

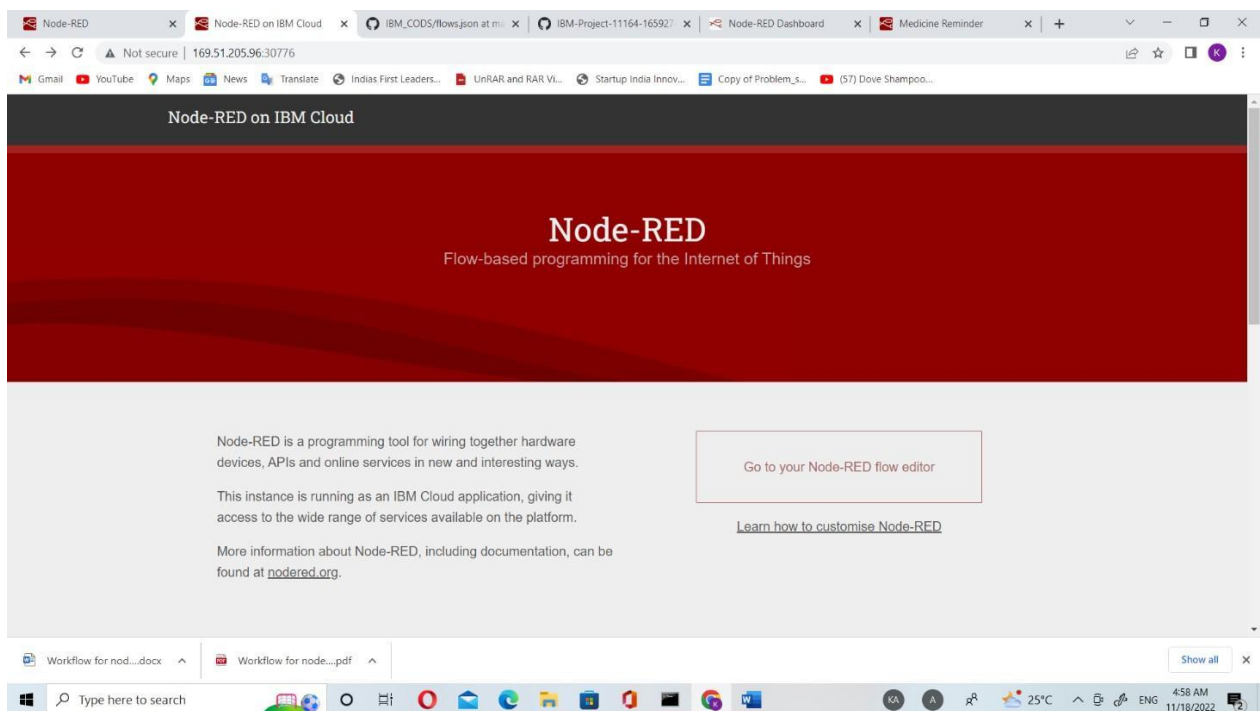
SPRINT DELIVERY – 3

TEAM ID : PNT2022TMID52021

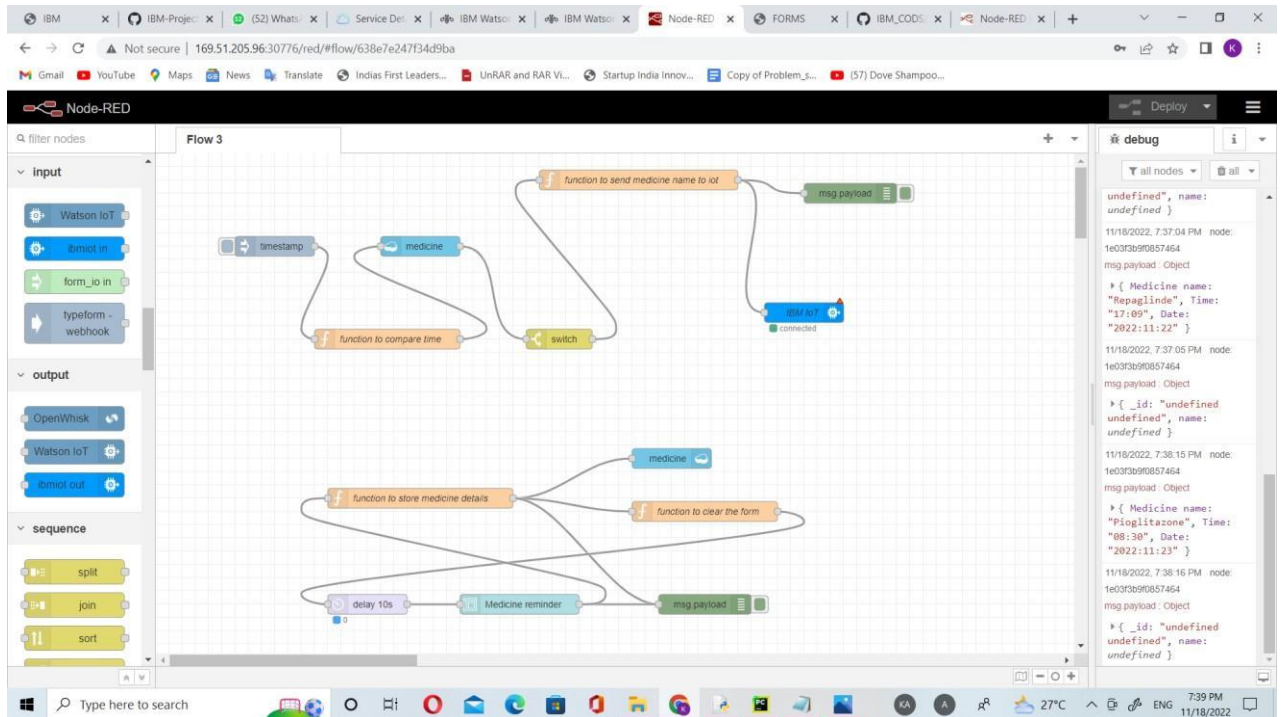
PROJECT NAME: PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF RELIANT

WORKFLOW FOR IOT SCENARIOS USING NODE RED(CREATING WEB UI)

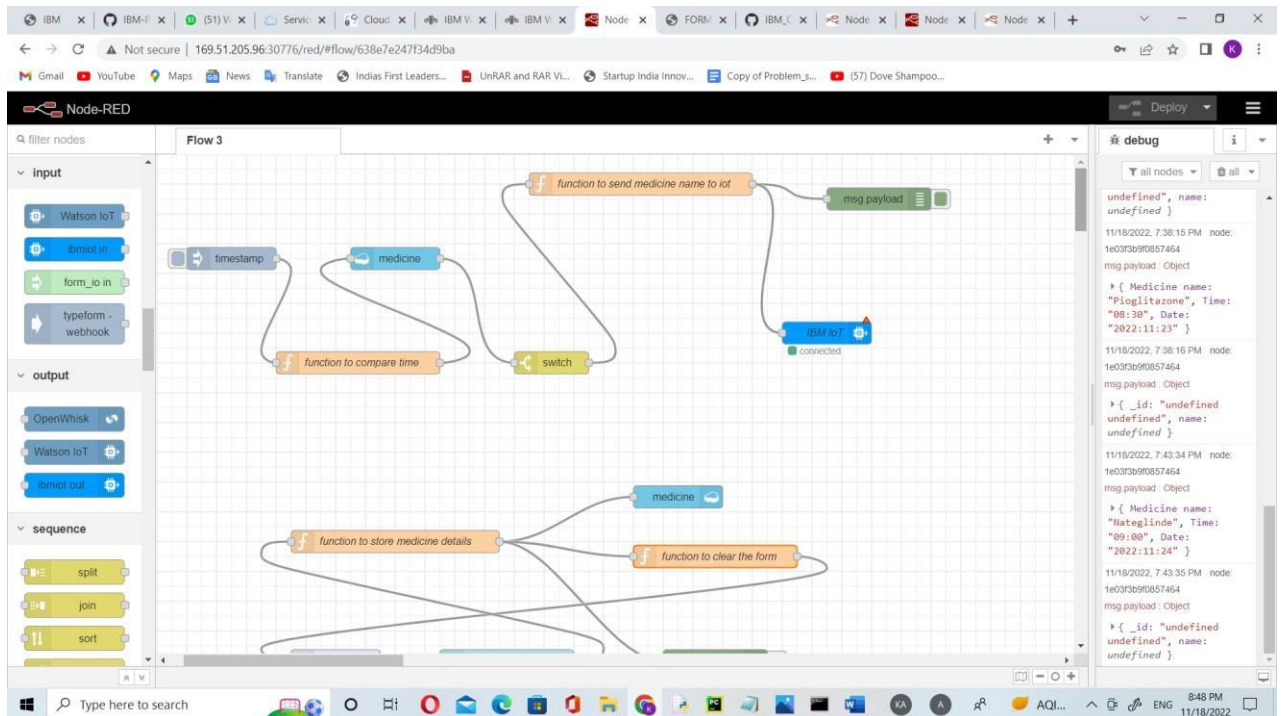
NODE-RED INSTALLATION:



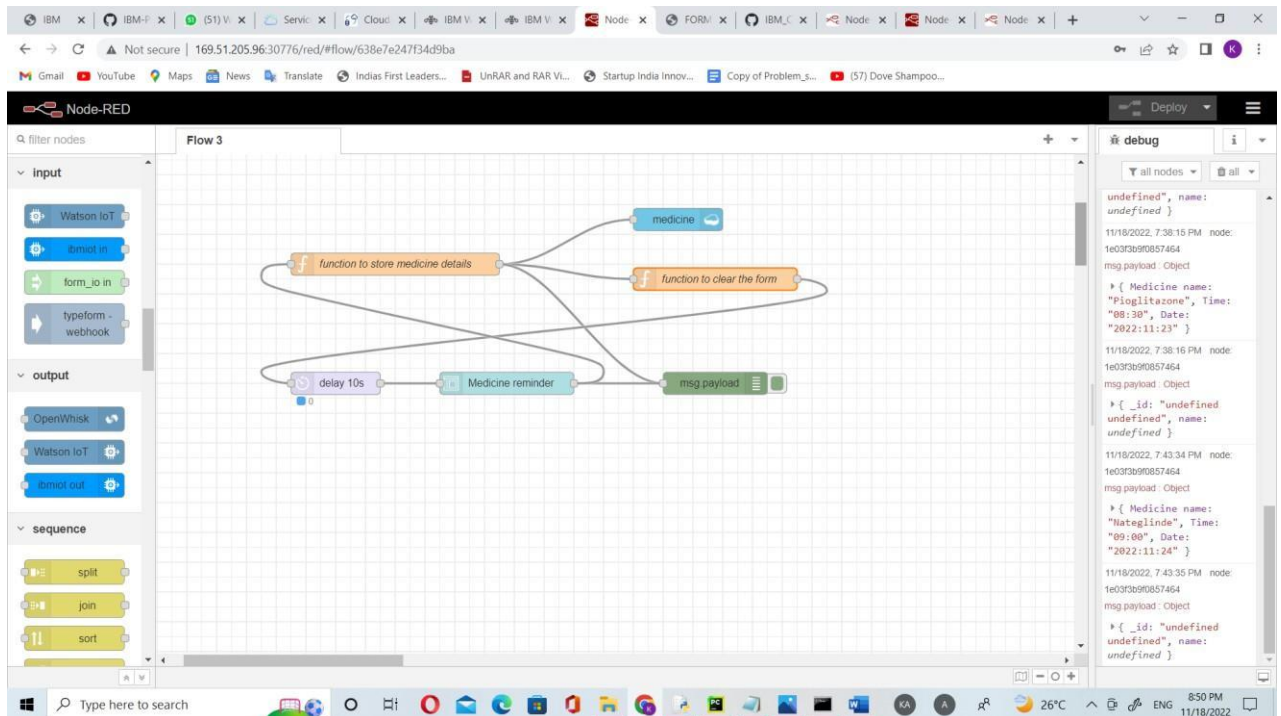
CREATING MEDICINE REMINDER FORM: (COMPLETE)



SENDING MEDICINE NAME AT APPROPRIATE TIME:

