

# **Intelligent Customer Help Desk with Smart Document Understanding**

A project report submitted for

**Internship at smartinternz.com@2020**

**Category : Artificial Intelligence**

By

**Joyson Gaurea**  
([joysongaurea@gmail.com](mailto:joysongaurea@gmail.com))

## Table of Contents

<b>Sr. No</b>	<b>Title</b>	<b>Page</b>
<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview	<b>1</b>
1.2	Purpose	<b>2</b>
<b>2</b>	<b>Literature Survey</b>	<b>3</b>
2.1	Existing Problem	<b>3</b>
2.2	Proposed Solution	<b>3</b>
<b>3</b>	<b>Theoretical Analysis</b>	<b>5</b>
3.1	Block Diagram	<b>5</b>
3.2	Hardware/Software Designing	<b>6</b>
<b>4</b>	<b>Experimental Investigations</b>	<b>7</b>
<b>5</b>	<b>Flowchart</b>	<b>10</b>
<b>6</b>	<b>Result</b>	<b>11</b>
<b>7</b>	<b>Advantages and Disadvantages</b>	<b>16</b>
<b>8</b>	<b>Applications</b>	<b>18</b>
<b>9</b>	<b>Conclusion</b>	<b>20</b>
<b>10</b>	<b>Future Scope</b>	<b>21</b>
<b>11</b>	<b>Bibliography</b>	<b>22</b>
<b>12</b>	<b>Appendix</b>	<b>24</b>
A	Source Code	<b>24</b>

# 1. Introduction

## 1.1 Overview

We will be able to build an application that leverages multiple IBM Cloud services such as Watson Assistant, Discovery, Cloud Functions, Node Red etc. By the end of this project. This project will help us learn best practices of combining Watson services and how they can build interactive information retrieval system.

### Project Requirements:

1. Functional Requirements : IBM Cloud Account, Internet.
2. Technical Requirements : Artificial Intelligence (Machine Learning).
3. Software Requirements : Any operating system, Web browser.
4. Project duration : 30 days.
5. Project team : Joyson Gaurea.

## **1.2 Purpose**

The purpose of this project is to provide a customer care chatbot to the end users of the project which can solve the customer queries hence saving the time and efforts of an employee manually trying to answer each of the customers. The chatbot would also incorporate smart document understanding using which the chatbot would scan the user manual of a product and reply the customer with the relevant piece of information that would otherwise require the customer to manually go through the user manual thereby providing a customer friendly service.

## **2. Literature Survey**

### **2.1 Existing 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. Often each employee is worth to a company and they can't afford for employees to spend all their time replying to customer queries instead of the work at hand.

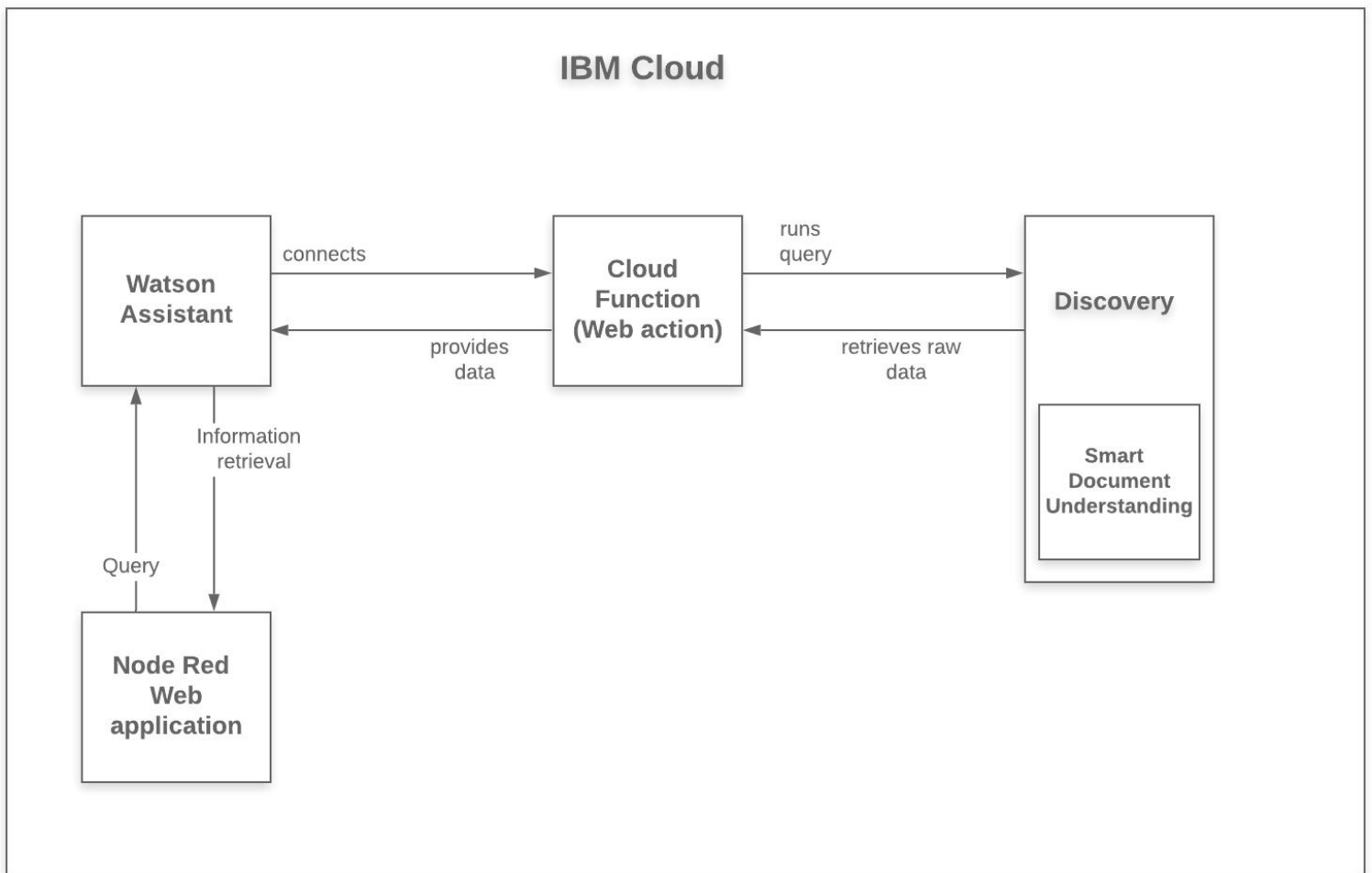
### **2.2 Proposed Solution**

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. To take it a step further, the project shall use the Smart Document Understanding

