

PROJECT REPORT
ON
**“Intelligent Customer Help Desk With Smart
Document Understanding”**

BY
MS.SAYALI NAMADEV PATIL

(sayalipatil1606@gmail.com)

ARTIFICIAL INTELLIGENCE

THE SMARTBRIDGE

2020-2021

INDEX

Index	1
1 INTRODUCTION	1
1.1 OVERVIEW	1
1.2 PURPOSE	1
1.2.1 SCOPE OF WORK	1
2 LITERATURE SURVEY	2
2.1 EXISTING PROBLEM	2
2.2 PROPOSED SOLUTION	2
3 THEORITICAL ANALYSIS	3
3.1 BLOCK DIAGRAM	3
3.2 HARDWARE/SOFTWARE DESIGNING	4
4 EXPERIMENTAL INVESTIGATIONS	5
4.1 CREATE IBM CLOUD SERVICES	5
4.2 CONFIGURE WATSON DISCOVERY	5
4.2.1 Import the document	5
4.2.2 Annotate with SDU	5
4.2.3 CREATE IBM CLOUD FUNCTION ACTIONS	8
4.2.4 CONFIGURE WATSON ASSISTANT	10
4.2.5 Add new intent	10
4.2.6 Create new dialog node	11
4.2.7 CREATE FLOW AND CONFIGURE NODE	13
5 FLOWCHART	25

6	RESULTS	27
7	PRONS AND CONS	28
7.1	PROS:	28
7.2	CONS:	28
8	APPLICATIONS	29
9	CONCLUSION	30
10	FUTURE SCOPE	31
	Bibliography	32
10.1	32

Chapter 1

INTRODUCTION

)

1.1 OVERVIEW

We will be able to write an application that leverages multiple Watson AI Services (Discovery , Assistant, Cloud function and Node Red). By the end of the project, we'll learn best practices of combining Watson services, and how they can build interactive information retrieval systems with Discovery + Assistant.

1.2 PURPOSE

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

Chapter 2

LITERATURE SURVEY

A problem definition states Project Scope which defines exactly what is in the process. Processes are usually continuous

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. And the question remains unanswered in most cases.

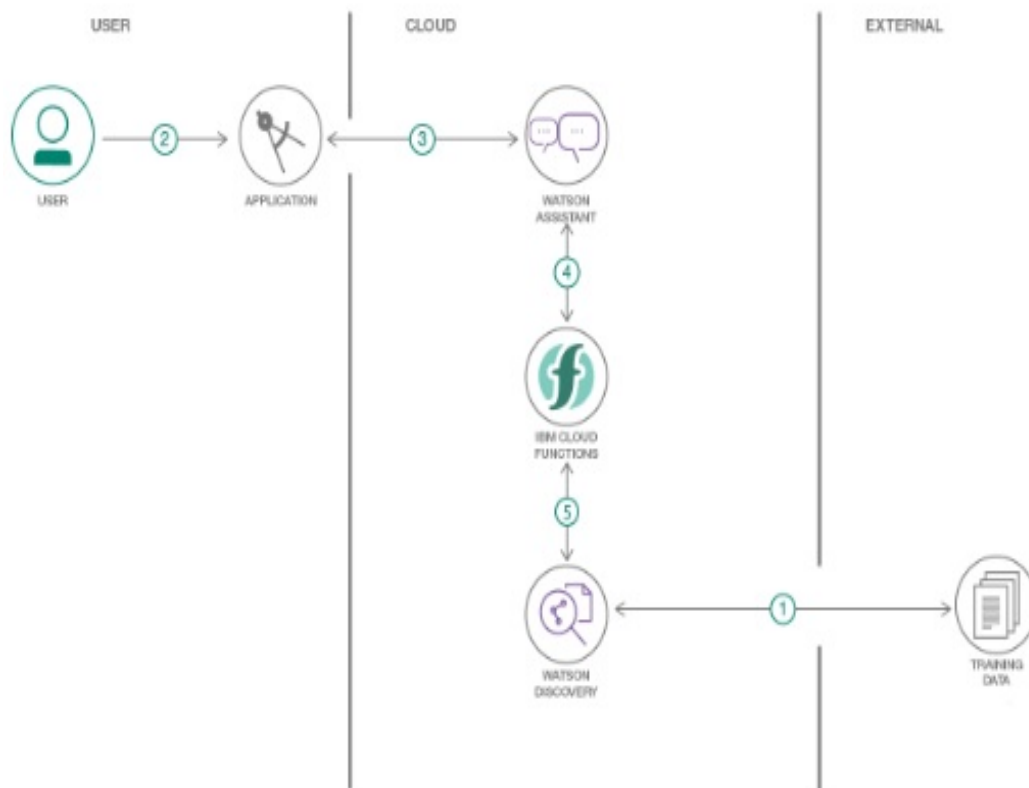
2.2 PROPOSED SOLUTION

To solve the above mentioned problem we will introduce a Watson Discovery feature that is Smart Document Understanding. 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.

Chapter 3

THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM



- 1. The document is annotated using Smart Document Understanding feature of Watson Discovery.

- 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 a Watson Assistant dialog skill.
- 4. If the user asks a query about product, a search query is passed to a predefined IBM Cloud Functions action.
- 5. 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.

Chapter 4

EXPERIMENTAL INVESTIGATIONS

Planning plays an important role in successful completion of a project. This plan acts as a checklist of the task done. It helps in stating the amount of work to be done in a stipulated period of time. The time chart and can set milestones for project.

4.1 CREATE IBM CLOUD SERVICES

Create the following services:

1. . Watson Discovery
2. Watson Assistant
3. Node Red

4.2 CONFIGURE WATSON DISCOVERY

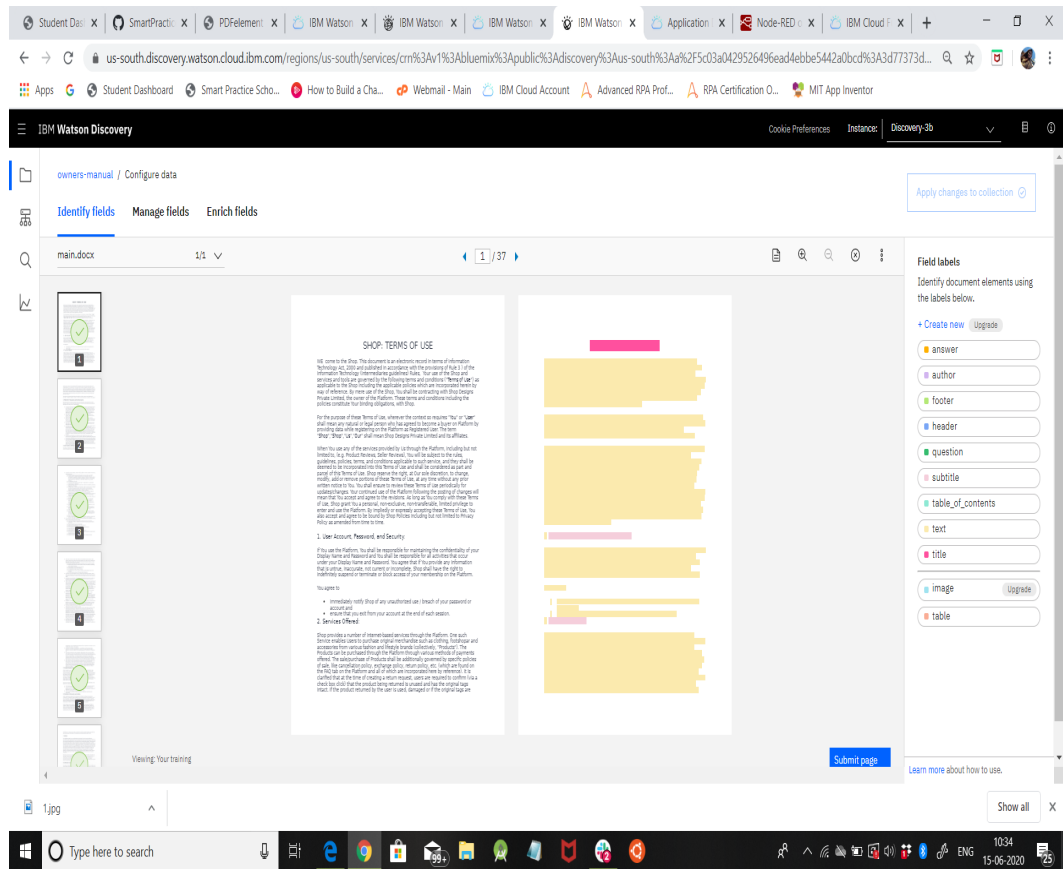
4.2.1 Import the document

Launch the Watson Discovery tool and create a new data collection by selecting the Upload your own data option. Give the data collection a unique name. When prompted, select and upload the Dataset of SDU.pdf file located in the data directory of your local repo. This Dataset of SDU.pdf file contains configuration of different parts in Dell Dataset of SDU.pdf laptop model.