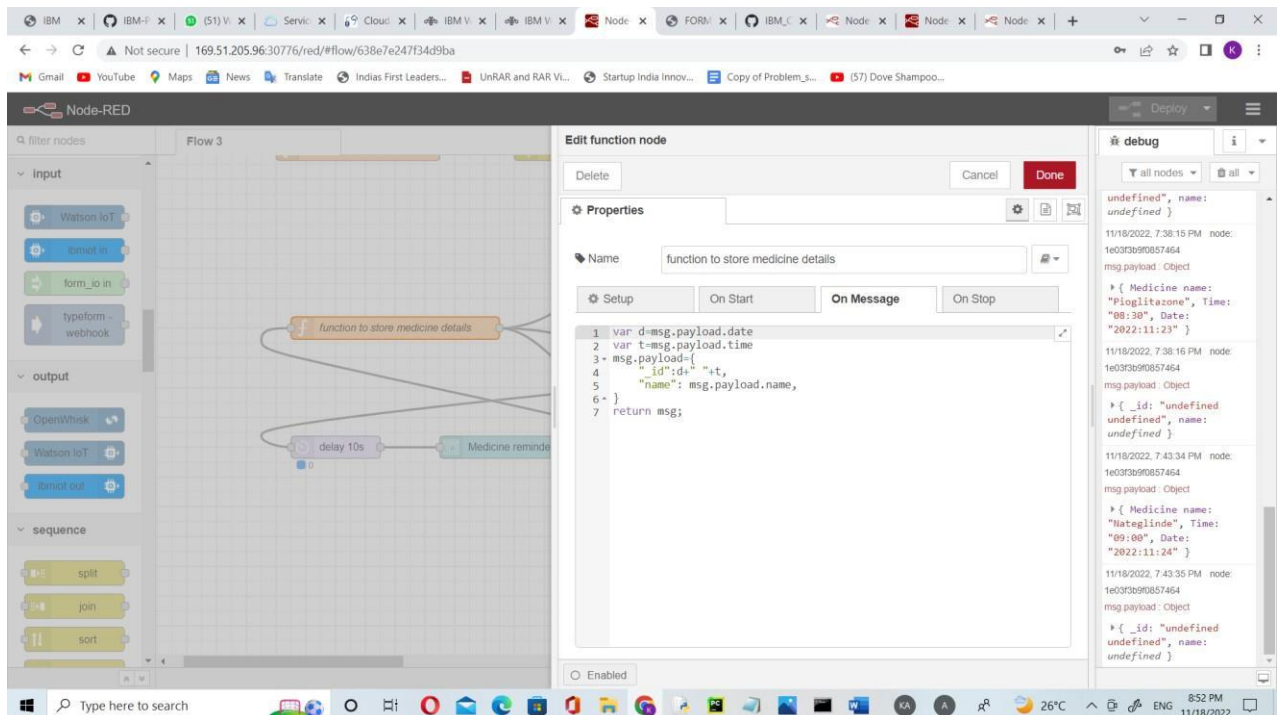


TO ENTER MEDICINE DETAILS:



FUNCTIONS:

1. FUNCTION TO STORE MEDICINE DETAILS



2. FUNCTION TO CLEAR THE FORM

The screenshot shows the Node-RED web interface. On the left, the 'Input' sidebar contains nodes like 'Watson IoT', 'formio in', 'typeform-webhook', and 'OpenWhisk'. The 'output' sidebar includes 'Watson IoT' and 'formio out'. The 'sequence' sidebar has 'split', 'join', and 'sort'. The main workspace shows a flow with a function node labeled 'function to clear the form' and a 'delay 10s' node. The 'Edit function node' panel is open, showing the function code:

```
1 - msg.payload = {
2   "date": "",
3   "name": "",
4   "time": ""
5 - }
6 return msg;
```

The 'debug' console on the right shows the message payload before and after the function execution, confirming that the form fields are cleared.

3. FUNCTION TO SEND MEDICINE NAME TO IOT

The screenshot shows the Node-RED web interface. The main workspace shows a flow with a function node labeled 'function to send medicine name to iot'. The 'Edit function node' panel is open, showing the function code:

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

The 'debug' console on the right shows the message payload before and after the function execution, confirming that the medicine name is sent to the IoT device.

4. FUNCTION TO COMPARE TIME

The screenshot shows the Node-RED web interface. A flow named 'Flow 3' is visible, containing a 'timestamp' node, a 'function to compare time' node, and a 'function to store medicine details' node. The 'function to compare time' node is selected, and its configuration panel is open. The configuration panel shows the node name 'function to compare time' and the following JavaScript code:

