PROJECT DEVELOPMENT PHASE

SPRINT 2

Development of Web User Interface in Node-RED platform

Date	05 November 2022
Team ID	PNT2022TMID18885
Project Name	Project - Personal Assistance for Seniors Who Are Self-Reliant

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	**Registration:** Creation of IBM serviceslike NodeRED, CloudantDB, TTS Service and design of IoT system	USN-1	As a user,I should login into my IBM Cloud account.	2	High	Anne Angelina J, Kawin M
Sprint-2	**Web UI:** Creating web UI using node-red and connect it toIBM Cloudant db	USN-2	As a user,I should be able to feed the medicine name and intake time in the web UI	2	High	Akshayasri S, Bhavani R K
Sprint-3	**Hardware implementation:** Developing Python code to retrieve data from cloudant db to send that data to IoT device at the appropriate time	USN-3	As a user, I should be able to send the medicine name to the IoT device at the scheduled time	2	High	Akshayasri S, Kawin M
Sprint-4	**Software implementation:** Converting the data received from cloud asvoice using IBM Text toSpeech service	USN-4	As a user, I must be able hear the medicine name which is to be taken at the appropriatetime	2	High	Anne Angelina J, Bhavani R K

Objectives:

i) To create a form UI in Node-Red platform to enter the medicine name and time of intake.

ii) To send the medicine name at the scheduled time.

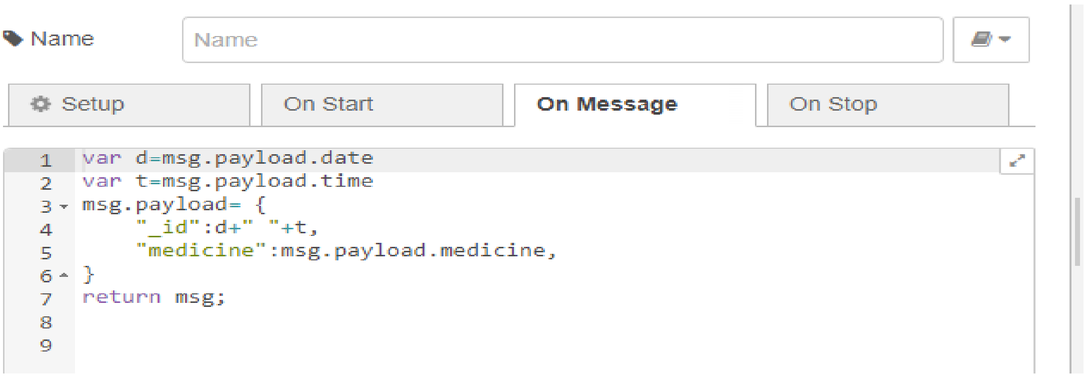
iii) Complete Web UI flow

iv) Deploying Web UI

i) Creating a form in Node-Red to enter medicine details:

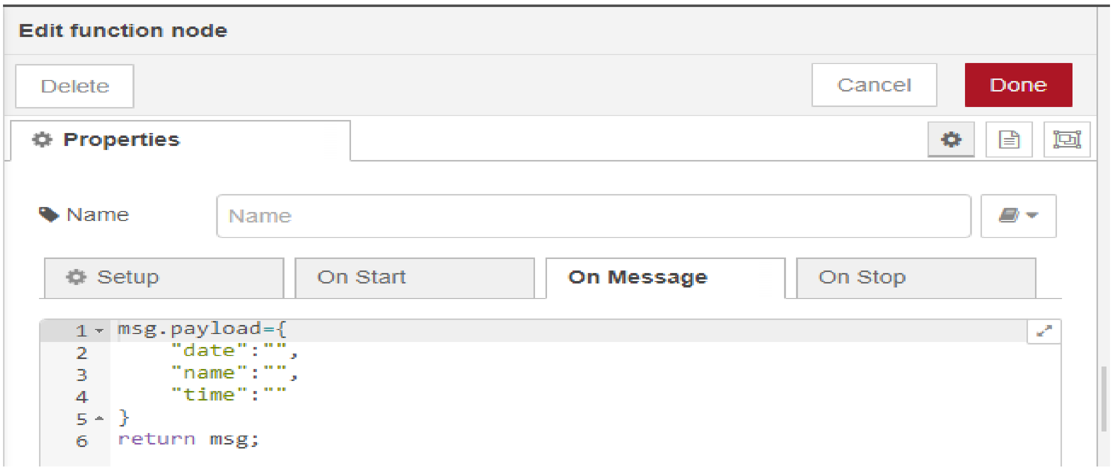
- ➢ Web UI can be created by using form UI from Node-Red Dashboard
- ➢ The Dashboard can be installed from node-red palette
- ➢ The medicine name, intake time and date is entered in the form

- The medicine details are wired and stored to the IBM cloudant db database with the help of a function.