4.2.2 Annotate with SDU

Now let's apply SDU to our document to see if we can generate some better query responses. From the Discovery collection panel, click the Configure data button

(located in the top right corner) to start the SDU process. Here is the layout of the Identify fields tab of the SDU annotation panel: The goal is to annotate all of the



pages in the document so Discovery can learn what text is important, and what text can be ignored

- 1. is the list of pages in the manual. As each is processed, a green check mark will appear on the page
- 2. is the current page being annotated.
- 3. is where you select text and assign it a label.
- 4. is the list of labels you can assign to the page text.
- Click [5] to submit the page to Discovery.
- Click [6] when you have completed the annotation process.

- As you go through the annotations one page at a time, Discovery is learning and should start automatically updating the upcoming pages. Once you get to a page that is already correctly annotated, you can stop, or simply click Submit [5] to acknowledge it is correct. The more pages you annotate, the better the model will be trained.
- For this specific owner's manual, at a minimum, it is suggested to mark the following:
 - The main title page as title
 - All headers and sub-headers as subtitle.
 - All page numbers as footers
 - All warranty and licensing information (located in the last few pages) as a footer.
 - All other text should be marked as text.
- Once you click the Apply changes to collection button [6], you will be asked to reload the document. Choose the same owner's manual .pdf document as before.
- Next, click on the Manage fields [1] tab.

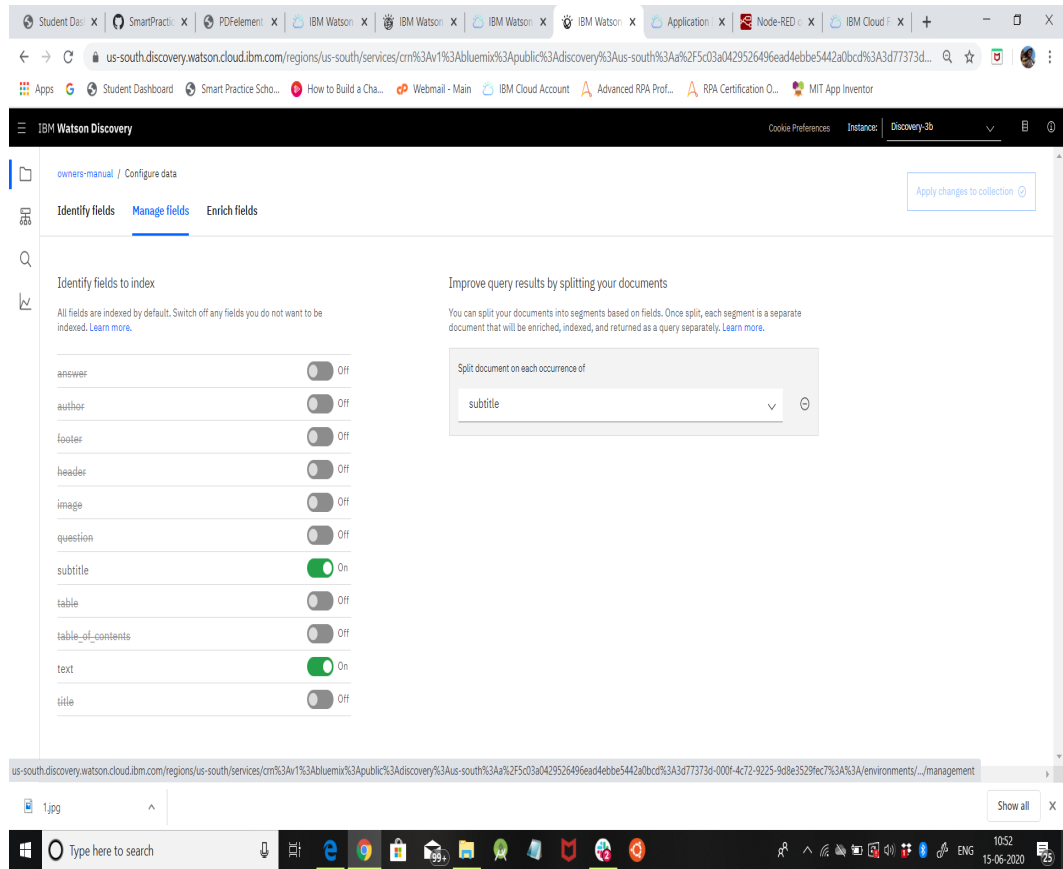
[2] Here is where you tell Discovery which fields to ignore. Using the on/off buttons, turn off all labels except text and title. [3] is telling Discovery to split the document apart, based on title. Click [4] to submit your changes. Once again, you will be asked to reload the document. Now, as a result of splitting the document apart, your collection will look different:

Go to Build your own query panel and see how much better the results are.

Store credentials for future use. In upcoming steps, you will need to provide the credentials to access your Discovery collection.

The values can be found in the following locations.

The Collection ID, Environment ID and Configuration ID values can be found by clicking the dropdown button located at the top right side of your collection panel:



4.2.3 CREATE IBM CLOUD FUNCTION ACTIONS

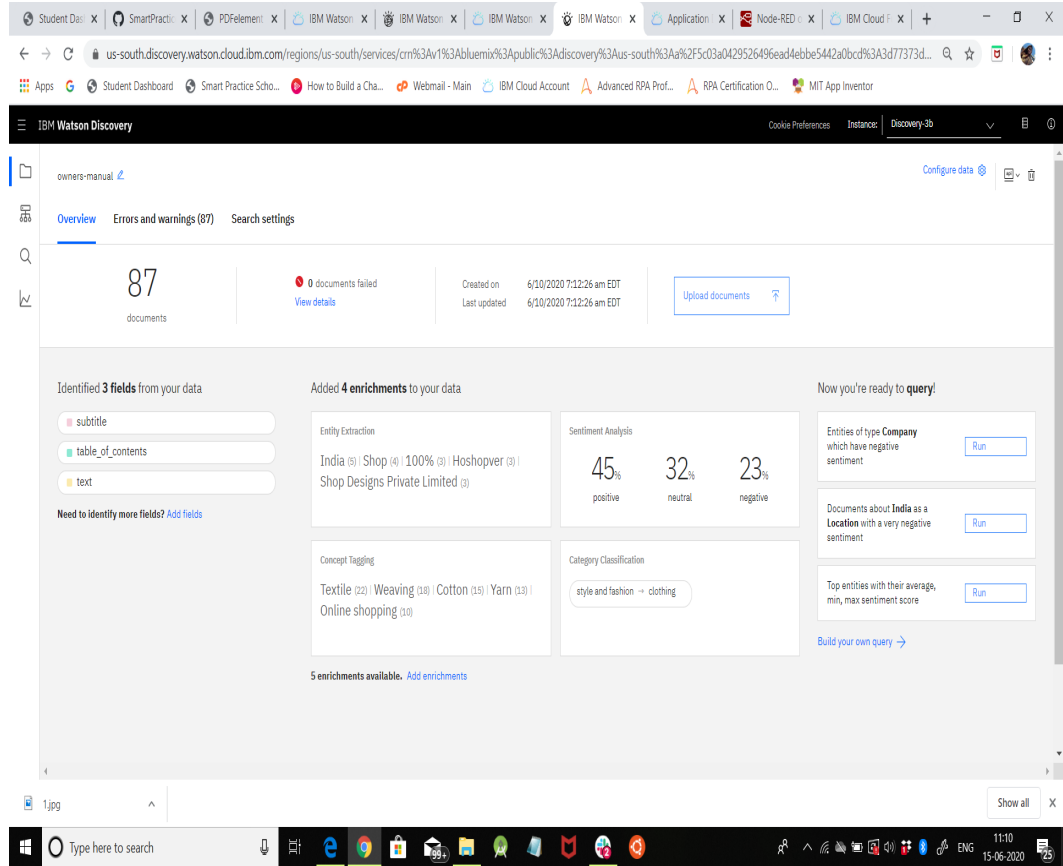
Now let's create the web action that will make queries for our Discovery collection. Go to IBM Cloud dashboard and in catalog section select Function to create the action