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

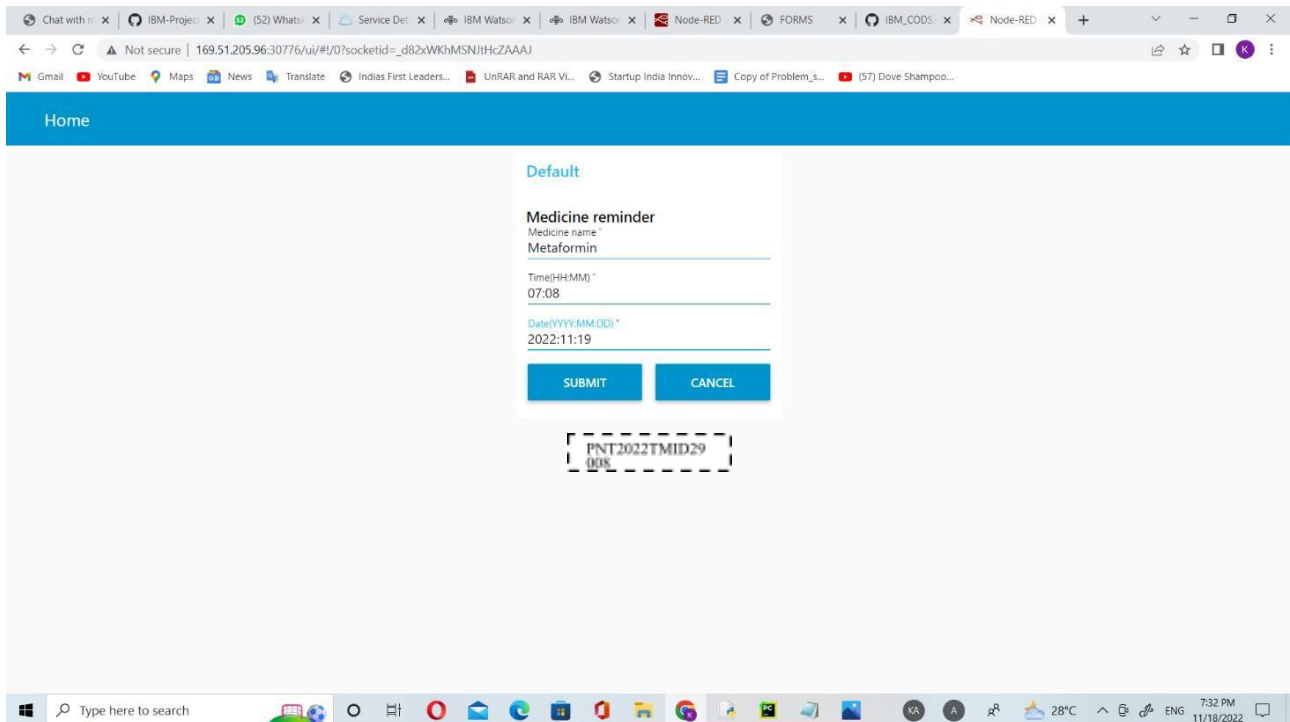
The debug console on the right shows the output of the function, displaying the date and time in ISO format.

WEB UI(MEDICINE REMINDER PAGE):

The screenshot shows the 'Medicine reminder' web UI. The page has a blue header with the text 'Home'. Below the header, there is a form titled 'Medicine reminder' with the following fields:

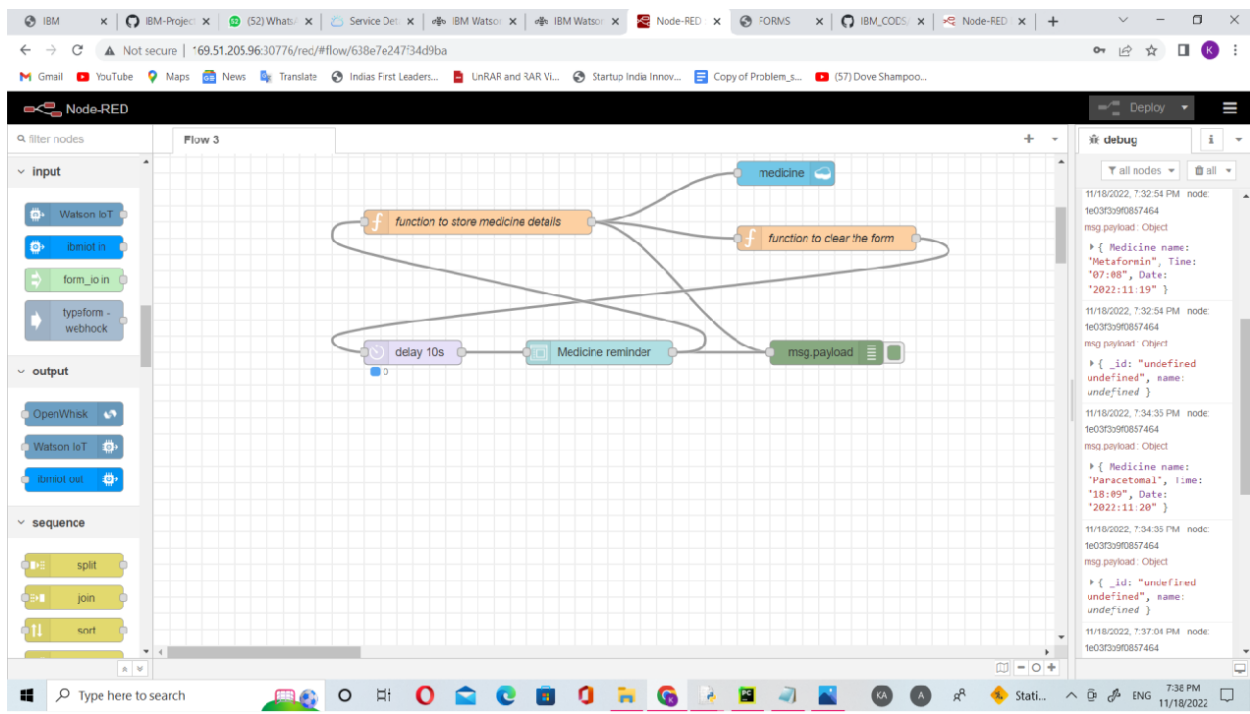
- Medicine name *
- Time(HH:MM) *
- Date(YYYY:MM:DD) *

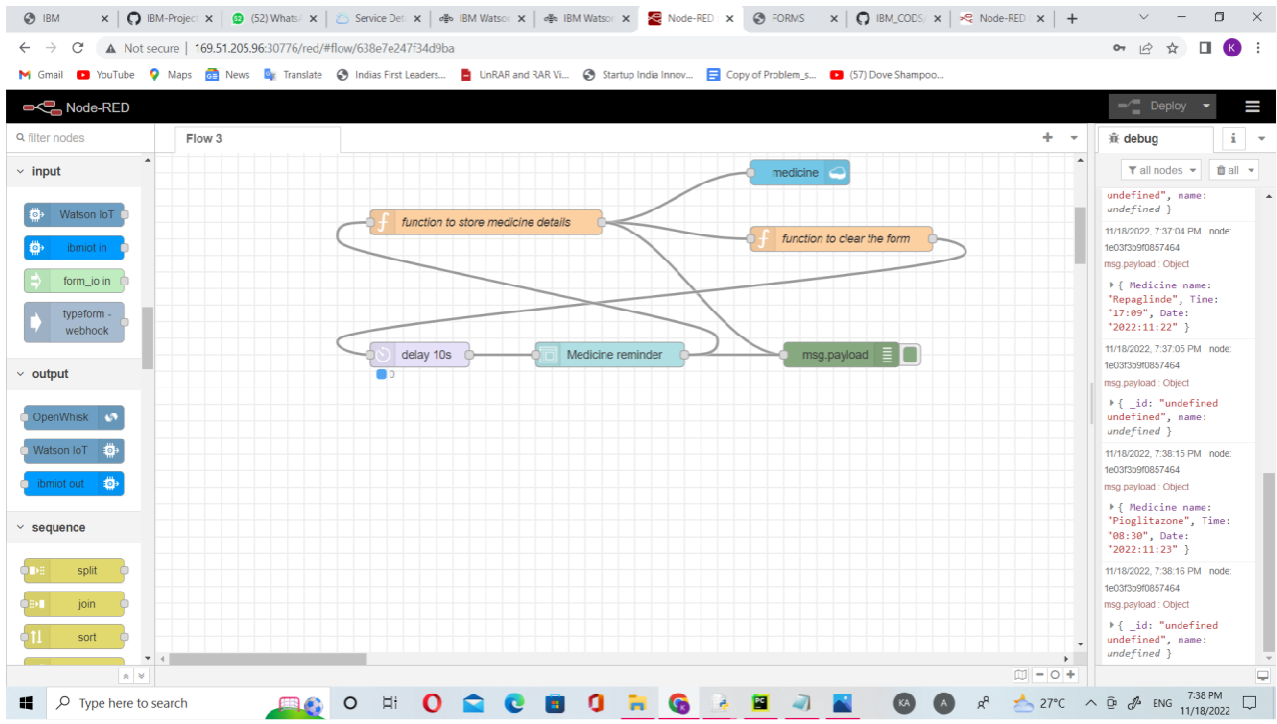
Below the form, there are two buttons: 'SUBMIT' and 'CANCEL'. The page is displayed in a web browser window with the URL 169.51.205.96:30776/ui/#/0?socketid=_d82xWKhMSNJIHcZAAA.