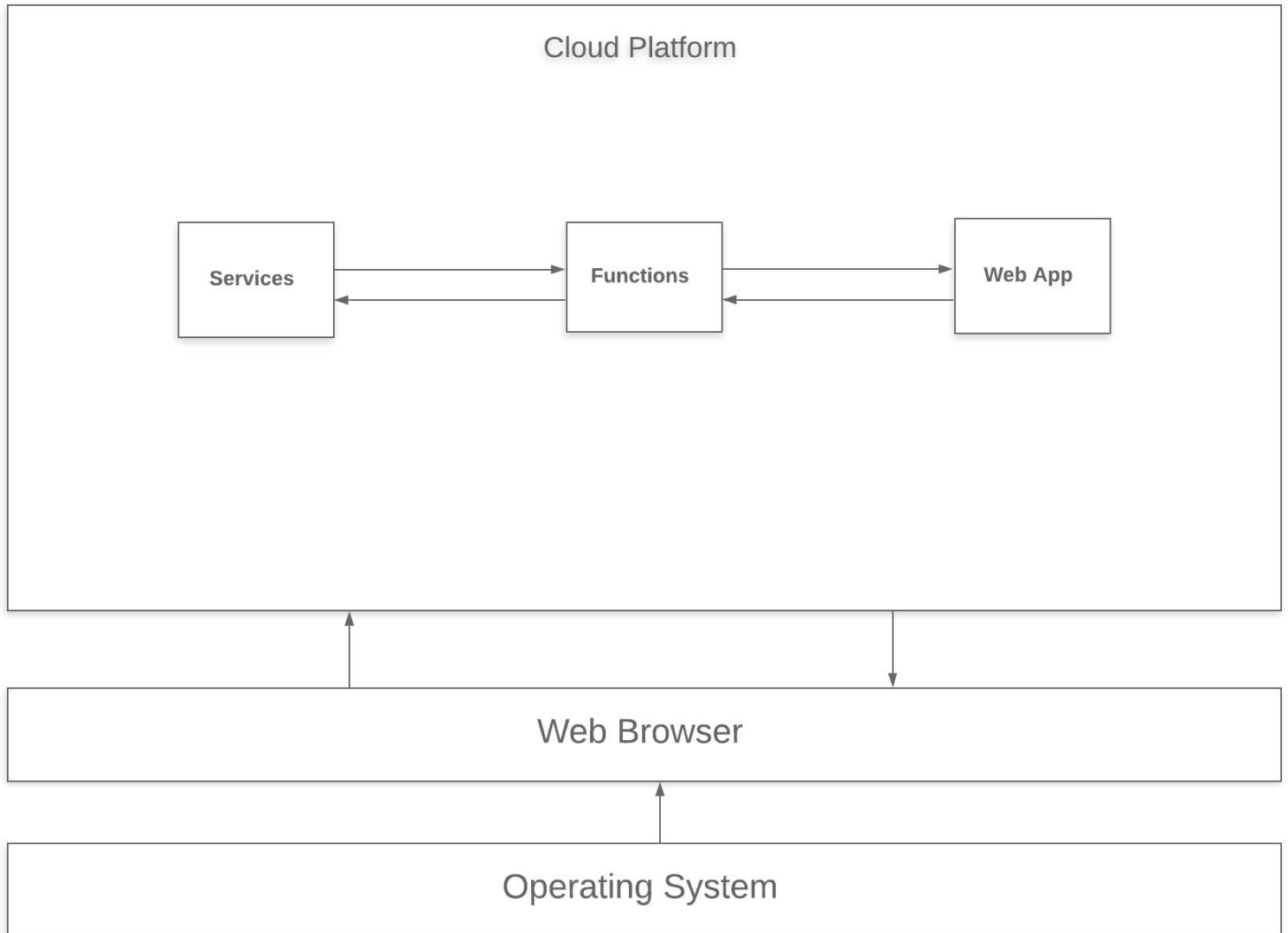
feature of Watson Discovery to train it on what text in the owners manual is important and what is not. This will improve the answers returned from the queries.

### 3. Theoretical Analysis

#### 3.1 Block Diagram



### 3.2 Software designing





## 4. Experimental Investigation

### **Non-Functional Requirements:**

Response time: The UI load time must not be more than 2 secs for users that access the website using an LTE mobile connection.

Throughput: The application should be capable enough to process 10,000 user requests per minute without affecting its performance.

Reliability: The database update process must roll back all related updates when any update fails.

Availability: New module deployment mustn't impact front page, product pages and check out pages availability and mustn't take longer than one hour. The rest of the pages that may experience problems must display a notification with a timer showing when the system is going to be up again.

## **Feasibility study:**

### Technical feasibility:

Intelligent Customer Help Desk with Smart Document Understanding is a complete cloud based application. The main technologies and tools that are associated with it are: HTML, CSS, JavaScript, IBM Cloud, diagram tools: LucidChart, Draw.io. Each of the technologies are freely available and the technical skills required are manageable. Time limitations of the product development and ease of implementing using these technologies are synchronized. Initially the website will be hosted on a free web hosting space, but for later implementation it will be hosted in a paid web hosting space with a sufficient bandwidth. Bandwidth required will be low since it doesn't require any multimedia aspect. Hence it proves that project OGMS is technically feasible.

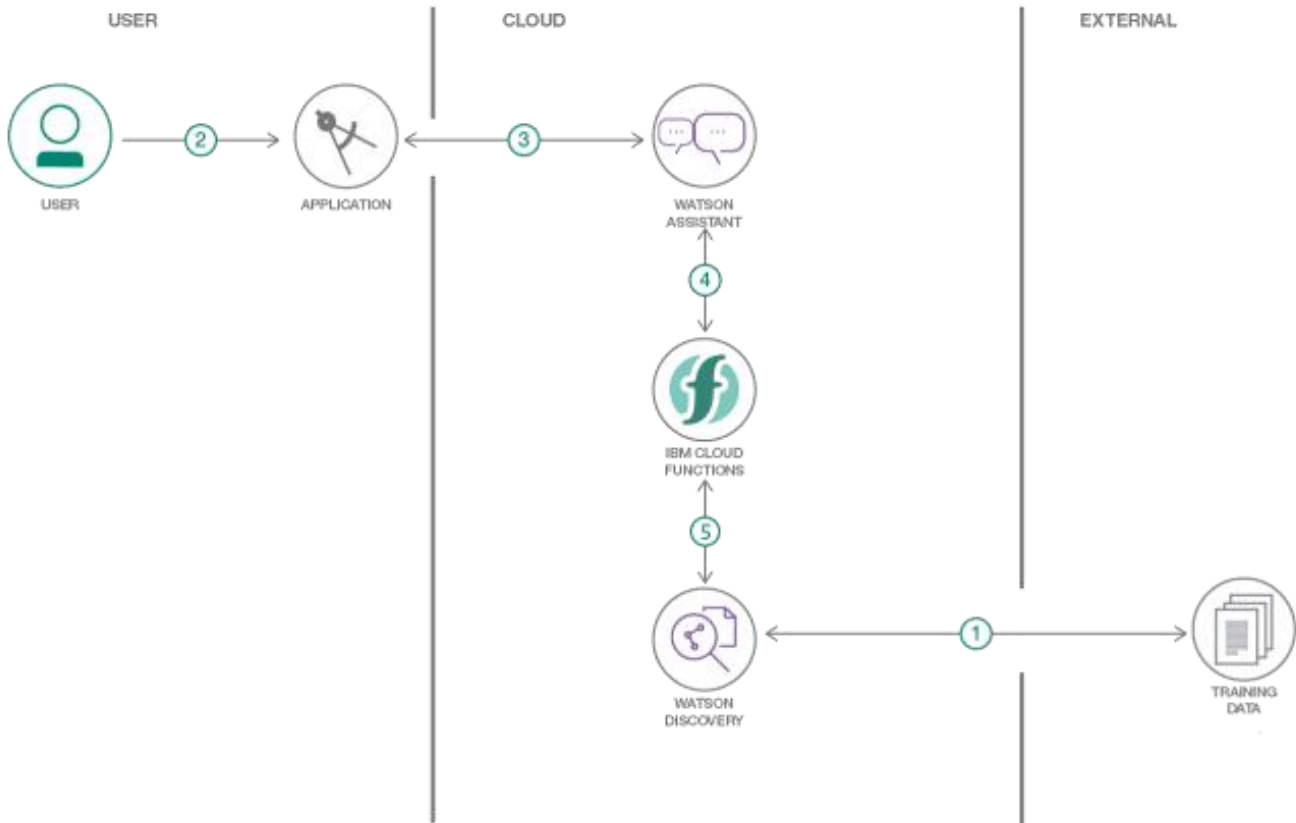
### Time feasibility:

