PROJECT REPORT

NAME - SRIKRISHNA VETURI (srikrishna.veturi18@siesgst.ac.in)

TITLE - Intelligent Customer Help Desk with Smart Document Understanding.

DOMAIN - Artificial Intelligence / Machine Learning

Internship at smartinternz.com@2020

TABLE OF CONTENTS :

1. INTRODUCTION

1.1 Overview :

We will build a chot bot that uses various Watson AI Services to deliver an effective Web based UI through which we can chat with the assistant . We will integrate the Watson Discovery service with Watson assistant using Webhooks.

- ➤Project Requirements : Node-Red,IBM Cloud,IBM Watson,Node JS
- ➤Functional Requirements : IBM Cloud
- ➤Technical Requirements : AI,ML,Watson AI,Node JS
- ➤Software Rquirements : Watson Assistant,Watson Discovery,Watson Cloud Functions,Node-RED.
- ➤Project Deliverables : Intelligent Chat Bot with Smart Document Understanding.
- ➤Project Duration : 1 month

1.2 Purpose :

The typical customer care chatbot 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 owners manual. So now, instead of “Would you like to speak to a customer representative?” we can return relevant sections of the owners 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 owners manual is important and what is not. This will improve the answers returned from the queries.

2. LITERATURE SURVEY :

2.1Existing Problem :

The typical customer care chatbot 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.

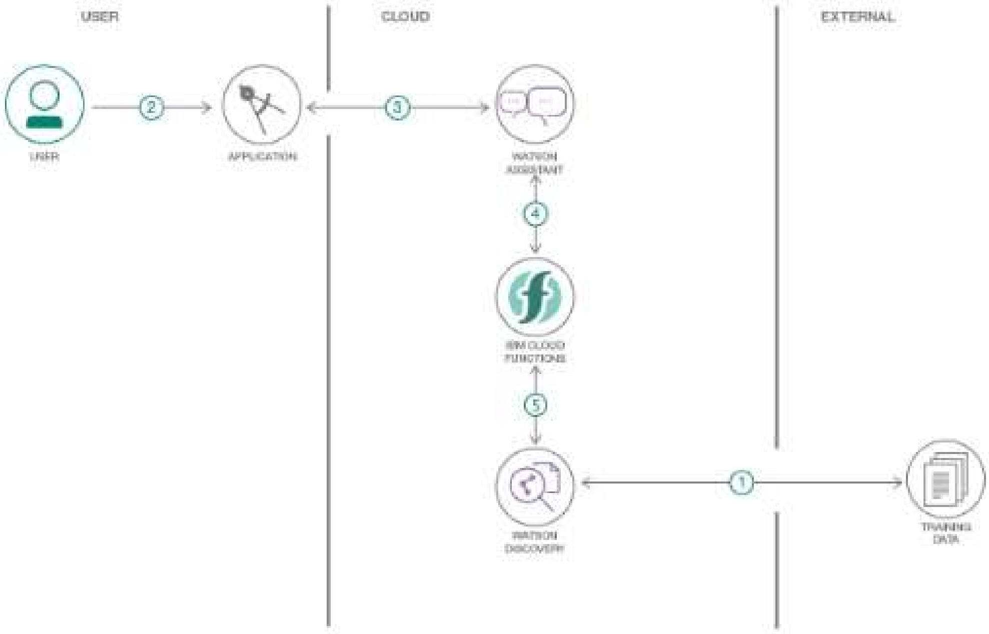
2. 2 Proposed Solution :

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 owners manual. So now, instead of "Would you like to speak to a customer representative?" we can return relevant sections of the owners manual to help solve our customers' problems.So unless the customer secifically asks for a customer representative, the bot will try to solve all of the user's queries.

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 owners manual is important and what is not.

