

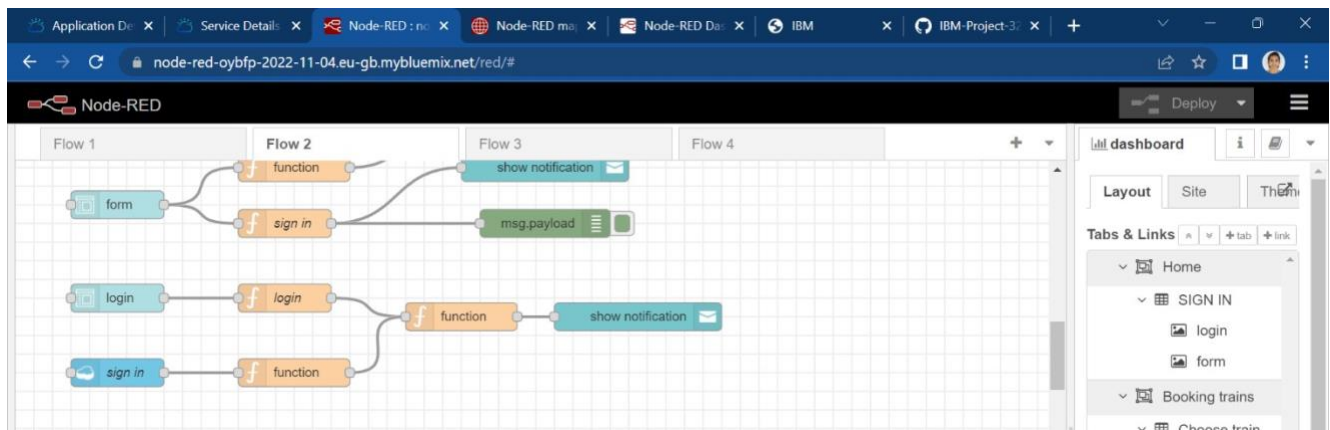
SPRINT-1

Date	19 NOVEMBER 2022
Team ID	PNT2022TMID47138
Project Name	Project – SMART SOLUTIONS FOR RAILWAYS

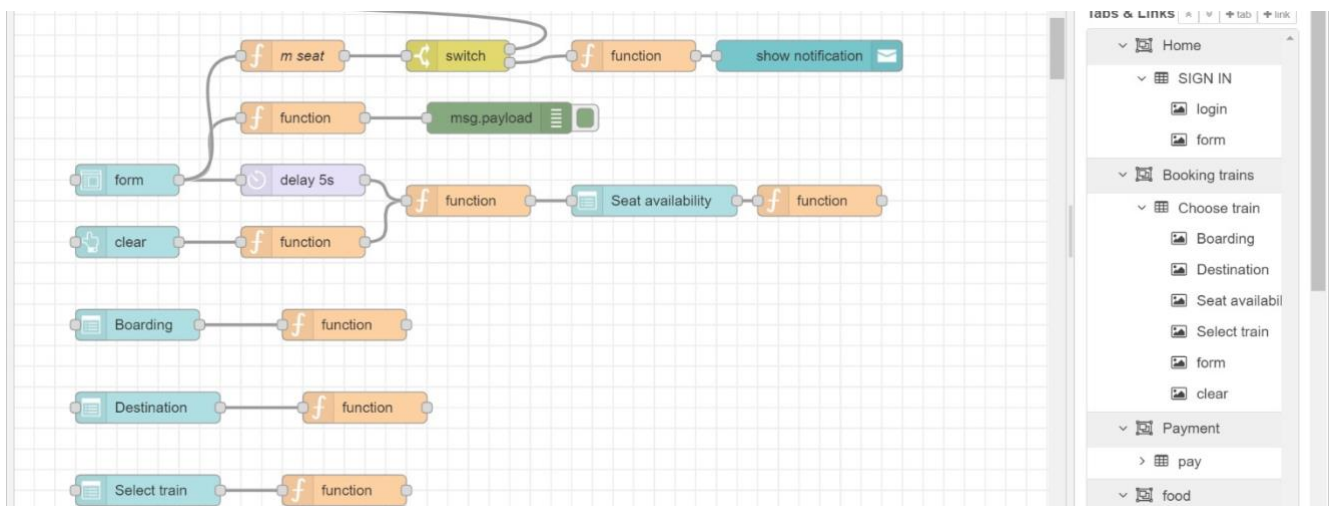
PROCEDURE:

