Project Report

on

Intelligent Customer Help Desk with Smart Document Understanding

in

Artificial Intelligence

by

G. VISHWANTHROY

(vishwanthroy1021@gmail.com)

Intelligent Customer Help Desk with Smart Document Understanding

1. Introduction

1.1 Overview

We will build a chatbot that uses various Watson AI Services (Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-Red) 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 Requirements: Watson Assistant, Watson Discovery, Watson Cloud Functions, Node-RED
- Project Deliverable: Intelligent Customer Help Desk with Smart Document Understanding
- Project Team: G. Vishwanth Roy
- Project Duration: 29 Days

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 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. So, 'unless and until customer specifically asks for a customer representative the bot will try to solve all your 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 owner's manual is important and what is not. This will improve the answers returned from the queries. Then using Watson actions as webhook, Watson Discovery can be

integrated with Watson assistant. Finally using Node-Red, Watson assistant can be integrated with a web UI. This UI can then be used to connect with Watson assistant and chat with it.

1.2.1 Scope of Work

- 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

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.

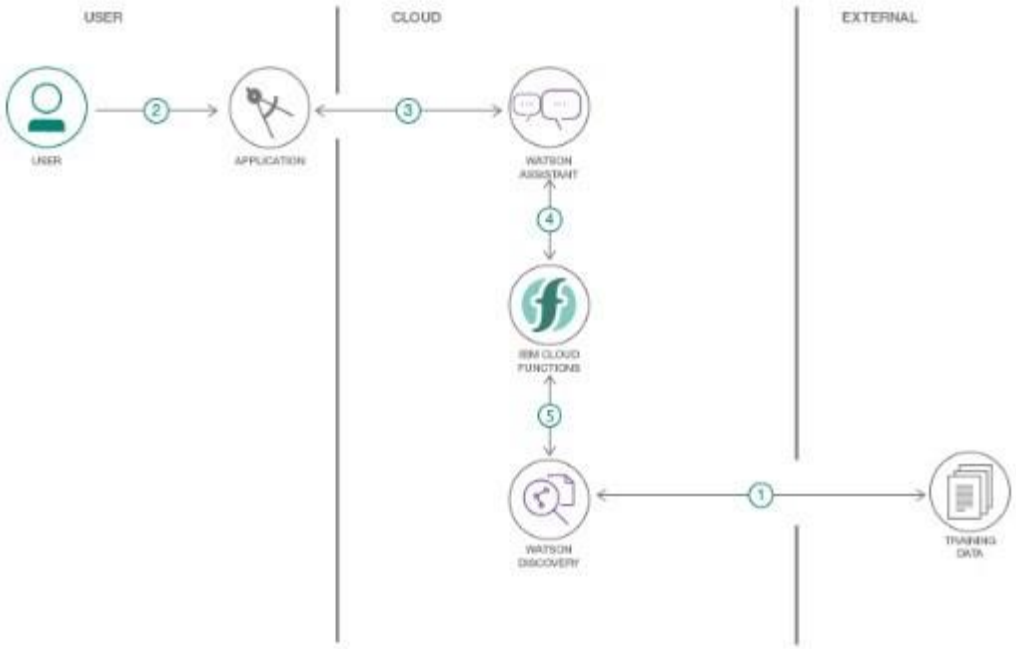
2.2 Proposed Solution

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 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. So, unless and until customer specifically asks for a customer representative the bot will try to solve all your 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 owner's manual is important and what is not. This will improve the answers returned from the queries. Then using Watson actions as web hook, Watson Discovery can be integrated with Watson assistant. Finally using Node-Red, Watson assistant can be integrated with a web UI. This UI can then be used to connect with Watson assistant and chat with it.