3 . THEORITICAL ANALYSIS

3.1. Block Diagram :



2.2. Proposed Solution :

- ○ Create IBM Cloud Services.
- ○ Configure Watson Discovery.
- ○ Create IBM Cloud Function Action
- ○ Configure Watson Assitant
- ○ Create flow and configure node
- ○ Deploy and run node red app

4. EXPERIMENTAL INVESTIGATIONS

1) Create IBM Cloud Services :

   a. Watson Discovery
   b. Watson Assistant
   c. Node Red

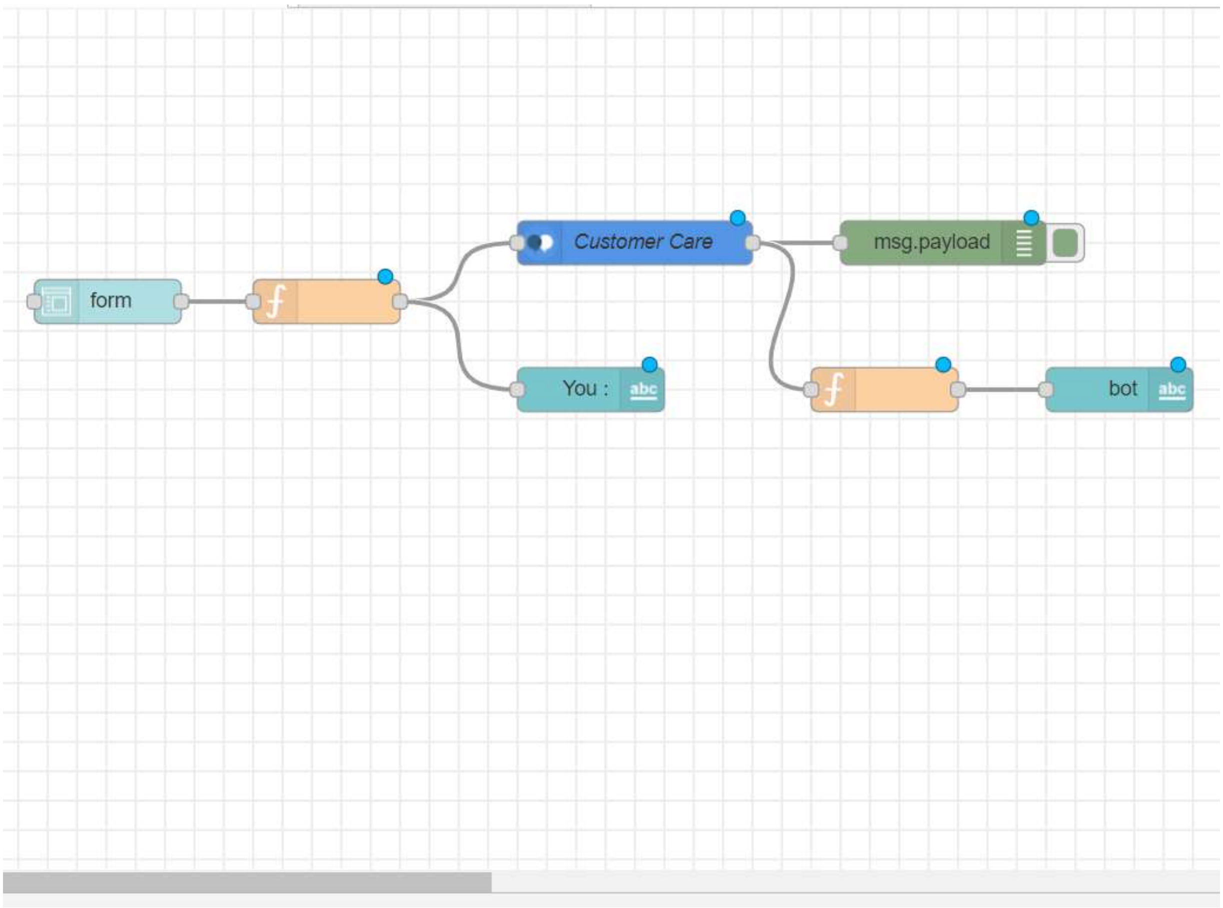
2) Configure Watson Discovery :

After creating and launching the discovery from the catalog, import the document on which we need to train the discovery service.we have selected ecobee3 user guide located in the data directory of our local repository. The ecobee is a popular residential thermostat which has WiFi and multiple configuration options.