- Step1: Develop node red application
- Step2: Install the required nodes from manage palette option
- Step3: Connect the node flow
- Step4: Deploy the flow

WEB APPLICATION :



NODES TO BOOK TRAIN:



FUNCTION NODE COMMAND TO INDICATE THE AVAILABLE SEATS:

```
var a=global.get('a') var
s= []
for(let i=0;i<a.length==0;i++){
s.push(a[i]) }
if(s.length==0){
    msg.options=[{"No seats available":0}]
}
else{
msg.options= s
}
msg.payload= s
return msg;
```

FUNCTION NODE COMMAND TO CHOOSE THE AVAILABLE SEATS:

```
var s=global.get('s') var
a=global.get('a') function
reg(x){  for(let
i=0;i<a.length;i++){
if(a[i]==x){
    a.splice(i,1)
    }
}
}
if(s==1){
global.set('s1',s)
reg(s)
}
else if(s==2){
global.set('s2',s)
reg(s)
}
else if(s==3){
global.set('s3',s)
reg(s)
}
else if(s==4){
global.set('s4',s)
reg(s)
}
else if(s==4){
global.set('s4',s)
reg(s)
}
return msg;
```

FUNCTION NODE COMMAND TO STORE DATA IN DATABASE:

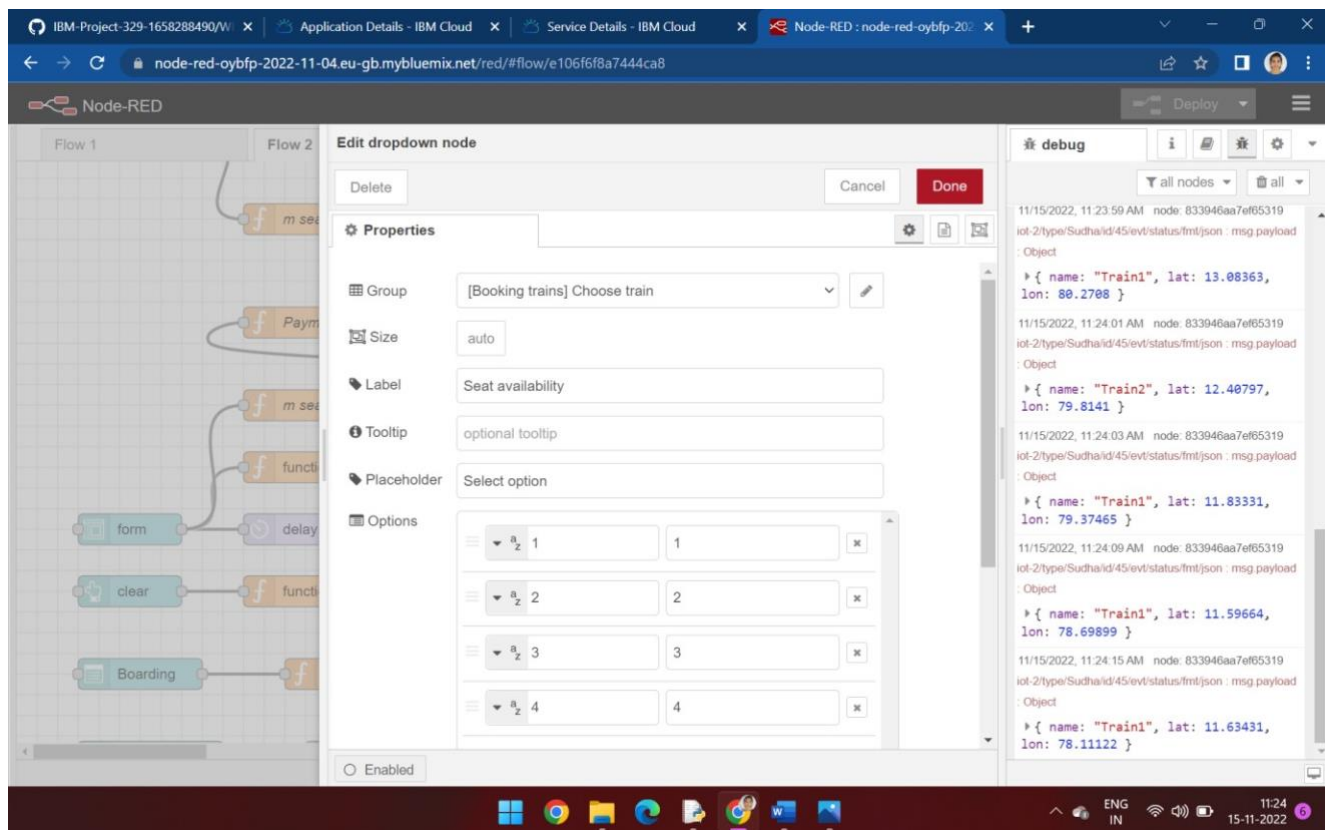
```
var m=global.get('m')
var d=new Date();
var utc=d.getTime()+ (d.getTimezoneOffset()*60000); var
offset=5.5;
newDate=new Date(utc+(3600000*offset));
var n=newDate.toISOString() var
date=n.slice(0,10) var time=n.slice(11,19)
var d1=date+', '+time msg.payload={
  "_id":d1,
  "Name":m.Name,
  "Age":m.Age,
  "Mobile":m.Num,
  "boarding":global.get('b'),
  "destination":global.get('d'),
  "Seat":global.get('s')
}
return msg;
```