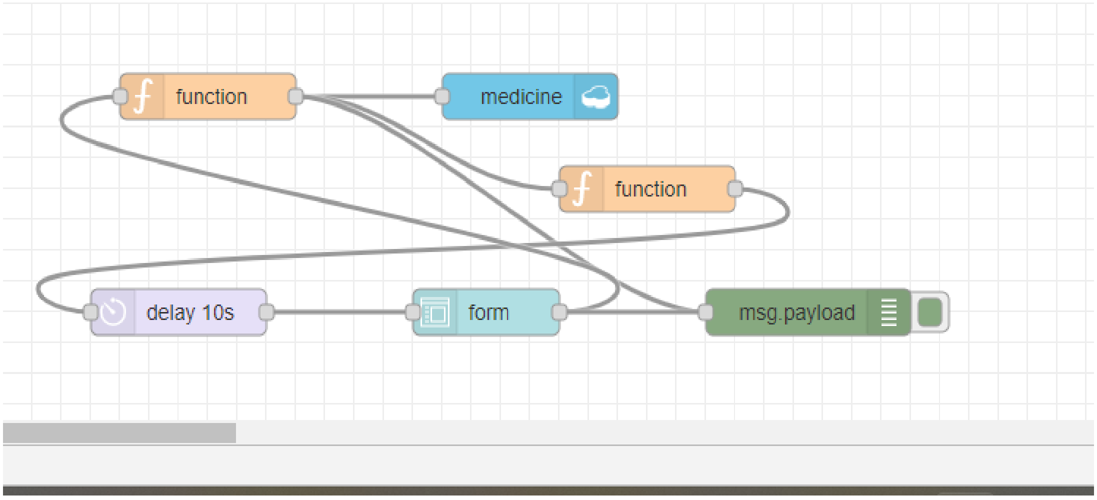
```
var d=msg.payload.date
var t=msg.payload.time
msg.payload= {
      "_id":d+" "+t,
      "medicine":msg.payload.medicine,
}
return msg;
```

- The below function is used to reset or clear the form after a delay to add more medicine details

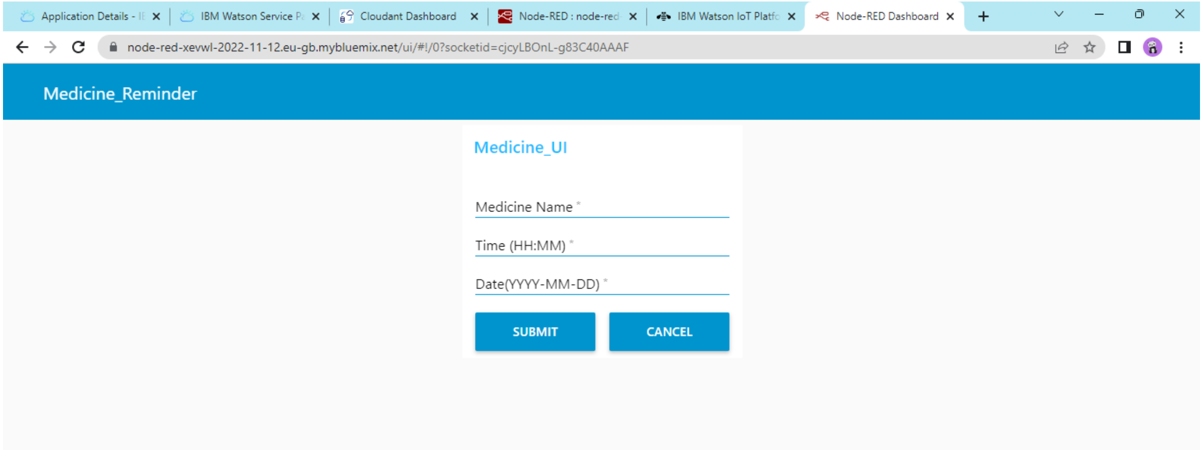


```
msg.payload={
    "date":"",
    "name":"",
    "time":""
}
return msg;
```

