

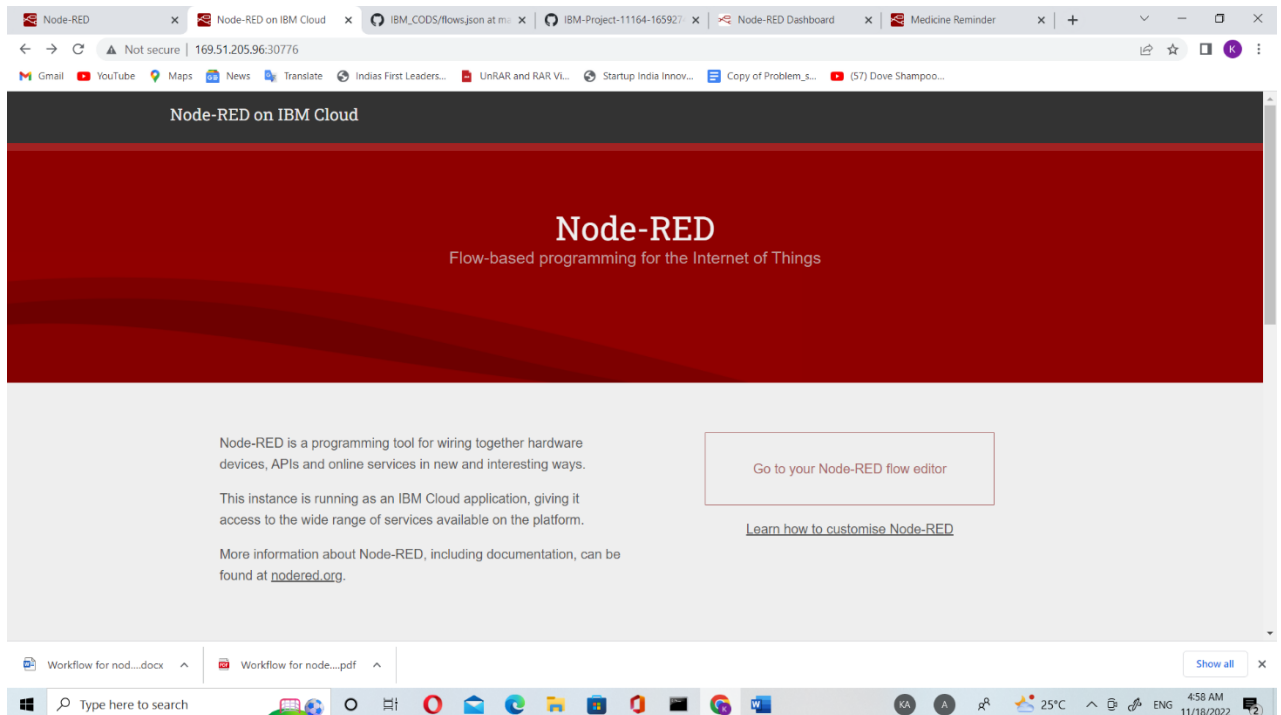
SPRINT DELIVERY – 3

TEAM ID : PNT2022TMID29008

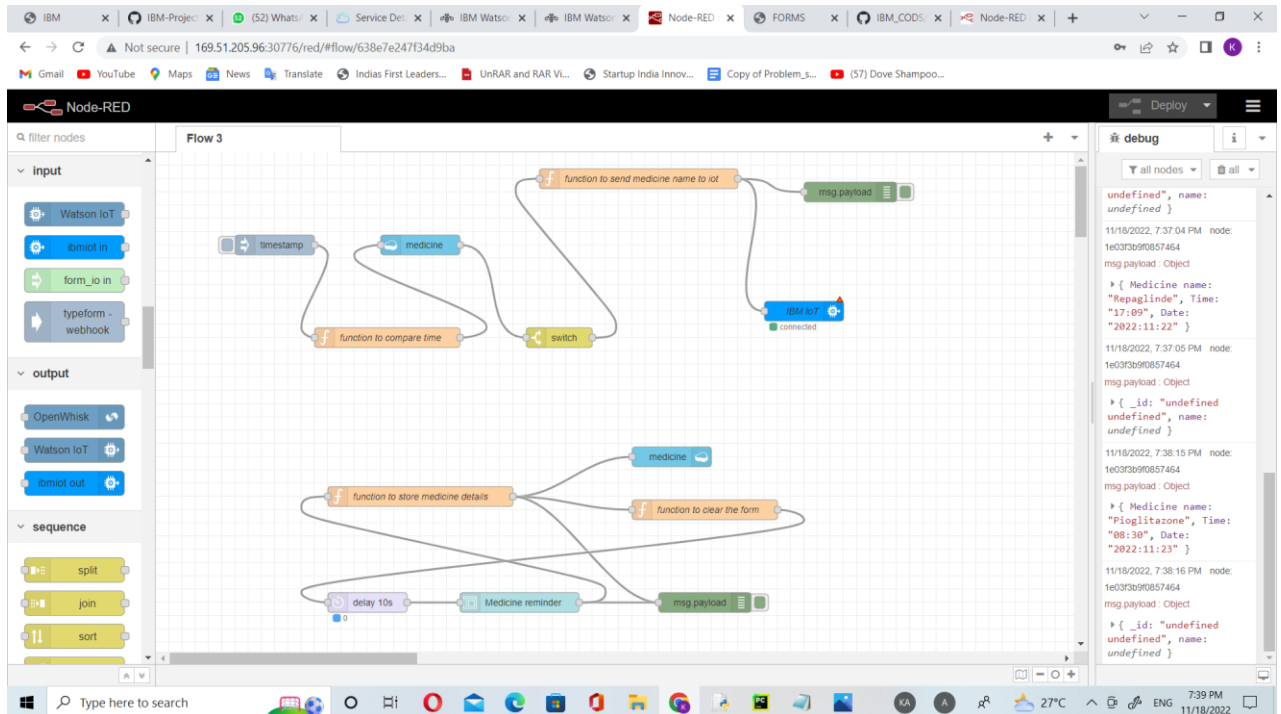
**PROJECT NAME: PERSONAL