

## WRITE A FUNCTION TO COMPARE THE TIME

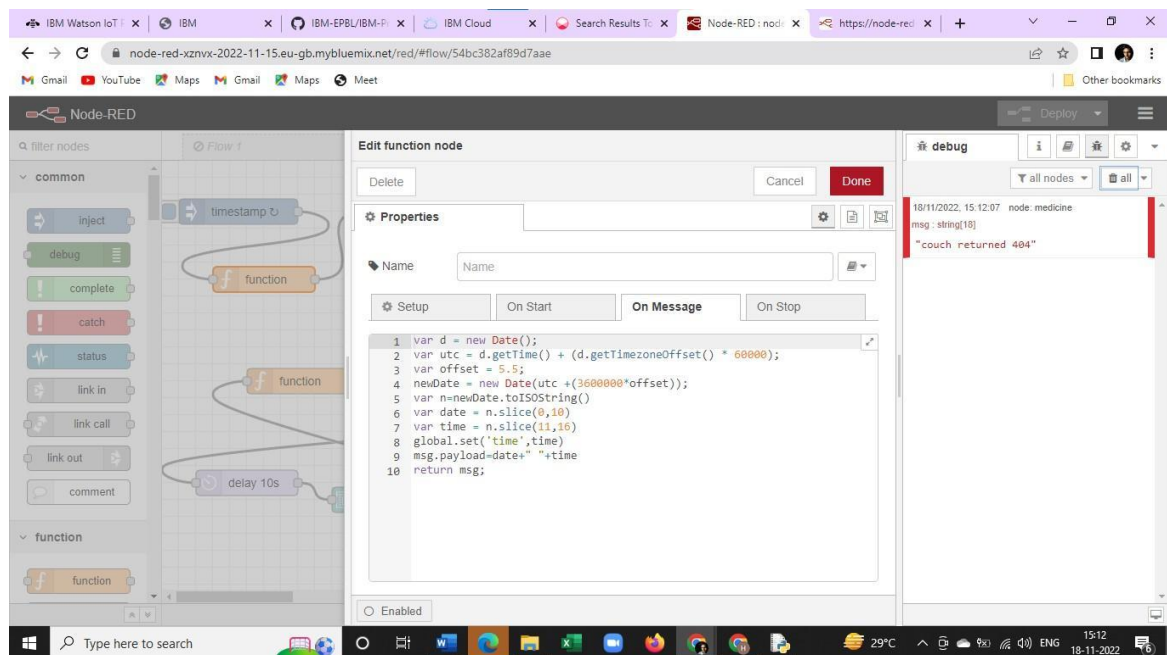
TEAM ID	PNT2022TMID26580
PROJECT NAME	Personal Assistance for Seniors Who Are Self-Reliant

### CODE :

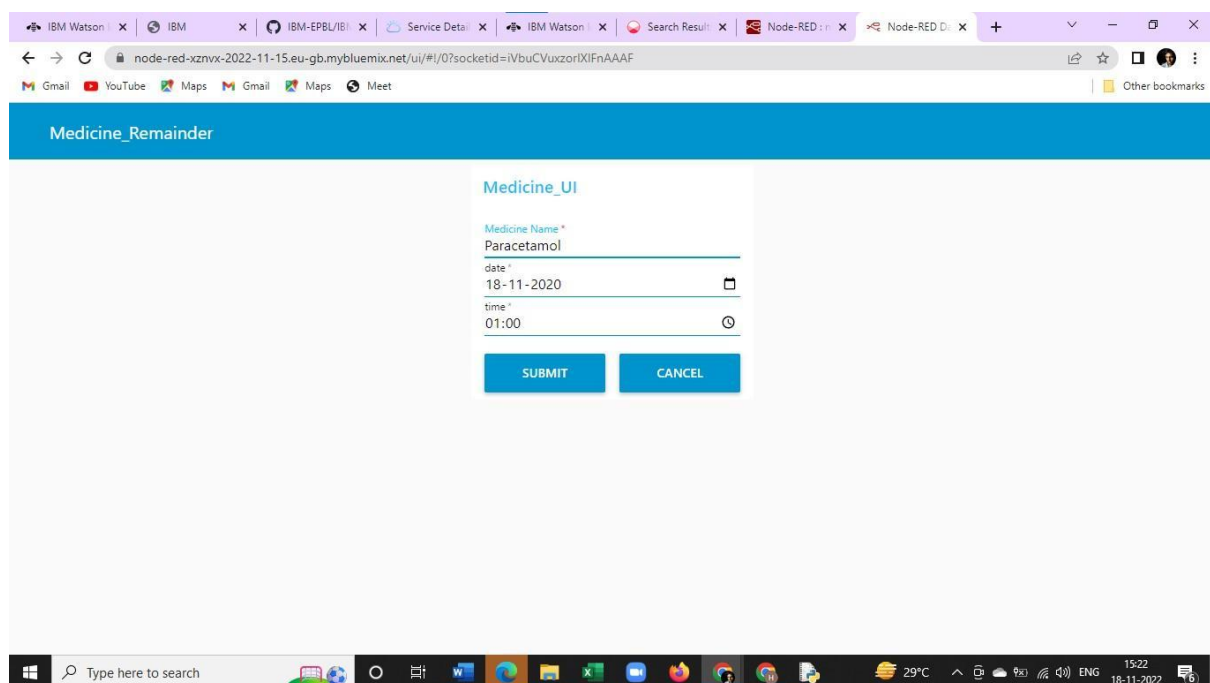
```
var d = new Date();
var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
var offset = 5.5;
newDate = new Date(utc +(3600000*offset));
var n=newDate.toISOString()
var date = n.slice(0,10)
var time = n.slice(11,16)
global.set('time',time)
msg.payload=date+" "+time
return msg;
```

