

Project Development Phase

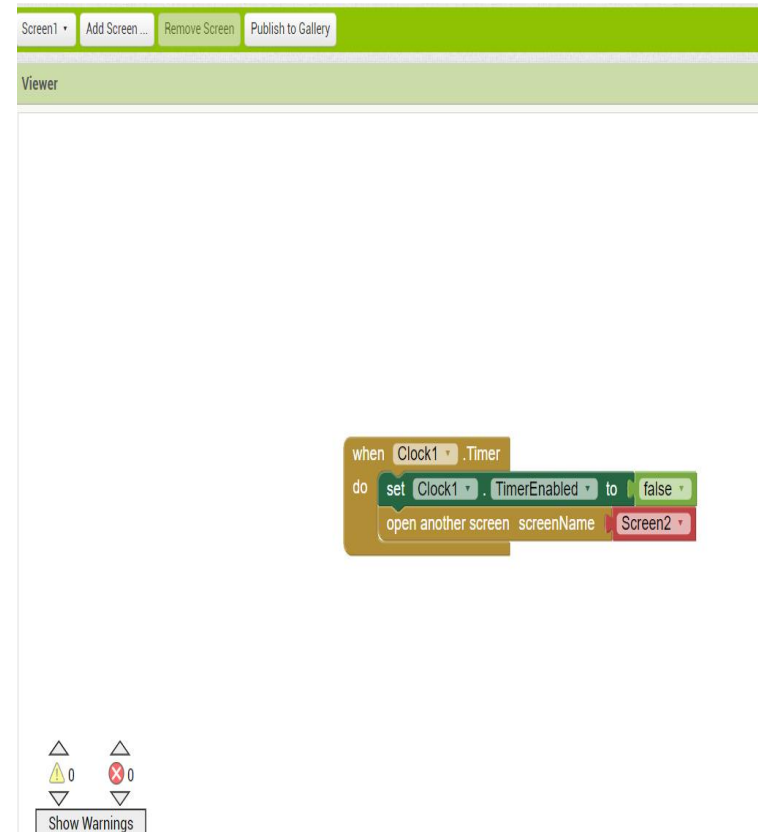
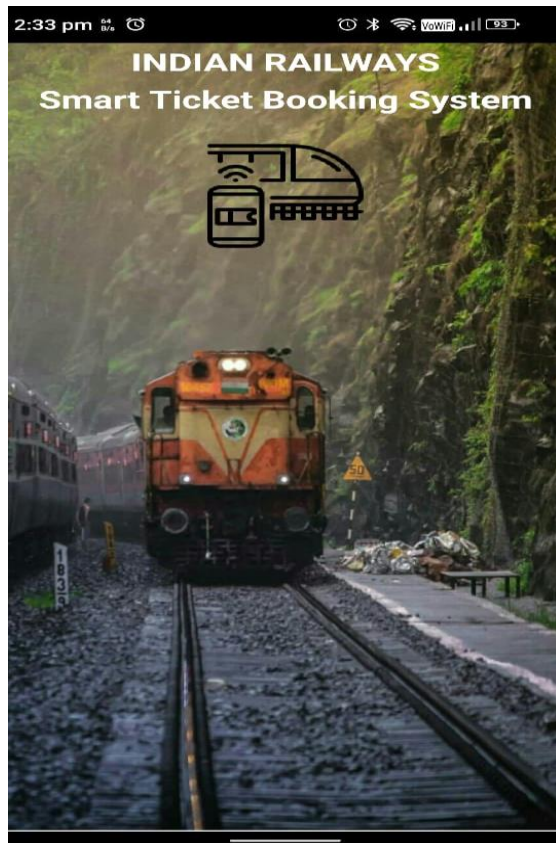
Date	18 October 2022
Team ID	PNT2022TMID35911
Project Name	Smart solution for Railways

Sprint Traget :

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priorit y	Team Members
Sprint-1	Registration	-USN-1	As a user, I can sign up using my email, password	2	High	Sanjay krishnan Abishek
Sprint-1		-USN-2	As a user, I will receive confirmation message once I have registered.	1	High	Sanjay krishnan Abishek
Sprint-1	Login	-USN-3	As a user, I can log into the application by entering email & password	1	High	Sanjay krishnan Abishek

INTRODUCTION :

- In this sprint we had completed an android application using MIT app inventor, which can be used for signing up and login to use the service.

