<u>Intelligent Customer Help Desk with Smart Document Understanding</u>

Internship Project Report

INTELLIGENT CUSTOMER HELP DESK WITH SMART DOCUMENT UNDERSTANDING

Submitted By-

Priyanshi Agarwal

(priyanshi.agarwal3405@gmail.com)

Thapar Institute of Engineering and Technology

INDEX

1. INTRODUCTION

1.1 Overview

We use the typical customer care chatbot experience but instead of relying on predefined responses, our dialog will provide a hook that can call out to other IBM Watson services for additional sources of information. In our case, it will be an owner's manual that has been uploaded into Watson Discovery.

Project Requirements

1. PC with high speed internet connectivity
2. Github account
3. IBM cloud account

Functional Requirements

1. When a user inputs a query, the intelligent help desk will answer the typical questions directly just like a chatbot.
2. Suppose the question that is asked is out of the scope of the chatbot, then the question will be passed on the Watson Discovery Service, which has been pre-loaded with the device's owner's manual.
3. Then returns relevant sections of the owner's manual to help solve customers' problems.

Technical Requirements

1. Knowledge of Python
2. Working knowledge with IBM Cloud
3. IBM Watson Services and IBM Cloud Functions.
4. Using GitHub and Slack.

Software Requirements

The project being based on IBM Cloud and GitHub, so as such no requirements on PC.

Project Deliverables

- Create a customer care dialog skill in Watson Assistant
- Use Smart Document Understanding to build an enhanced Watson Discovery collection
- Create an IBM Cloud Functions web action that allows Watson Assistant to post queries to Watson Discovery

Build a web application with integration to all these services & deploy the same on IBM Cloud Platform

1.2 Purpose

To create a chat-bot which can answer simple questions as well as return more and more accurate answers. Also not only answer the simple questions, but when a question falls outside of the scope of the pre-determined question set, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded with the device’s owners manual.

2. LITERATURE SURVEY

2.1 Existing Problem

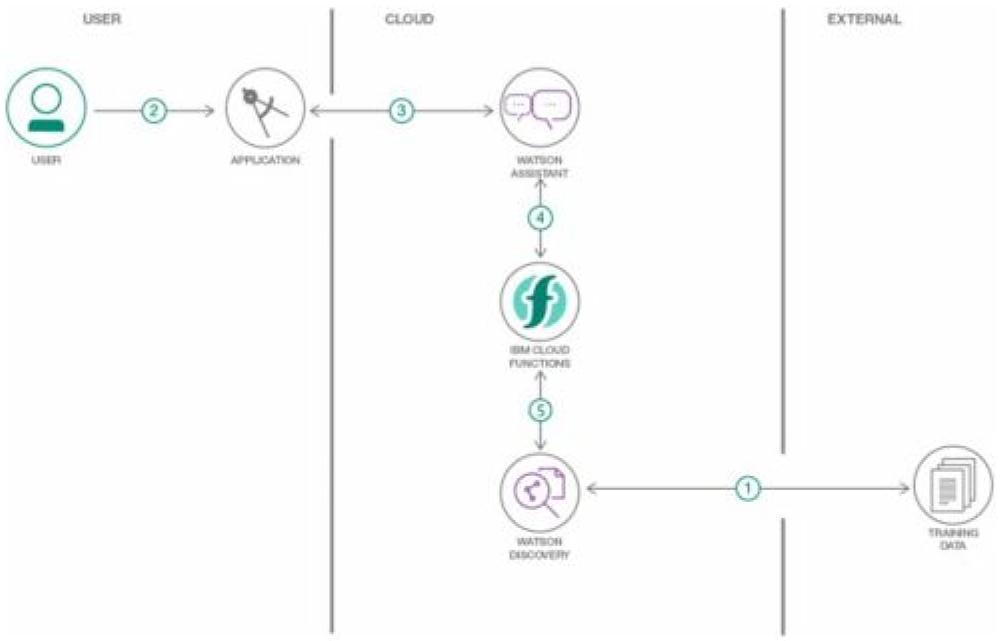
The typical customer care chat-bot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. When a question falls outside of the scope of the pre-determined question set, the option is typically to tell the customer the question isn’t valid or offer to speak to a real person.

2.2 Proposed Solution

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. To take it a step further, the project shall use the Smart Document Understanding feature of Watson Discovery to train it on what text in the owner's manual is important and what is not. This will improve the answers returned from the queries.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



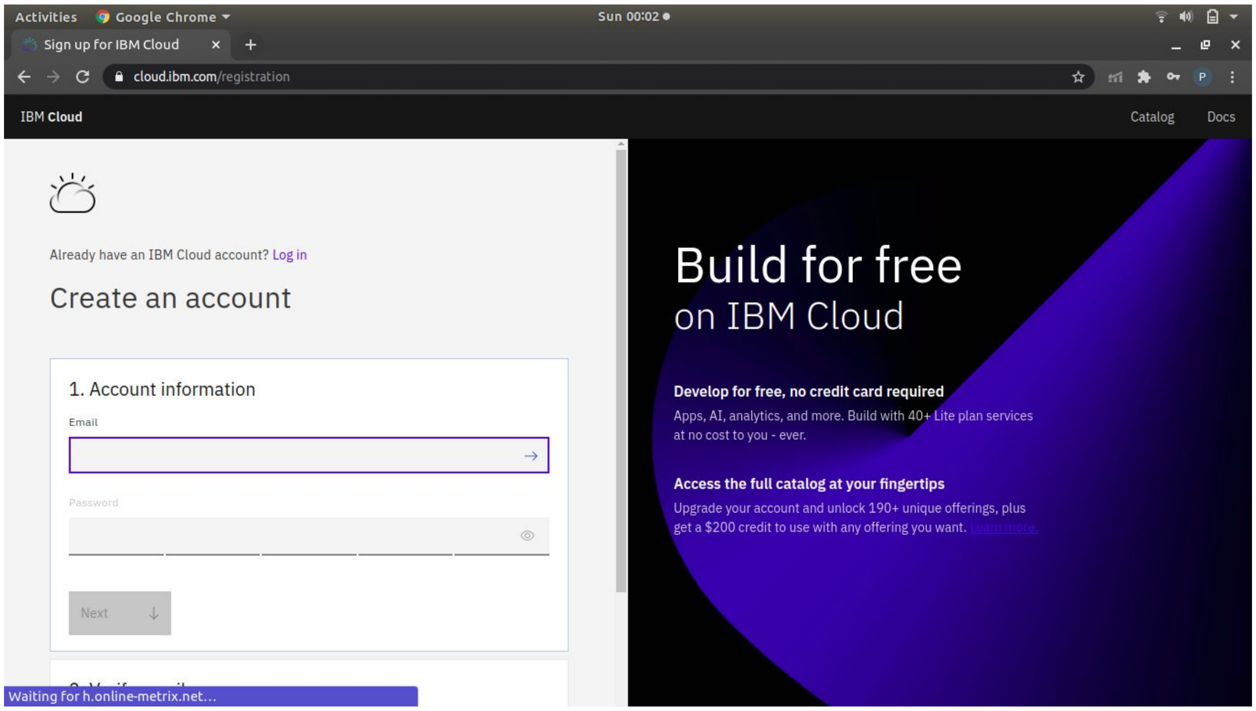
- The document is annotated using Watson Discovery SDU (Smart Document Understanding).
- The user interacts with the backend server via the app UI. The frontend app UI is a chatbot that engages the user in a conversation.
- Dialog between the user and backend server is coordinated using a Watson Assistant dialog skill.
- If the user asks a product operation question, a search query is passed to a predefined IBM Cloud Functions action.
- The Cloud Functions action will query the Watson Discovery service and return the results.