ASSISTANCE FOR SENIORS WHO
ARE SELF RELIANT**

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

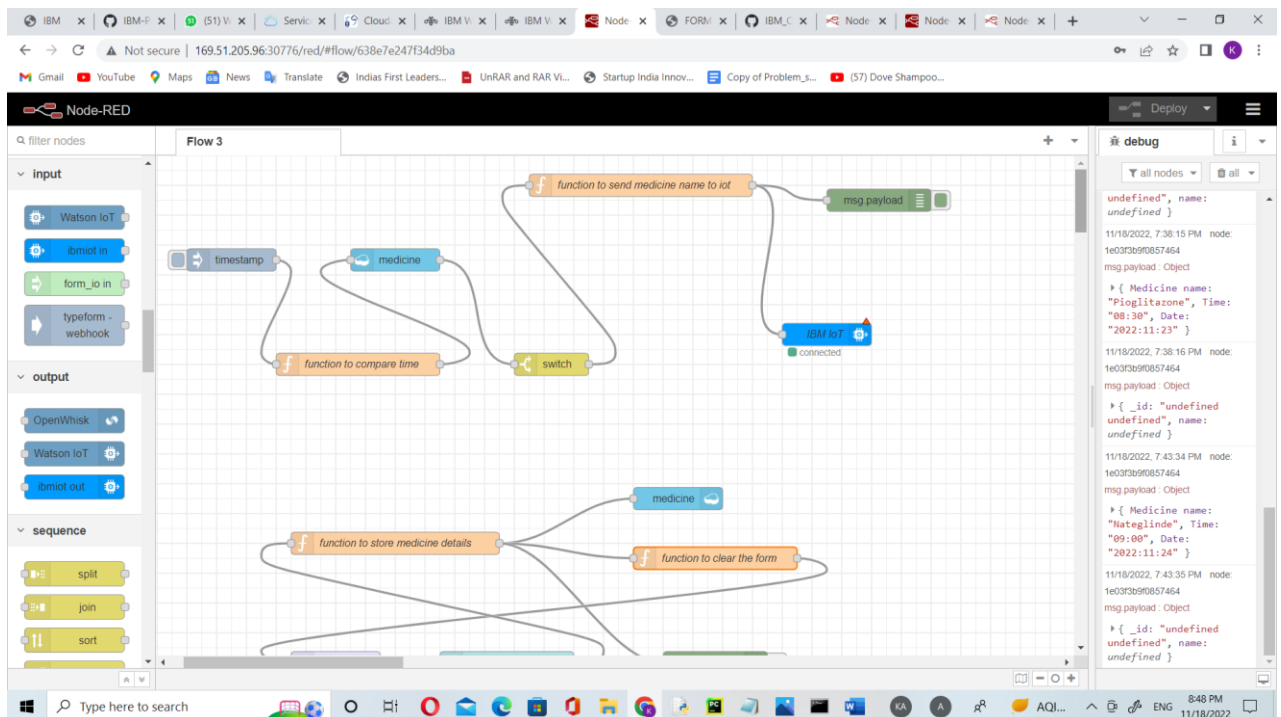
NODE-RED INSTALLATION:



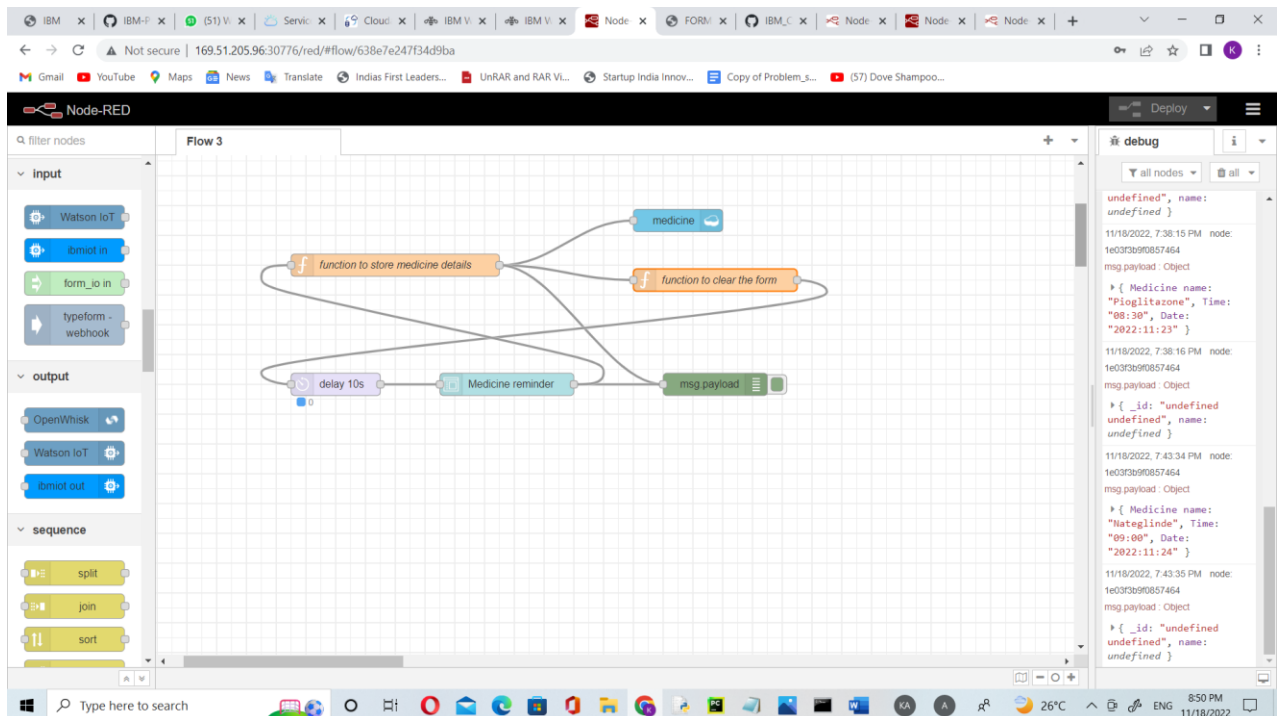
CREATING MEDICINE REMINDER FORM: (COMPLETE)



SENDING MEDICINE NAME AT APPROPRIATE TIME:

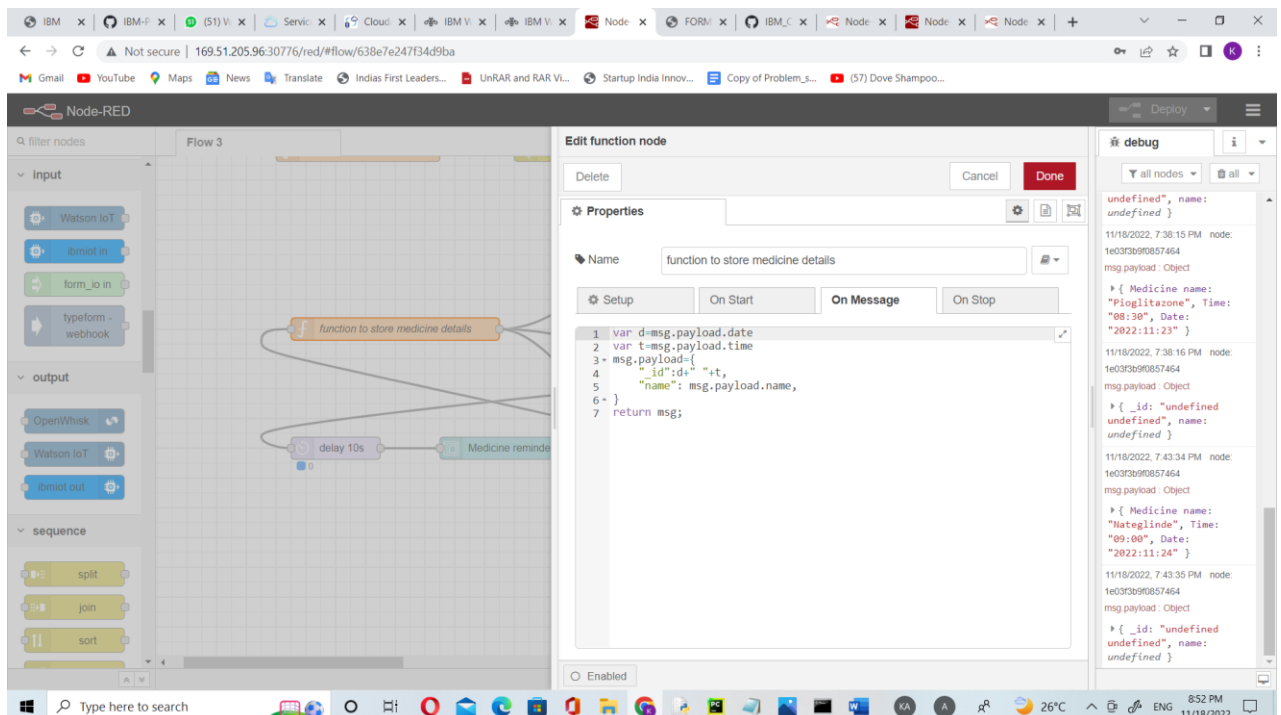


TO ENTER MEDICINE DETAILS:



FUNCTIONS:

1. FUNCTION TO STORE MEDICINE DETAILS



2. FUNCTION TO CLEAR THE FORM

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with a function node labeled "function to store medicine details" and a "delay 10s" node. The "Edit function node" panel is open, showing the name "function to clear the form" and the following JavaScript code:

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

The debug console on the right shows a series of messages, including an "undefined" message and a message with a payload object containing medicine details: { "Medicine name": "Pioglitazone", "Time": "08:30", "Date": "2022:11:23" }.

3.FUNCTION TO SEND MEDICINE NAME TO IOT

The screenshot shows the Node-RED web interface. The main workspace displays a flow with a "timestamp" node, a "medicine" node, a "function to compare time" node, and a "function to store medicine details" node. The "Edit function node" panel is open, showing the name "function to send medicine name to iot" and the following JavaScript code:

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

The debug console on the right shows a series of messages, including an "undefined" message and a message with a payload object containing medicine details: { "Medicine name": "Nateglinde", "Time": "09:00", "Date": "2022:11:24" }.

