Project Report
on

Intelligent Customer Help Desk with Smart Document Understanding

by

Name : Drishti Sabhaya
drishtisabhaya@gmail.com

Contents

Introduction

Overview

In this project, we will be learning about how to build a customer help desk by using various tools and actions of IBM cloud which includes Watson Discovery, Watson Assistant, Cloud Function and Node Red.
We will be buliding a intelligent chatbot by integrating Waton services and cloud functions with node-red.

- Project Requirements: Python, IBM Cloud, IBM Watson, Node-Red
- Functional Requirements: IBM Cloud
- Technical Requirements: Python, Watson AI, Node Js
- Software Requirements: Watson Discovery, Watson Assistant, Cloud functions
- Project Deliverables: Intelligent Chatbot
- Project Duration: 1 Month

Purpose

The purpose of this project is to build a intelligent chatbot which can answer all the questions of a customer. This will help in lowering the barrier of the questions which the normal chatbots can't respond when given to the questions out of it's scope. Either it would return as immaterial or connect to the agent.
But in this project we will build a chatbot by training it from a document

which it will interpret in a smart way by using Watson Discovery and will be integrated with Watson Assistant by skills to respond in a perfect answer.

Scope

1. Create a customer care dialog skill in Watson Assistant.
2. Use Watson Discovery to build an enhanced smart document understanding.
3. Create an IBM Cloud Function action that allows Watson Assistant to post queries to Watson Discovery.
4. Build a web application with integration of all the services with node-red and deploy the same on IBM Cloud Platform.

Literature Survey

Existing Problem

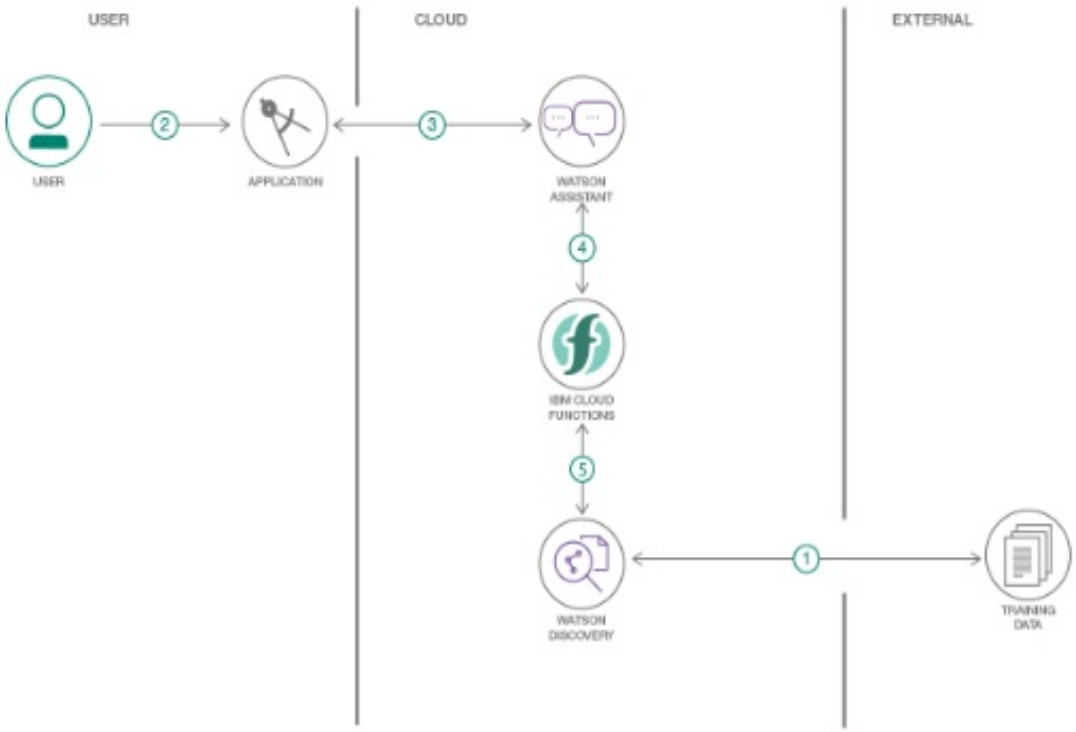
Generally a chatbot is built to answer only to specific questions of the customer and if asked the one's which are out of scope then it becomes obscure for them and marks as irrelevant or to a extent may acknowledge customer to seek out an agent help. But in this case every customer would be facing this problem and would be directly contacting to agenst which in turn increases traffic. This would create problems for agents to answer to each and every customer at a time. Hence we an efficient chatbot which reduces the task of an agent.