Every software development process model has an associated time period for development cycle. These allotted time periods can be minimum 2 weeks. A minimum viable product (MVP) can be designed in a day and improvements can be done in future regularly. Hence it proves that the proposed system has time feasibility.

### Economical feasibility:

Since it is a web/ cloud application, Intelligent Smart Help Desk with Smart Document Understanding will have an associated hosting cost. The free tier for IBM Cloud however provides a heavy duty service but it won't be ideal for production use. Since the system doesn't consist of any multimedia data transfer, bandwidth required for the operation of this application is very low. The system will follow the freeware software standards. No cost will be charged from the potential customers. Bug fixes and maintaining tasks will have an associated cost. Besides the associated cost, there will be many benefits for the customers. Hence it proves that the project is economically feasible.

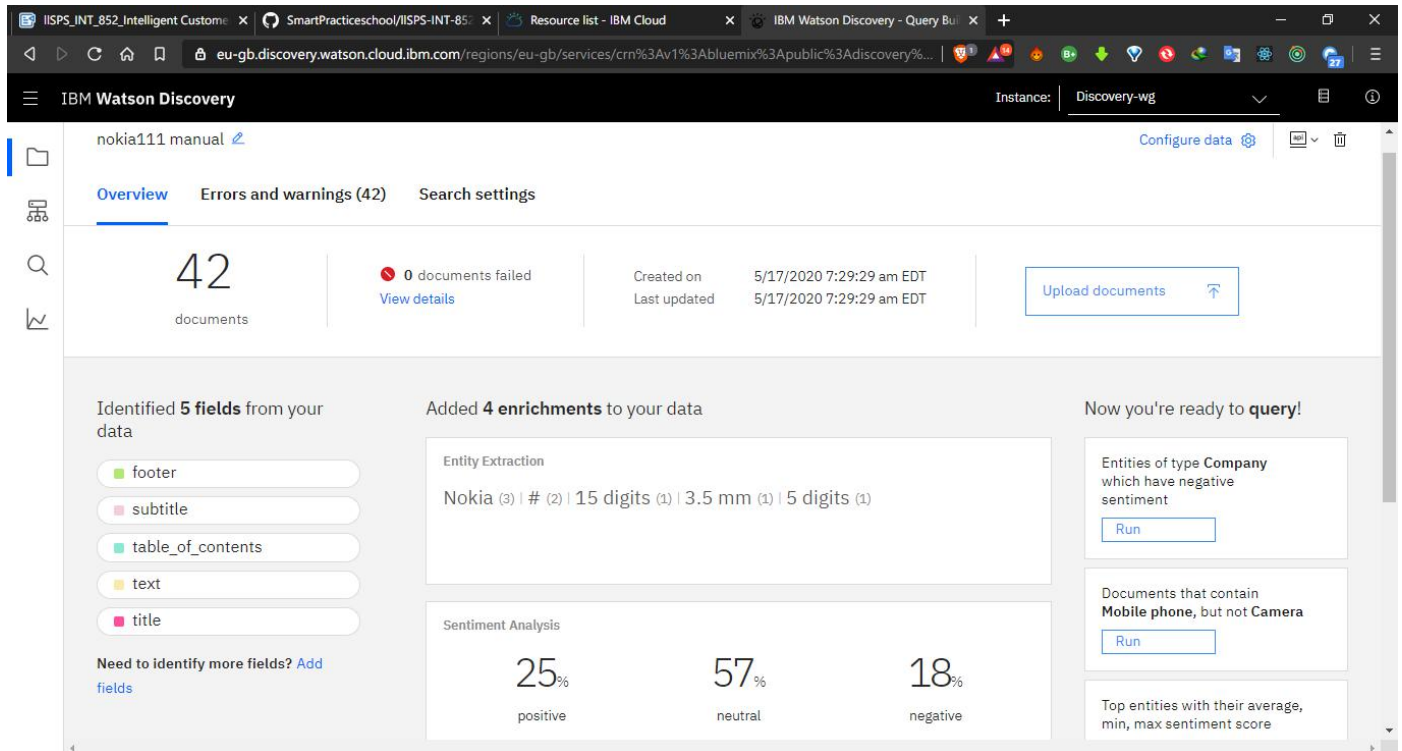
## 5. Flowchart



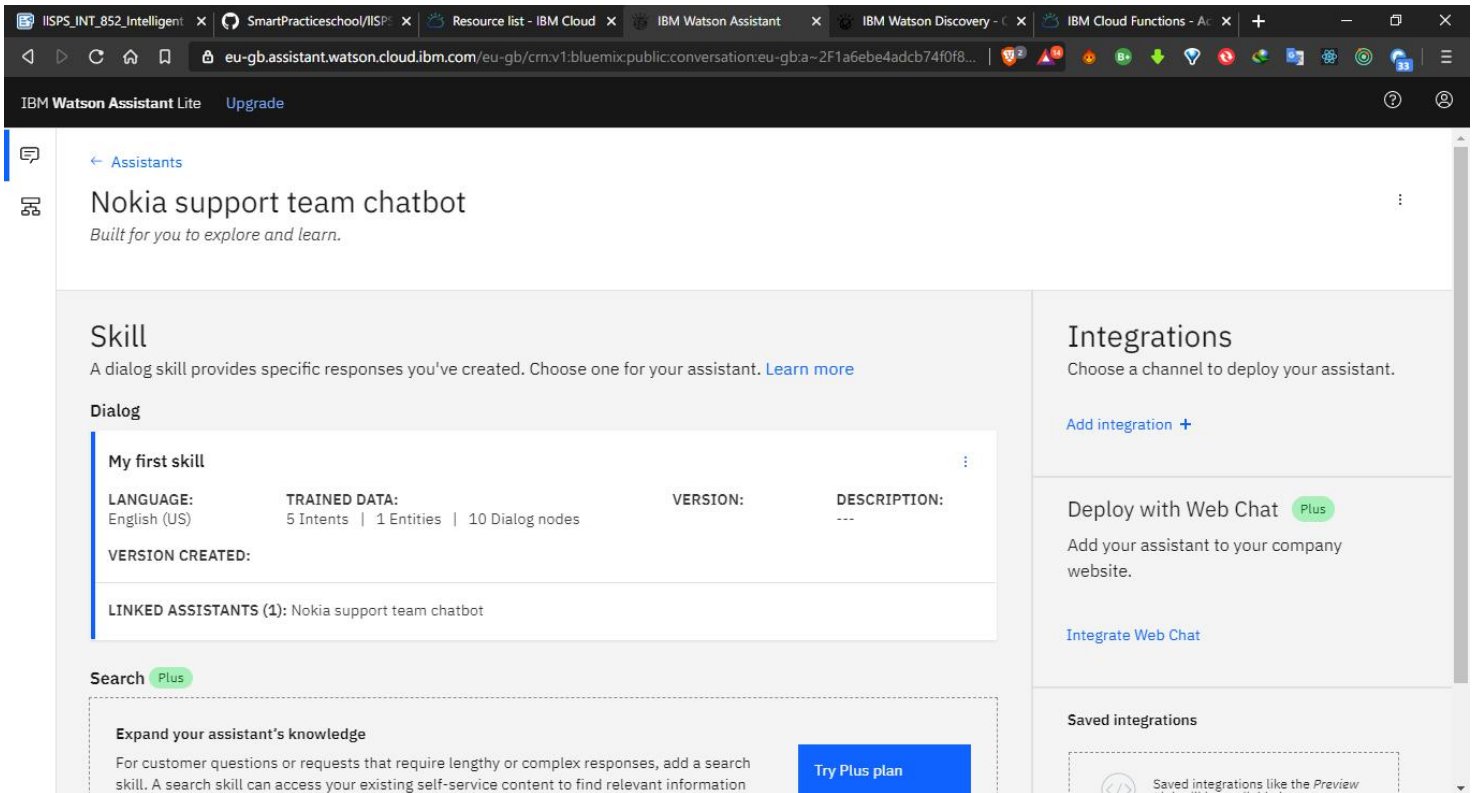
1. The document is annotated using Watson Discovery SDU.
2. The user interacts with the backend server via the app UI. The frontend app UI is a chatbot that engages the user in a conversation.
3. Dialog between the user and backend server is coordinated using Watson dialog skill.
4. If the user asks a product operation, a search query is passed to a predefined IBM Cloud functions action.
5. The Cloud Functions action will query the Watson Discovery service and returns the results.

## 6. Results

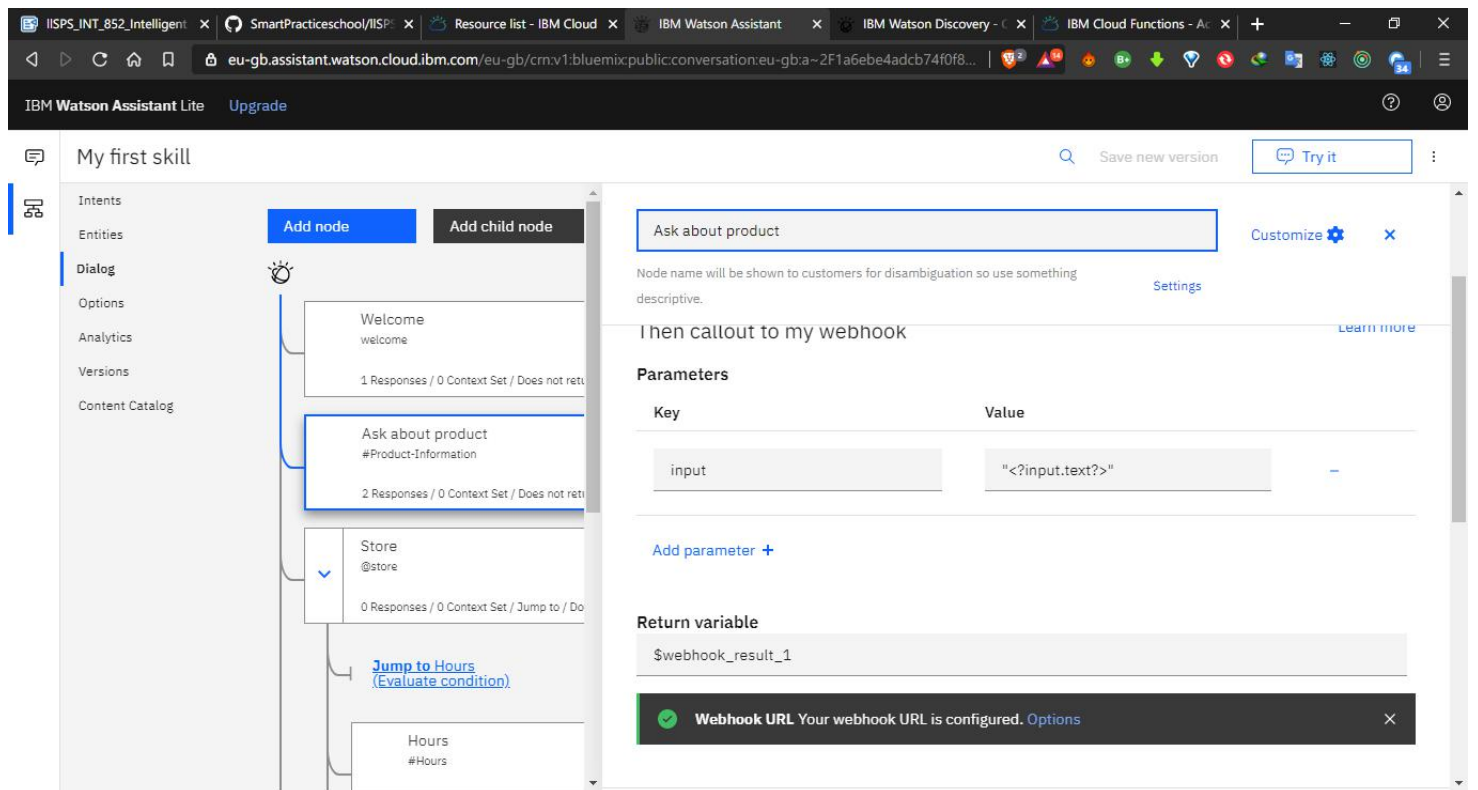
The successful implementation screenshots are shown below:



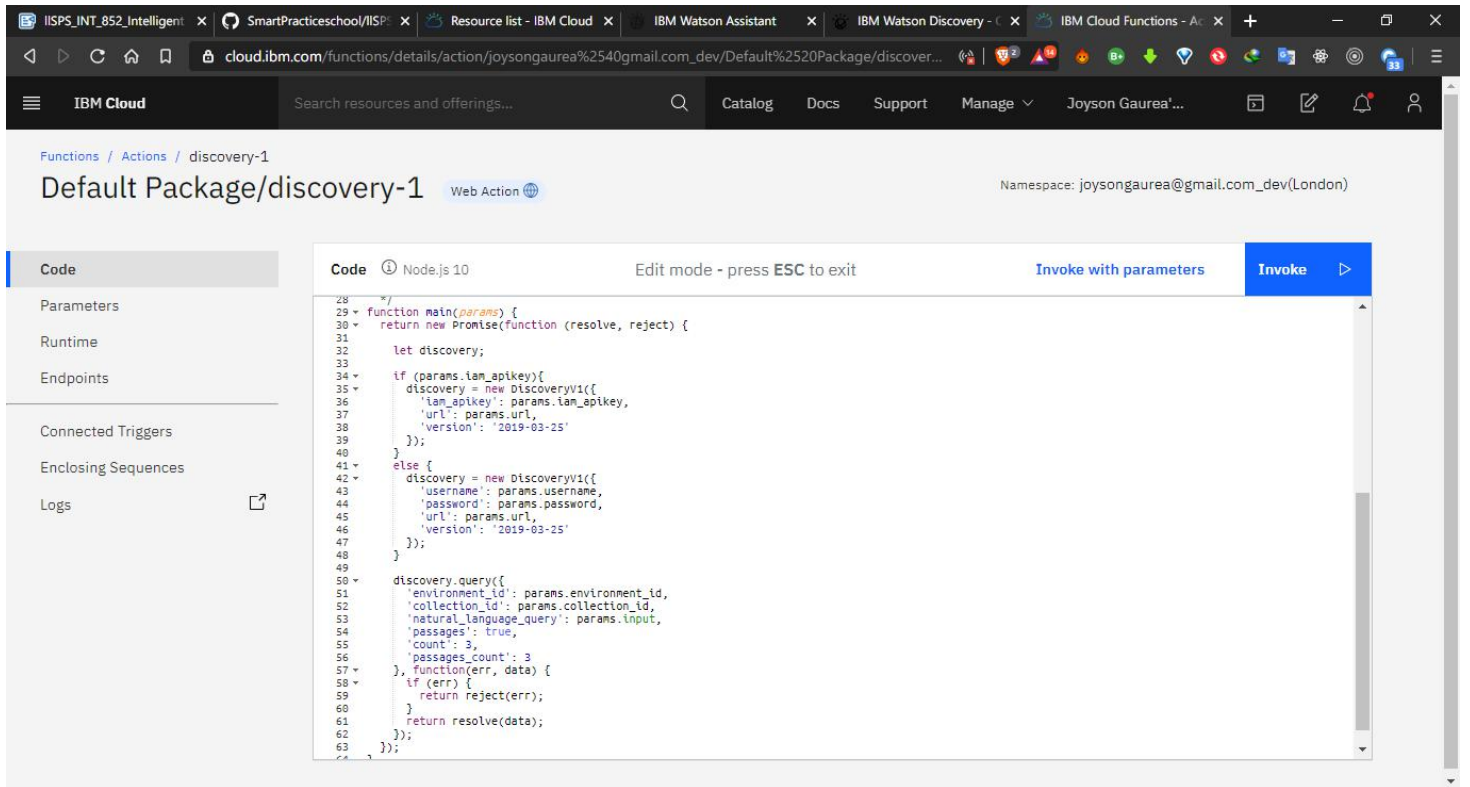
(Figure : IBM Discovery)



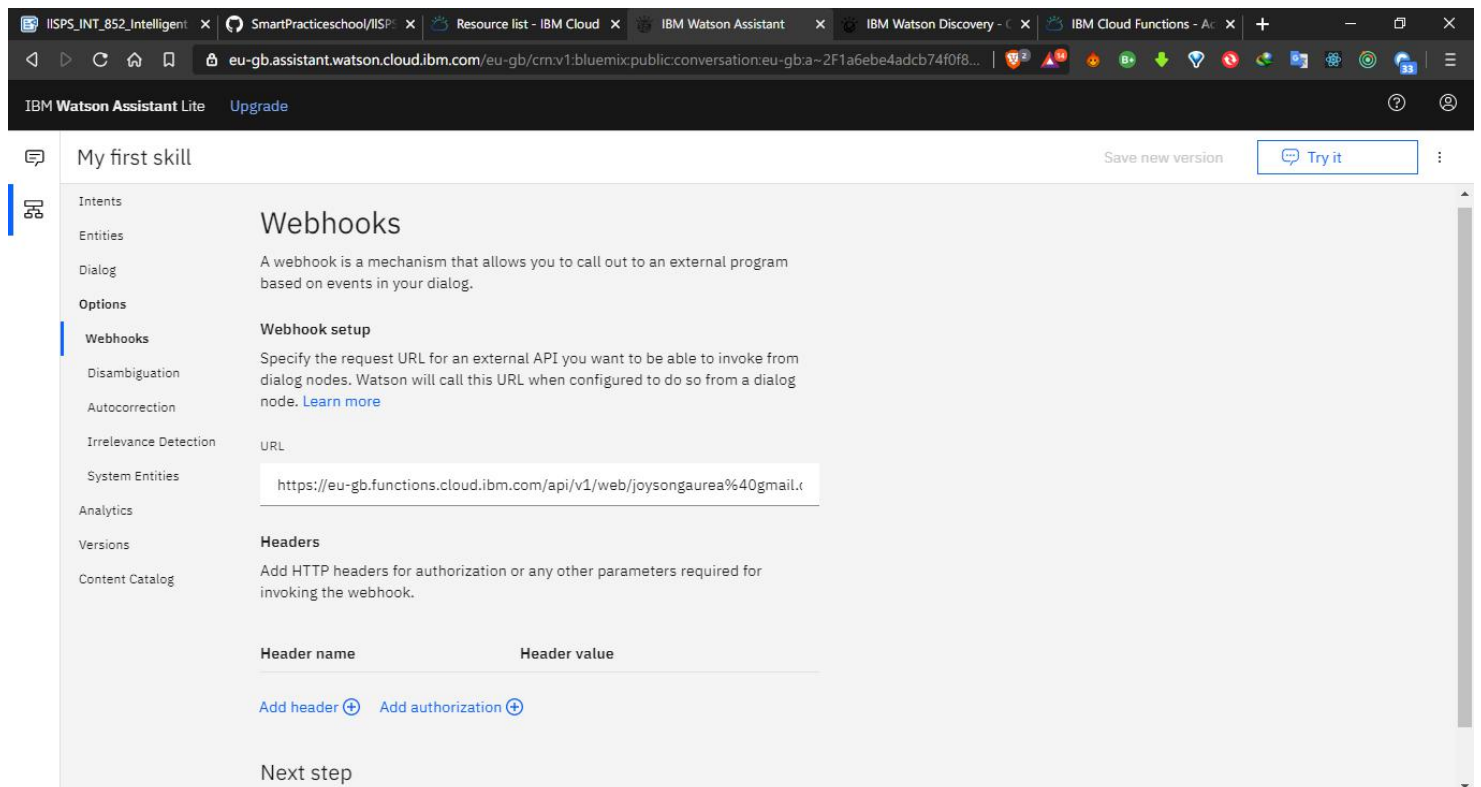
(Figure : Watson Assistant)