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 'medicine' node, and a 'function to compare time' 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 trigger 'On Message'. The function code is as follows:

```
1 var d= new Date();
2 var utc= d.getTime()+ (d.getTimezoneOffset() *60000);
3 var offset=-5.5;
4 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;
```

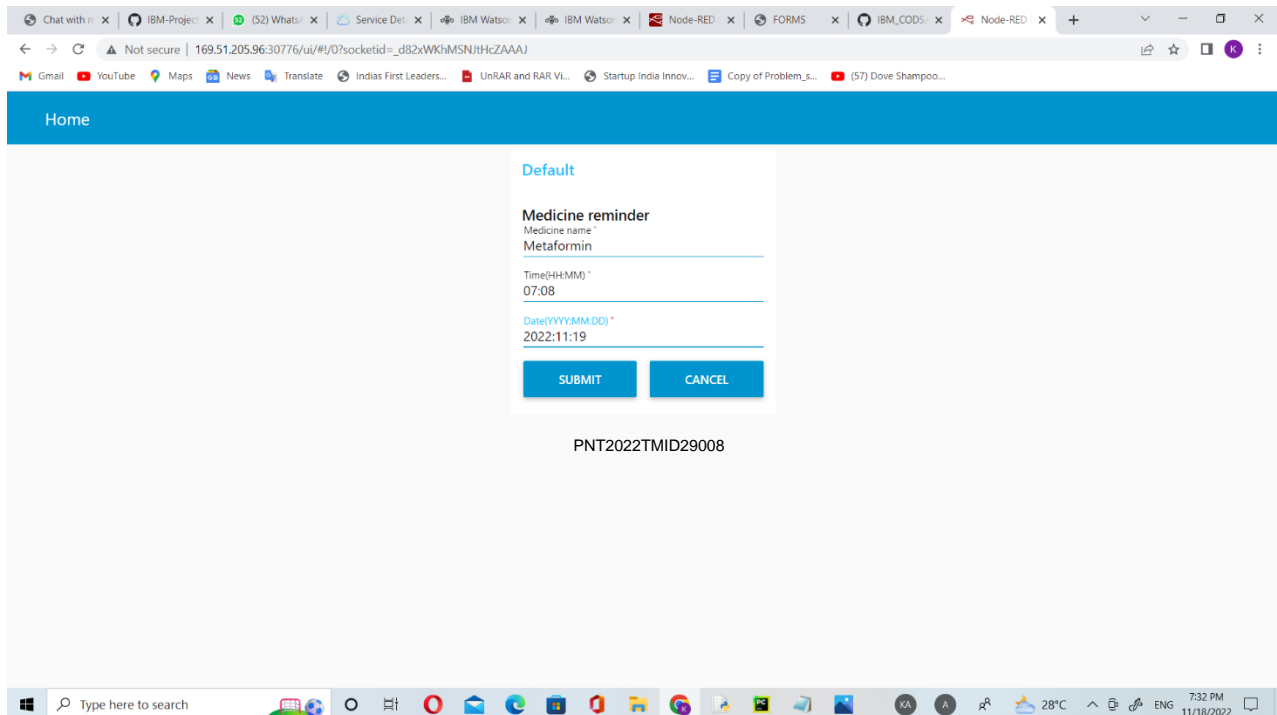
The debug console on the right shows the output of the function, which is a JSON object with 'name' and 'time' fields.

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 with the following fields:

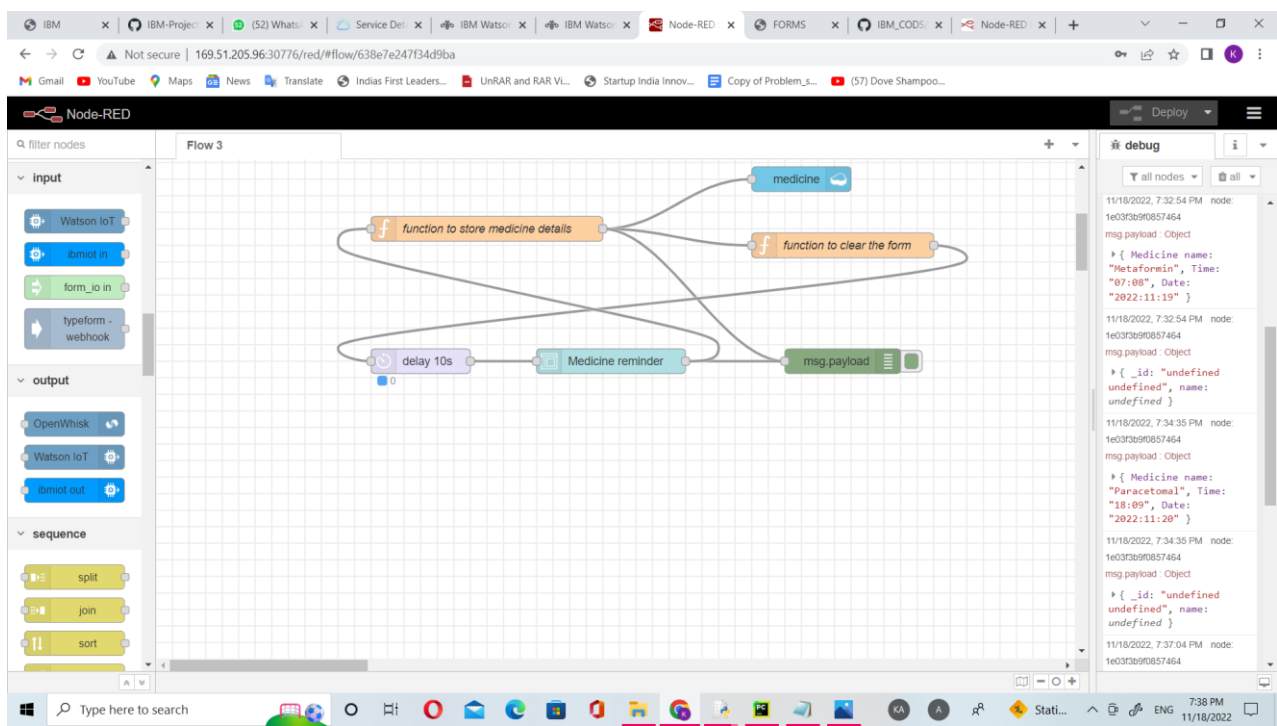
- Medicine name *
- Time(HH:MM) *
- Date(YYYY-MM-DD) *

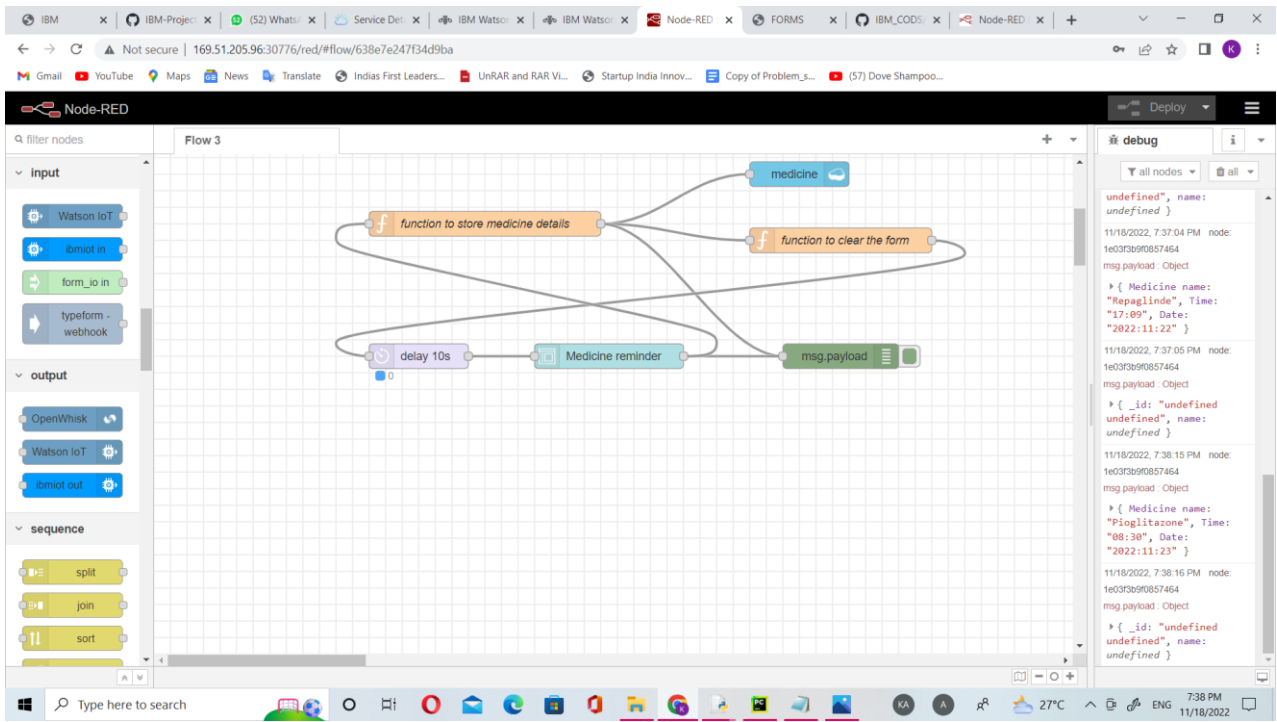
Below the form, there are two buttons: 'SUBMIT' and 'CANCEL'. The page also displays the text 'PNT2022TMID29008'.