Proposed Solution

We have proposed a chatbot with help of which customers queries are directly processed by chatbot and there is no need to involve an agent. This is done by building a chabot which is feeded by a smart document with the help of Watson Discovery and given a flow by Watson Assistant skills which are managed by Cloud Functions and thus arre all integrated by Node-Red.

Theoretical Analysis

Block Diagram



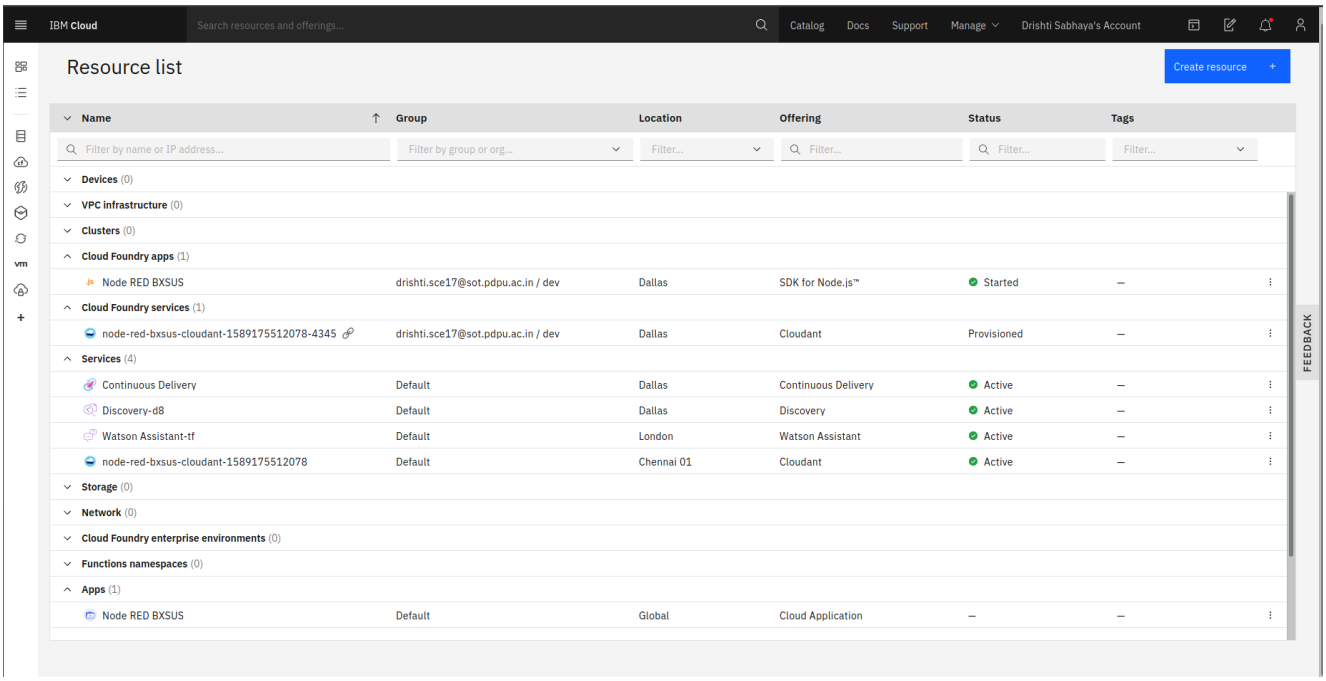
1. The document is annotated using Watson Discovery SDU.
2. The user interacts with the backend server via the app UI. The frontend app UI is a chatbot that is build by Node-Red.
3. Dialog and flows between the user and backend server is coordinated using a Watson Assistant dialog skill.
4. If the user asks the product operation question, a search query is passed to a predefined IBM Cloud Function action.
5. Cloud function action will query the Watson Discovery service and return the results.

