @param {string} params.input
 *
 * @return {object}
 *
 */

const assert = require('assert');
const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');

/**
 *
 * main() will be run when you invoke this action
 *
 * @param Cloud Functions actions accept a single parameter, which must be a
JSON object.
 *
 */
```

```

* @return The output of this action, which must be a JSON object.
*
*/
function main(params) {
  return new Promise(function (resolve, reject) {

    let discovery;

    if (params.iam_apikey){
      discovery = new DiscoveryV1({
        'iam_apikey': params.iam_apikey,
        'url': params.url,
        'version': '2020-05-09'
      });
    }
    else {
      discovery = new DiscoveryV1({
        'username': params.username,
        'password': params.password,
        'url': params.url,
        'version': '2020-05-11'
      });
    }

    discovery.query({
      'environment_id': params.environment_id,
      'collection_id': params.collection_id,
      'natural_language_query': params.input,
      'passages': true,
      'count': 3,
      'passages_count': 3
    }, function(err, data) {
      if (err) {
        return reject(err);
      }
    });
  });
}

```

```

    }
    return resolve(data);
  });
});
}

```

■ Dataset used- ecobee userguide3

■ Node-red(flow.json)

```

[{"id":"8e260e36.af87b","type":
  "tab","label":"Flow
  1","disabled":false,"info":""
},

```

```

{"id":"e37a3244.eachb5","type":"ui_form","z":"8e260e
36.af87b","name":"","label":"","group":"fd994811.73
7c88","order":1,"width":0,"height":0,
  "options":[{"label":"Enter      your
input","value":"text","type":"text","required":true
,"rows":null}],

```

```

"formValue":{"text":""},"payload":"","submit":"subm
it","cancel":"cancel","topic":"","x":130,"y":200,
  "wires":[["aceca082.e809a"]]},

```

```

{"id":"aceca082.e809a","type":"function","z":"8e260
e36.af87b","name":"","
  "func":"msg.payload=msg.payload.text;\nreturn
msg;","outputs":1,"noerr":0,"x":290,"y":140,
  "wires":[["aa762517.282f58","ba7f0471.1c7df8"]]},

```

```

{"id":"fc0b5dbc.ba276","type":"function","z":"8e260
e36.af87b","name":"","

```

```

"func":"msg.payload=msg.payload.output.text[0];\nre
turn msg;","outputs":1,"noerr":0,"x":630,"y":140,
  "wires":[["93d455bf.8f39b8"]]},

```

```

{"id":"aa762517.282f58","type":"watson-conversation
-v1","z":"8e260e36.af87b",
  "name":"Customer      Care
","workspaceid":"1046b103-e6ea-472e-90d2-38c60692c1
5c","multiuser":false,"context":true,

```

```
"empty-payload":false,

"service-endpoint":"https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/b8becd33-36fc-42bf-85d0-174e766675ec",

"timeout":"","optout-learning":false,"x":480,"y":80,
"wires":[["9f8482ce.bcb88","fc0b5dbc.ba276"]]],

{"id":"ba7f0471.1c7df8","type":"ui_text","z":"8e260e36.af87b",

"group":"fd994811.737c88","order":2,"width":0,"height":0,"name":"","label":"You :",

"format":"{{msg.payload}}","layout":"row-left","x":410,"y":240,"wires":[]},

{"id":"9f8482ce.bcb88","type":"debug","z":"8e260e36.af87b","name":"","

"active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":830,"y":40,"wires":[]},

{"id":"93d455bf.8f39b8","type":"ui_text","z":"8e260e36.af87b","group":"fd994811.737c88","order":3,"width":0,"height":0,"name":"","

"label":"Bot","format":"{{msg.payload}}","layout":"col-center","x":770,"y":240,"wires":[]},

{"id":"fd994811.737c88","type":"ui_group","z":"","name":"Chatbot","tab":"97d3d5de.249e58","order":1

,"disp":true,"width":11,"collapse":false},{

"id":"97d3d5de.249e58","type":"ui_tab","z":"","name":"Customer Care Helpdesk",

"icon":"dashboard","disabled":false,"hidden":false}

]
```

References:

1. <https://www.allbusinesstemplates.com/download/?filecode=2KBA4&lang=en&iuid=9f9faa69-9fab-40ee-8457-ea0e5df8c8de>
2. <https://www.youtube.com/embed/LOcKv-mENq8>
3. <https://my15.digitalexperience.ibm.com/b73a5759-c6a6-4033-ab6b-d9d4f9a6d65b/dxsites/151914d1-03d2-48fe-97d9-d21166848e65/>
4. <https://cloud.ibm.com/login>
5. <https://www.ibm.com/cloud/get-started>
6. <https://www.ibm.com/watson/products-services>
7. <https://www.youtube.com/watch?v=hitUOFNne14>
8. <https://developer.ibm.com/articles/introduction-watson-discovery/>
9. <https://www.youtube.com/embed/G3bqRndQtQg>
10. <https://developer.ibm.com/patterns/customer-help-desk-with-smart-document-understanding/>