DEBUG WINDOW:

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





MEDICINE DATABASE IN CLOUDANT DB:

The screenshot shows the Cloudant database interface in a browser. The left sidebar shows the 'medicine' database selected. The main area displays a table of documents. The table has columns for '_id' and 'name'. The documents are listed with their '_id' and 'name' values.

_id	name
Time:07:08 Date:2022:11:19	{ "name": "metaformin" }
Time:08:30 Date:2022:11:23	{ "name": "Pioglitazone" }
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": "paracetamol" }

Screenshot of the IBM Watson IoT Platform interface showing a database of medicine documents. The interface includes a sidebar with navigation options like All Documents, Query, Permissions, Changes, and Design Documents. The main area displays a list of documents, with the selected document's JSON content shown in a text editor.

Document ID: [Dropdown]

Options: [JSON] [Table] [Metadata]

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-c84f3838b765b1361da38a09d438d8d"
  },
  "doc": {
    "_id": "Time:07:08 Date:2022:11:19",
    "_rev": "2-c84f3838b765b1361da38a09d438d8d",
    "name": {
      "name": "metaformin"
    }
  }
}
```

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": "1-971feeb2e492ac767f8bca24c53fe9f2"
  },
  "doc": {
    "_id": "Time:08:30 Date:2022:11:23",
    "name": {
      "name": "metaformin"
    }
  }
}
```

Showing document 1 - 5. Documents per page: 20

IBM WATSON IOT PLATFORM:

Screenshot of the IBM Watson IoT Platform interface showing a list of devices. The interface includes a sidebar with navigation options like Browse, Action, Device Types, and Interfaces. The main area displays a table of devices, with the selected device's details shown in a modal window.

IBM Watson IoT Platform

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

Device ID: 12345

Status: Disconnected

Device Type: Iotsensor

Class ID: Device

Date Added: 17 Nov 2022 20:29

Descriptive Location: [Dropdown]

Identity: [Dropdown]

Device Information: [Dropdown]

Recent Events: [Dropdown]

State: [Dropdown]

Logs: [Dropdown]

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": "Nateglinde", "Time": "09:00", ...}	json	a few seconds ago
event_1	{"Medicine name": "Pioglitazone", "Time": "08:30", ...}	json	a minute ago
event_1	{"Medicine name": "Repaglinde", "Time": "17:09", ...}	json	2 minutes ago
event_1	{"Medicine name": "Paracetamol", "Time": "18:09", ...}	json	2 minutes ago
event_1	{"Medicine name": "Metaformin", "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. On the left, the 'network' tab is selected in the node palette. The main workspace displays a flow with the following nodes: a '[get] /signup' trigger node, a 'function 2' node, a 'delay 5s' node, a 'signup' node, 'function 3' node, and a 'debug 1' node. The flow is configured as follows: '[get] /signup' connects to 'function 2', which then connects to 'function 3'. 'function 3' connects to the 'signup' node. The 'signup' node connects to the 'debug 1' node. A 'delay 5s' node is also present but not connected to the main flow. The right-hand 'debug' console shows a list of messages. The first two messages are empty objects. The subsequent three messages are JSON objects containing the following data:

```
{ Username: "keerthana", Password: "12345" }
```

```
{ Username: "keerthana", Password: "12345", email: "keerthananandh4@gmail.com", age: "20" }
```