- Node-Red Flow to create a form

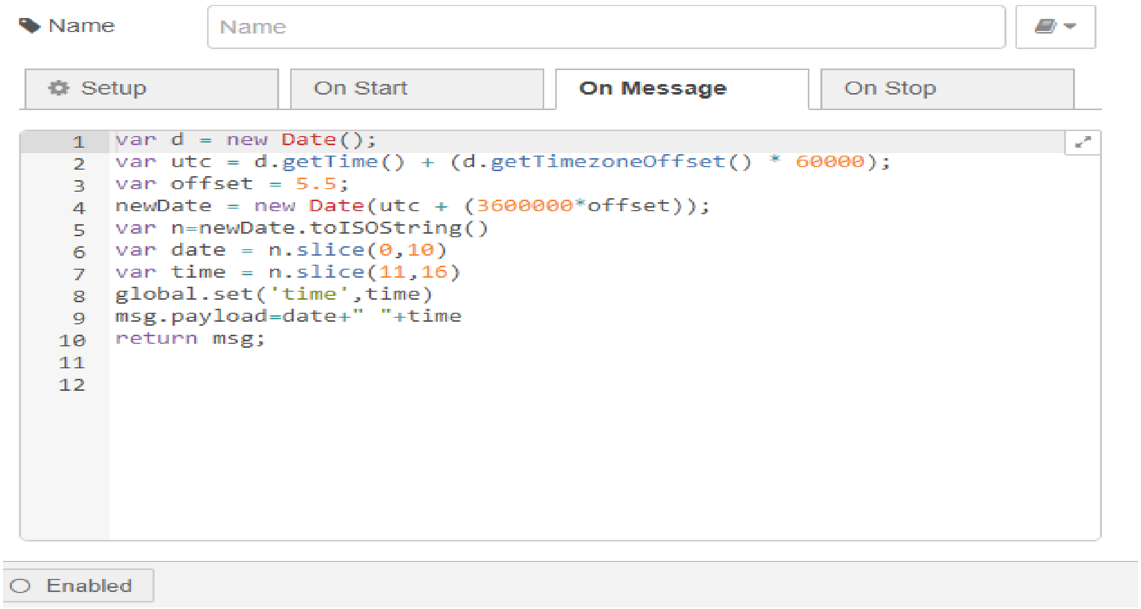


- Web User Interface for entering medicine details



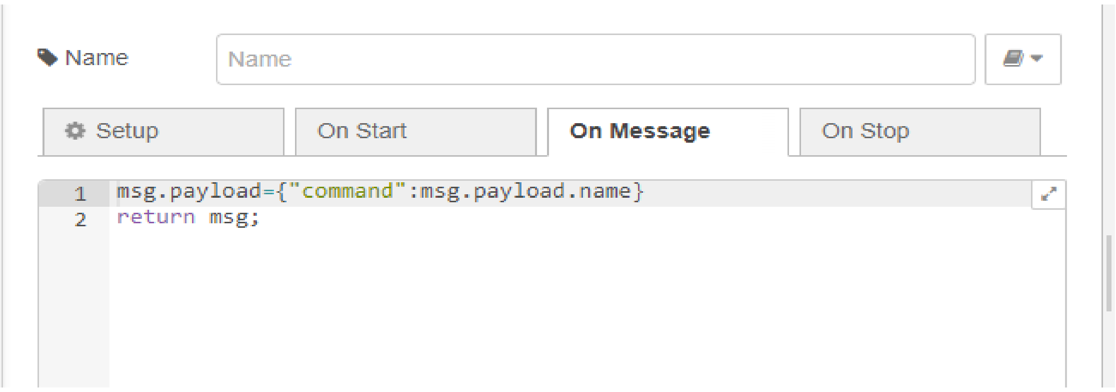
ii) Sending the medicine name at the appropriate time:

- A function is created to compare the present time with the scheduled time.
- The function to get the present time is shown below.