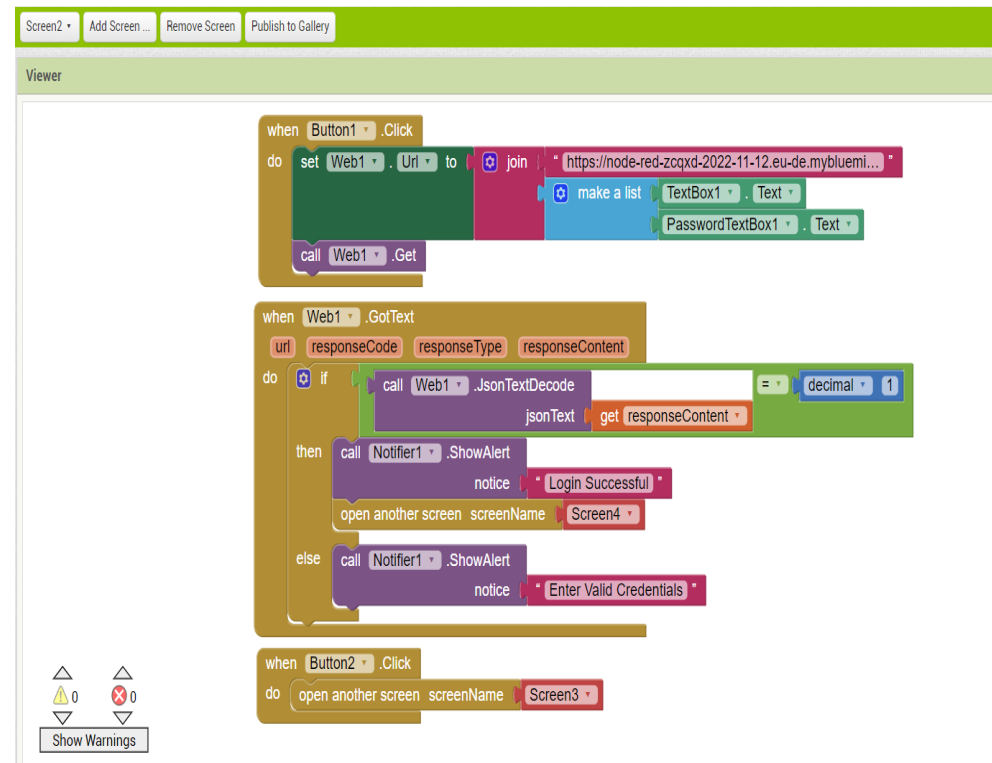
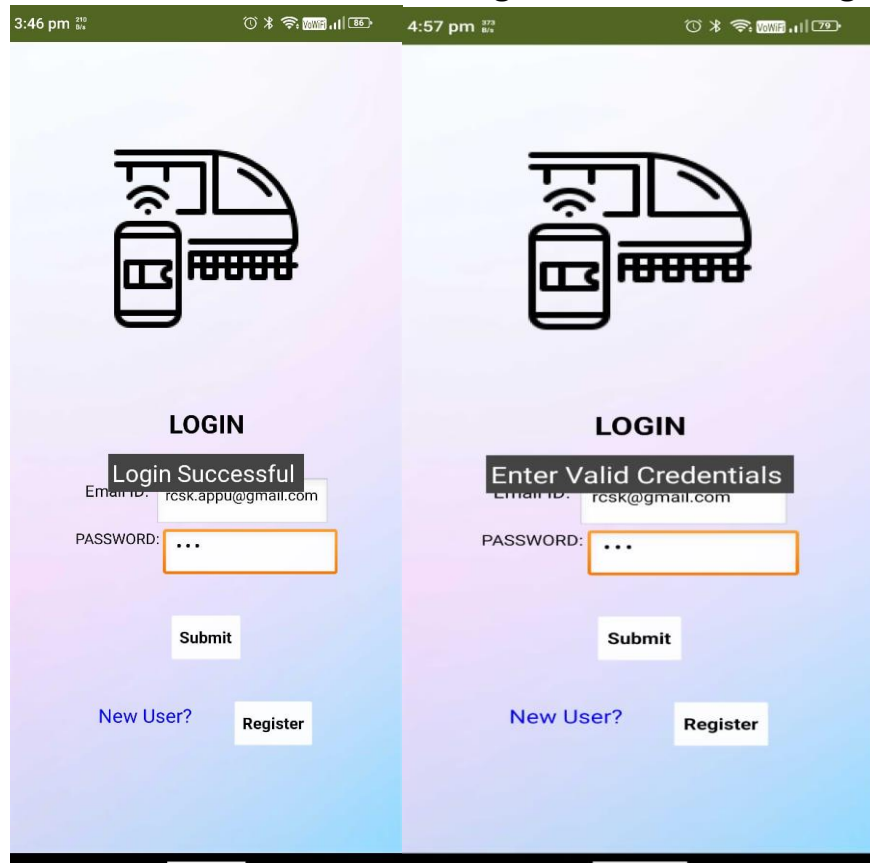