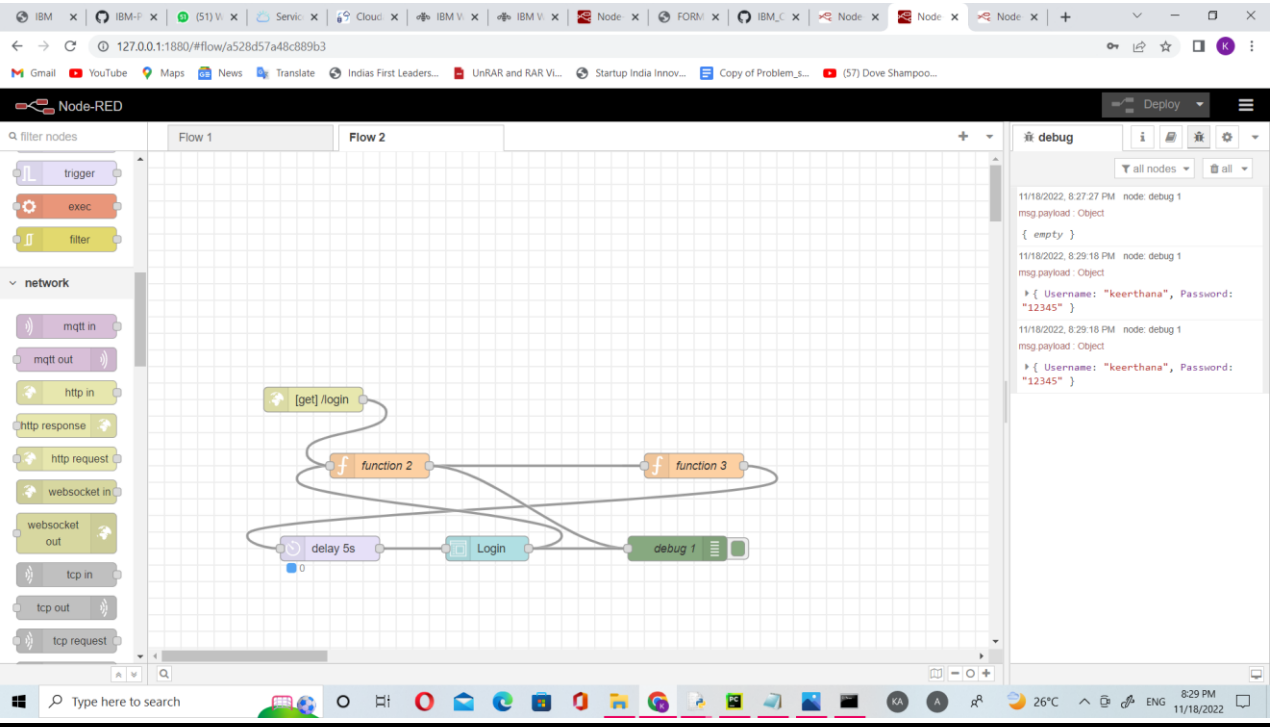
WEBUI(SIGNUP):

The screenshot shows a web browser displaying a 'Signup' form. The form is titled 'Default' and contains the following fields:

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

Below the form are two buttons: 'SUBMIT' and 'CANCEL'. At the bottom of the page, the text 'PNT2022TMID29008' is displayed. The browser's address bar shows the URL '127.0.0.1:1880/ui/#/0?socketid=5aWJRCVNg8uphtMhAAAA'.

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:** A text input field with the value "keerthana".
- Password:** A password input field with the value "12345".

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

Below the form, the text "PNT2022TMID29008" is displayed.

MEDICINE REMINDER PAGE:

<http://169.51.205.96:30776/ui/#!/0?socketid= d82xWKhMSNJtHcZAAAJ>

SIGNUP PAGE:

<http://127.0.0.1:1880/ui/#!/0?socketid=5aWJRQVNq8uphtMhAAAA>