3.2 Hardware / Software designing

1. Create IBM Cloud Services
2. Configure Watson Discovery
3. Create IBM Cloud Functions Action
4. Configure Watson Assistant
5. Create flow and configure node
6. Deploy and run node red app.

4. EXPERIMENTAL INVESTIGATIONS

1. Necessary IBM Cloud services created



Already have an IBM Cloud account? Log in

Create an account

1. Account information

Email

Password

Next

Build for free on IBM Cloud

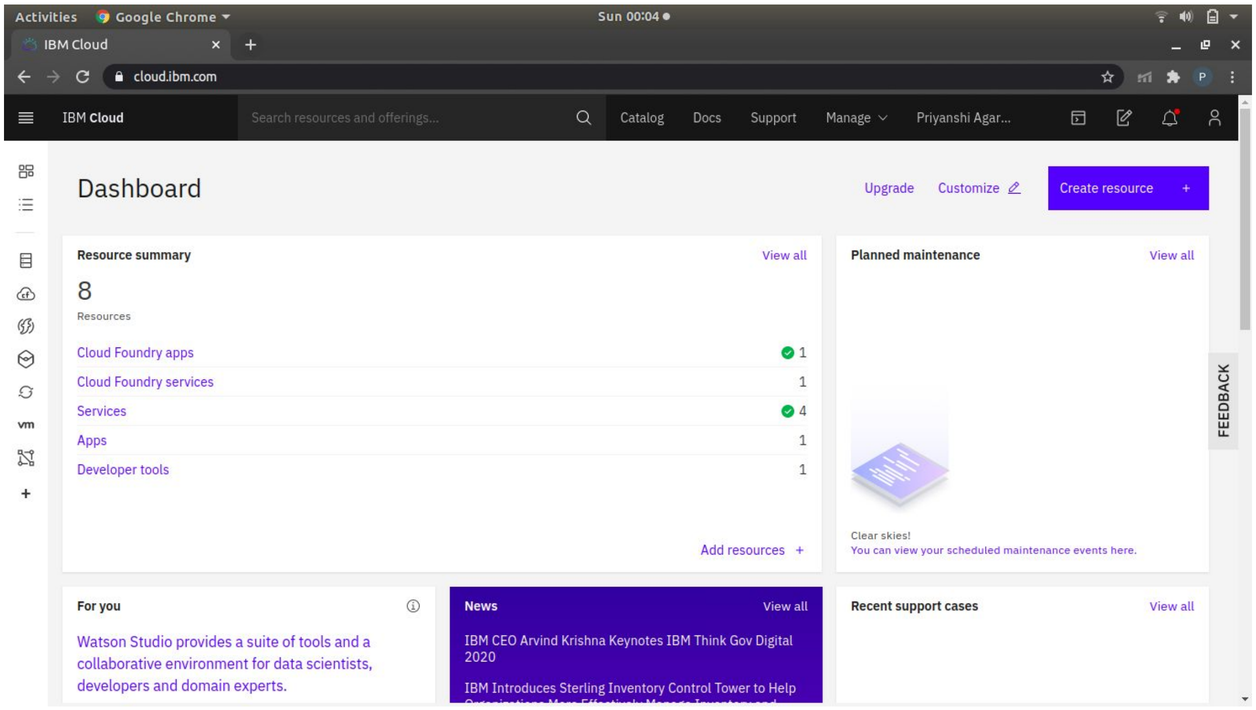
Develop for free, no credit card required

Apps, AI, analytics, and more. Build with 40+ Lite plan services at no cost to you - ever.

Access the full catalog at your fingertips

Upgrade your account and unlock 190+ unique offerings, plus get a $200 credit to use with any offering you want. Learn more.

Waiting for h.online-metrix.net...



Dashboard

Resource summary

8 Resources

Cloud Foundry apps	1
Cloud Foundry services	1
Services	4
Apps	1
Developer tools	1

Planned maintenance

Clear skies!
You can view your scheduled maintenance events here.

For you

Watson Studio provides a suite of tools and a collaborative environment for data scientists, developers and domain experts.