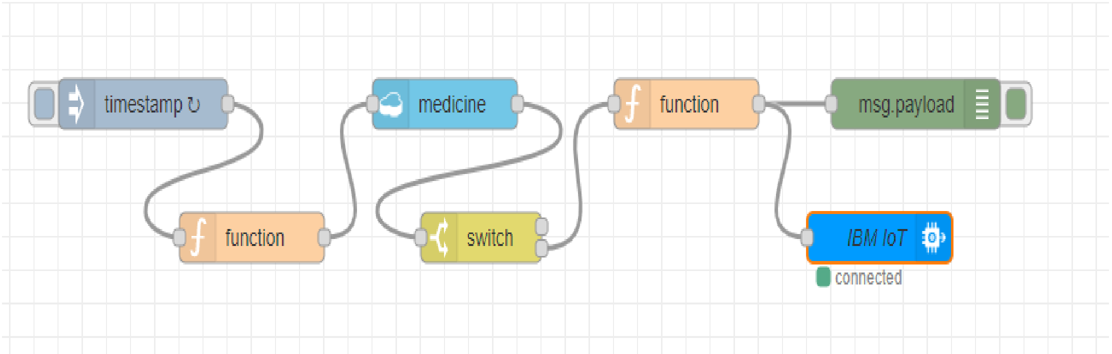
```
var d = new Date();
var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
var offset = 5.5;
newDate = new Date(utc + (3600000*offset));
var n=newDate.toISOString()
var date = n.slice(0,10)
var time = n.slice(11,16)
global.set('time',time)
msg.payload=date+" "+time
return msg;
```

- After obtaining the present time, cloudant node is connected to the function in order search the scheduled time by _id
- The cloudant node is connected to the IBM cloudant database.
- If the present time and scheduled time in the database matches, then the name of the medicine will be sent to the IoT device using switch node and a function following that node.

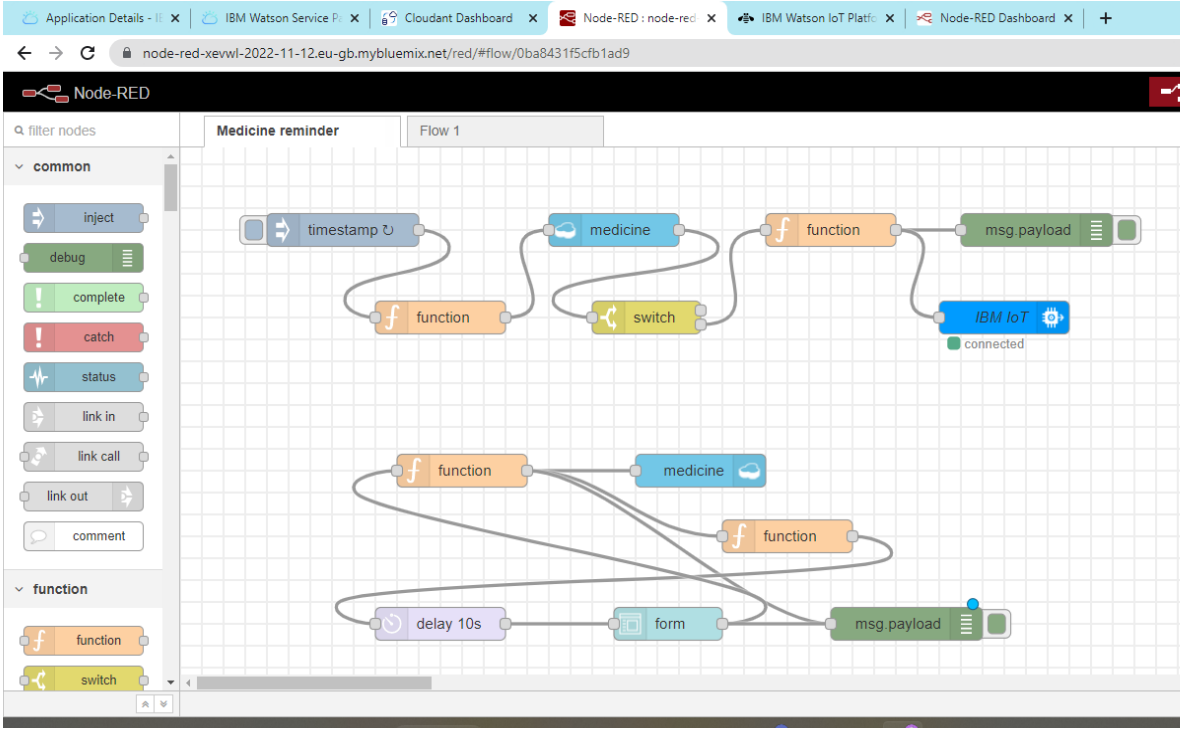


```
msg.payload={"command":msg.payload.name}
return msg;
```