3. Theoretical Analysis

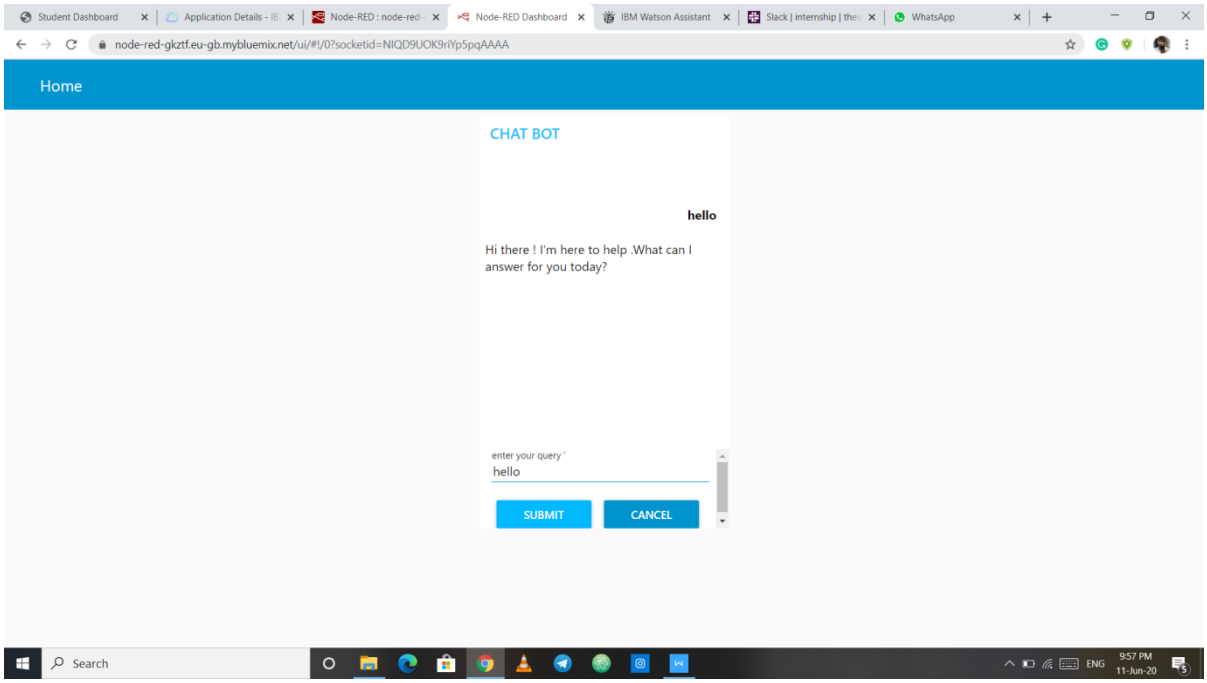
Block / Flow Diagram

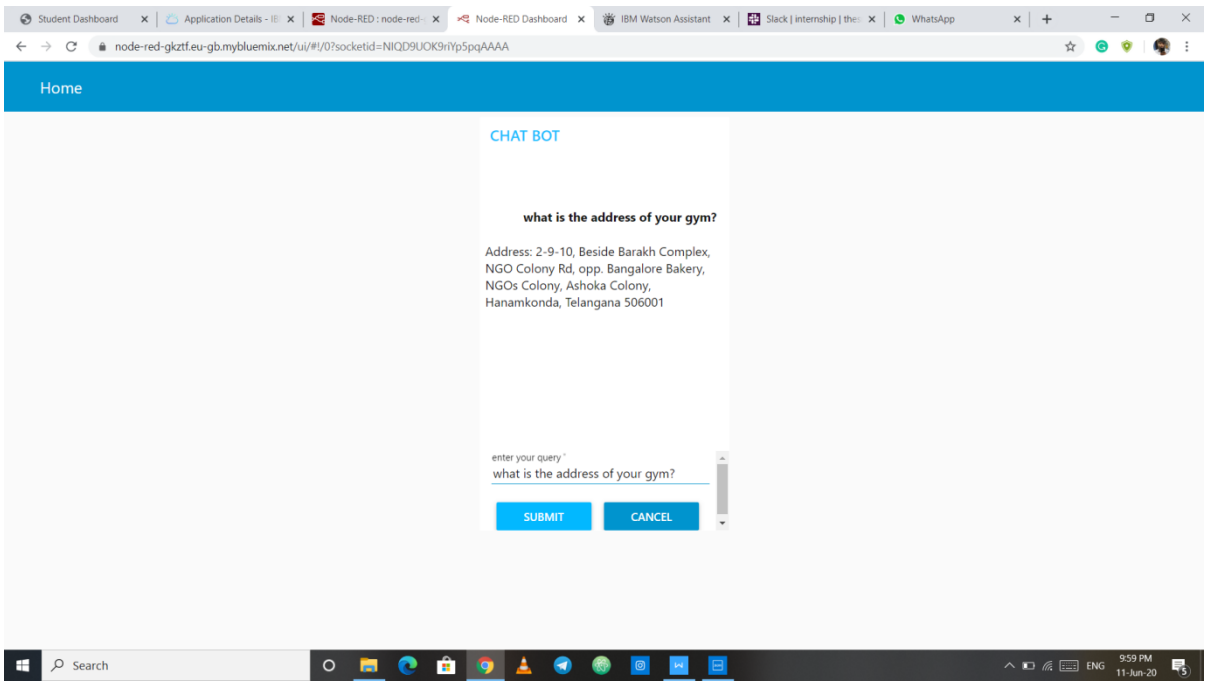


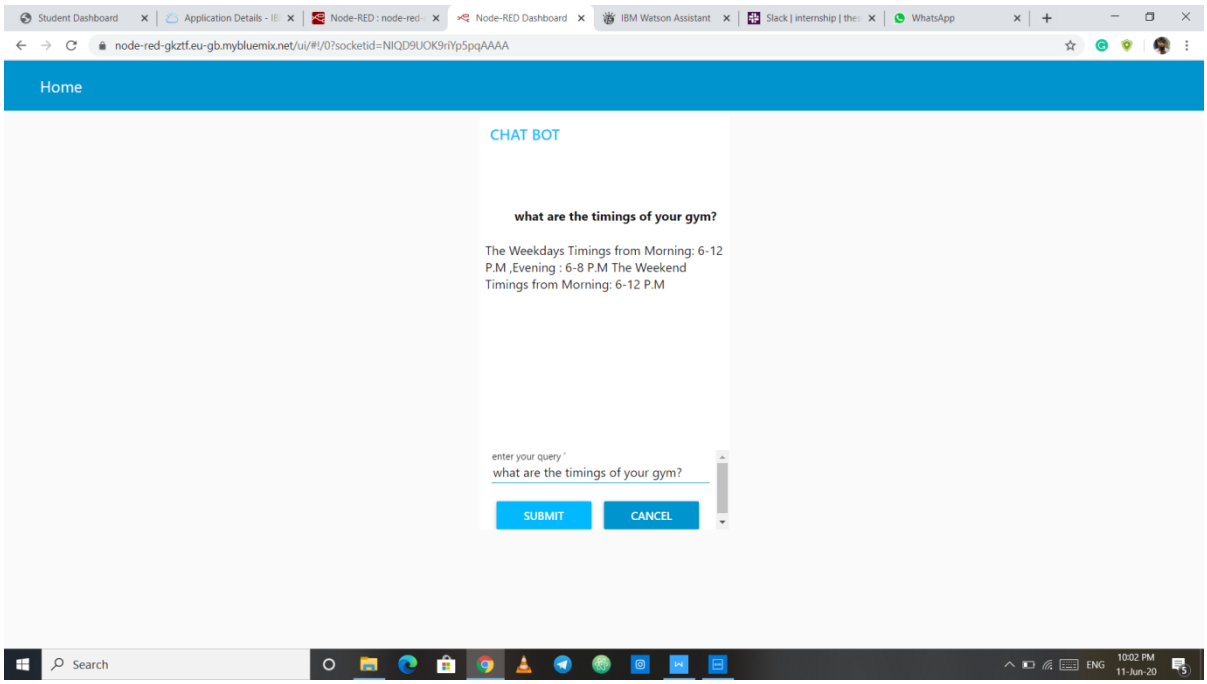