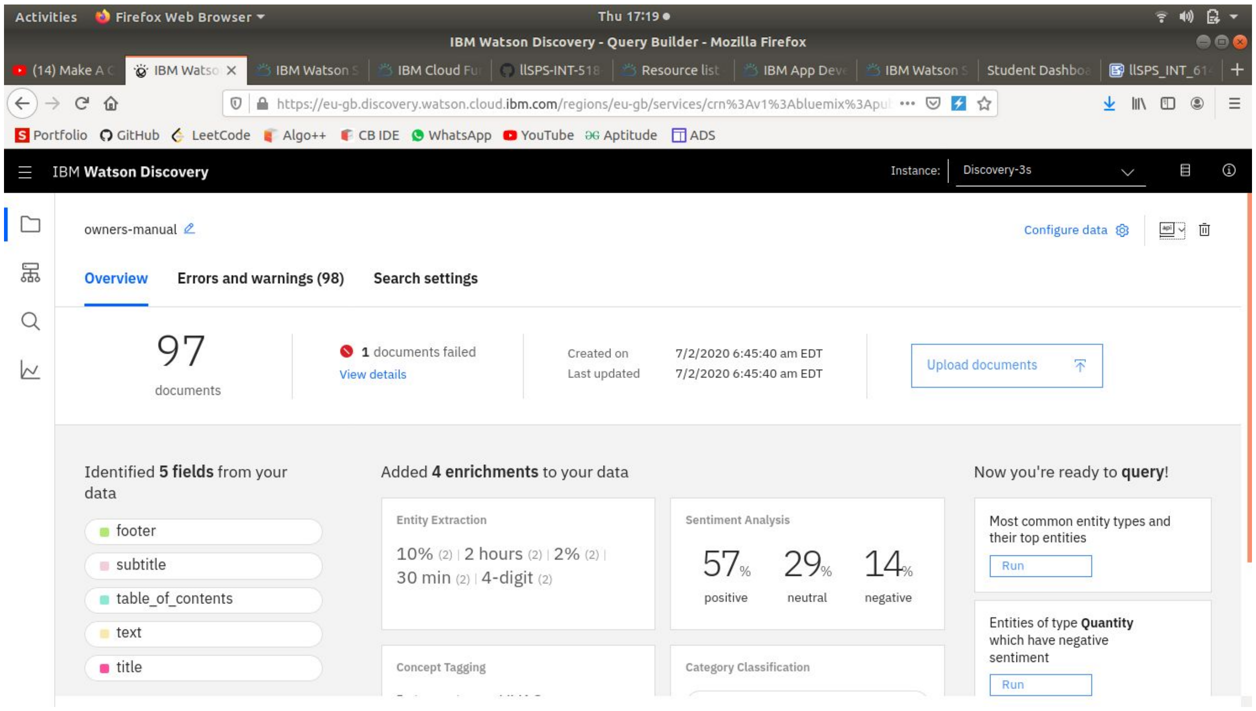
News

IBM CEO Arvind Krishna Keynotes IBM Think Gov Digital 2020

IBM Introduces Sterling Inventory Control Tower to Help

Recent support cases

2. Watson Discovery Service configured



97 documents

1 documents failed
View details

Created on 7/2/2020 6:45:40 am EDT
Last updated 7/2/2020 6:45:40 am EDT

Identified **5 fields** from your data

- footer
- subtitle
- table_of_contents
- text
- title

Added **4 enrichments** to your data

Entity Extraction
10% (2) | 2 hours (2) | 2% (2) | 30 min (2) | 4-digit (2)

Sentiment Analysis
57% positive 29% neutral 14% negative

Concept Tagging

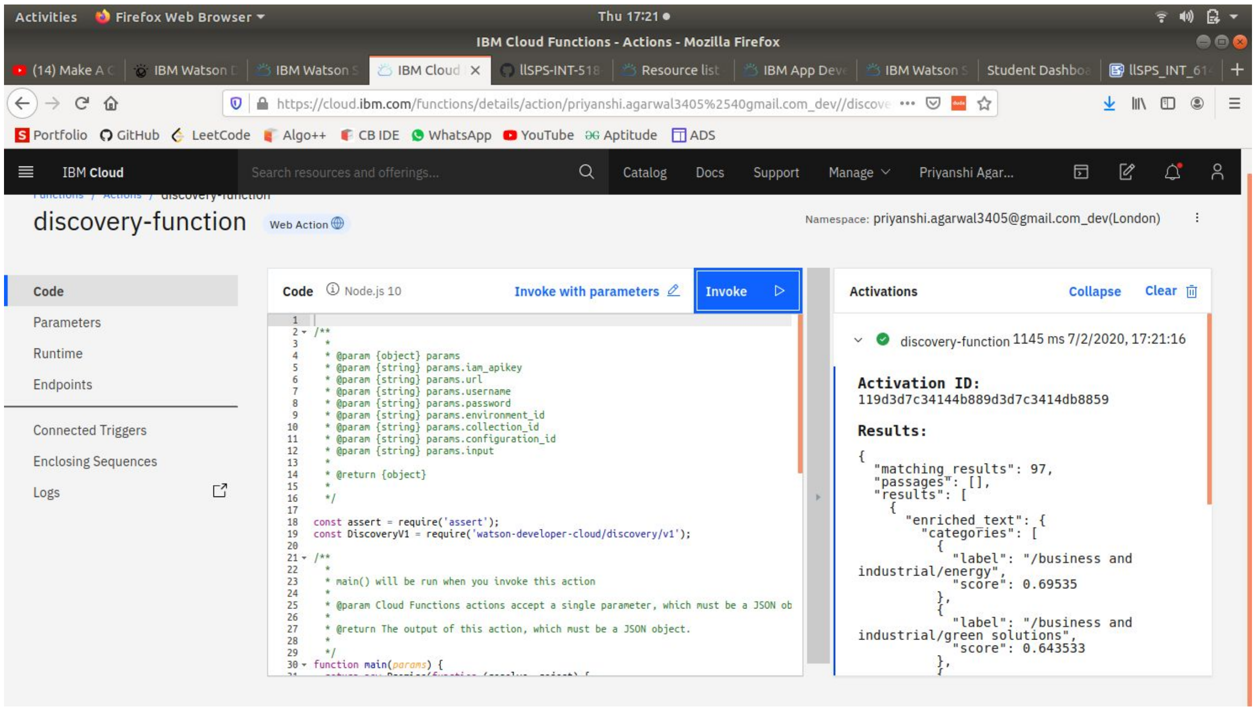
Category Classification

Now you're ready to **query**!

Most common entity types and their top entities

Entities of type **Quantity** which have negative sentiment

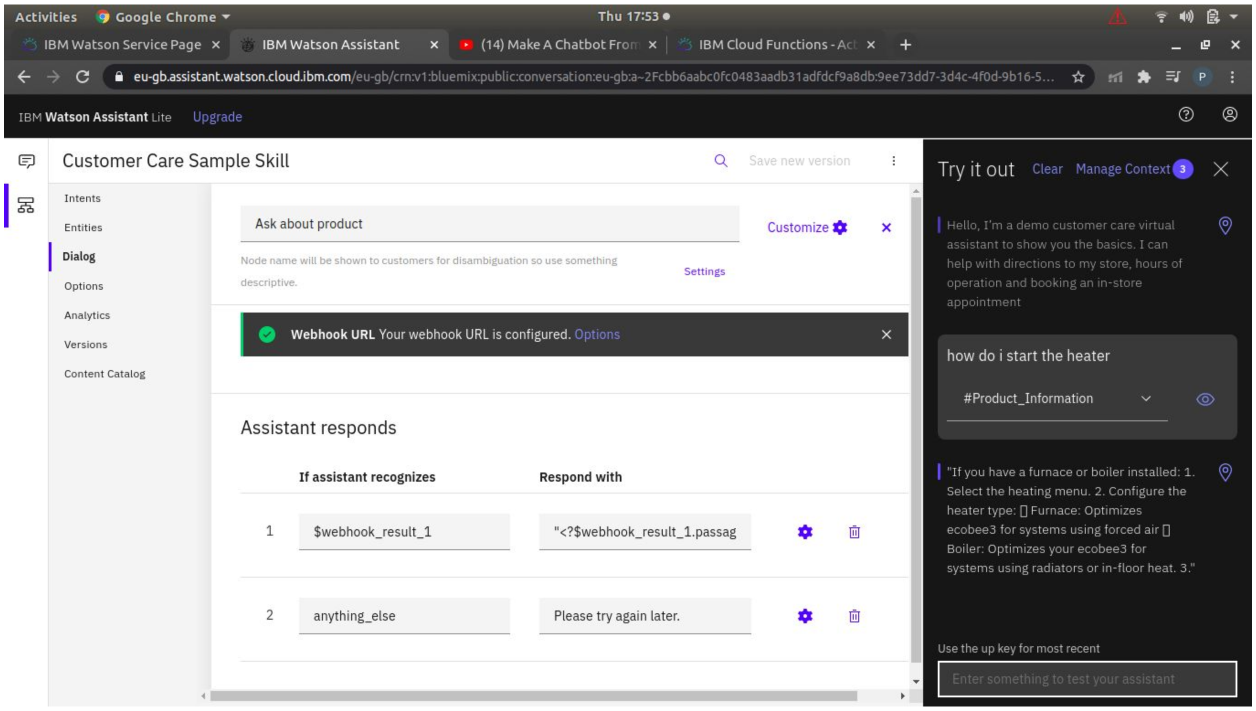
3. Create Cloud Function Action



```
/**
  *
  * @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 ob
  *
  * @return The output of this action, which must be a JSON object.
  *
  */
function main(params) {
```

```
discovery-function 1145 ms 7/2/2020, 17:21:16

Activation ID:
119d3d7c34144b889d3d7c3414db8859

Results:
{
  "matching_results": 97,
  "passages": [],
  "results": [
    {
      "enriched_text": {
        "categories": [
          {
            "label": "/business and
industrial/energy",
            "score": 0.69535
          },
          {
            "label": "/business and
industrial/green solutions",
            "score": 0.643533
          },
```

