

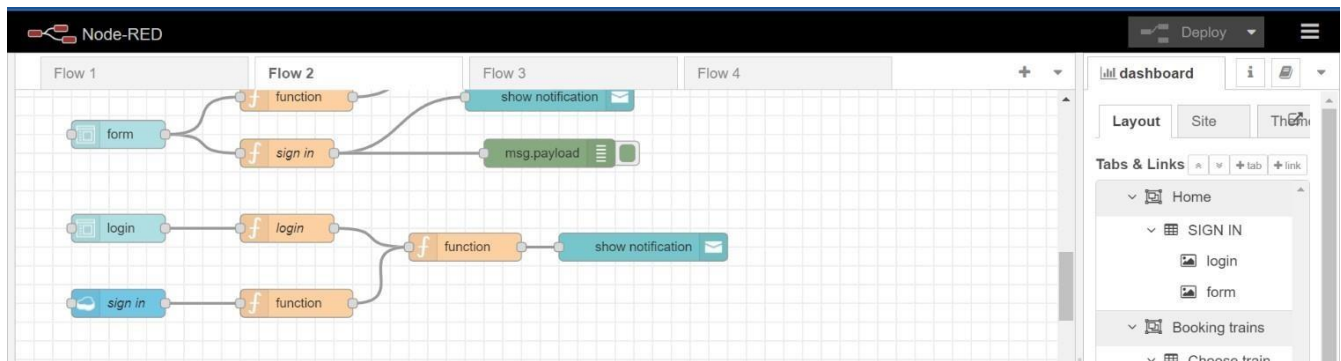
SPRINT-1

Date	19 NOVEMBER 2022
Project Name	SMART SOLUTIONS FOR RAILWAYS

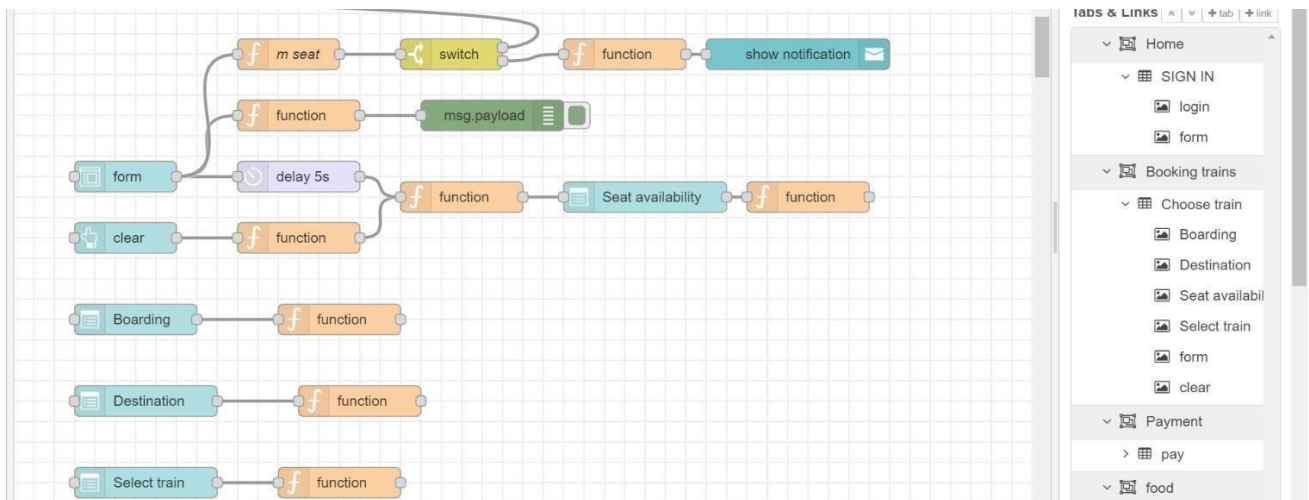
PROCEDURE:

- Srep1: 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. On the left, the 'Edit form node' panel is open, showing the configuration for a form node. The 'Properties' section includes a 'Group' dropdown set to '[Booking trains] Choose train', a 'Size' dropdown set to 'auto', and a 'Label' text field containing 'optional label'. Below these, the 'Form elements' section contains a table with three rows of form elements:

Label	Name	Type	Required	UiRows	Remove
Name	name	Text	<input checked="" type="checkbox"/>		
Age	age	Number	<input checked="" type="checkbox"/>		
Mobile num	num	Number	<input checked="" type="checkbox"/>		

At the bottom of the form node configuration, there is an 'Enabled' checkbox which is currently checked.

On the right side of the interface, the 'debug' console is open, showing a log of messages. The messages are JSON objects containing train data, such as:

```
{ name: "Train1", lat: 13.08363, lon: 80.2708 }
```

```
{ name: "Train2", lat: 12.40797, lon: 79.8141 }
```

```
{ name: "Train1", lat: 11.83331, lon: 79.37465 }
```

```
{ name: "Train1", lat: 11.59664, lon: 78.69899 }
```

```
{ name: "Train1", lat: 11.63431, lon: 78.11122 }
```

SEAT DROPDOWN BOX:

The screenshot shows the Node-RED web interface. On the left, a flow is visible with nodes like 'form', 'clear', and 'Boarding'. The main panel is titled 'Edit dropdown node'. It contains the following configuration:

- Group:** [Booking trains] Choose train
- Size:** auto
- Label:** Seat availability
- Tooltip:** optional tooltip
- Placeholder:** Select option
- Options:** A list of four options, each with a dropdown arrow, a label, and a value:
 - ▼ a_z 1 | 1
 - ▼ a_z 2 | 2
 - ▼ a_z 3 | 3
 - ▼ a_z 4 | 4

At the bottom left of the configuration panel is a checkbox labeled 'Enabled'. On the right, the 'debug' console shows a series of JSON messages from an IoT device, including train names and coordinates.

WEB UI OUTPUT:

[Home](#)

SIGN IN

user name *

sudha

password *

55

confirm password *

55

SUBMIT

CANCEL

login

email *

password *

SUBMIT

CANCEL

WEB UI FOR TRAIN BOOKING:

≡ Booking trains

Choose train

Boarding

Coimbatore

Destination

Chennai

Seat availability

Select option

Select train

Blue mountain

Name *

sudha

Age *

20

Mobile num *

9876543210

SUBMIT

CANCEL

CLEAR