➢ Overall node flow for sending the medicine name to IoT device

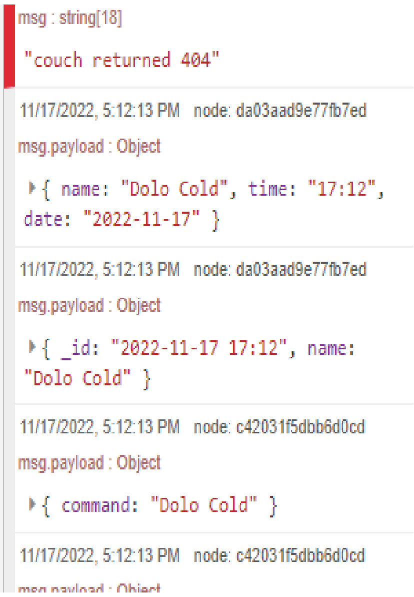


iii) Complete Web UI flow:

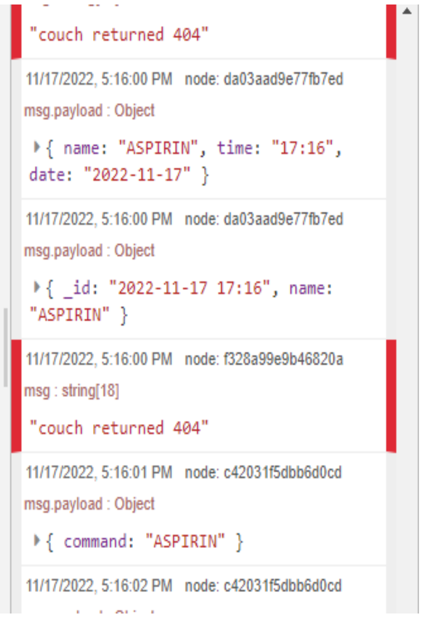


iv) Deploying Web UI:

The debug window in Node-Red depicts the medicine details.