4. Watson Assistant configured


IBM Watson Assistant
eu-gb.assistant.watson.cloud.ibm.com/eu-gb/crn:v1:bluemix:public:conversation:eu-gb:a~2Fcbb6aabc0fc0483aadb31adfdcf9a8db:9ee73dd7-3d4c-4f0d-9b16-5...
IBM Watson Assistant Lite
Upgrade
Customer Care Sample Skill
Save new version
Intents
Entities
Dialog
Options
Analytics
Versions
Content Catalog
Customize
Settings
Try it out
Clear
Manage Context
Use the up key for most recent
Enter something to test your assistant

Ask about product

Node name will be shown to customers for disambiguation so use something descriptive.

Webhook URL Your webhook URL is configured. Options

Assistant responds

	If assistant recognizes	Respond with
1	$webhook_result_1	"<?$webhook_result_1.passag
2	anything_else	Please try again later.

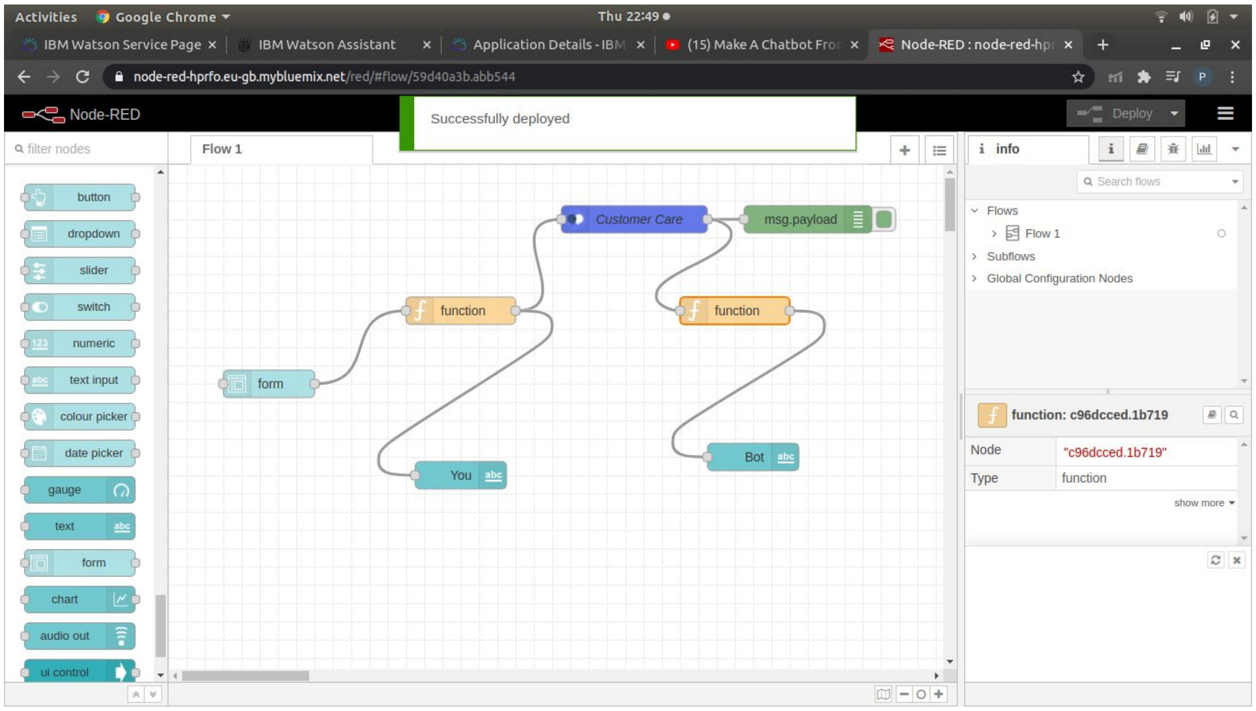
Hello, I'm a demo customer care virtual assistant to show you the basics. I can help with directions to my store, hours of operation and booking an in-store appointment

how do i start the heater

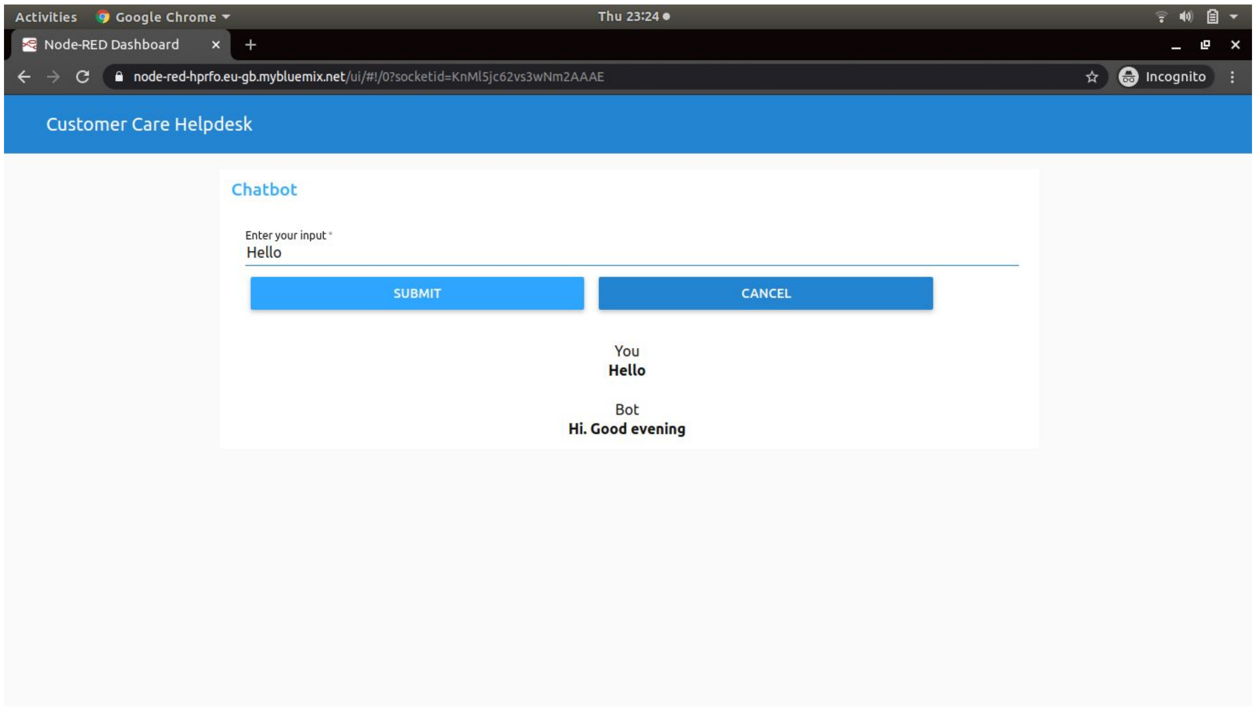
#Product_Information

"If you have a furnace or boiler installed: 1. Select the heating menu. 2. Configure the heater type: ▯ Furnace: Optimizes ecobee3 for systems using forced air ▯ Boiler: Optimizes your ecobee3 for systems using radiators or in-floor heat. 3."