DEBUG WINDOW:

The details of medicine name, date and time that are entered are shown in debug window.





MEDICINE DATABASE IN CLOUDANT DB:

The Cloudant database interface shows a list of documents in the 'medicine' database. The documents are displayed in a table view with columns for '_id' and 'name'. The documents are as follows:

_id	name
Time:07:08 Date:2022:11:19	{ "name": "metaformin" }
Time:08:30 Date:2022:11:23	{ "name": "Floglitazone" }
Time:09:00 Date:2022:11:24	{ "name": "Nateline" }
Time:17:09 Date:2022:11:22	{ "name": "Repaglinde" }
Time:18:09 Date:2022:11:18	{ "name": "paracetomal" }

The interface also includes a sidebar with navigation options (All Documents, Query, Permissions, Changes, Design Documents) and a bottom status bar showing 'Showing 2 of 3 columns' and 'Showing document 1 - 5'.

medicine

Document ID

Options

JSON

Create Document

id "Time:07:08 Date:2022:11:19"

```
{
  "id": "Time:07:08 Date:2022:11:19",
  "key": "Time:07:08 Date:2022:11:19",
  "value": {
    "rev": "2-c64f3638b765b1361da38a69c438d8d"
  },
  "doc": {
    "_id": "Time:07:08 Date:2022:11:19",
    "_rev": "2-c64f3638b765b1361da38a69c438d8d",
    "name": {
      "name": "metformin"
    }
  }
}
```

id "Time:08:30 Date:2022:11:23"

```
{
  "id": "Time:08:30 Date:2022:11:23",
  "key": "Time:08:30 Date:2022:11:23",
  "value": {
    "rev": "1-571feeb2e492ac5797f8bce24c53fe9f"
  },
  "doc": {
    "_id": "Time:08:30 Date:2022:11:23",
  }
}
```

Showing document 1 - 5. Documents per page: 20

IBM WATSON IOT PLATFORM:

IBM Watson IoT Platform

71115.internetofthings.ibmcloud.com/dashboard/devices/browse

Device ID Status Device Type Class ID Date Added Descriptive Location

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Disconnected	Iotsensor	Device	17 Nov 2022 20:29	

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	("Medicine name":"Nateglindol","Time":"09:00",...	json	a few seconds ago
event_1	("Medicine name":"Pioglitazone","Time":"08:30",...	json	a minute ago
event_1	("Medicine name":"Repaglinide","Time":"17:09",...	json	2 minutes ago
event_1	("Medicine name":"Paracetamol","Time":"13:09",...	json	2 minutes ago
event_1	("Medicine name":"Metformin","Time":"07:08",...	json	3 minutes ago

1 Simulation running

SIGNUP FORM IN NODE RED:

In debug window the details are stored.

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with the following nodes: a `[get] /signup` node, a `function 2` node, a `function 3` node, a `delay 5s` node, a `signup` node, and a `debug 1` node. The flow starts with the `[get] /signup` node, which connects to `function 2`. `function 2` connects to `function 3`, which then connects to the `debug 1` node. A `delay 5s` node is also connected to the `signup` node, which in turn connects to `function 3`. The left sidebar shows the 'filter nodes' and 'network' categories. The right sidebar shows the 'debug' console with the following log entries:

```
11/18/2022, 8:27:27 PM node: debug 1
msg.payload: Object
{ empty: }
11/18/2022, 8:29:18 PM node: debug 1
msg.payload: Object
{ Username: "keerthana", Password: "12345" }
11/18/2022, 8:29:18 PM node: debug 1
msg.payload: Object
{ Username: "keerthana", Password: "12345" }
11/18/2022, 8:37:43 PM node: debug 1
msg.payload: Object
{ Username: "keerthana", Password: "12345", email: "keerthanaandh4@gmail.com", age: "20" }
11/18/2022, 8:37:43 PM node: debug 1
msg.payload: Object
{ Username: "keerthana", Password: "12345", email: "keerthanaandh4@gmail.com", age: "20" }
```