FORM DETAILS:

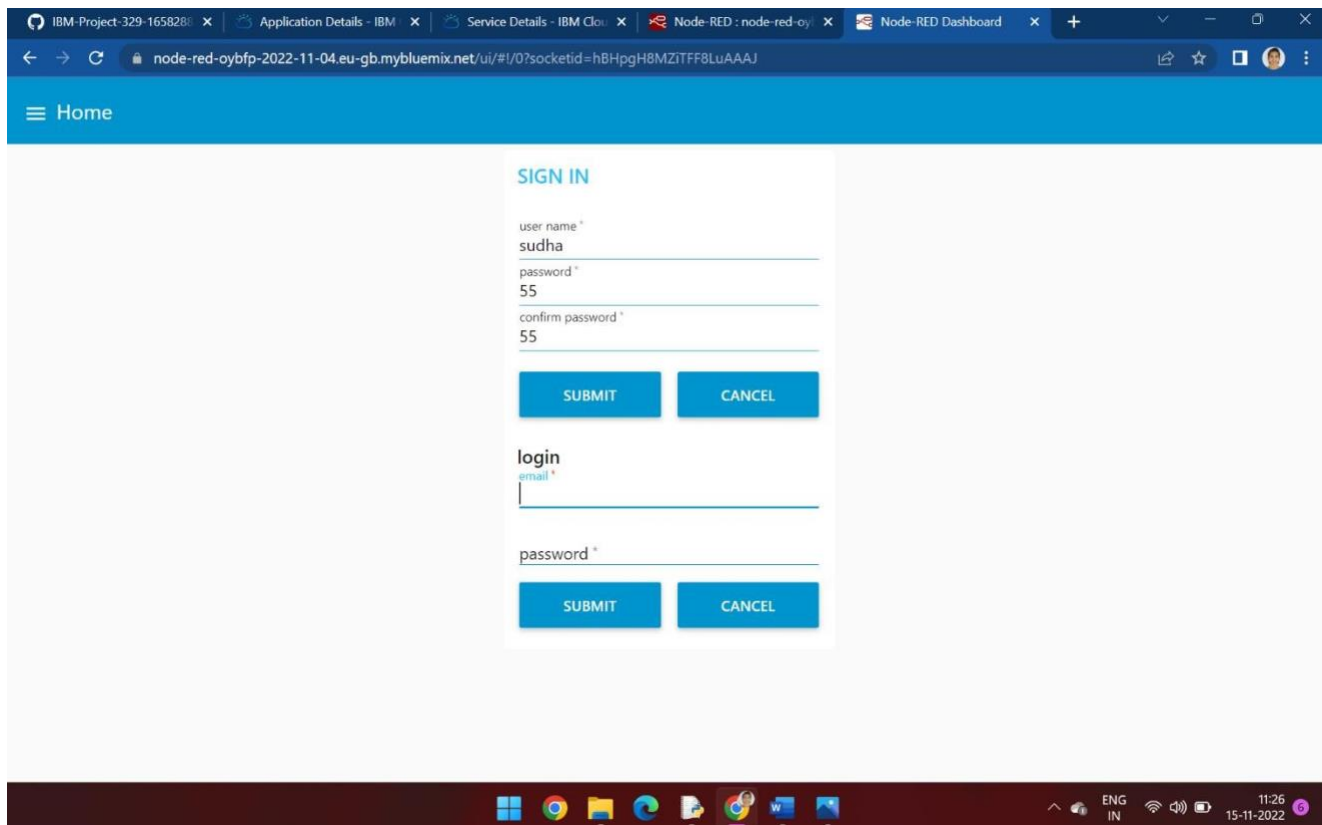
The screenshot displays the Node-RED web interface. The main panel shows the 'Edit form node' configuration for a form titled '[Booking trains] Choose train'. The form has three visible elements: 'Name' (Text), 'Age' (Number), and 'Mobile num' (Number). The 'Required' column for all three is checked. The 'UiRows' column shows '1' for Name, '1' for Age, and '1' for Mobile num. The 'Remove' column has trash icons for each element. The 'Properties' panel on the left shows the form's group, size, and label. The 'debug' console on the right shows a sequence of messages from the 'node: 833946aa7ef65319' node, each containing a JSON payload with 'name', 'lat', and 'lon' fields. The messages are as follows:

- 11/15/2022, 11:23:25 AM: `{ name: "Train1", lat: 13.08363, lon: 80.2708 }`