5. Built node-red flow to integrate all services



6. Built a web dashboard

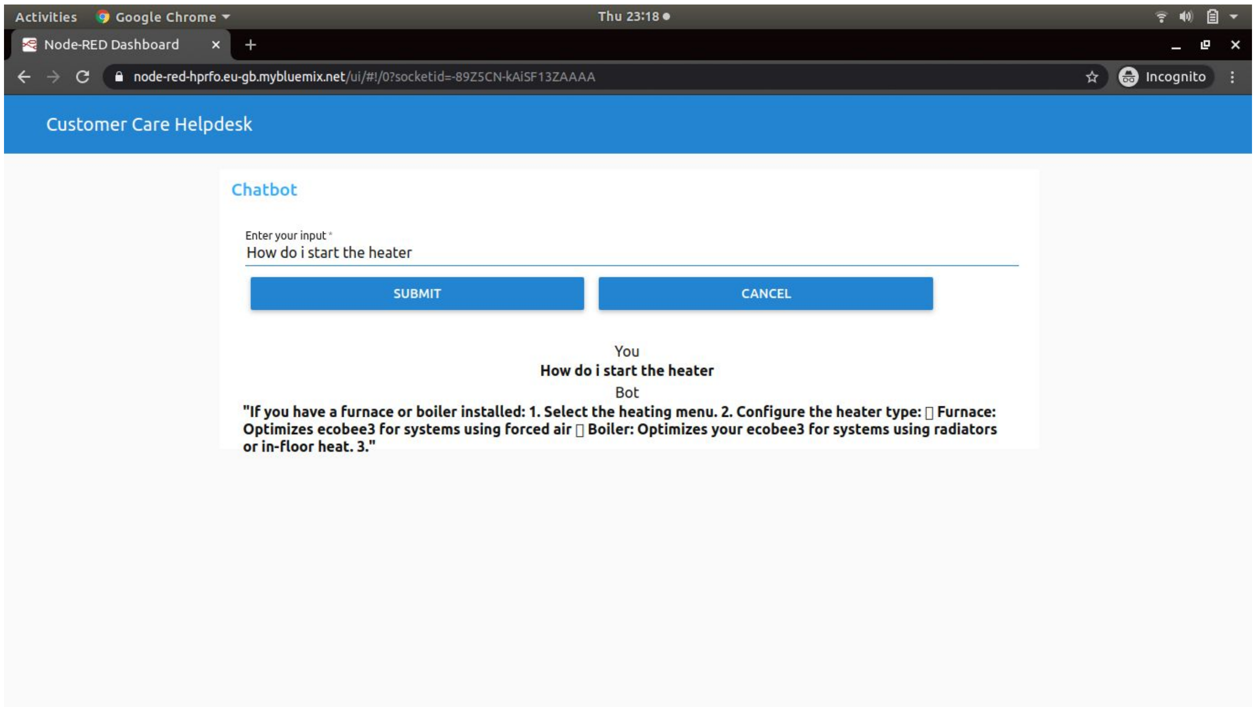


Hello

You
Hello

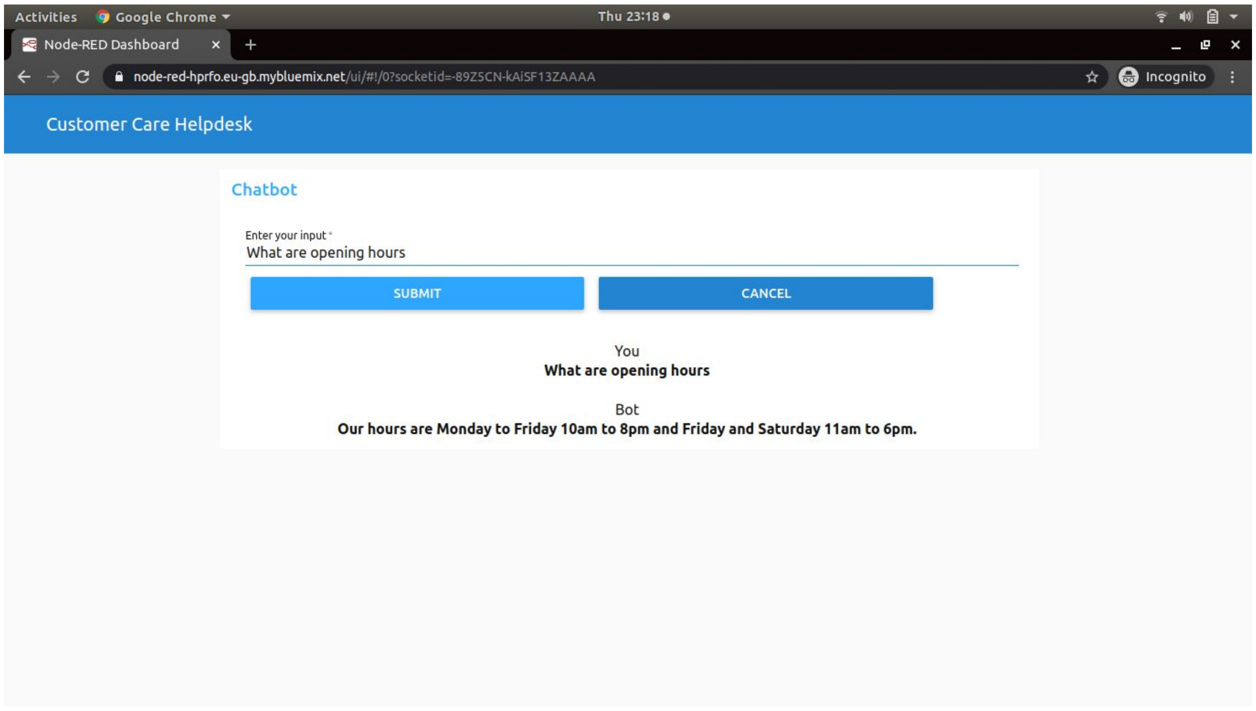
Bot
Hi. Good evening

7. Tested and captured results



How do i start the heater

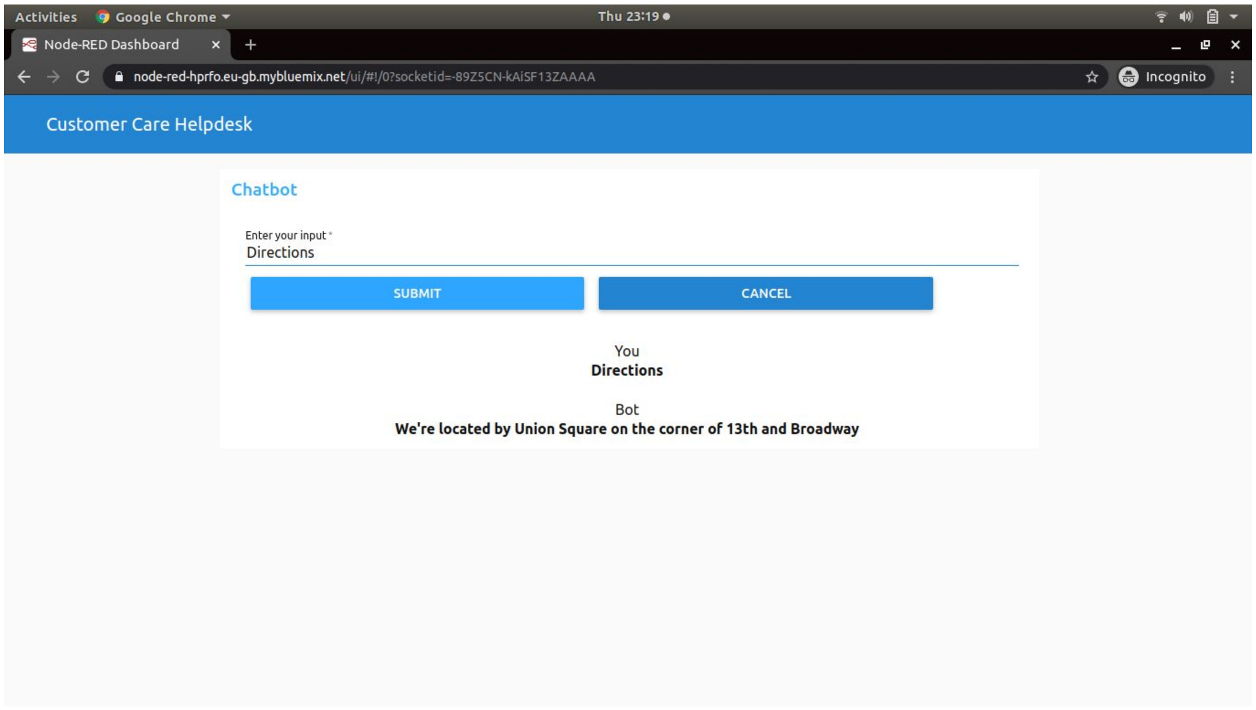
You
How do i start the heater
Bot