From the Functions main panel, click on the Actions tab. Then click on Create. From the Create panel, select the Create Action option. On the Create Action panel, provide a unique Action Name [1], keep the default package [2], and select the Node.js 10 [3] runtime. Click the Create button [4] to create the action.

From the Functions main panel, click on the Actions tab. Then click on Create. From the Create panel, select the Create Action option. On the Create Action panel, provide a unique Action Name [1], keep the default package [2], and select the Node.js 10 [3] runtime. Click the Create button [4] to create the action.

Once your action is created, click on the Code tab:

In the code editor window, cut and paste in the respective code.



The code is pretty straightforward - it simply connects to the Discovery service, makes a query against the collection, then returns the response.

If you press the Invoke button, it will fail due to credentials not being defined yet. We'll do this next. Select the Parameters tab

Add the following keys – `url`, `environmentid`, `collectionid`, `configurationid` and `iamaapikey`.

For values, please use the values associated with the Discovery service you created in the previous step.

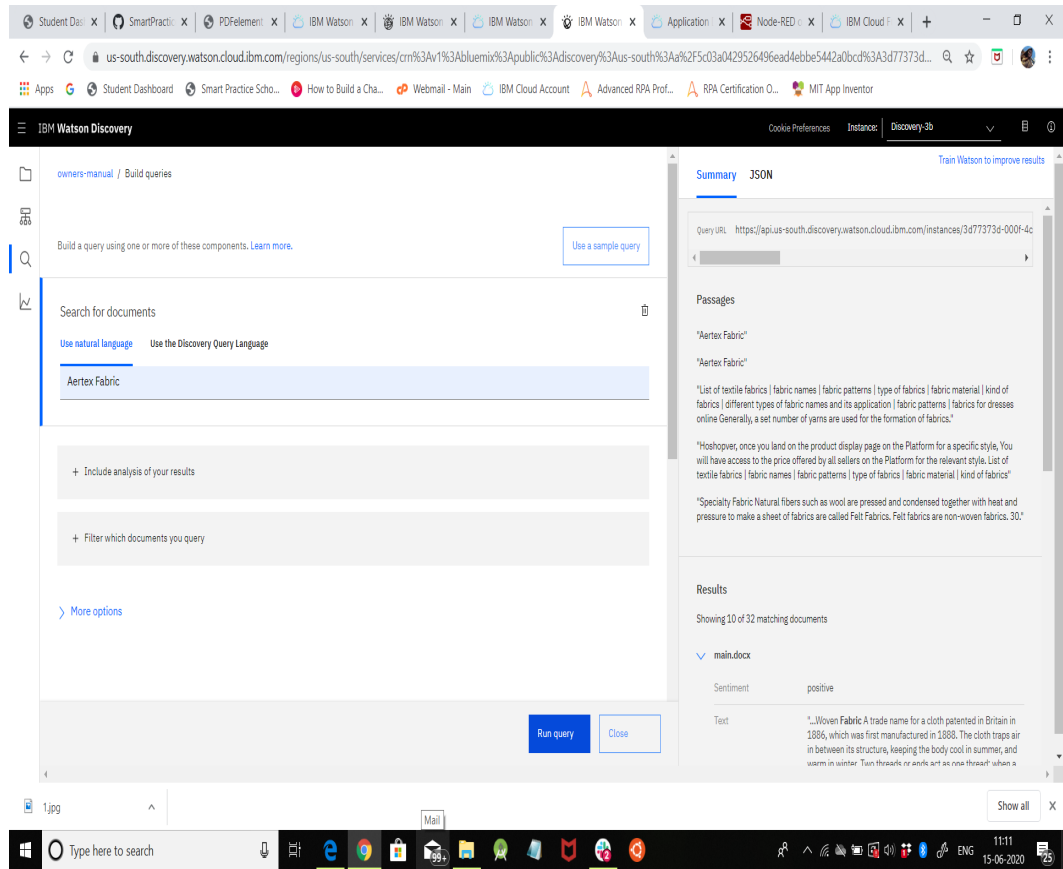
Now that the credentials are set, return to the Code panel and press the Invoke button again.

Now you should see actual results returned from the Discovery service:

Next, go to the Endpoints panel [1]

Click the checkbox for Enable as Web Action [2]. This will generate a public endpoint URL [3].

Take note of the URL value [3], as this will be needed by Watson Assistant in a



future step.

4.2.4 CONFIGURE WATSON ASSISTANT

Launch the Watson Assistant tool and create a new dialog skill. This dialog skill contains all of the nodes needed to have a typical call center conversation with a user.

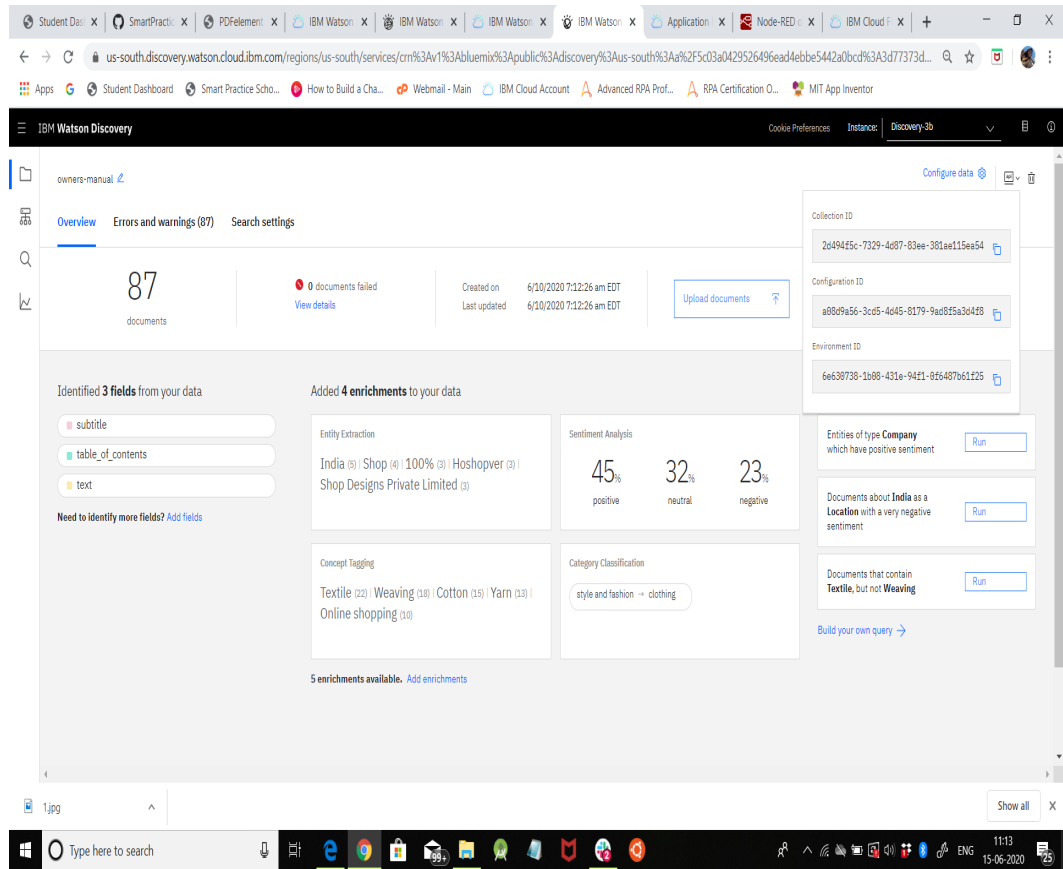
4.2.5 Add new intent

The default customer care dialog does not have a way to deal with any questions involving outside resources, so we will need to add this.