Hardware/Software Design

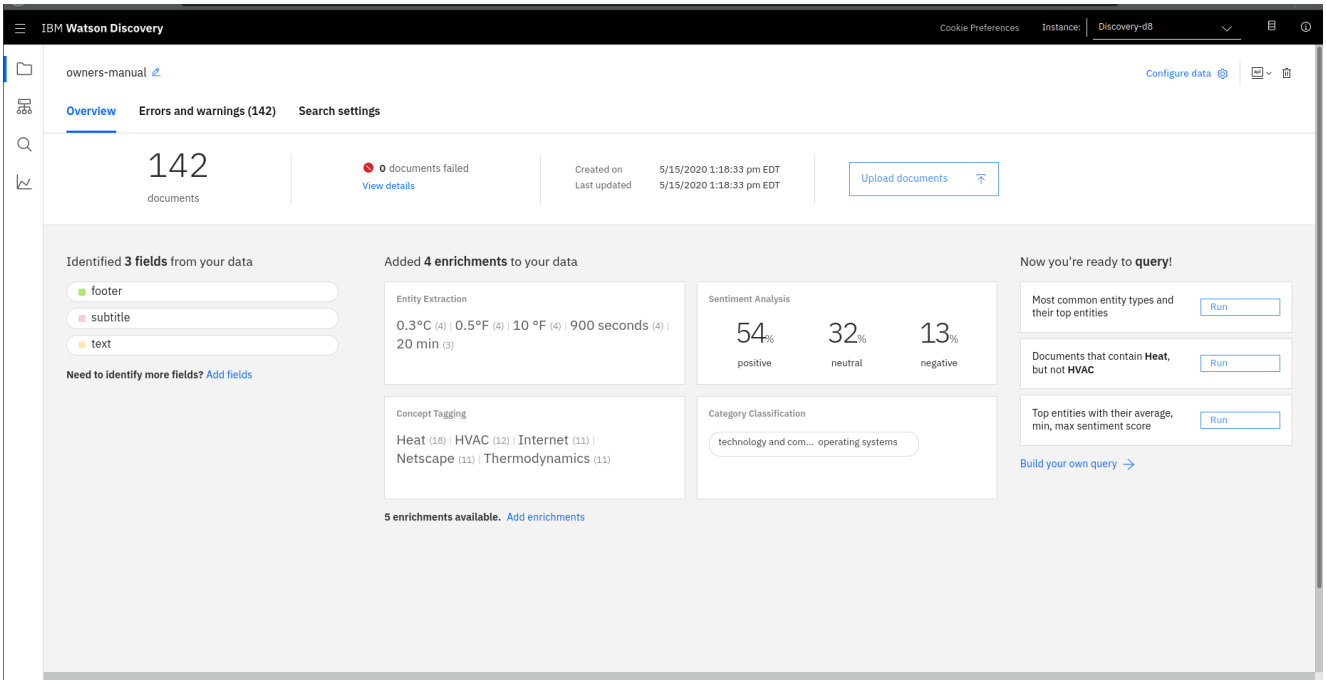
- IBM Cloud services
- Watson Discovery
- Cloud Functions
- Watson Assistant
- Node-Red