Hardware / Software Designing

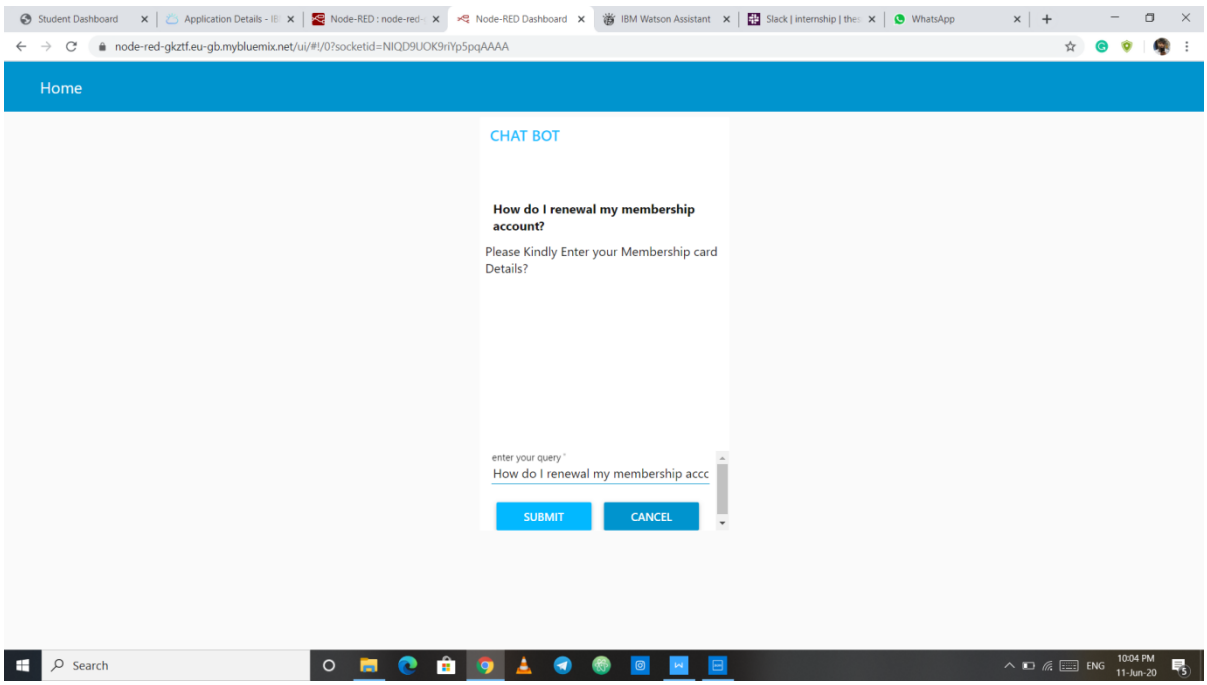
1. Create necessary Watson Services.
2. Configure Watson Discovery.
3. Create Watson Cloud Functions Action.
4. Configure Watson Assistant.
5. Integrate Watson Discovery with Watson Assistant using webhook.
6. Build Node-RED flow to integrate Watson Assistant and Web Dashboard.