5 . FLOWCHART :

Insert the following nodes into the node red flow :

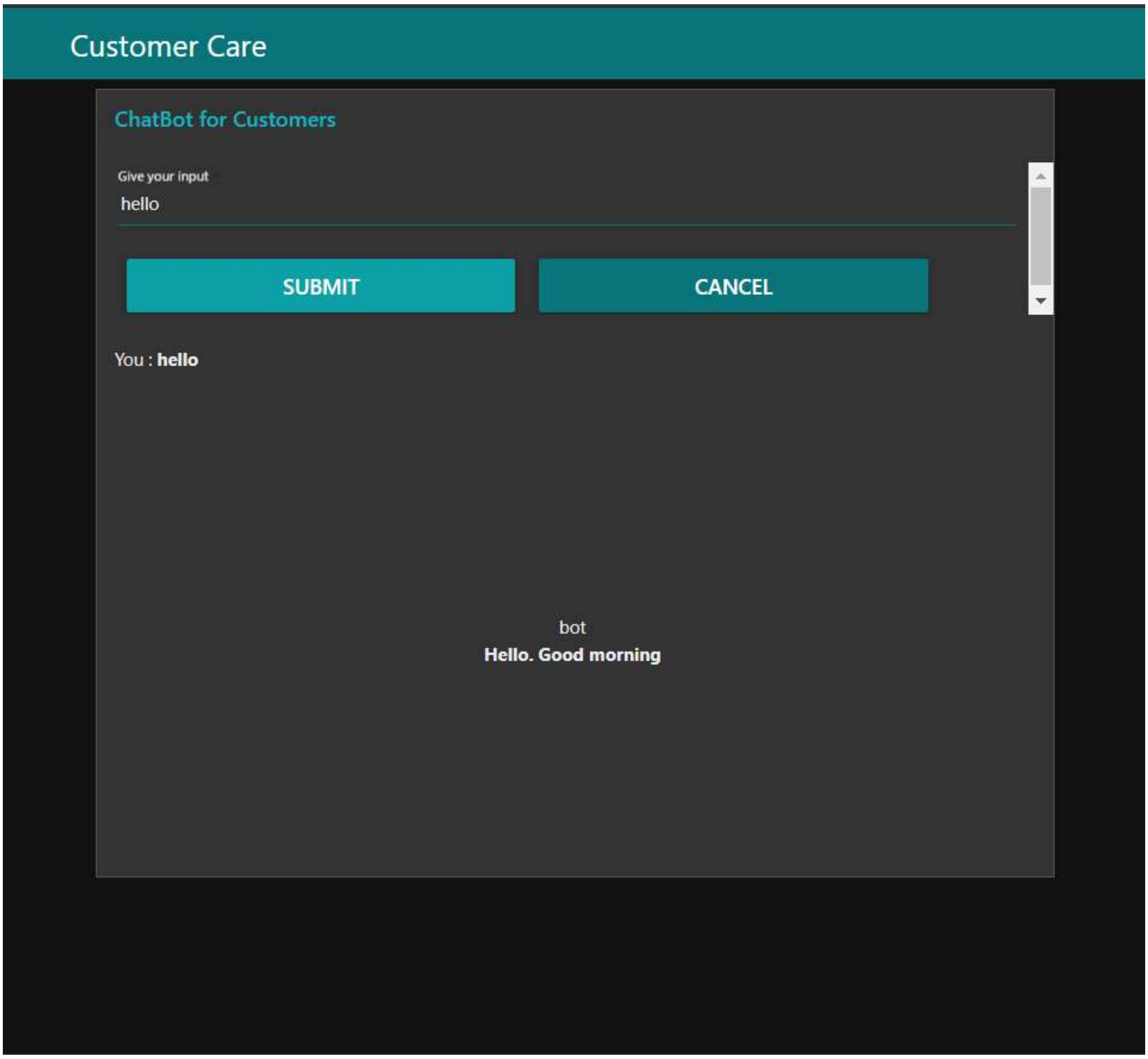
1) Form
2) Function
3) Text
4) Watson Assistant
5) Debug



visit the link for the flow :