"If you have a furnace or boiler installed: 1. Select the heating menu. 2. Configure the heater type: Furnace: Optimizes ecobee3 for systems using forced air Boiler: Optimizes your ecobee3 for systems using radiators or in-floor heat. 3."



What are opening hours

You
What are opening hours

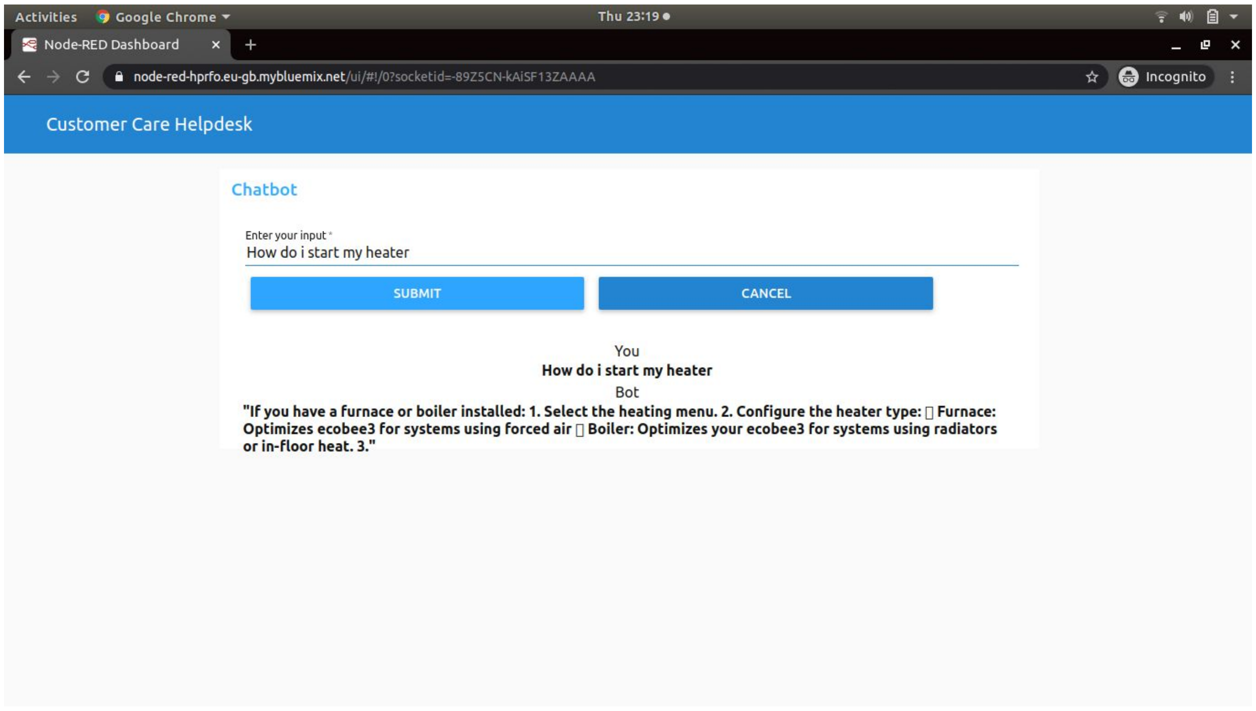
Bot
Our hours are Monday to Friday 10am to 8pm and Friday and Saturday 11am to 6pm.