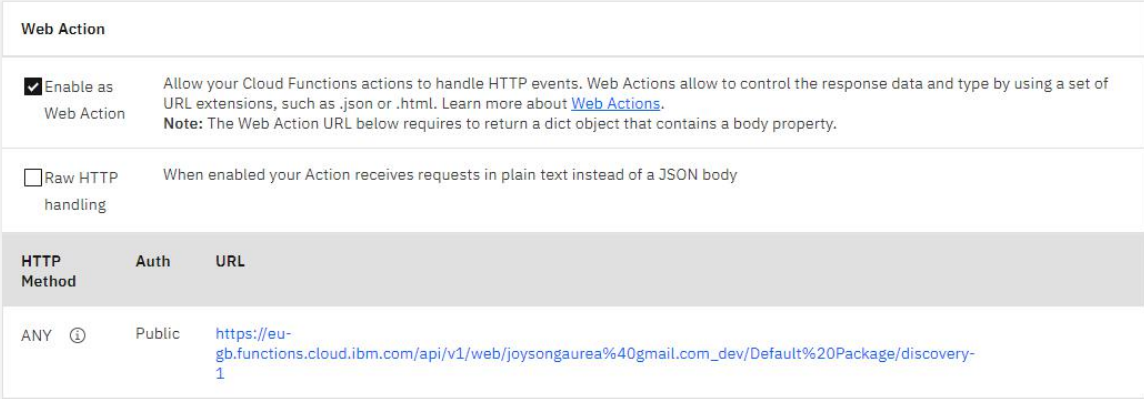
(Figure : Watson Assistant Dialog Skill)



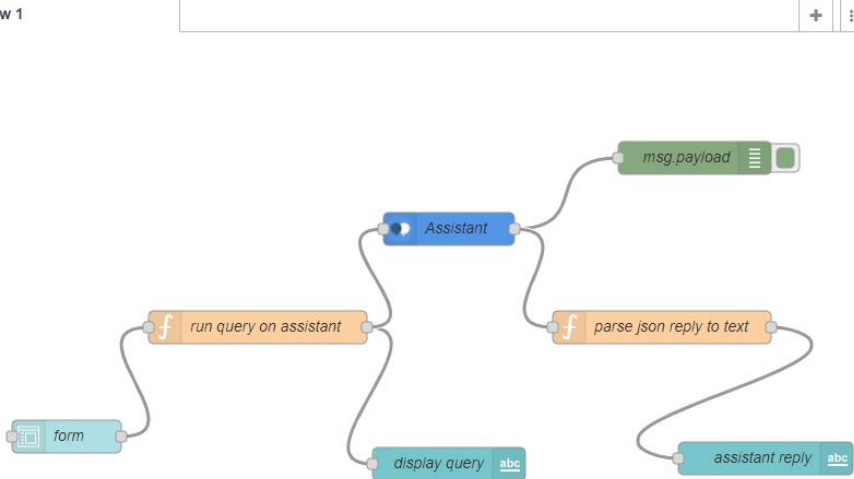
(Figure : Web action code)



(Figure : Webhook)

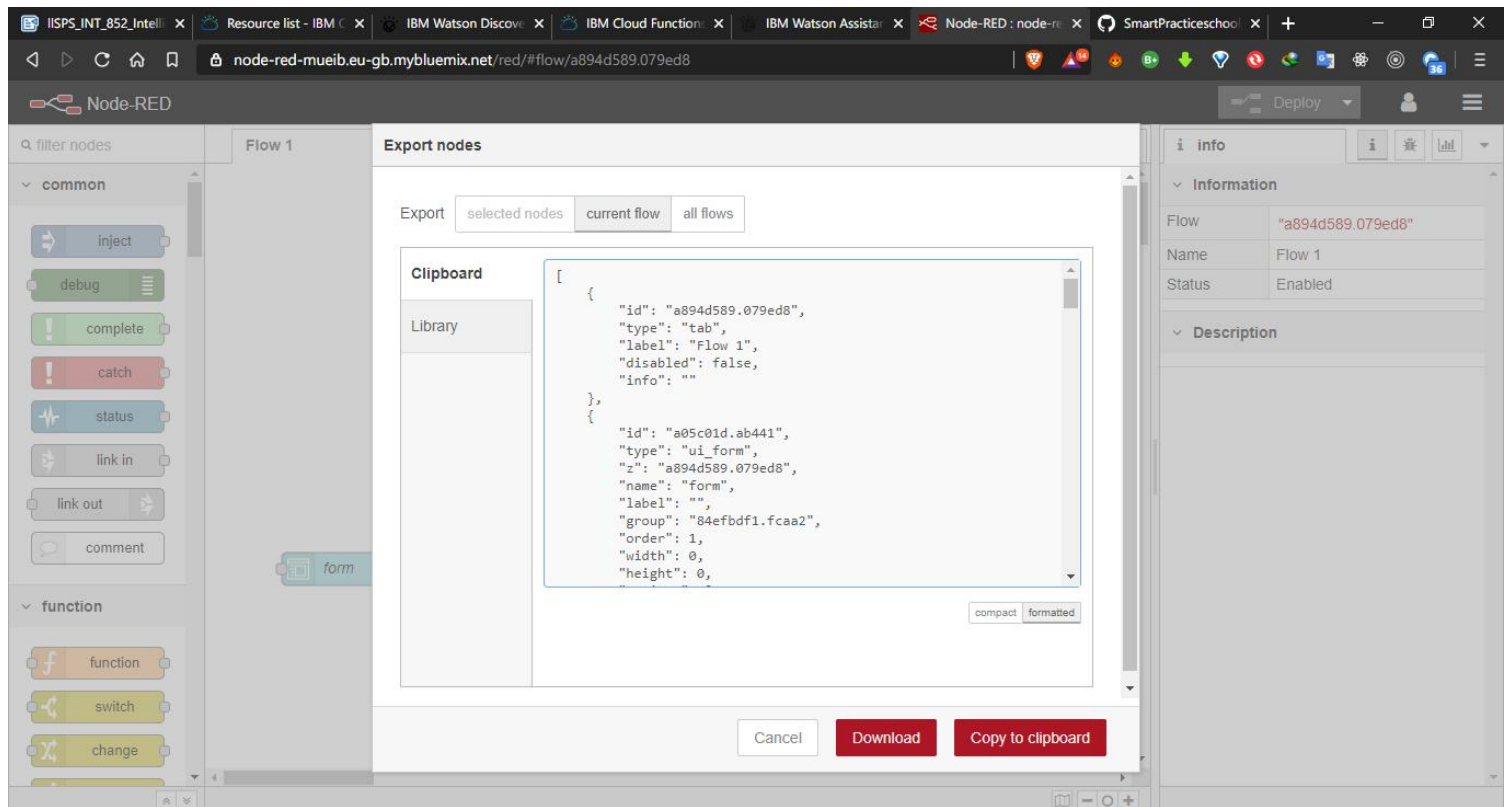


(Figure : Web action endpoint)