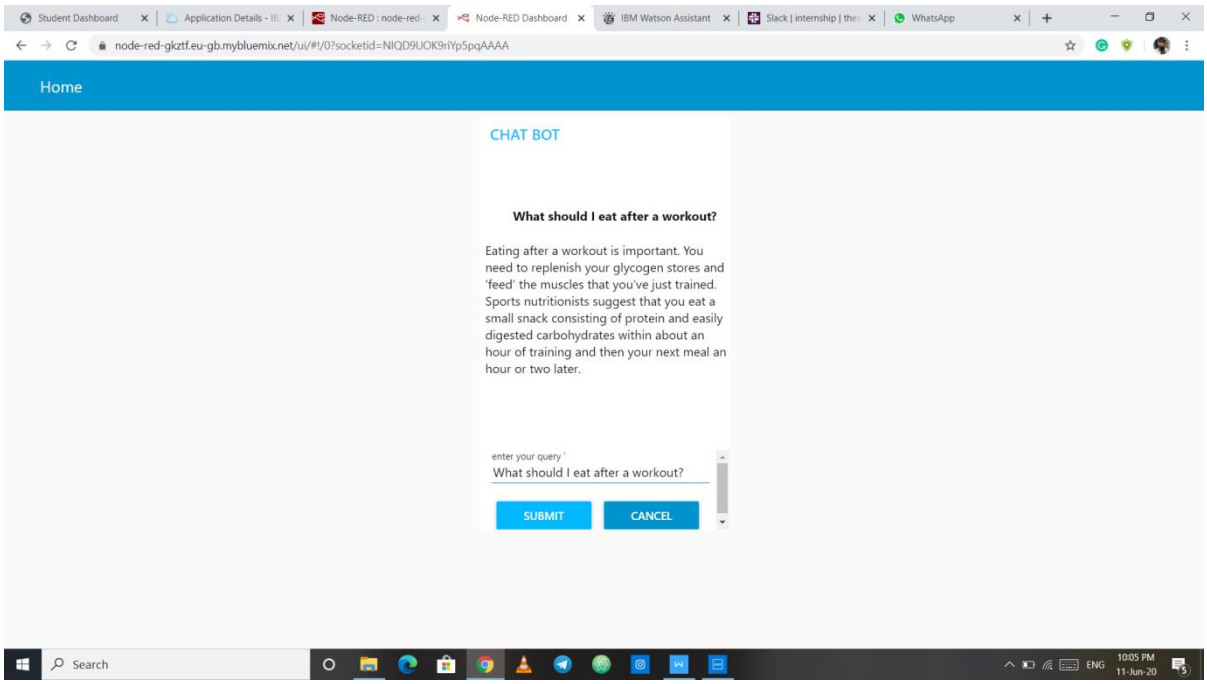
4. Experimental Investigation

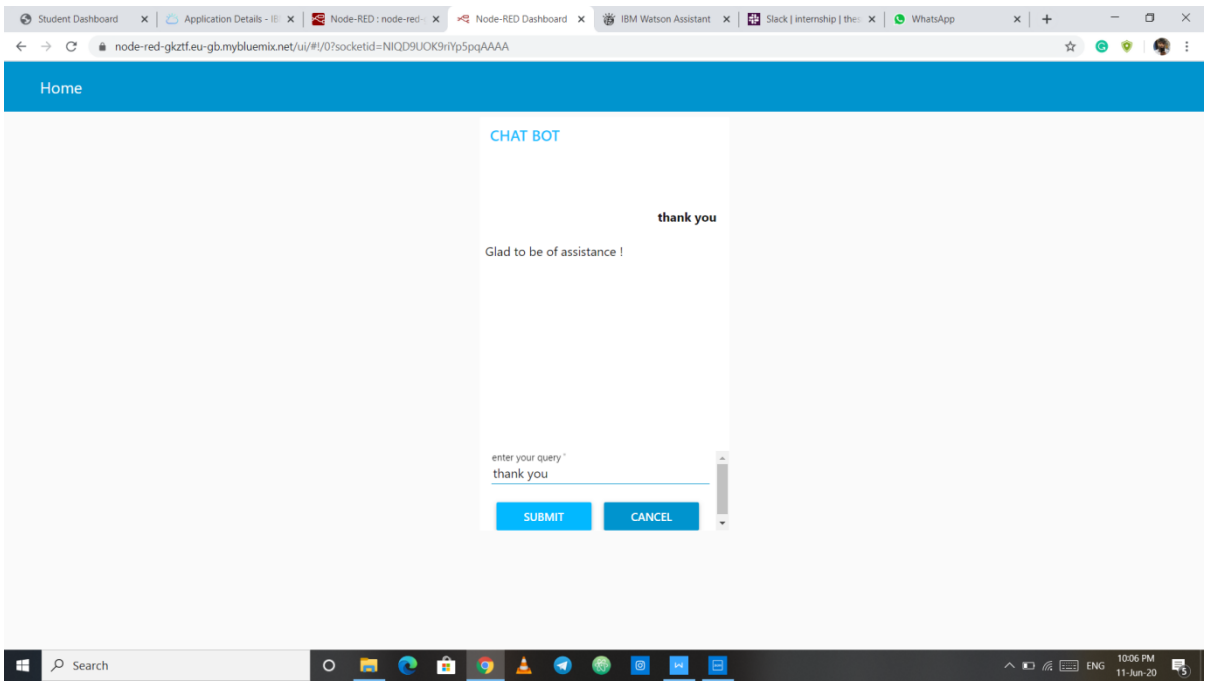





Student Dashboard
Application Details - IB
Node-RED : node-red-
Node-RED Dashboard
IBM Watson Assistant
Slack | internship | the
WhatsApp
node-red-gkztf.eu-gb.mybluemix.net/ui/#!/0?socketid=NIQD9UOK9riYp5pqAAAA
Home
CHAT BOT
what are the timings of your gym?
The Weekdays Timings from Morning: 6-12 P.M ,Evening : 6-8 P.M The Weekend Timings from Morning: 6-12 P.M
enter your query *
what are the timings of your gym?
SUBMIT
CANCEL
Search
ENG
10:02 PM
11-Jun-20


Student Dashboard
Application Details - IB
Node-RED : node-red-
Node-RED Dashboard
IBM Watson Assistant
Slack | internship | the
WhatsApp
node-red-gkztf.eu-gb.mybluemix.net/ui/#!/0?socketid=NIQD9UOK9riYp5pqAAAA
Home
CHAT BOT
How do I renewal my membership account?
Please Kindly Enter your Membership card Details?
enter your query *
How do I renewal my membership accc
SUBMIT
CANCEL
Search
ENG
10:04 PM
11-Jun-20