https://node-red-ubobq.eu-gb.mybluemix.net/red/#flow/3914e65a.4fd64a

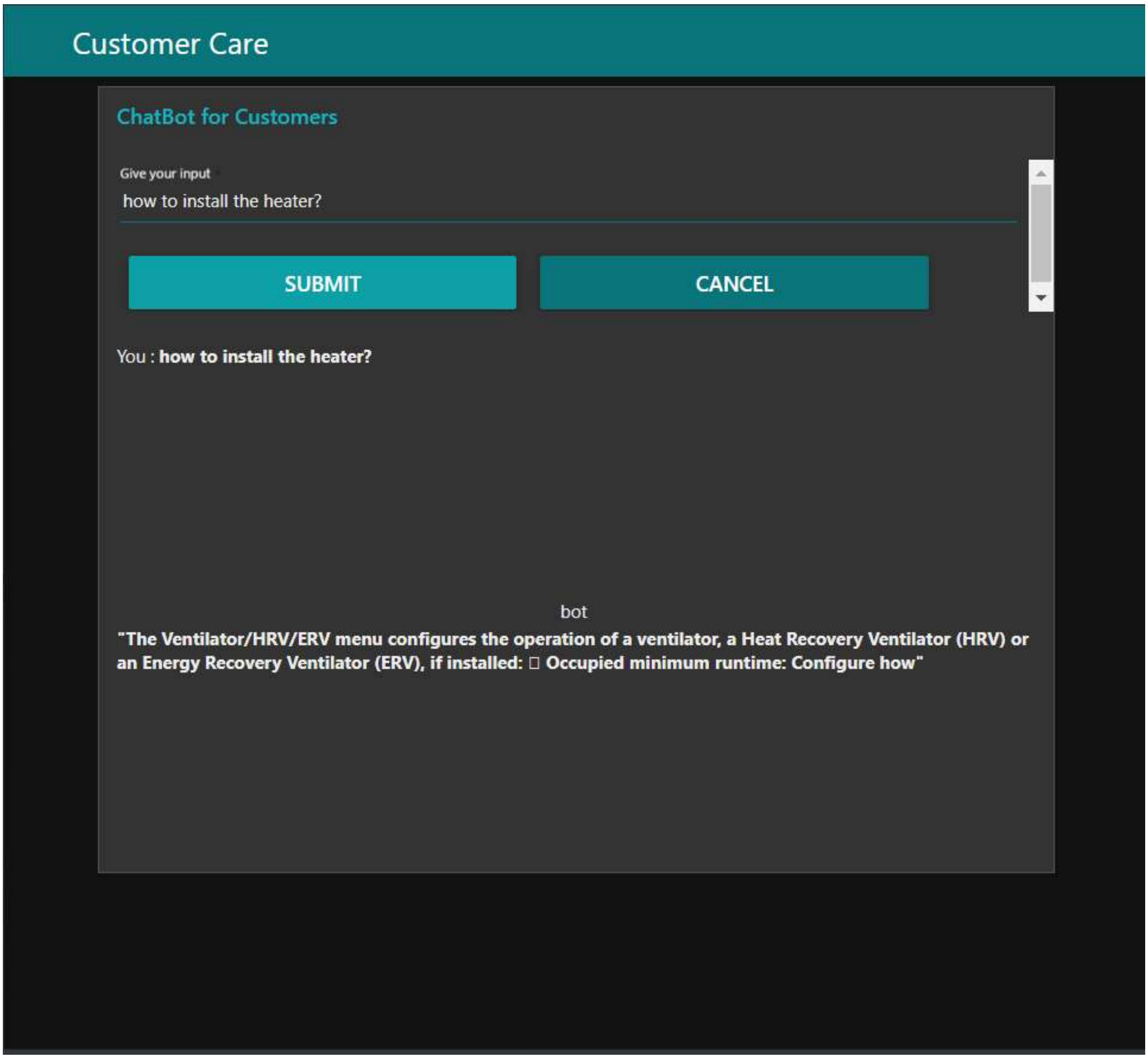
6. RESULT :