Create a new intent that can detect when the user is asking about operating the Laptop.

From the Customer Help Desk Skill panel, select the Intents tab. Click the Create intent button.

Name the intent *Product information*, and at a minimum, enter the following example question to



4.2.6 Create new dialog node

Now we need to add a node to handle our intent. Click on the Dialog [1] tab, then click on the drop down menu for the Greetings node [2], and select the Add node below [3] option.

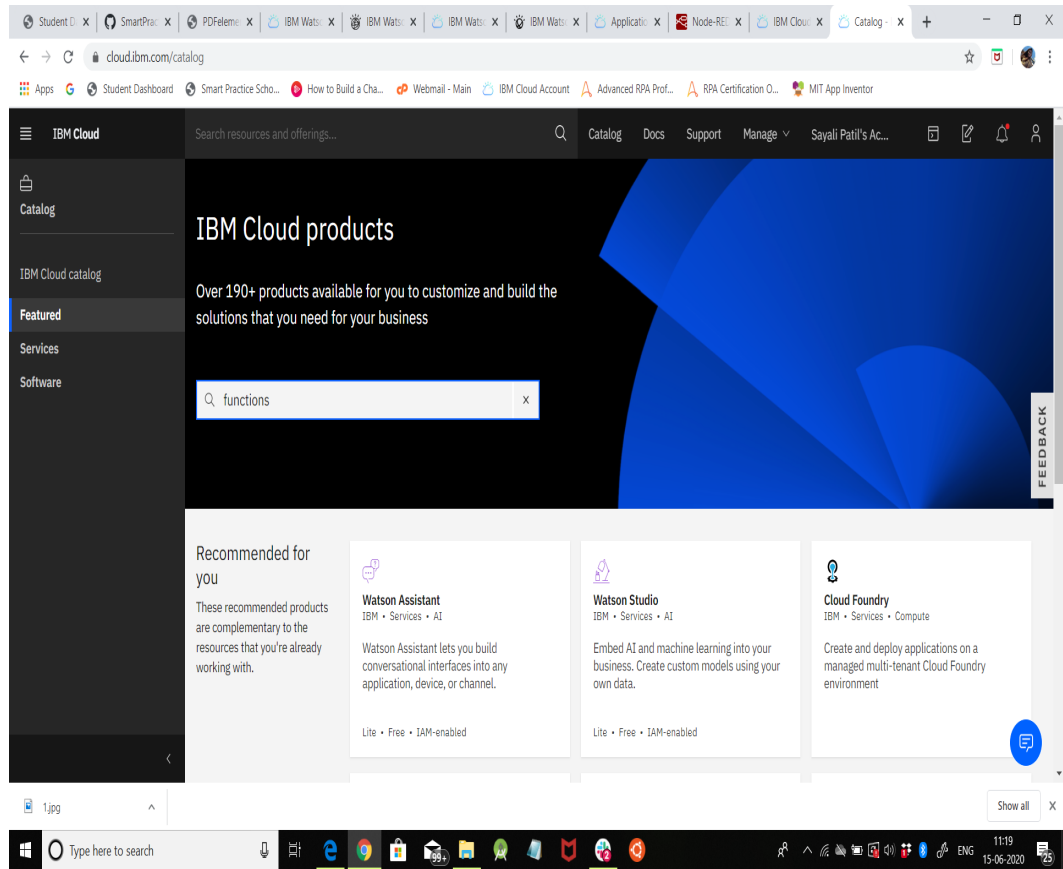
Name the node "Ask Queries" [1] and assign it our new intent [2].

This means that if Watson Assistant recognizes a user input such as "how to flash the bios?", it will direct the conversation to this node.

ENABLE WEBHOOK FROM ASSISTANT

Set up access to our WebHook for the IBM Cloud Functions action you created in Step 4. Select the Options tab [1]:

Enter the public URL endpoint for your action [2]. Return to the Dialog tab, and click on the Ask Queries node. From the details panel for the node, click on Customize, and enable Webhooks for this node:



Click Apply. The dialog node should have a Return variable [1] set automatically to $webhook_result_1$. This is the variable name you can use to access the result from the Discovery service query.

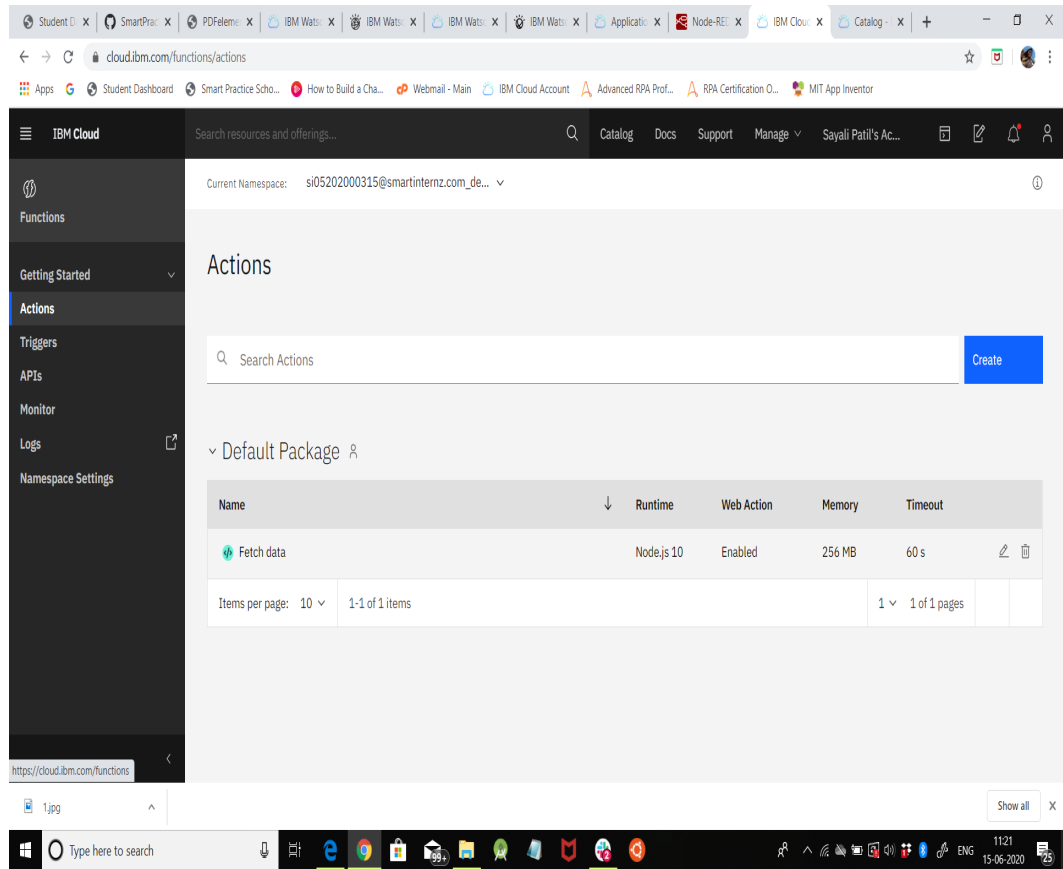
You will also need to pass in the user's question via the parameter input [2]. The key needs to be set to the value: `"input.text?"`. If you fail to do this, Discovery will return results based on a blank query. Optionally, you can add these responses to aid in debugging:

Test in Assistant Tooling

From the Dialog panel, click the Try it button located at the top right side of the panel. Enter some user input:

Note that the input "how to remove wireless card?" has triggered our Ask Queries dialog node, which is indicated by the $Product_information_response$.