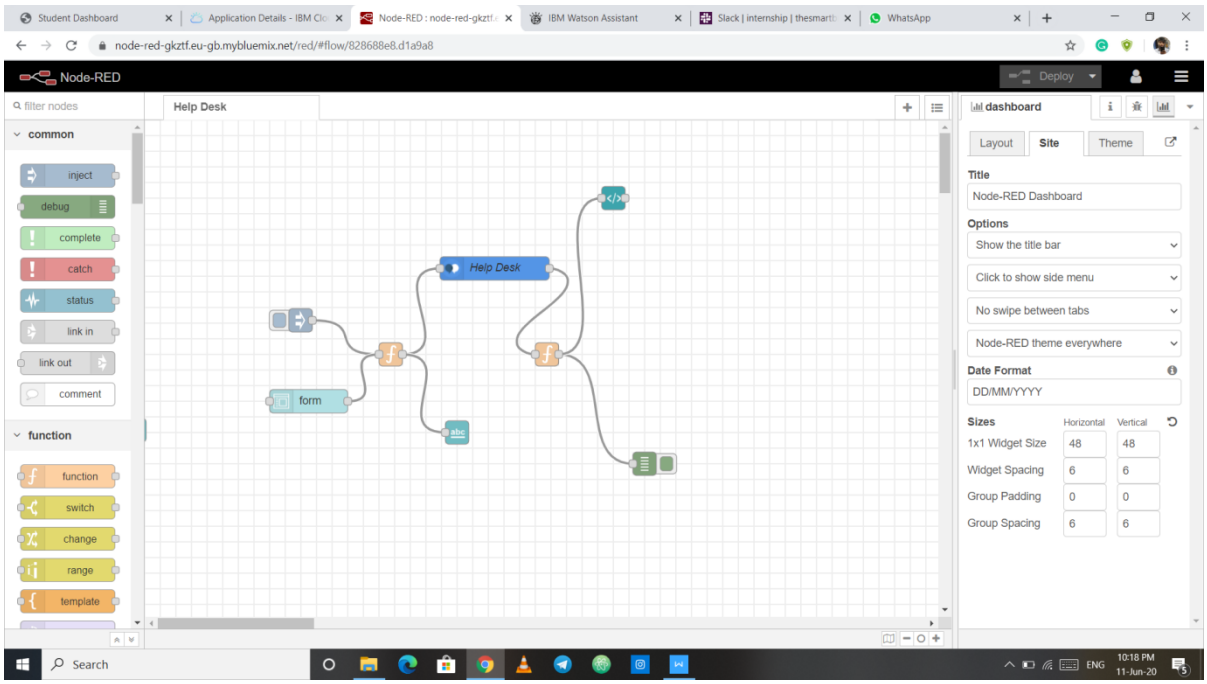

Student Dashboard
Application Details - IB
Node-RED : node-red-
Node-RED Dashboard
IBM Watson Assistant
Slack | internship | the
WhatsApp
node-red-gkztf.eu-gb.mybluemix.net/ui/#!/0?socketid=NIQD9UOK9riYp5pqAAAA
Home
CHAT BOT
What should I eat after a workout?
Eating after a workout is important. You need to replenish your glycogen stores and 'feed' the muscles that you've just trained. Sports nutritionists suggest that you eat a small snack consisting of protein and easily digested carbohydrates within about an hour of training and then your next meal an hour or two later.
enter your query
What should I eat after a workout?
SUBMIT
CANCEL
Search
ENG
10:05 PM
11-Jun-20


Student Dashboard
Application Details - IB
Node-RED : node-red-
Node-RED Dashboard
IBM Watson Assistant
Slack | internship | the
WhatsApp
node-red-gkztf.eu-gb.mybluemix.net/ui/#!/0?socketid=NIQD9UOK9riYp5pqAAAA
Home
CHAT BOT
thank you
Glad to be of assistance !
enter your query
thank you
SUBMIT
CANCEL
Search
ENG
10:06 PM
11-Jun-20

5. Flowchart

Insert the following nodes into the flow in Node-RED.

- Inject
- Debug
- ui_form
- ui_text
- Function
- Watson-conversation-v1
- ui_template



6. Results

Web based UI was developed by integrating all the services using Node-RED.

URL for UI Dashboard : https://node-red-gkztf.eu-gb.mybluemix.net/ui/

7. Advantages & Disadvantages

Advantages

1. Reduces Man Power
2. Cost Efficient
3. Less and less calls will be diverted to Customer Representatives.

Disadvantages

1. Sometimes it can mislead customers as it tries to search irrelevant information in the manual.
2. It may also give same answers to different queries.

8. Applications

1. This chatbot can be deployed to various websites as it can solve a lot of basic questions.
2. It can be used to deploy as Customer Help desk for small scale products as their manual usually has the solution for the user's problems.

9. Conclusion

An Intelligent Customer Help desk Chat-bot was created using various Watson services like Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-RED.

10. Future Scope

In the future, various other Watson services like Text-To-Speech and Speech-To-Text can be integrated in the chatbot. This can make the chatbot Hands-free.