Directions

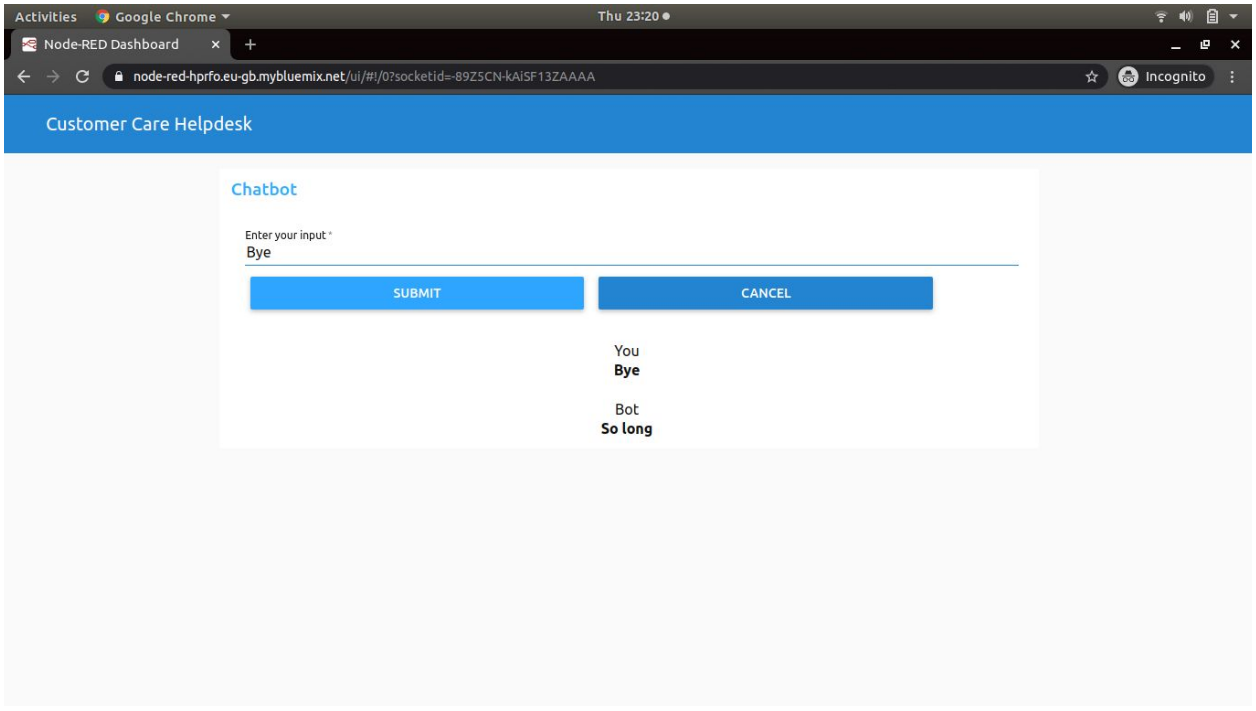
You
Directions

Bot
We're located by Union Square on the corner of 13th and Broadway



How do i start my heater

You
How do i start my heater
Bot
"If you have a furnace or boiler installed: 1. Select the heating menu. 2. Configure the heater type: ▯ Furnace: Optimizes ecobee3 for systems using forced air ▯ Boiler: Optimizes your ecobee3 for systems using radiators or in-floor heat. 3."