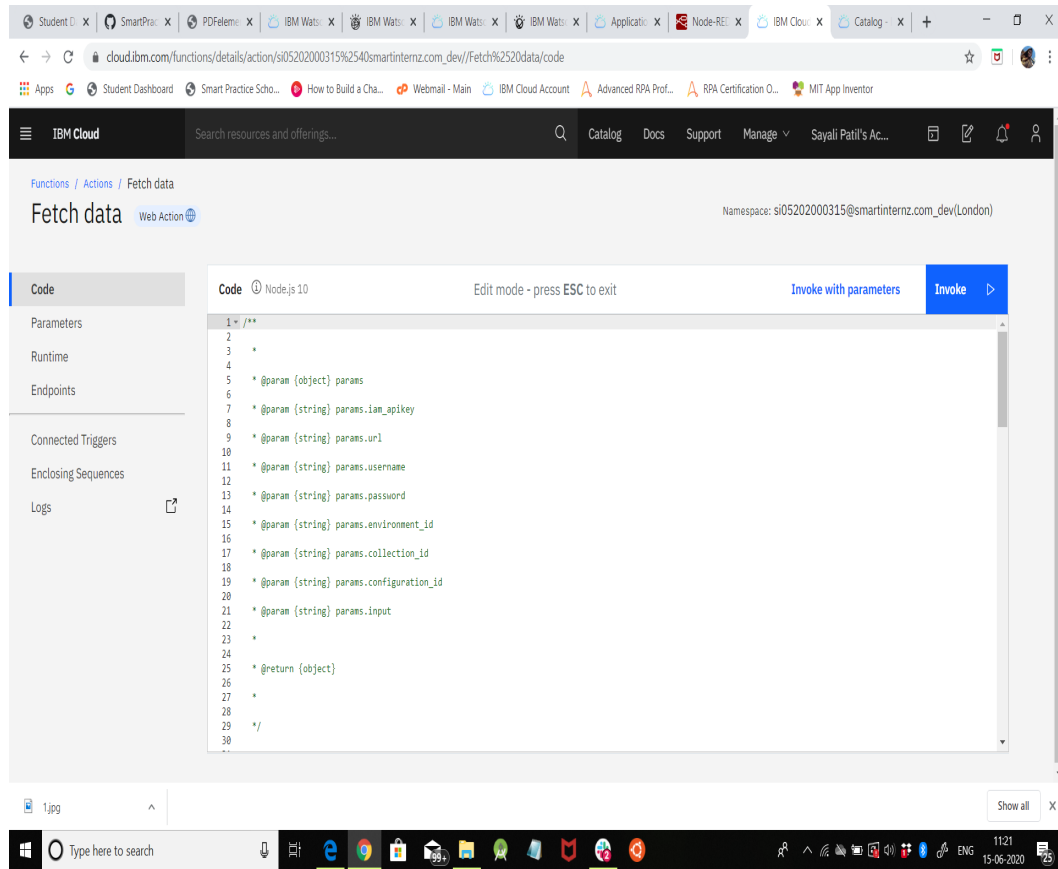
And because we specified that $webhook_result_1.passages$ be the response, that value is displayed also as $?webhook_result_1.passages[0].passage_text? >$ and try results again.