LOGIN:

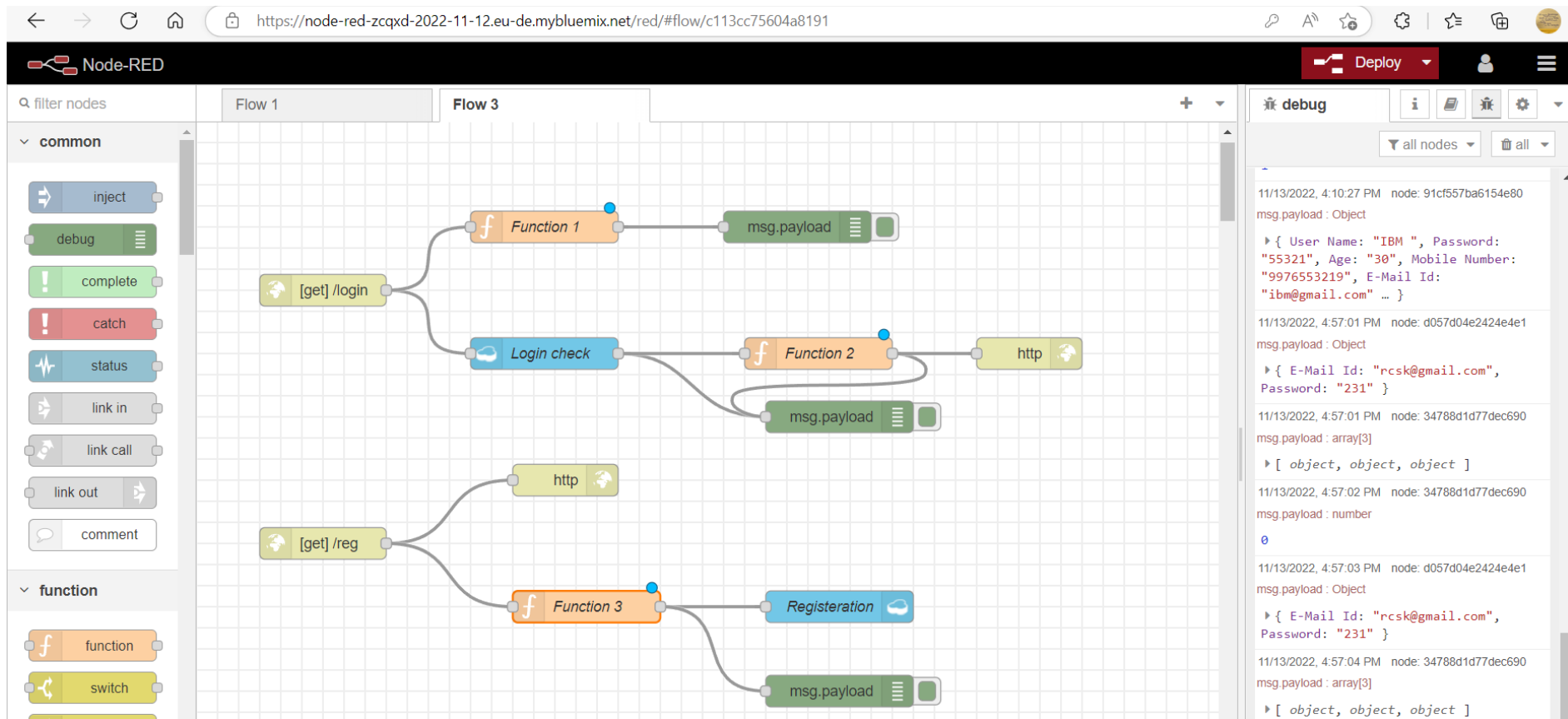
Registered users can enter their email address and password to login. For new user, click Register button for Registration.