hello

You : **hello**

bot
Hello. Good morning

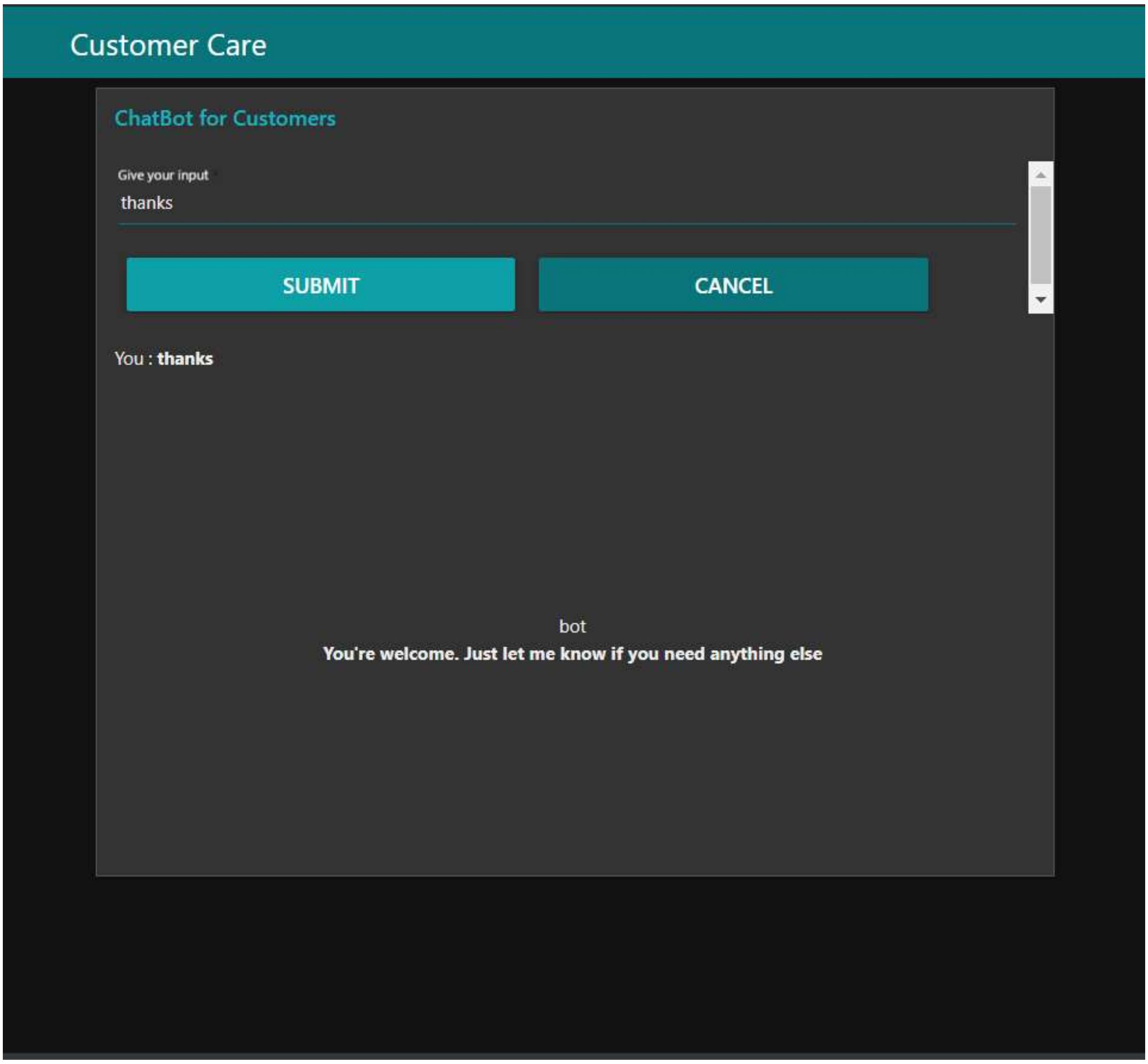


how to install the heater?

You : **how to install the heater?**

bot

"The Ventilator/HRV/ERV menu configures the operation of a ventilator, a Heat Recovery Ventilator (HRV) or an Energy Recovery Ventilator (ERV), if installed: □ Occupied minimum runtime: Configure how"



thanks

You : **thanks**

bot

You're welcome. Just let me know if you need anything else

visit the link for the chat bot :

https://node-red-ubobq.eu-gb.mybluemix.net/ui/#!/0?socketid=b9iROC81iMmDVkWCAAAU

7. ADVANTAGES AND DISADVANTAGES

Advantages :

- Companies can use these to decrease the work flow to the representatives.
- reduce the number of representatives.
- cost efficient.
- decrease In the number of calls diverted to representatives.
- less work load on emmployees

Disadvantages :

- sometimes the chatbot misleads the customers.
- the discovery returns wrong results when not properly configured.
- giving same answer for different questions.
- sometimes it is unable to connect to the customer's intents

8. APPLICATIONS :

- ○ It can be used as customer care serives for small businesses.
- ○ it can bee deployed as social media chat bots on platforms like facebook,slack and telegram.
- ○ chat bot can be deployed to serve any basic questions or queries that the users might have.