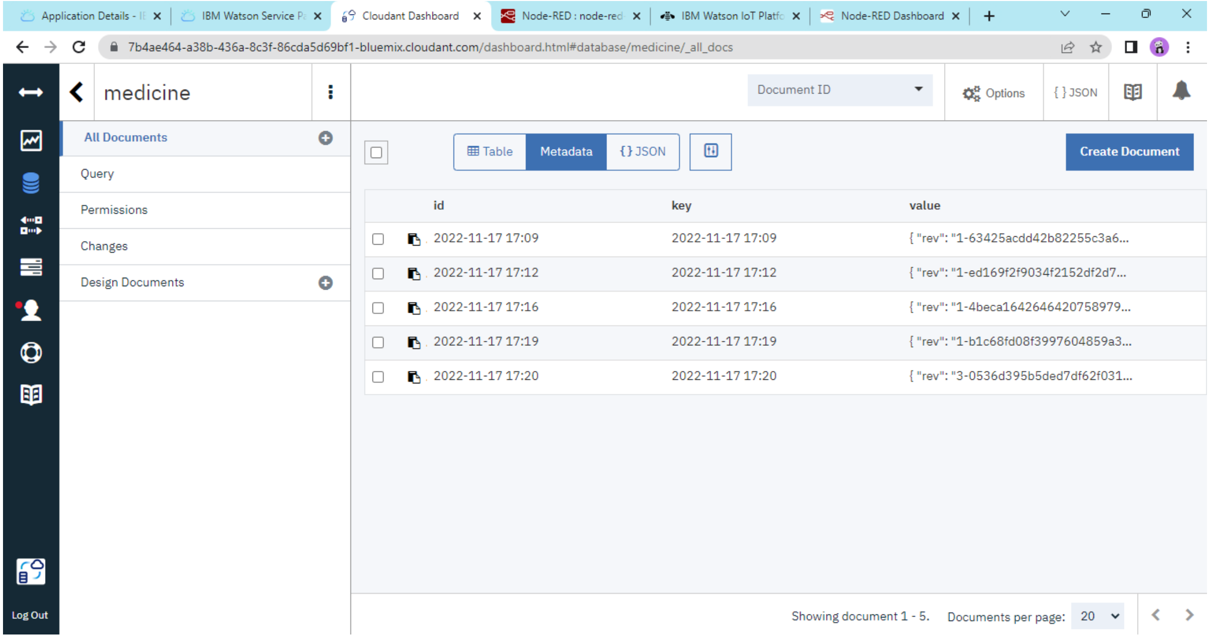


```
msg : string[18]
"couch returned 404"

11/17/2022, 5:12:13 PM   node: da03aad9e77fb7ed
msg.payload : Object
▸{ name: "Dolo Cold", time: "17:12",
date: "2022-11-17" }

11/17/2022, 5:12:13 PM   node: da03aad9e77fb7ed
msg.payload : Object
▸{ _id: "2022-11-17 17:12", name:
"Dolo Cold" }

11/17/2022, 5:12:13 PM   node: c42031f5dbb6d0cd
msg.payload : Object
▸{ command: "Dolo Cold" }

11/17/2022, 5:12:13 PM   node: c42031f5dbb6d0cd
msg.payload : Object
```



```
"couch returned 404"

11/17/2022, 5:16:00 PM   node: da03aad9e77fb7ed
msg.payload : Object
▸{ name: "ASPIRIN", time: "17:16",
date: "2022-11-17" }

11/17/2022, 5:16:00 PM   node: da03aad9e77fb7ed
msg.payload : Object
▸{ _id: "2022-11-17 17:16", name:
"ASPIRIN" }

11/17/2022, 5:16:00 PM   node: f328a99e9b46820a
msg : string[18]
"couch returned 404"

11/17/2022, 5:16:01 PM   node: c42031f5dbb6d0cd
msg.payload : Object
▸{ command: "ASPIRIN" }

11/17/2022, 5:16:02 PM   node: c42031f5dbb6d0cd
```

- ➢ The following image shows the medicine details stored in the cloudant db under the medicine database.