## STEPS TO BE FOLLOWED :

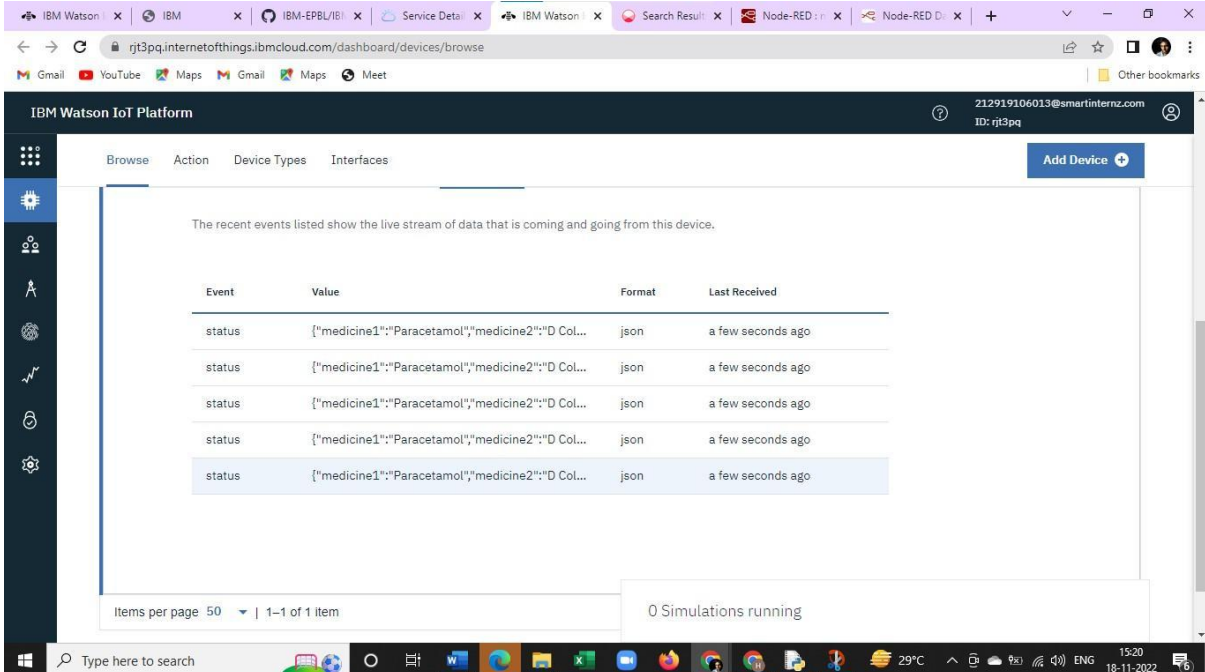
### STEP 1 : EDIT THE FUNCTION CODE IN THE FUNCTION NODE



### STEP 2 : AFTER CHANGING THE FUNCTION SUBMIT THE DETAILS IN WEB APPLICATION UI



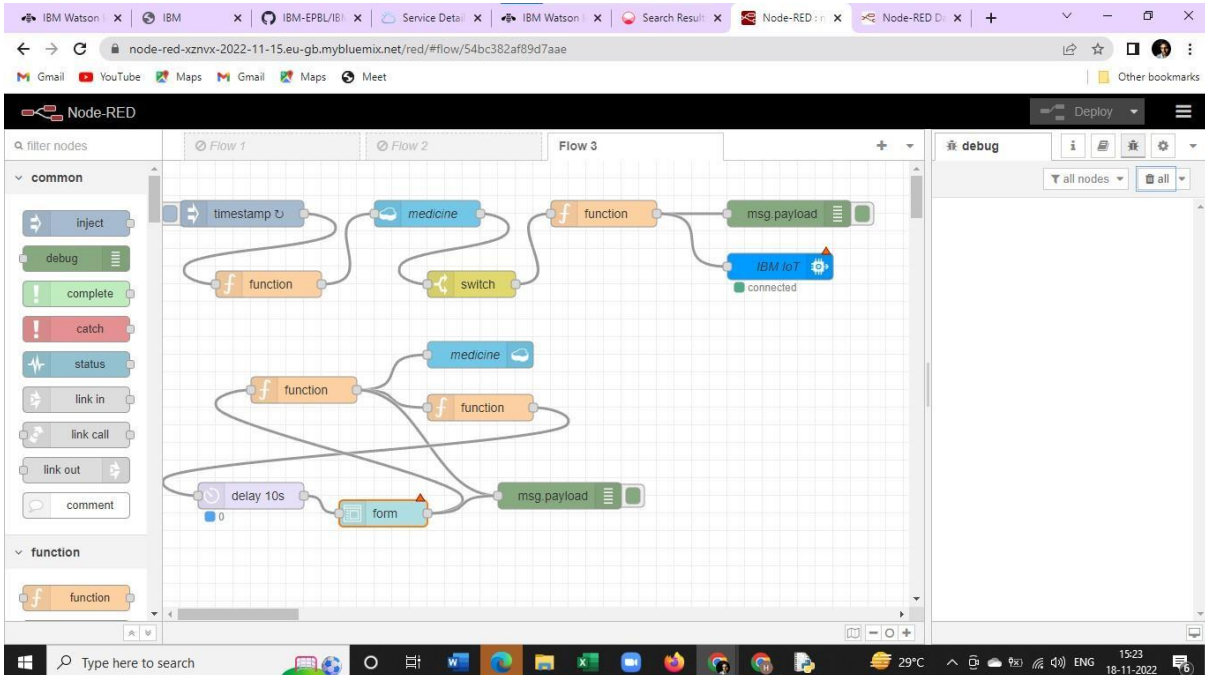
## STEP 3 : CHECK THE SUBMITTED DATA WAS STORED IN IOT PLATFORM



The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for IBM Watson, IBM, IBM-EPBL/IBI, Service Details, and Search Results. The main content area displays a table of recent events. The table has four columns: Event, Value, Format, and Last Received. The events are listed as 'status' with a JSON value containing medicine information, in 'json' format, received 'a few seconds ago'. The bottom status bar indicates '0 Simulations running'.

Event	Value	Format	Last Received
status	{"medicine1": "Paracetamol", "medicine2": "D Col..."}	json	a few seconds ago
status	{"medicine1": "Paracetamol", "medicine2": "D Col..."}	json	a few seconds ago
status	{"medicine1": "Paracetamol", "medicine2": "D Col..."}	json	a few seconds ago
status	{"medicine1": "Paracetamol", "medicine2": "D Col..."}	json	a few seconds ago
status	{"medicine1": "Paracetamol", "medicine2": "D Col..."}	json	a few seconds ago

## STEP 4 : NOW WE HAVE READY TO TAKE THE OUTPUT IN THE GIVEN NODE



The screenshot shows the Node-RED interface. The left sidebar contains a palette of nodes categorized into 'common' and 'function'. The main workspace displays a flow diagram with several nodes connected. The flow starts with a 'timestamp' node, followed by a 'function' node, then a 'medicine' node, a 'switch' node, and another 'function' node. The flow then splits into two paths: one leading to a 'msg.payload' node and another leading to a 'form' node. The 'form' node is connected to a 'delay 10s' node, which then connects to another 'msg.payload' node. The right sidebar shows a 'debug' console.