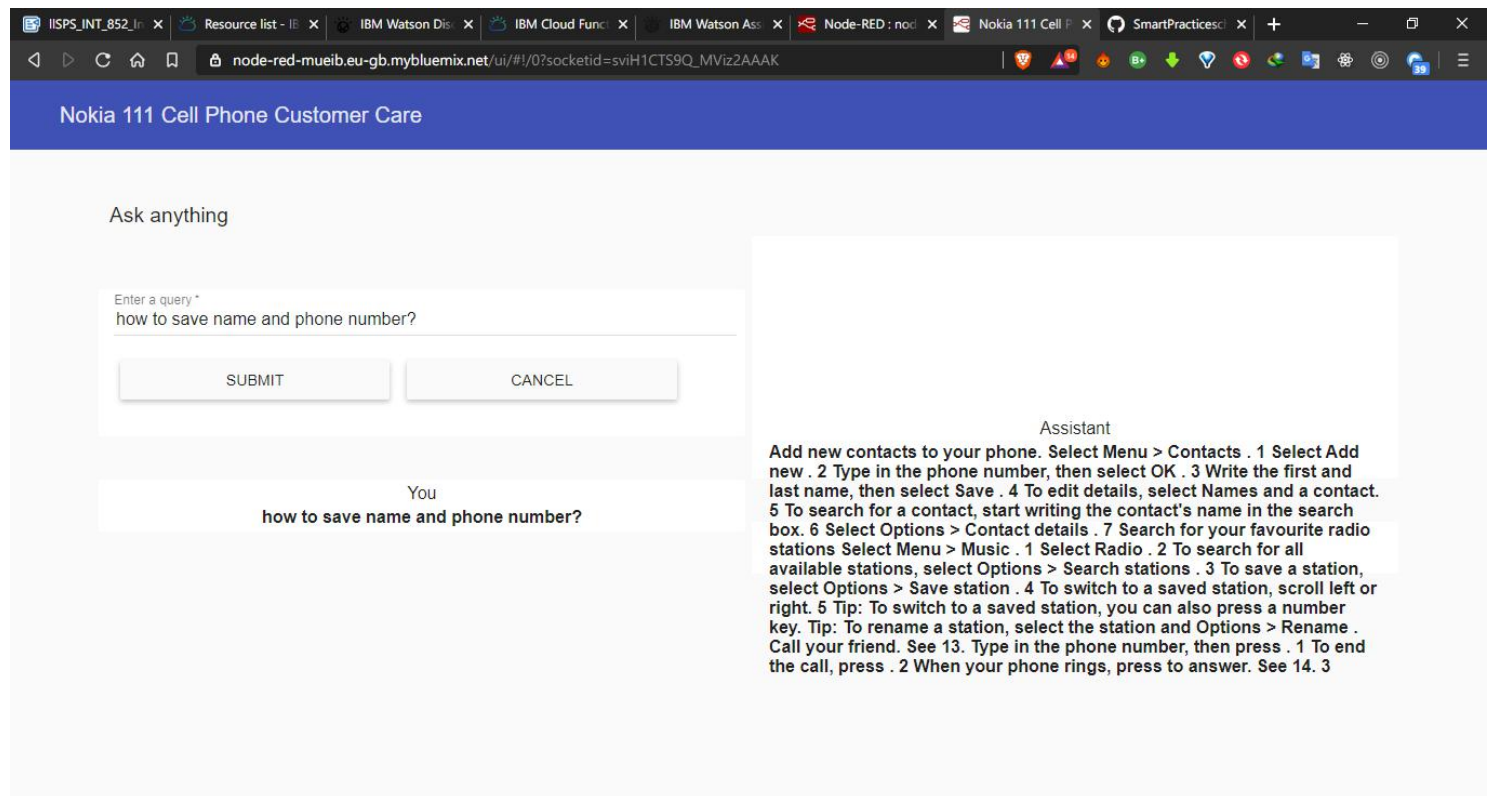


(Figure : Node-red flow)





(Figure : Node Red flow code)



(Figure : Node red UI)

## 7. Advantages and Disadvantages

### Advantages

Traditional search can't fully understand the structure of your complex business documents because they contain industry-specific content and exist in different formats – like PDFs, Excel tables, and PowerPoints – that are visually difficult for traditional search to interpret. With Smart Document Understanding (SDU), you can visually label the text within your enterprise documents (from headers and footers, to tables, and so on) allowing Watson Discovery to surface the exact answer from a body of text or section. Thanks to machine learning, once you train SDU on a few pages, you can begin to apply that understanding to hundreds.

Your search tool needs to continue to learn and improve over time. Watson Discovery's Integrated Machine Learning capabilities allow it to continuously improve from user interactions and enables users to create and deploy custom machine learning models based on domain expertise faster than ever before. Watson Discovery Machine Learning features include relevancy ranking, text analytics, custom model optimization, and more.

Most chatbots frustrate users when asked detailed questions, relaying that they don't understand and can't provide an accurate answer. Watson Discovery's integration with IBM's award-winning conversational AI, Watson Assistant, enables users to retrieve both pre-trained answers as well as more complex content found in their knowledge bases, product manuals, project reports, and more. It's so much more than a chatbot.

### **Disadvantages**

1. It's available only in English (limits areas of use).
2. It is often seen as a disruptive technology.
3. High maintenance.
4. Doesn't process structured data directly.
5. Increasing rate of data with limited resources.
6. High switching costs
7. Takes time to integrate Watson and its services into a company.
8. Targeting towards bigger organisations that can afford Watson.
9. Takes time and effort to teach Watson in order to use it to its full potential.

## 8. Applications

Businesses today have a lot of data and that too data in the cloud, data on premises, unstructured data, structured data. It's not uncommon for enterprise AI training plans to contain terabytes of documents that must be ingested, annotated and the enrichments tested before useful results can be delivered. Search processes like that can consume a lot of resources before beginning to generate value. If only there was a way to make search intelligent. By intelligent search we mean, if a technology can analyse heaps of documents and retrieve the most relevant data within a fraction of second. This is where IBM Discovery with smart document understanding comes into play which can be accessed with Watson Assistant.