11. Bibliography

1. Node-RED Starter Application : https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-applicat ion/

2. Build your open AI assistant : https://www.youtube.com/watch?v=hitUOFNne14

3. How to use Watson Assistant with Webhooks : https://www.youtube.com/embed/5z3i5IsBVnk

4. Watson Discovery : https://developer.ibm.com/articles/introduction-watson-discovery/

5. Enhance customer help desk with Watson Smart Document Understanding: https://www.youtube.com/watch?v=-yniuX-Poyw

Appendix

Source Code

Node-Red Flow Code


```
[

. {
• "id":"828688e8.d1a9a8",
• "type":"tab",
• "label":"Help Desk",
• "disabled":false,
• "info":""

},

. {

• "id":"a1420d79.49e73",
• "type":"function",
• "z":"828688e8.d1a9a8",
• "name":"Input Function",
• "func":"msg.payload=msg.payload.input;\nreturn msg;",
```

```
"outputs":1,
"noerr":0,
"x":315,
"y":300,
"wires":[
[
"4e0ec9d.8f1bf38",
"83266398.63ddf"
]
],
"l":false
},
{
"id":"990d0699.9e9f68",
"type":"function",
"z":"828688e8.d1a9a8",
"name":"Output Function",
"func":"\n msg.payload=msg.payload.output.text[0]\nreturn msg;",
"outputs":1,
"noerr":0,
"x":515,
"y":300,
"wires":[
[
"ee3c8ed4.0647a",
"3ddb36a2.45ff6a"
]
],
"l":false
},
{
"id":"edcdcdab.17e6c",
"type":"inject",
"z":"828688e8.d1a9a8",
"name":"",
"topic":"ChatBot",
"payload":"",
"payloadType":"date",
"repeat":"",
```

```
"crontab":"",
"once":false,
"onceDelay":0.1,
"x":199.60000610351562,
"y":256.4000015258789,
"wires":[
[
"a1420d79.49e73"

]
],

"l":false

},

{

"id":"69c1ee5.b342f1",
"type":"ui_form",
"z":"828688e8.d1a9a8",
"name":"",
"label":"",
"group":"69e32062.62d97",
"order":4,
"width":"0",
"height":"0",
"options":[
{
"label":"enter your query",
"value":"input",
"type":"text",
"required":true,
"rows":null

}
],

"formValue":{
"input":""

},

"payload":"",
"submit":"submit",
"cancel":"cancel",
"topic":"",
"x":210,
"y":360,
```

```
"wires":[

[
"a1420d79.49e73"

]
]
},

{
"id":"4c35294f.7cea28",
"type":"ui_text",
"z":"828688e8.d1a9a8",
"order":0,
"width":0,
"height":0,
"name":"",
"label":"text",
"format":"{{msg.payload}}",
"layout":"row-spread",
"x":-13,
"y":397,
"wires":[

],

"l":false

},

{

"id":"4e0ec9d.8f1bf38",
"type":"watson-conversation-v1",
"z":"828688e8.d1a9a8",
"name":"Help Desk",
"workspaceid":"96ad6ddb-7a35-47ab-b31d-f389e3e36bab",
"multiuser":false,
"context":true,
"empty-payload":false,
"service-endpoint":"https://api.eu-
gb.assistant.watson.cloud.ibm.com/instances/ff2f41f7-a90c-479f-bf26-
f23bcd599df3",
"timeout":"",
"optout-learning":false,
"x":448.00000762939453,
"y":190.00000476837158,
"wires":[
[
```

```
        "990d0699.9e9f68"

    ]
    ]
    },

    {

    "id":"83266398.63ddf",
    "type":"ui_text",
    "z":"828688e8.d1a9a8",
    "group":"69e32062.62d97",
    "order":2,
    "width":6,
    "height":1,
    "name":"",
    "label":"",
    "format":"{{msg.payload}}",
    "layout":"row-spread",
    "x":400,
    "y":400,
    "wires":[

    ],

    "l":false

    },

    {

    "id":"ee3c8ed4.0647a",
    "type":"debug",
    "z":"828688e8.d1a9a8",
    "name":"",
    "active":true,
    "tosidebar":true,
    "console":false,
    "tostatus":false,
    "complete":"payload",
    "targetType":"msg",
    "x":640,
    "y":440,
    "wires":[

    ],

    "l":false

    },
```

```
. {
• "id":"3ddb36a2.45ff6a",
• "type":"ui_template",
• "z":"828688e8.d1a9a8",
• "group":"69e32062.62d97",
• "name":"",
• "order":3,
• "width":6,
• "height":5,
• "format":"<!DOCTYPE html>\n<html>\n<head>\n<style>\ndiv.scrl {\n
  background-color: white;\n width: 310px;\n height: 400px;\n overflow: auto\n
  margin-bottom:20px;\n}\n</style>\n</head>\n<body>\n<div
  class=\"scrl\">{{msg.payload}}</div>\n</body>\n</html>\n",
• "storeOutMessages":true,
• "fwdInMessages":true,
• "resendOnRefresh":true,
• "templateScope":"local",
• "x":600,
• "y":100,
• "wires":[
. [

  ]
  ],

• "l":false

  },

. {

• "id":"69e32062.62d97",
• "type":"ui_group",
• "z":"",
• "name":"CHAT BOT",
• "tab":"74d4b73f.edff78",
• "order":1,
• "disp":true,
• "width":"6",
• "collapse":false

  },

. {

• "id":"74d4b73f.edff78",
• "type":"ui_tab",
• "z":"",
• "name":"Home",
```

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

}
]

CLOUD FUNCTION 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);
    });
  });
}
```