- 11/15/2022, 11:23:27 AM: `{ name: "Train2", lat: 12.40797, lon: 79.8141 }`
- 11/15/2022, 11:23:29 AM: `{ name: "Train1", lat: 11.83331, lon: 79.37465 }`
- 11/15/2022, 11:23:35 AM: `{ name: "Train1", lat: 11.59664, lon: 78.69899 }`
- 11/15/2022, 11:23:41 AM: `{ name: "Train1", lat: 11.63431, lon: 78.11122 }`

SEAT DROPDOWN BOX:



WEB UI OUTPUT:



WEB UI FOR TRAIN BOOKING:

IBM-Project-329-165826Application Details - IBMService Details - IBM CloNode-RED : node-red-oyNode-RED Dashboard

node-red-oybfp-2022-11-04.eu-gb.mybluemix.net/ui/#!/?socketid=h8HpgH8MZITFF8LuAAAJ

node-red-oybfp-2022-11-04.eu-gb.mybluemix.net/ui/#!/?socketid=h8HpgH8MZITFF8LuAAAJ

Booking trains

Choose train

BoardingCoimbatore

DestinationChennai

Seat availabilitySelect option

Select trainBlue mountain

Name *
sudha

Age *
20

Mobile num *
9876543210

SUBMIT

CANCEL

CLEAR

Windows Taskbar

11:27 15-11-2022