Experimental Investigation

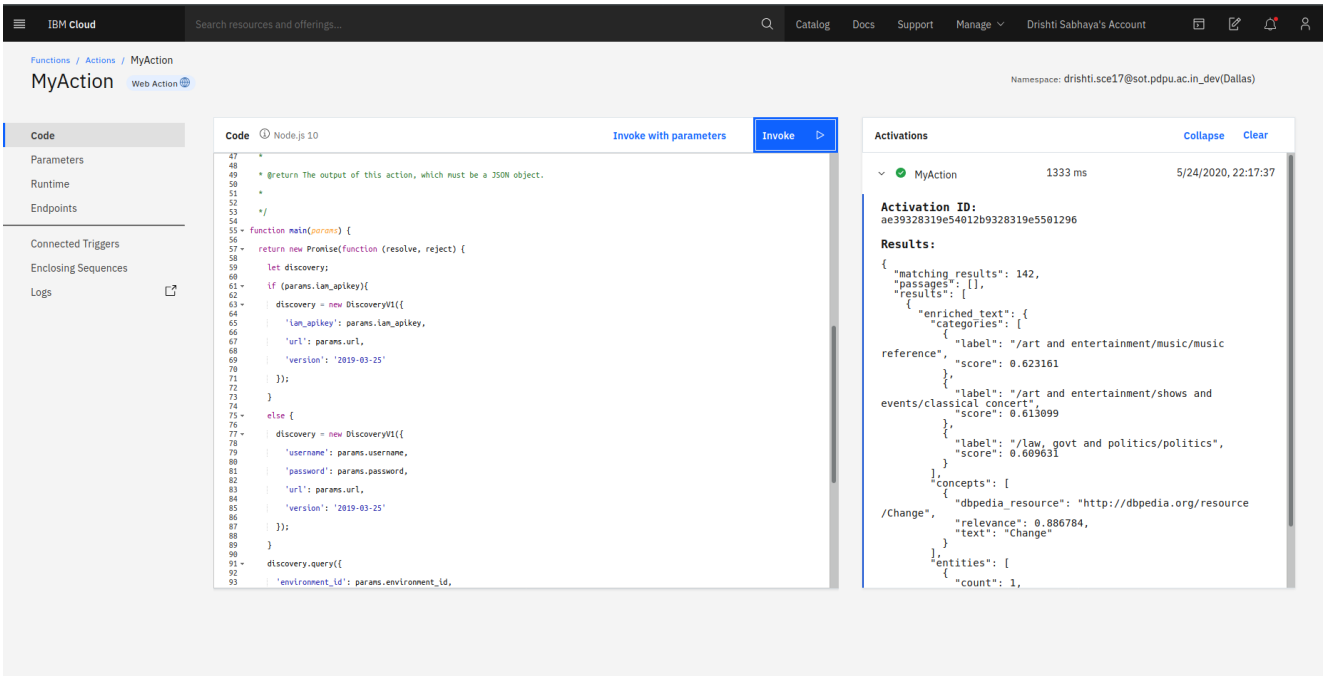
Create the necessary IBM cloud functions.



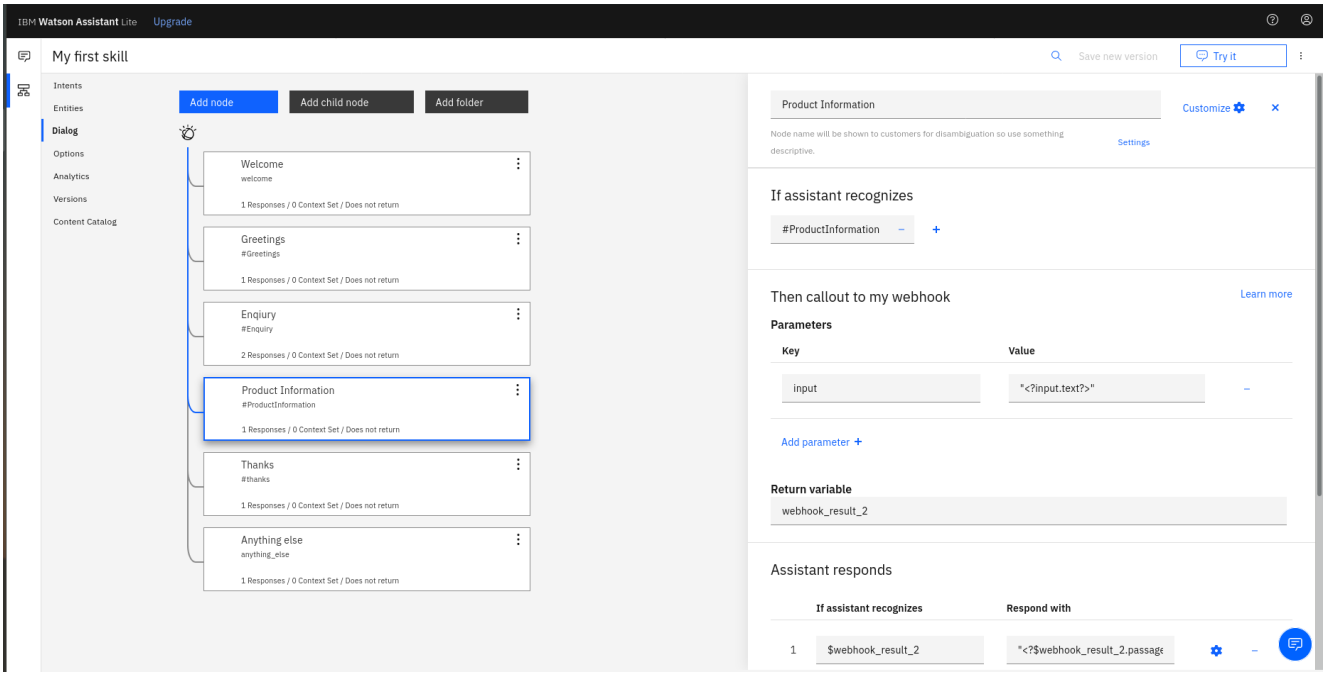
Create a Watson Discovery service and configure it.