9. CONCLUSION :

By following the above mentioned steps, we can create a basic chat bot which can help us to answer the basic questions of the custoer or user related to location of the office, working hours and the information bout the product. We successfully created the Intelligent help desk smart chatbot using Watson Assistant, Watson Cloud Function,Watson Discovery and Node-RED.

10. FUTURE SCOPE :

We can improve our accuracy in giving the correct replies by asking the users if they found the chatbot to be useful at the end of every conversation and if not, we can ask them why.
We can import the pre-built node-red flow and can improve our UI, moreover, we can make a data base and use it to show the recent conversations that the users had with the bot.
There is another possibility of making the bot recognize voice which can help some disabled users.

11. BILIOGRAPHY :

Node-RED starter application :

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

Build your own AI assitant :
https://www.youtube.com/embed/W3iPbFTAAds

Watson Discovery and smart document understanding:

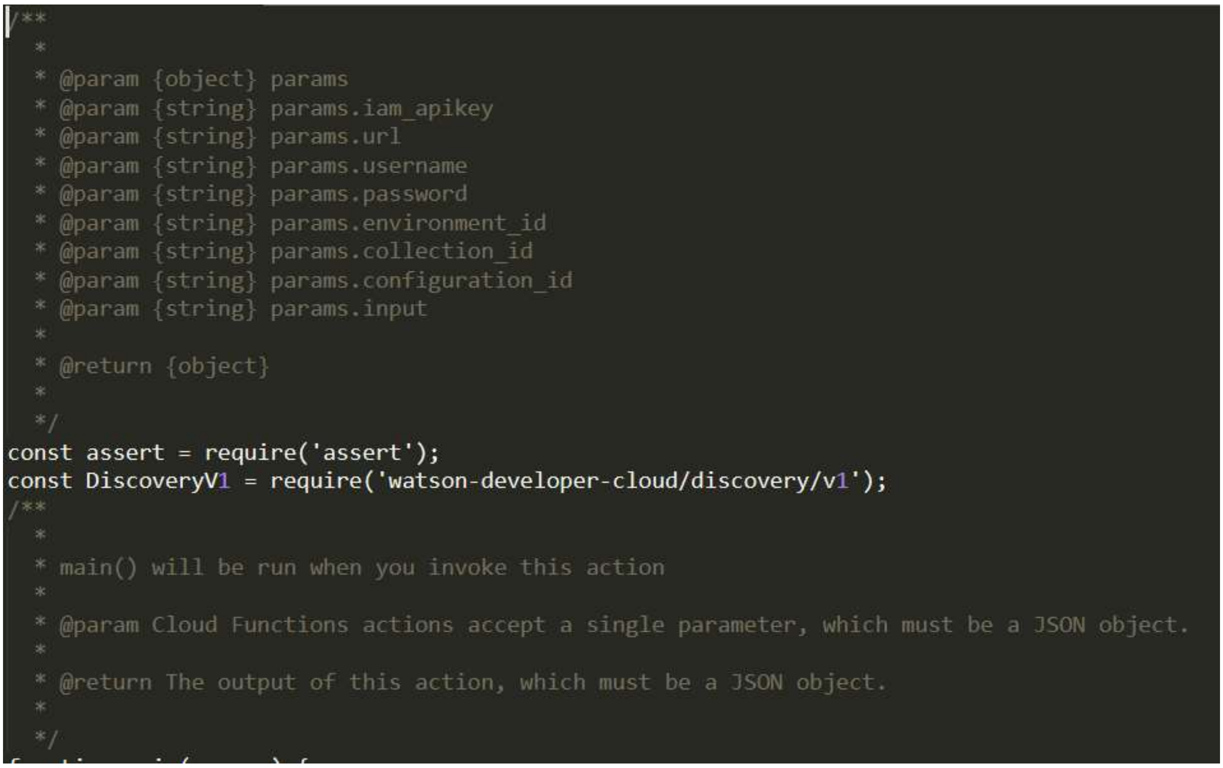
https://www.youtube.com/embed/Jpr3wVH3FVA