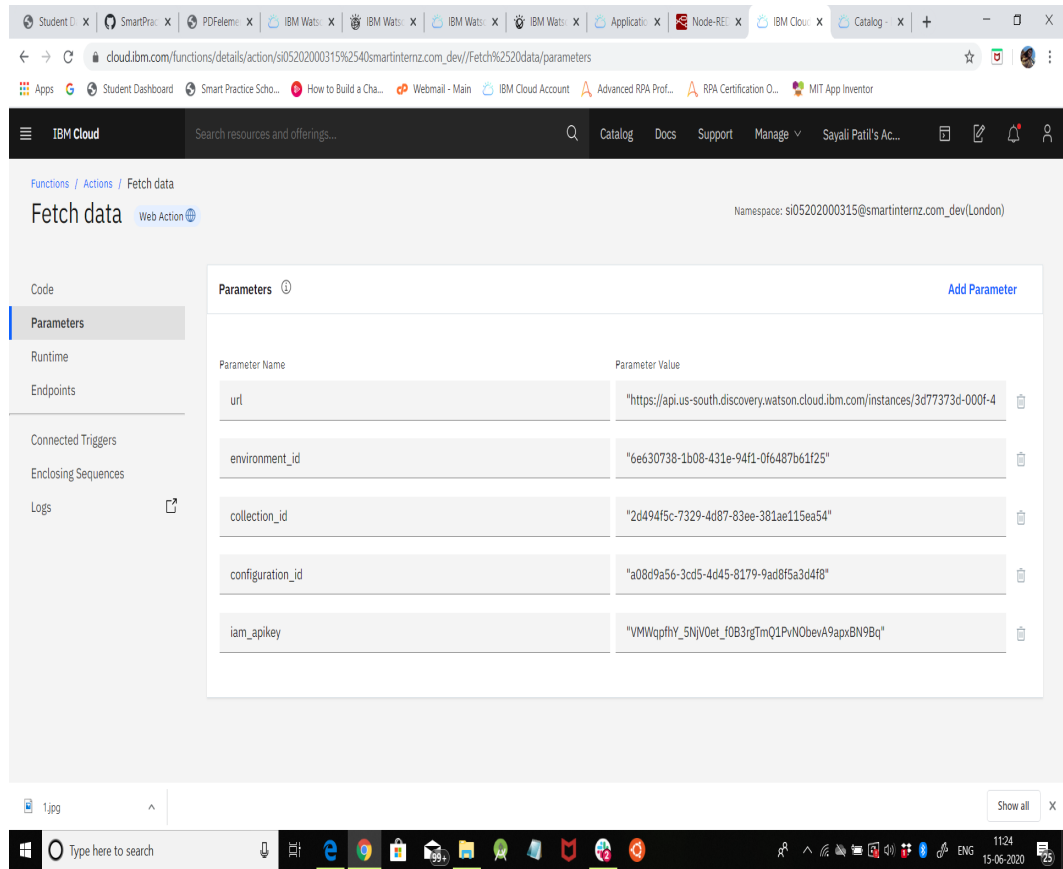
4.2.7 CREATE FLOW AND CONFIGURE NODE

Integration of watson assistant in Node-RED

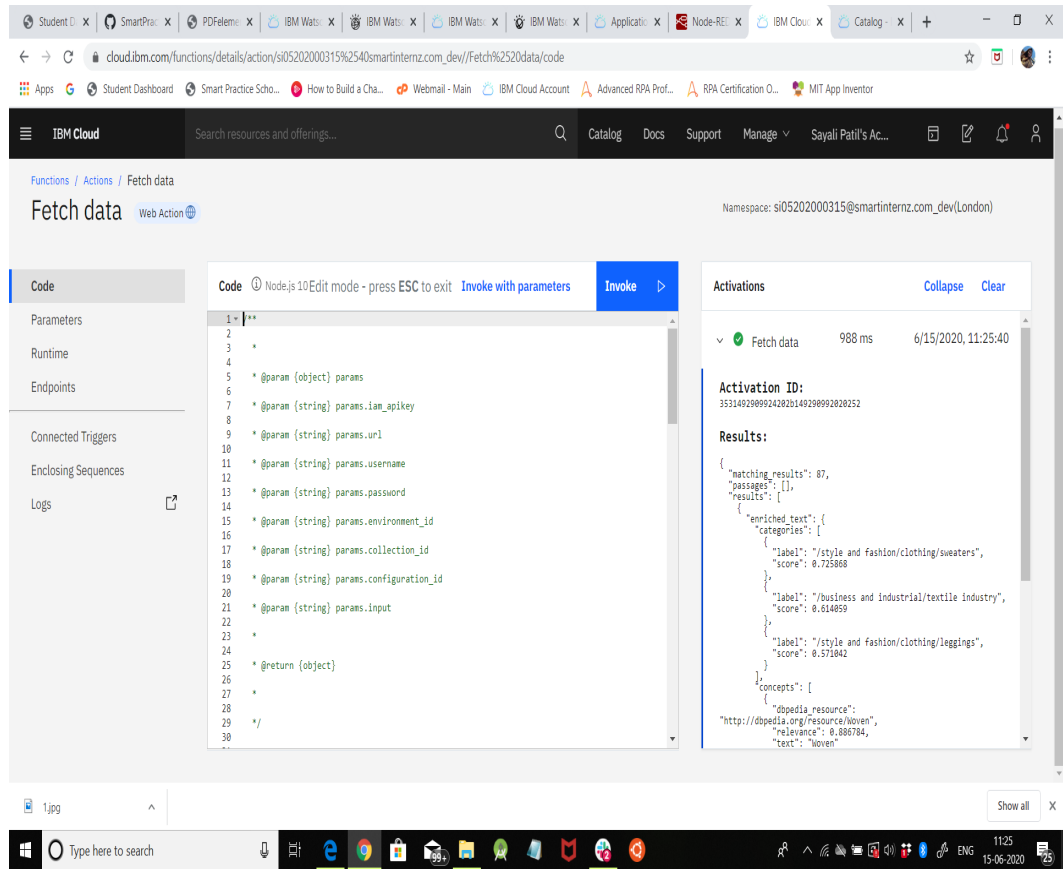
- Double-click on the Watson assistant node
- workspace id of your Watson assistant service
- After entering all the information click on Done.
- Drag inject node on to the flow from the Input section .
- Drag Debug on to the flow from the output section
- Double-click on the inject node
- Select the payload as a string



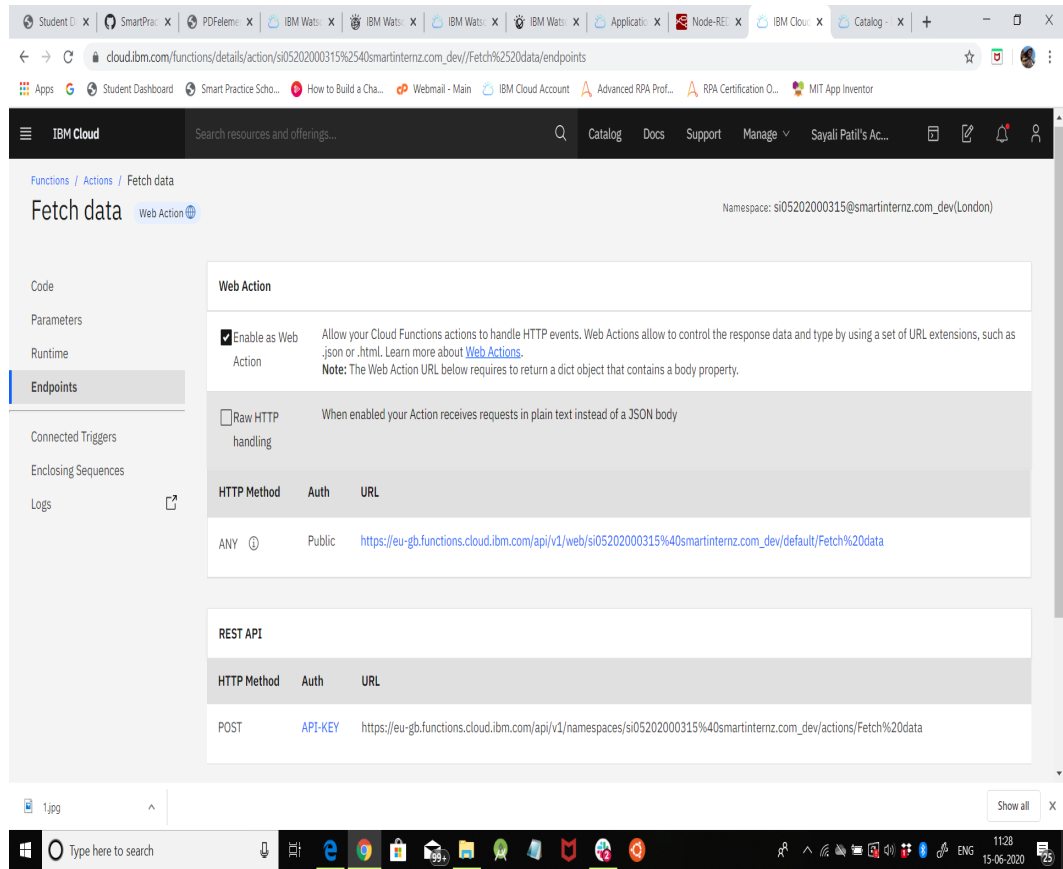
- Enter a sample input to be sent to the assistant service and click on done
- Connect the nodes and click on Deploy
- Click on the button to send input text to the assistant node
- Observe the output from the assistant service node
- The Bot output is located inside “output.text”
- Drag the function node to parse the JSON data and get the bot response
- Double click on the function node and enter the JSON parsing code as shown below and click on done
- Connect the nodes as shown below and click on Deploy



- Re-inject the flow and observe the parsed output
- For creating a web application UI we need “dashboard” nodes which should be installed manually
- Go to navigation pane and click on manage palette
- Click on install
- Search for “node-red-dashboard” and click on install and again click on install on the prompt
- The following message indicates dashboard nodes are installed, close the manage palette
- Search for “Form” node and drag on to the flow
- Double click on the “form” node to configure



- Click on the edit button to add the “Group” name and “Tab” name
- Click on the edit button to add tab name to web application
- Give sample tab name and click on add do the same thing for the group
- Give the label as “Enter your input”, Name as “text” and click on Done
- Drag a function node, double-click on it and enter the input parsing code as shown below
- Click on done
- Connect the form output to the input of the function node and output of the function to input of assistant node



- Search for “text” node from the “dashboard” section
- Drag two “text” nodes on to the flow
- Double click on the first text node, change the label as “You” and click on Done
- Double click on the second text node, change the label as “Bot” and click on Done
- Connect the output of “input parsing” function node to “ You” text node and output of “Parsing” function node to the input of “Bot” text node
- Click on deploy

The screenshot displays the IBM Watson Assistant Lite web interface. The browser's address bar shows the URL: `eu-gb.assistant.watson.cloud.ibm.com/eu-gb/cmv1:bluemixpublicconversation:eu-gb-a-2f5c03a0429526496ead4ebbe5442a0bcdcbd6f9c1-c6a0-4822-97b4-e9b7a292fd8e/skills/59f820d4-ca7c-47...`. The page title is "IBM Watson Assistant Lite" with an "Upgrade" link. The main heading is "#product_information", with a "Last updated: 4 days ago" timestamp and a "Try it" button. The interface is divided into sections for configuring an intent:

- Intent name:** A text box containing "#product_information".
- Description (optional):** A text box with the placeholder "Add a description to this intent".
- User example:** A section with the instruction "Add unique examples of what the user might say. (Pro tip: Add at least 5 unique examples to help Watson understand)". It includes a text input field with the placeholder "Type a user example here, e.g. I want to pay my credit card bill". Below this are "Add example" and "Show recommendations" buttons.
- Entity Annotation:** A toggle switch for "Annotate entities" with a "What's this?" link.
- User examples list:** A section titled "User examples (14) ↑" with a sort icon "Added ↑↓". It shows "Showing 1-14 of 14 examples" and a pagination control "1 of 1 pages".