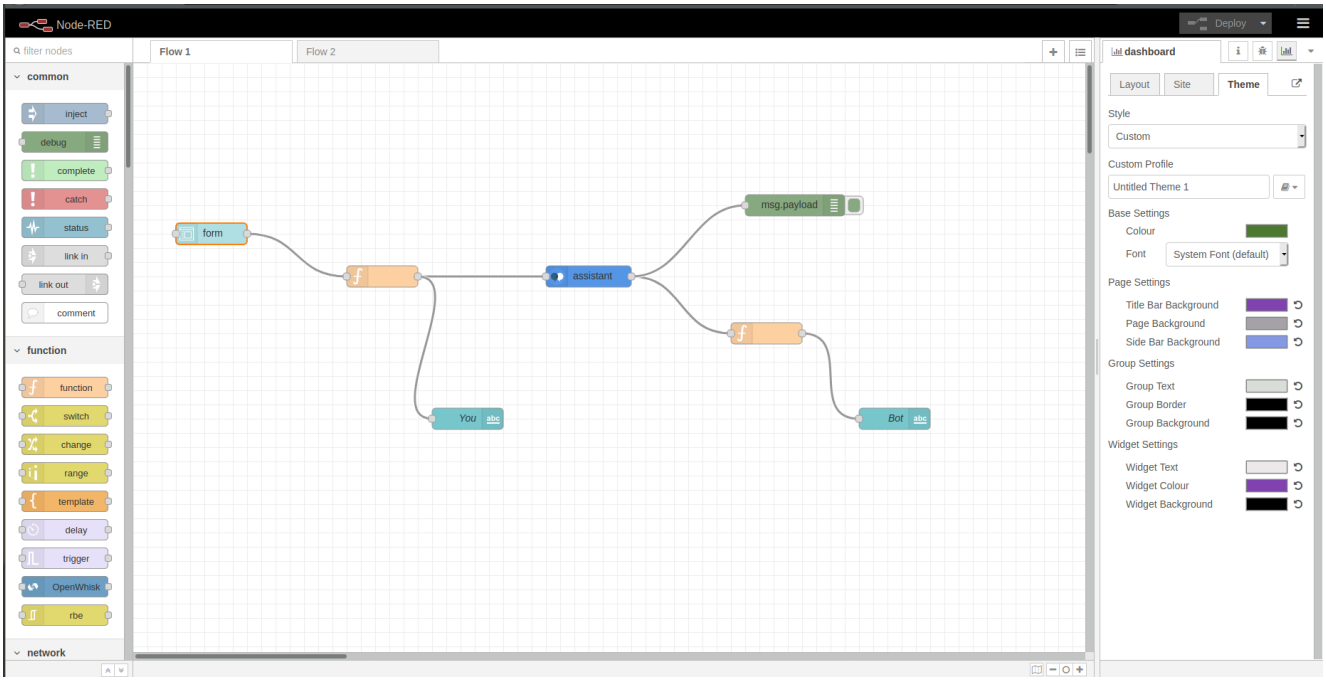
Implement Cloud Functions and write the necessary code for processing queries for Discovery.