This chatbot can be deployed as a standalone app which may no longer need a customer to manually visit the company's website.

More models of the device can be added which will require user manuals and other documents regarding those devices to be used by Watson Discovery and Smart Document Understanding thereby increasing the support for a larger customer base.

The chatbot can be used to get customer sentiment insights from product reviews.

## 9. Conclusion

In this project, we used the IBM Watson chatbot experience and instead of relying on predefined responses, our Watson dialog skill provided a web hook that calls out to other IBM Watson services for additional sources of information. In this case it was an owners manual for the cell phone Nokia 111 that has been uploaded into Watson Discovery. We used Smart Document Understanding feature of Watson Discovery wherein we specified each section of the document initially while the ML model learnt and implemented the rest of the section identification. We used a Node Red application to integrate all these services and created a UI dashboard for the customer to use the chatbot.

## 10. Future Scope

This chatbot can be deployed as a standalone app which may no longer need a customer to manually visit the company's website.

As of now, the chatbot supports only one device Nokia 111, the support for more devices can be added by adding more user instruction manual or related documents or any form of data for the discovery service. In this way the chatbot can support wide variety of customers.

The chatbot can be used to get customer sentiment insights from product reviews.

The chatbot can be further implemented as a progressive web app for widening its reach with customers alongwith offline features.

The chatbot will also display queries and replies of other users with a similar intent and hence enhancing customer's expected results.

## 11. Bibliography

### Internet Documents :

[1] Create a Node-RED starter application available at <https://developer.ibm.com/components/node-red/tutorials/how-to-create-a-node-red-starter-application/>

[2] Node-RED website available at <https://nodered.org/>

[3] Getting started with Watson Assistant available at <https://developer.ibm.com/components/watson-assistant/series/learning-path-watson-assistant>

[4] Introduction to Watson Discovery available at <https://developer.ibm.com/articles/introduction-watson-discovery/>

[5] Watson Discovery documentation available at <https://cloud.ibm.com/docs/services/discovery?topic=discovery-getting-started>



[6] Node-RED Watson Discovery Chatbot Telegram available at <https://medium.com/@rimaibrahim/node-red-watson-discovery-chatbot-telegram-ce616ddcd0d9>

[7] RESTful Web Services available at <https://developer.ibm.com/technologies/web-development/articles/ws-restful>

[8] IBM Cloud Functions documentation available at <https://cloud.ibm.com/docs/openwhisk?topic=cloud-functions-getting-started>

## 12. Appendix

### Source Code

IBM Cloud functions web action 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);
    });
  });
}

```

Node-RED flow code :

```
[
  {
    "id": "a894d589.079ed8",
    "type": "tab",
    "label": "Flow 1",
    "disabled": false,
    "info": ""
  },
  {
    "id": "a05c01d.ab441",
    "type": "ui_form",
    "z": "a894d589.079ed8",
    "name": "form",
    "label": "",
    "group": "84efbdf1.fcaa2",
    "order": 1,
    "width": 0,
    "height": 0,
    "options": [
      {
        "label": "Enter a query",
        "value": "query",
        "type": "text",
        "required": true,
        "rows": null
      }
    ],
    "formValue": {
      "query": ""
    },
    "payload": "",
    "submit": "submit",
    "cancel": "cancel",
    "topic": "",
  }
]
```

```

    "x": 120,
    "y": 370,
    "wires": [
      [
        "a3211f98.cdb8c"
      ]
    ]
  },
  {
    "id": "a3211f98.cdb8c",
    "type": "function",
    "z": "a894d589.079ed8",
    "name": "function",
    "func": "msg.payload = msg.payload.query;\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 285,
    "y": 275,
    "wires": [
      [
        "162d311d.e82e6f",
        "455a7552.6d669c"
      ]
    ]
  },
  {
    "id": "455a7552.6d669c",
    "type": "watson-conversation-v1",
    "z": "a894d589.079ed8",
    "name": "Assistant",
    "workspaceid": "9e3b568e-b59a-4508-b81d-4ed7273f0407",
    "multiuser": false,
    "context": false,
    "empty-payload": false,
    "service-endpoint": "",
    "timeout": "",

```

```

    "optout-learning": false,
    "x": 470,
    "y": 180,
    "wires": [
      [
        "5a78a9a3.f9a118",
        "48eeafb6.5e146"
      ]
    ]
  },
  {
    "id": "48eeafb6.5e146",
    "type": "function",
    "z": "a894d589.079ed8",
    "name": "function",
    "func":
      "msg.payload.text=\"\";\nif(msg.payload.context.webhook_result_1)
{\n  for(var i in msg.payload.context.webhook_result_1.results)
{\n
msg.payload.text=msg.payload.text+\"\\n\"+msg.payload.context.w
ebhook_result_1.results[i].text;\n  }\n
msg.payload=msg.payload.text;\n}\nelse msg.payload =
msg.payload.output.text[0];\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 625,
    "y": 270,
    "wires": [
      [
        "7177a04a.f32f9"
      ]
    ]
  },
  {
    "id": "7177a04a.f32f9",
    "type": "ui_text",

```

```

    "z": "a894d589.079ed8",
    "group": "40774cf4.6ec124",
    "order": 4,
    "width": 0,
    "height": 0,
    "name": "assistant reply",
    "label": "Assistant",
    "format": "{{msg.payload}}",
    "layout": "col-center",
    "x": 760,
    "y": 390,
    "wires": []
  },
  {
    "id": "5a78a9a3.f9a118",
    "type": "debug",
    "z": "a894d589.079ed8",
    "name": "msg.payload",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "x": 695,
    "y": 115,
    "wires": []
  },
  {
    "id": "162d311d.e82e6f",
    "type": "ui_text",
    "z": "a894d589.079ed8",
    "group": "84efbdf1.fcaa2",
    "order": 2,
    "width": 0,
    "height": 0,

```

```

    "name": "display query",
    "label": "You",
    "format": "{{msg.payload}}",
    "layout": "col-center",
    "x": 470,
    "y": 395,
    "wires": []
  },
  {
    "id": "84efbdf1.fcaa2",
    "type": "ui_group",
    "z": "",
    "name": "Ask anything",
    "tab": "541816f8.a44468",
    "order": 1,
    "disp": true,
    "width": "12",
    "collapse": false
  },
  {
    "id": "40774cf4.6ec124",
    "type": "ui_group",
    "z": "",
    "name": "Assistant",
    "tab": "541816f8.a44468",
    "order": 3,
    "disp": false,
    "width": "12",
    "collapse": false
  },
  {
    "id": "541816f8.a44468",
    "type": "ui_tab",
    "z": "",
    "name": "Nokia 111 Cell Phone Customer Care",
    "icon": "dashboard",

```



```
    "disabled": false,  
    "hidden": false  
  }  
]
```