The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 11:47 on 15-06-2020.

IBM Watson Assistant Lite Upgrade

Cloth Center Customer Service

Save new version Try it

Intents

Entities

Dialog

Options

Analytics

Versions

Content Catalog

Add node Add child node Add folder

Welcome
welcome
1 Responses / 0 Context Set / Does not return

greeting
#greetings || @greet
4 Responses / 0 Context Set / Does not return

ask for products
#product_information
2 Responses / 0 Context Set / Does not return

products
#items || @products
1 Responses / 0 Context Set / Does not return

enquiry

1.jpg

Show all X

Type here to search

11:49 15-06-2020

IBM Watson Assistant Lite Upgrade

Cloth Center Customer Service

Intents

Entities

Dialog

Options

Analytics

Versions

Content Catalog

Add node Add child node Add folder

Welcome
welcome
1 Responses / 0 Context Set / Does not return

greeting
#greetings || @greet
4 Responses / 0 Context Set / Does not return

ask for products
#product_information
2 Responses / 0 Context Set / Does not return

products
#items || @products
1 Responses / 0 Context Set / Does not return

enquiry

ask for products

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

If assistant recognizes

#product_information

Then callout to my webhook

Parameters

Key	Value
input	<?input_text?>

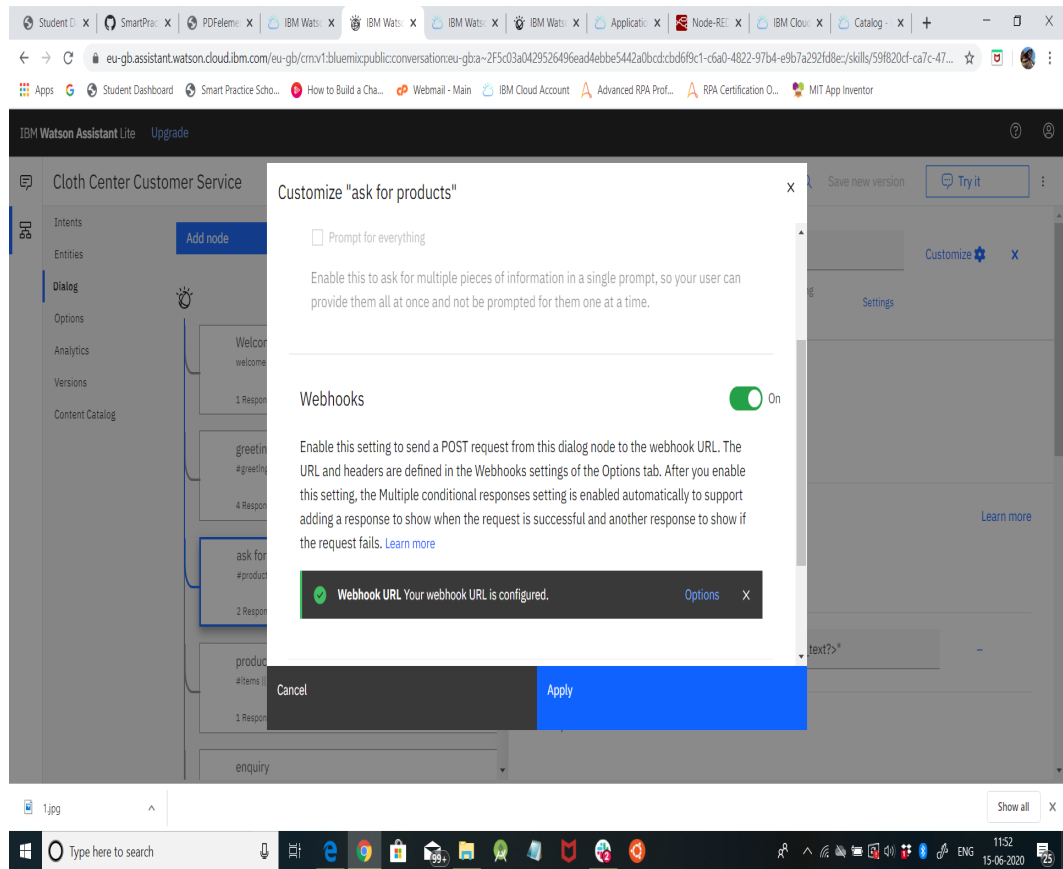
Add parameter +

1.jpg

Type here to search

11:50 15-06-2020

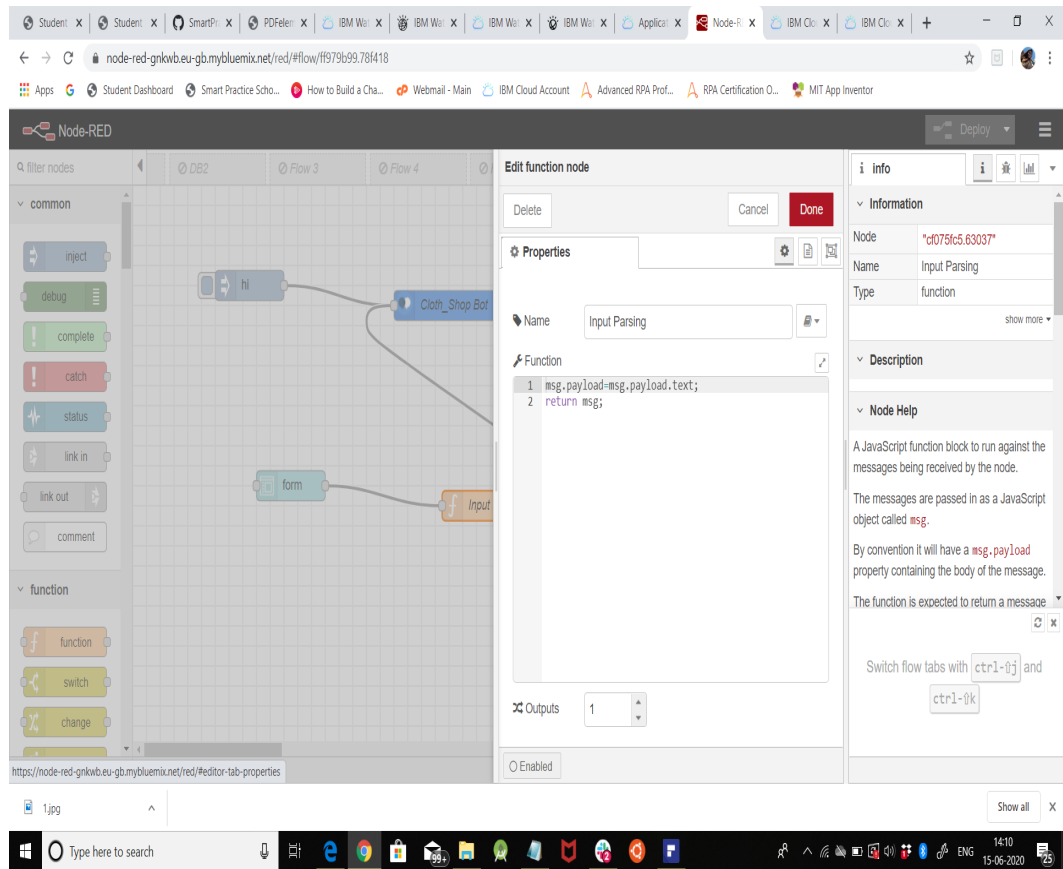
The screenshot displays the IBM Watson Assistant Lite web interface. The browser's address bar shows the URL: `eu-gb.assistant.watson.cloud.ibm.com/eu-gb/cmv1:bluemixpublicconversation:eu-gb-a~2f5c03a0429526496ead4ebbe5442a0bcdcbd6f9c1-c6a0-4822-97b4-e9b7a292fd8e/skills/59f820d4-ca7c-47...`. The page title is "Cloth Center Customer Service". On the left, a sidebar menu lists various configuration options: Intents, Entities, Dialog, Options, Webhooks (highlighted), Disambiguation, Autocorrection, Irrelevance Detection, System Entities, Analytics, Versions, and Content Catalog. The main content area is titled "Webhooks" and includes a description: "A webhook is a mechanism that allows you to call out to an external program based on events in your dialog." Below this, the "Webhook setup" section prompts the user to "Specify the request URL for an external API you want to be able to invoke from dialog nodes." A text box contains the URL: `https://eu-gb.functions.cloud.ibm.com/api/v1/web/si05202000315%40smc`. The "Headers" section instructs the user to "Add HTTP headers for authorization or any other parameters required for invoking the webhook." It features a table with columns "Header name" and "Header value", and buttons for "Add header" and "Add authorization". A "Next step" section is visible at the bottom of the main content area. The interface also includes a "Save new version" button and a "Try it" button in the top right corner. The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 11:51 on 15-06-2020.