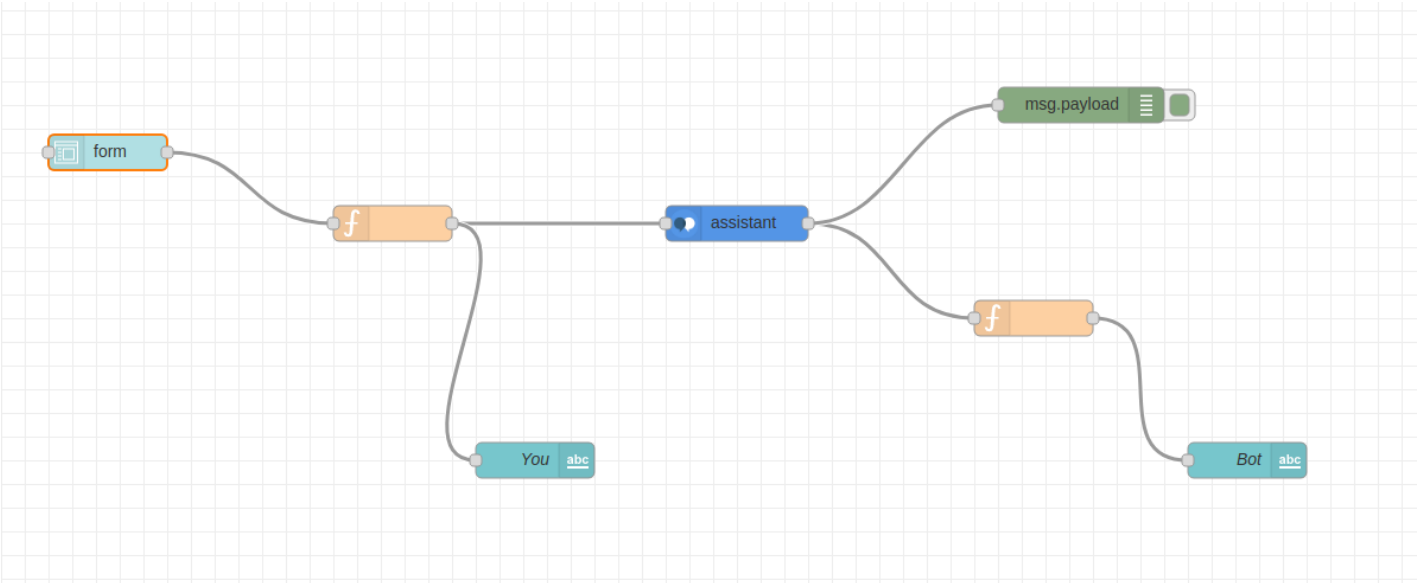
Create a Watson Assistant skill by adding intents, entities and dialog.