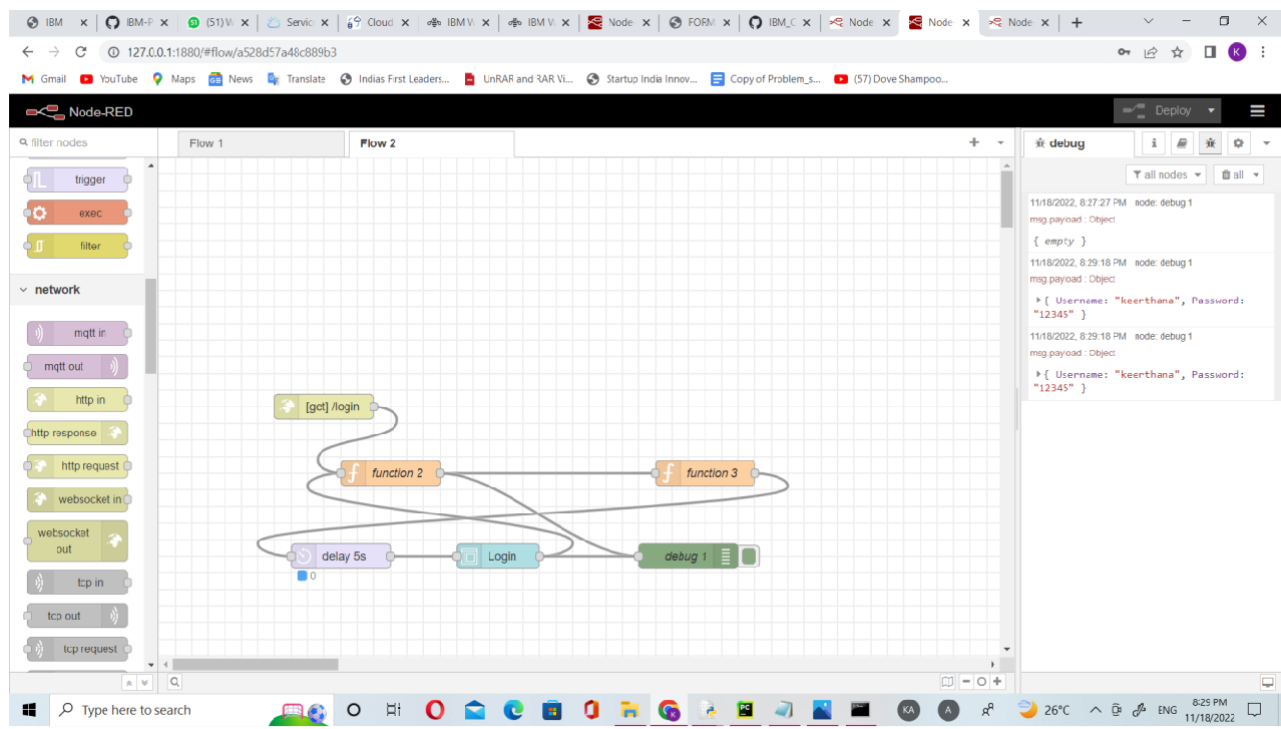
WEBUI(SIGNUP):

The screenshot shows a web browser displaying a signup form. The form is titled "Signup" and has a "Default" tab. The form fields are:

- Username:
- Password:
- Email:
- Age:

Below the form are two buttons: "SUBMIT" and "CANCEL". At the bottom of the page, there is a small box containing the text "PNT2022TMID29 (100)".

LOGIN FORM USING NODE RED:



The screenshot shows a web browser displaying a login form. The form is titled "Login" and has a "Default" tab. The form fields are:

- Username:** keertana
- Password:** *****

Below the form fields are two buttons: **SUBMIT** and **CANCEL**.

Below the form, there is a dashed box containing the text: **PNT2022TMID29**

MEDICINE REMINDER PAGE:

http://169.51.205.96:30776/ui/#!/0?socketid=_d82xWKhMSNJtHcZAAAJ

SIGNUP PAGE:

<http://127.0.0.1:1880/ui/#!/0?socketid=5aWJRQVNg8uphtMhAAAA>