The screenshot displays the IBM Watson Assistant Lite web interface. The browser's address bar shows the URL: `eu-gb.assistant.watson.cloud.ibm.com/eu-gb/cmv1/bluemixpublicconversation-eu-gb-a~2f5c03a0429526496ead4ebbe5442a0bcdcbd6f9c1-c6a0-4822-97b4-e9b7a292fd8e/skills/59f820dc-ca7c-47...`. The interface is titled "Cloth Center Customer Service" and includes a sidebar with navigation options: Intents, Entities, Dialog, Options, Analytics, Versions, and Content Catalog. The "Dialog" section is currently selected, showing a flowchart with a node named "ask for products". Below this node, a condition is set: "If assistant recognizes #product_information". The action for this node is "Then callout to my webhook", with a parameter table defined as follows:

Key	Value
input	*?input_text?*

To the right of the dialog editor is a "Try it out" panel. It contains a text input field with the value "admin" and a dropdown menu set to "#product_information". Below this, a note states: "An Admin has some extra privilege including all privilege of visitor and user. Admin can add products, edit product information and add/remove product. Admin can add user, edit user information and can remove user. Admin can ship order to user based on order placed by sending confirmation mail. Design and Implementation of E-Commerce Site for Online Shopping". At the bottom of the interface, a Windows taskbar is visible, showing the search bar and various application icons. The system clock indicates the time is 14:07 on 15-06-2020.



Chapter 5

FLOWCHART

Firstly, go to manage pallette and install dashboard.

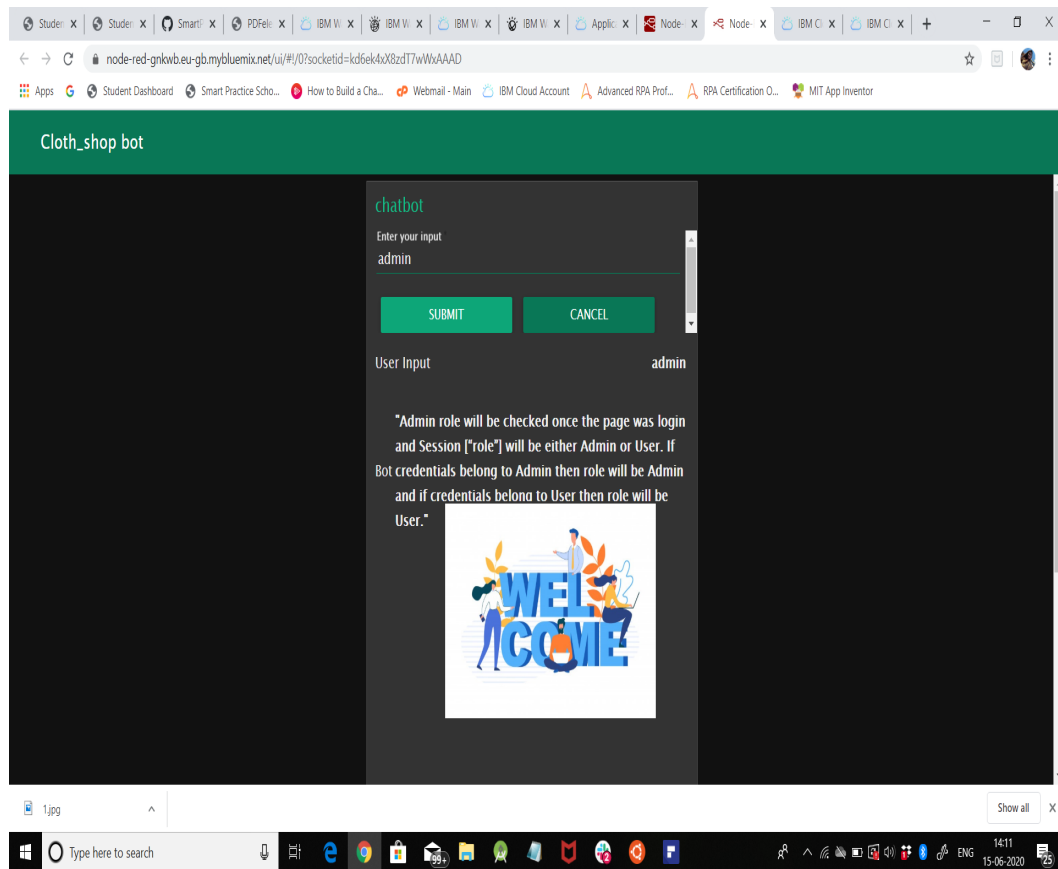
Now create the flow with the help of following node:

- **UI Form**
- **Function**
- **Assistant**
- **Debug**
- **UI text**
- **Template**
- **Audio out**
- **Language Translator**
- **Play audio**

Chapter 6

RESULTS

Finally our Node-RED dash board integrates all the components and displayed in the Dashboard UI by typing URL - <https://node-red-qgwak.eu-gb.mybluemix.net/ui> in browser.



Chapter 7

PRONS AND CONS

A summary of what scenarios will be tested, how they will be tested, how often they will be tested, and so on.

7.1 PROS:

- Companies can deploy chatbots to rectify simple and general human queries.
- Reduces man power
- Cost efficient
- No need to divert calls to customer agent and customer agent can look on other works.
- No need to divert calls to customer agent and customer agent can look on other works.

7.2 CONS:

- Some times chatbot can mislead customers
- Giving same answer for different sentiments
- Sometimes cannot connect to customer sentiments and intentions.

Chapter 8

APPLICATIONS

- It can deploy in popular social media applications like facebook messenger, slack, telegram
- Chatbot can deploy any website to clarify basic doubts of viewers.

Chapter 9

CONCLUSION

By doing the above procedure and all we successfully created Intelligent helpdesk smart chatbot using Watson assistant, Watson discovery, Node-RED and cloud-functions.

Chapter 10

FUTURE SCOPE

We can include watson studio text to speech and speech to text services to access the chatbot handsfree. This is one of the future scope of this project.

Bibliography

10.1

```
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.  
@param string params.input ** @return object ** / const assert = require('assert'); const DiscoveryV  
require('watson-developer-cloud/discovery/v1'); /**** main() will be run when you invoke this action  
* @param CloudFunctions.actions.acceptSingleParameter, which must be a JSON object. *  
* @return The output of this action, which must be a JSON object. ** / function main(params) return ne  
// developer.ibm.com/articles/introduction-watson-discovery/https : //medium.com/@rimaib  
red-watson-discovery-chatbot-telegram-ce616ddcd0d9https : //www.youtube.com/embed/Jp  
//cloud.ibm.com/docs/openwhisk?topic=cloud-functions-getting-startedhttps :  
//www.ibm.com/cloud/watson-assistant/
```