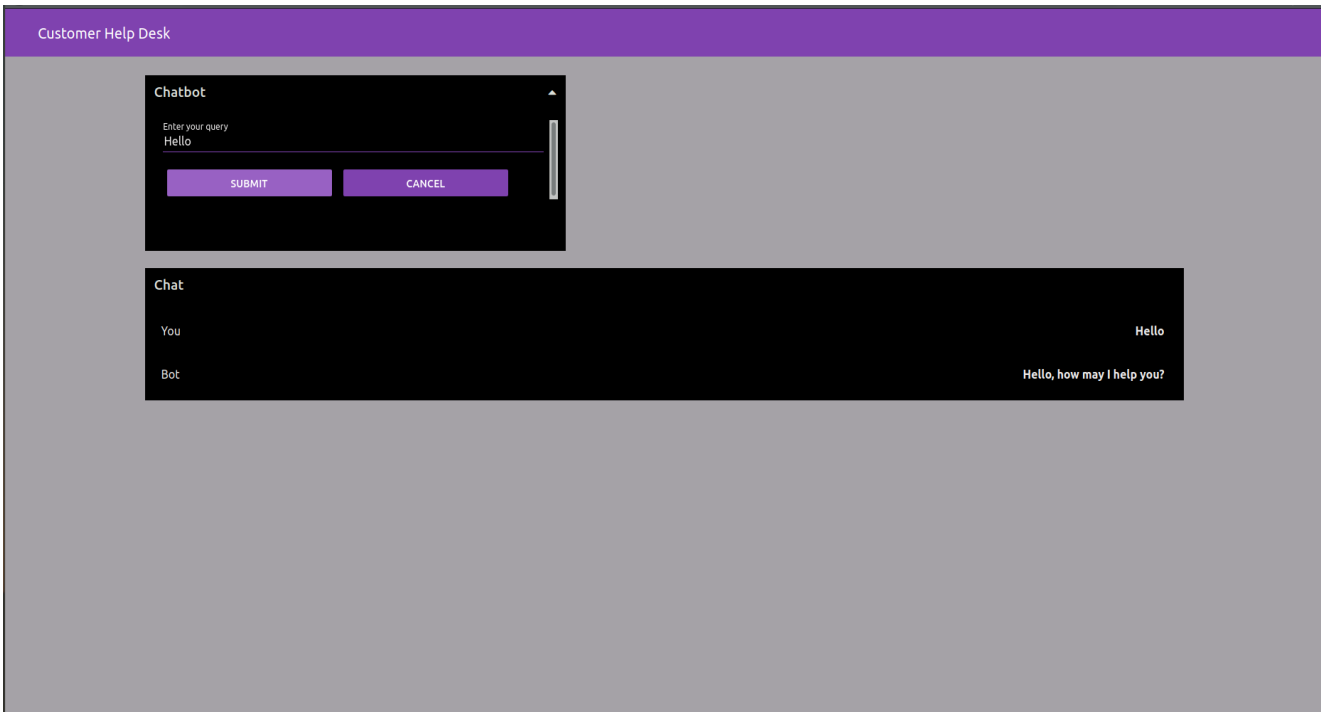
Create flow diagram by Node-Red dashboard and display the WEB UI.