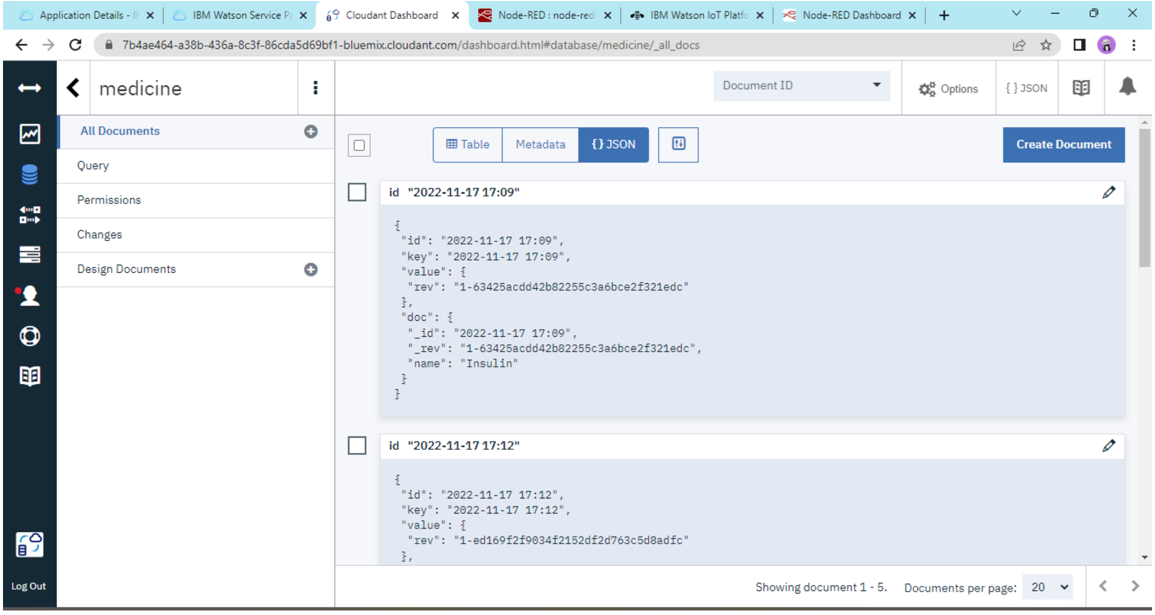


7b4ae464-a38b-436a-8c3f-86cda5d69bf1-bluemix.cloudant.com/dashboard.html#database/medicine/_all_docs

	id	key	value
□	2022-11-17 17:09	2022-11-17 17:09	{"rev": "1-63425acdd42b82255c3a6...
□	2022-11-17 17:12	2022-11-17 17:12	{"rev": "1-ed169f2f9034f2152df2d7...
□	2022-11-17 17:16	2022-11-17 17:16	{"rev": "1-4beca1642646420758979...
□	2022-11-17 17:19	2022-11-17 17:19	{"rev": "1-b1c68fd08f3997604859a3...
□	2022-11-17 17:20	2022-11-17 17:20	{"rev": "3-0536d395b5ded7df62f031...

Showing document 1 - 5. Documents per page: 20

- The following image shows the details of the medicine in the medicine database in JSON format



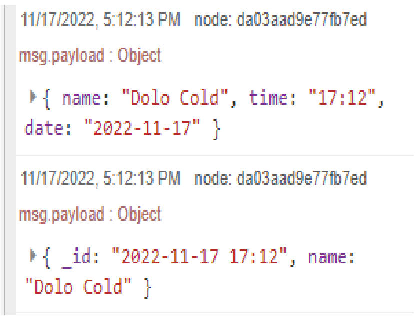
id "2022-11-17 17:09"

```
{
 "id": "2022-11-17 17:09",
 "key": "2022-11-17 17:09",
 "value": {
  "rev": "1-63425acdd42b82255c3a6bce2f321edc"
 },
 "doc": {
  "_id": "2022-11-17 17:09",
  "_rev": "1-63425acdd42b82255c3a6bce2f321edc",
  "name": "Insulin"
 }
}
```

id "2022-11-17 17:12"

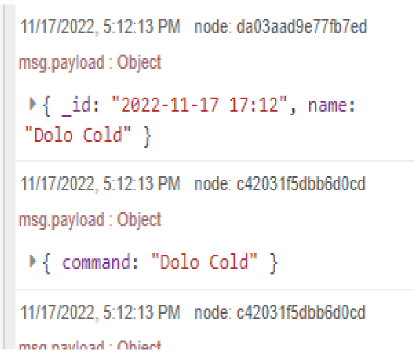
```
{
 "id": "2022-11-17 17:12",
 "key": "2022-11-17 17:12",
 "value": {
  "rev": "1-ed169f2f9034f2152df2d763c5d8adfc"
 },
```

- **The medicine reminder operation is depicted below:**
    - The medicine name Dolo Cold and its intake time 17:12 and the date 17.11.2022 is stored in the database.



```
11/17/2022, 5:12:13 PM   node: da03aad9e77fb7ed
msg.payload : Object
 ▸{ name: "Dolo Cold", time: "17:12",
 date: "2022-11-17" }

11/17/2022, 5:12:13 PM   node: da03aad9e77fb7ed
msg.payload : Object
 ▸{ _id: "2022-11-17 17:12", name:
 "Dolo Cold" }
```

    - When it’s time to take the medicine, the medicine name is sent as a command to the IoT device.



```
11/17/2022, 5:12:13 PM   node: da03aad9e77fb7ed
msg.payload : Object
 ▸{ _id: "2022-11-17 17:12", name:
 "Dolo Cold" }

11/17/2022, 5:12:13 PM   node: c42031f5dbb6d0cd
msg.payload : Object
 ▸{ command: "Dolo Cold" }

11/17/2022, 5:12:13 PM   node: c42031f5dbb6d0cd
msg.payload : Object
```