IBM Cloud Functions :

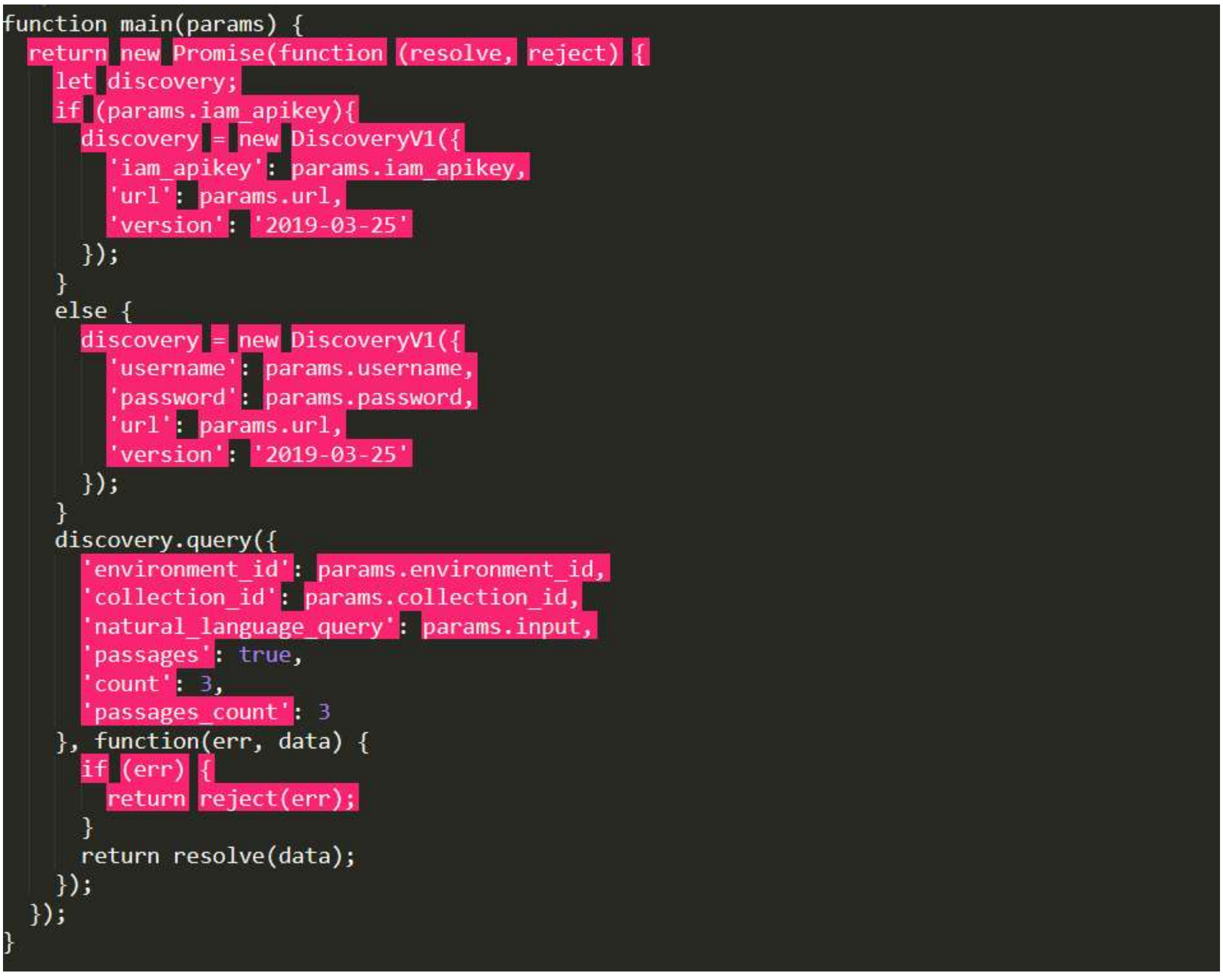
https://www.youtube.com/embed/G3bqRndQtQg

APPENDIX :

Source Code for cloud functions :



```
/**
  *
  * @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);
    });
  });
}
```