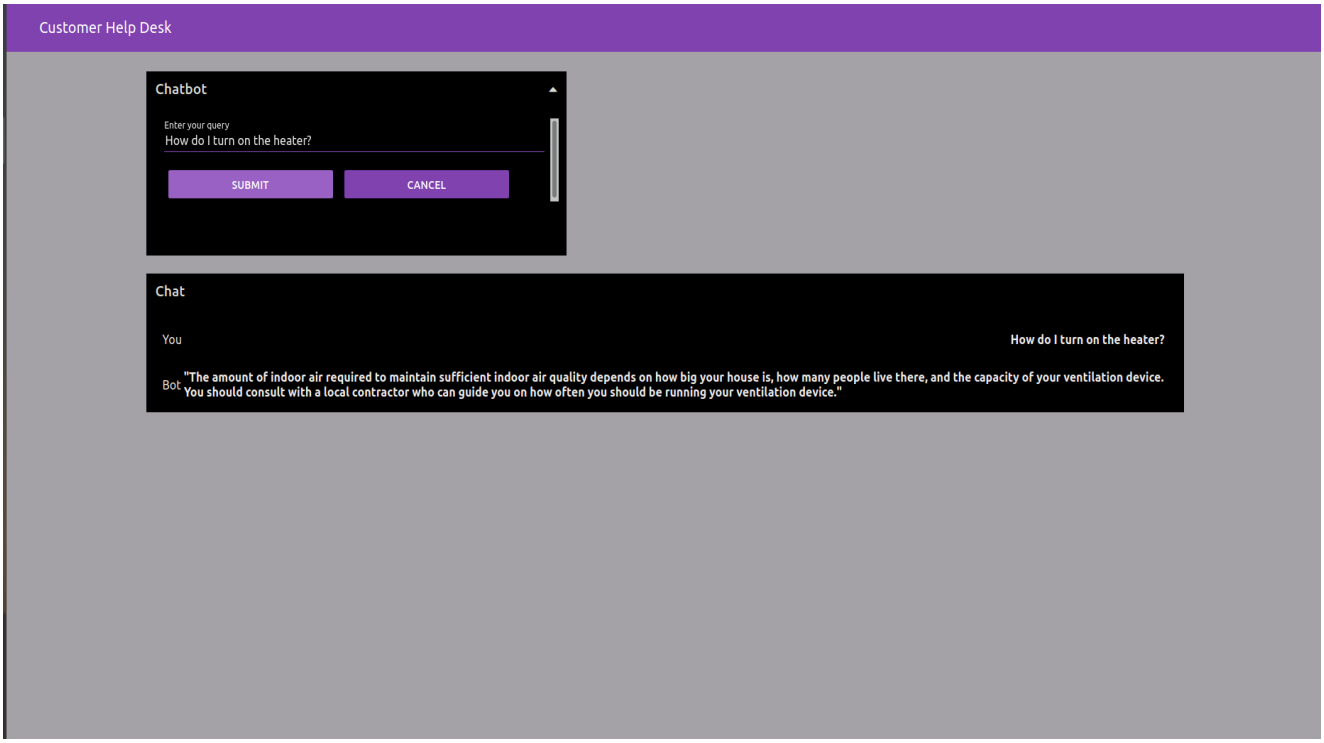


Flowchart



Result


Customer Help Desk
Chatbot
Enter your query
Hello
SUBMIT
CANCEL
Chat
You
Hello
Bot
Hello, how may I help you?


Customer Help Desk
Chatbot
Enter your query
How do I turn on the heater?
SUBMIT
CANCEL
Chat
You
How do I turn on the heater?
Bot
"The amount of indoor air required to maintain sufficient indoor air quality depends on how big your house is, how many people live there, and the capacity of your ventilation device. You should consult with a local contractor who can guide you on how often you should be running your ventilation device."

Advantages & Disadvantages

Advantages

1. Chatbots eliminate the requirement of any manpower during online interaction and are hence seen as a big advantage by companies receiving multiple queries at once.
2. Unlike humans, chatbots once installed can attend queries at any time of the day.
3. These type of chatbots are capable of learning from interactions and updating themselves on their own.
4. Humans have a limit to the number of clients they can handle at once. However, with chatbots, there is no such constraint and they can handle as many queries as required at once.
5. They are cost effective.

Disadvantages

1. Chatbots are often seen to be complicated and require a lot of time to understand user’s requirement.
2. Chatbots are installed with the motive to speed-up the response and improve customer interaction. However, due to limited data-availability and time required for self-updating, this process appears more time-taking and expensive.
3. Sometimes if not properly configured can return wrong results.

Applications

1. AI chatbots are quickly becoming a popular choice for many businesses.
2. It can be deployed in popular social media applications like Facebook, Slack and Telegram.
3. Tasks like hiring a cab, ordering food online, or even checking the weather can be easily accomplished via chatbots.
4. Customers who have completed a purchase are more likely to purchase again. Chatbots can interact with these customers and leverage the opportunity of upselling to them.

Conclusion

By following the above-mentioned steps, we can create a basic chatbot which can help us to answer the basic questions of the customer or user related to queries and the information about the product. We successfully created the intelligent helpdesk smart chatbot using Watson Assistant, Watson Cloud Function, Watson Discovery and Node-Red.

Future Scope

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 chats to the customer. We can also improve the results of discovery by enriching it with

more fields and doing the Smart Data Annotation more accurately.
We can get the premium version to increase the scope of our chatbot in terms of the call and requests. We can also include Watson text to audio and Speech to text services to access the chatbot handsfree. These are few of the future scopes which are possible.

Appendix

Source Code

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

Reference

- https://www.ibm.com/cloud/architecture/tutorials/cognitive_discovery
- http://www.iotgyan.com/learning-resource/integration-of-watson-assistant-to-node-red
- https://github.com/IBM/watson-discovery-sdu-with-assistan
- https://www.youtube.com/watch?v=Jpr3wVH3FVA