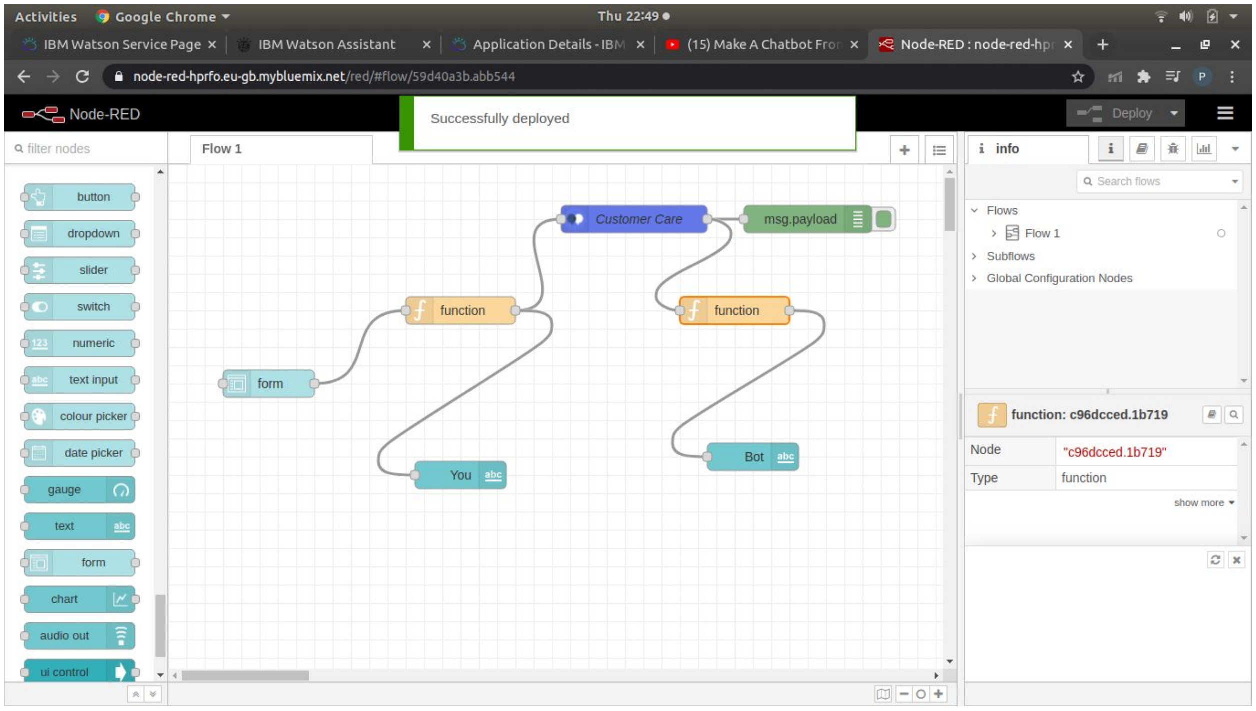


Bye

You
Bye

Bot
So long

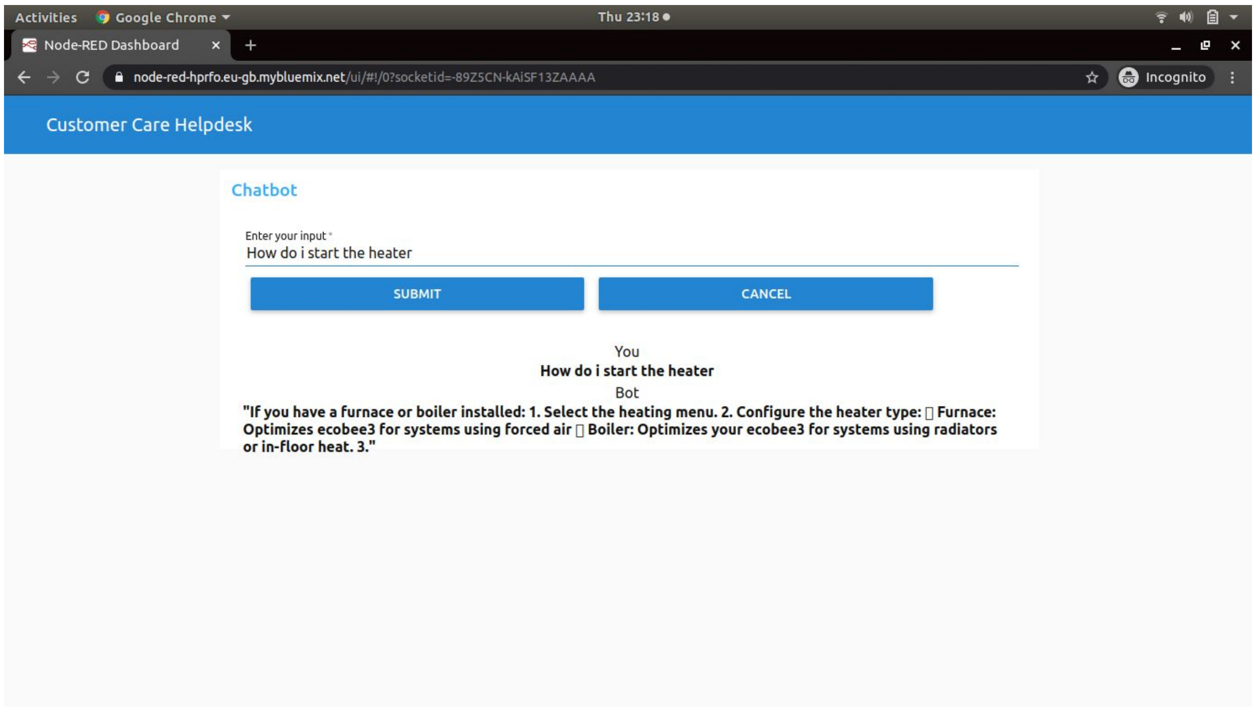
5.FLOWCHARTS



6. RESULT

Finally all the components are integrated using node-red dashboard. They can be displayed on dashboard UI by using URL-

https://node-red-hprfo.eu-gb.mybluemix.net/ui/#!/0?socketid=zrfxwKUNr0-5VpzrAAAF



How do i start the heater

You

How do i start the heater

Bot

"If you have a furnace or boiler installed: 1. Select the heating menu. 2. Configure the heater type: Furnace: Optimizes ecobee3 for systems using forced air Boiler: Optimizes your ecobee3 for systems using radiators or in-floor heat. 3."

7. ADVANTAGES AND DISADVANTAGES

Advantages

1. Reduces man power
2. Cost efficient
3. Faster customer service
4. 24/7 availability
5. Increased customer satisfaction
6. Customer agent can focus on other tasks as no need to divert any calls to him/her.

Disadvantages

1. Maintainance.
2. Limited response for customers.
3. May give same answers for different sentiments.
4. Sometimes cannot connect to customer emotions and sentiments.

8. APPLICATIONS

- It can be deployed on social media platforms like facebook, slack etc.

- Can also be deployed on any website to clarify basic doubts like coding blocks, udemy etc.

9. CONCLUSION

By following proper steps , I was successfully able to create Intelligent Customer Help Desk smart chatbot using Watson assistant, Watson discovery, Node-RED and Cloud-functions.

10. FUTURE SCOPE

To extend it further, we can also include Watson Studio text to speech and speech to text services and can also integrate it with a robot.

11. BIBLIOGRAPHY

Appendix:

a) Source code

1. Cloud function action

```
/**

 *

 * @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': '2019-03-25'
   });
  }
  else {
   discovery = new DiscoveryV1({
    'username': params.username,
    'password': params.password,
    'url': params.url,
    'version': '2019-03-25'
   });
  }
  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);
  });
 });
}
```

2. Node-red flow

```
[
 {
  "id": "59d40a3b.abb544",
  "type": "tab",
  "label": "Flow 1",
  "disabled": false,
  "info": ""
 },
 {
  "id": "937d65c0.69c048",
  "type": "ui_form",
  "z": "59d40a3b.abb544",
  "name": "",
  "label": "",
  "group": "e3800fb4.c9873",
  "order": 1,
  "width": 0,
  "height": 0,
  "options": [
   {
    "label": "Enter your input",
    "value": "text",

    "type": "text",

    "required": true,

    "rows": null

   }

  ],

  "formValue": { "text": "" },

  "payload": "",

  "submit": "submit",

  "cancel": "cancel",

  "topic": "",

  "x": 110,

  "y": 240,

  "wires": [["f1eca286.1d75d"]]

 },

 {

  "id": "f1eca286.1d75d",

  "type": "function",

  "z": "59d40a3b.abb544",

  "name": "",

  "func": "msg.payload=msg.payload.text;\nreturn msg;",
  "outputs": 1,

  "noerr": 0,

  "initialize": "",

  "finalize": "",

  "x": 320,

  "y": 160,

  "wires": [["1a9db72c.c2dc79", "ec9fc561.a77e28"]]

},

{

  "id": "c96dcced.1b719",

  "type": "function",

  "z": "59d40a3b.abb544",

  "name": "",

  "func": "msg.payload=msg.payload.output.text[0];\nreturn msg;",

  "outputs": 1,

  "noerr": 0,

  "initialize": "",

  "finalize": "",

  "x": 620,

  "y": 160,

  "wires": [["b707a4ca.6434d8"]]
```

```
},
{
  "id": "1a9db72c.c2dc79",
  "type": "watson-conversation-v1",
  "z": "59d40a3b.abb544",
  "name": "Customer Care",
  "workspaceid": "YOUR_ID",
  "multiuser": false,
  "context": true,
  "empty-payload": false,
  "service-endpoint": "YOUR_URL",
  "timeout": "",
  "optout-learning": false,
  "x": 510,
  "y": 60,
  "wires": [["b5976af6.d52148", "c96dcced.1b719"]]
},
{
  "id": "ec9fc561.a77e28",
  "type": "ui_text",
  "z": "59d40a3b.abb544",
  "group": "e3800fb4.c9873",

  "order": 2,

  "width": 0,

  "height": 0,

  "name": "",

  "label": "You",

  "format": "{{msg.payload}}",

  "layout": "col-center",

  "x": 320,

  "y": 340,

  "wires": []

},

{

  "id": "b5976af6.d52148",

  "type": "debug",

  "z": "59d40a3b.abb544",

  "name": "",

  "active": true,

  "tosidebar": true,

  "console": false,

  "tostatus": false,
  "complete": "false",
  "statusVal": "",
  "statusType": "auto",
  "x": 700,
  "y": 60,
  "wires": []
},
{
  "id": "b707a4ca.6434d8",
  "type": "ui_text",
  "z": "59d40a3b.abb544",
  "group": "e3800fb4.c9873",
  "order": 3,
  "width": 0,
  "height": 0,
  "name": "",
  "label": "Bot",
  "format": "{{msg.payload}}",
  "layout": "col-center",
  "x": 640,
  "y": 320,
  "wires": []

},

{

  "id": "e3800fb4.c9873",

  "type": "ui_group",

  "z": "",

  "name": "Chatbot",

  "tab": "69a8b472.3b4cec",

  "order": 1,

  "disp": true,

  "width": 14,

  "collapse": false

},

{

  "id": "69a8b472.3b4cec",

  "type": "ui_tab",

  "z": "",

  "name": "Customer Care Helpdesk",

  "icon": "dashboard",

  "disabled": false,

  "hidden": false
```

```
}

]
```

b) References

https://www.ibm.com/cloud/get-started

https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/

https://nodered.org/

https://github.com/watson-developer-cloud/node-red-labs

https://www.youtube.com/embed/s7wmiS2mSXY

https://www.w3schools.com/howto/howto_make_a_website.asp

https://www.ibm.com/watson/products-services
https://developer.ibm.com/technologies/web-development/articles/ws-restful

https://www.youtube.com/embed/G3bqRndQtQg