Once they hit Submit, The credentials will be checked in Cloudant DB via Node red .

If credentials are available, Login Successful message will be displayed and open Booking page.



NODE RED FLOW:



- From MIT app, we get registration credential using “get” block.
- Pass the information through a function where list is converted to proper Json format and load in Cloudant DB.
- For Login, the login credential is checked in DB and if credential are available, “success” message is displayed or “Enter valid credential” is displayed.

Code inside Functions:

LOGIN:

Function 1:

```
Edit function node

Delete Cancel Done

Properties

Name Function 1

Setup On Start On Message On Stop

1 a=msg.payload.login
2 s=a.replace("[",',') // replacing list box [] with the empty string
3 s=s.replace("]",',')
4 s=s.split(',') // splitting each word string with ','
5 count=0 // to count the length of particular word to
6 c1=0 // seaprte them from ""
7- for(i in s[0]){ //calculating length of first word
8-   count=count+1}
9- for (j in s[0]){ //taking only content iside string excluding ""
10-   email=s[0].slice(1,count-1)}
11- for(i in s[1]){ // the same process is done for all input credential
12-   c1=c1+1}
13- for (var j in s[1]){
14-   pass=s[1].slice(2,c1-1)} // from 2 because to eliminate , and ""
15- msg.payload={
16-   "E-Mail Id":email,
17-   "Password":pass // convert to json format ,store in msg.payload
18- }
19 global.set('e',email) //the above things are used for checking in DB
20 //TAKING e and p as global to use in rest functions
21 global.set('p',pass)
22 global.set('m',msg.payload)
23 //the same procedure is done in Function 2 and Function 3
24 return msg;
```

Function 2:

```
Edit function node

Delete Cancel Done

Properties

Name Function 2

Setup On Start On Message On Stop

1 var a=msg.payload
2 var e=global.get('e') // getting global e
3 var p=global.get('p')
4 count=0
5- for(j in a){
6-   count=count+1}
7 flag=0
8 flag1=0
9 // checking emai and pass in DB and returning as flag,
10- for (i=0;i<count;i++){
11-   if((e==a[i]["E-Mail Id"])&(p==a[i]["Password"]))
12-   {
13-     flag=1//if flag =1, credential availblle in DB ,send to App.
14-   }
15-   else{
16-     flag1=1
17-   }
18- }
19- if(flag==1){
20-   msg.payload=1
21- }
22- else if(flag==0){
23-   msg.payload=0
24- }
25 return msg;
```

REGISTRATION: Function 3:

Edit function node

Delete Cancel Done

Properties

Name Function 3

Setup On Start On Message On Stop

```
1 a=msg.payload.reg //same procedure as Function 1
2 s=a.replace("[",',')
3 s=s.replace("]",',')
4 s=s.split(',')
5 count=0
6 c1=0
7 c2=0
8 c3=0
9 c4=0
10 c5=0
11 for(i in s[0]){
12     count=count+1
13 }
14 for (j in s[0]){
15     name=s[0].slice(1,count-1)
16 }
17 for(i in s[1]){
18     c1=c1+1
19 }
20 for (var j in s[1]){
21     pass=s[1].slice(2,c1-1)
22 }
23 for(i in s[2]){
24     c2=c2+1
25 }
26 for (var j in s[2]){
27     age=s[2].slice(2,c2-1)
```

Edit function node

Delete Cancel Done

Properties

Name Function 3

Setup On Start On Message On Stop

```
24 }
25 for (var j in s[2]){
26     age=s[2].slice(2,c2-1)
27 }
28 for(i in s[3]){
29     c3=c3+1
30 }
31 for (var j in s[3]){
32     mno=s[3].slice(2,c3-1)}
33 for(i in s[4]){
34     c4=c4+1
35 }
36 for (var j in s[4]){
37     email=s[4].slice(2,c4-1)}
38 for(i in s[5]){
39     c5=c5+1
40 }
41 for (var j in s[5]){
42     adrno=s[5].slice(2,c5-1)
43 }
44 msg.payload={
45     "User Name": name,
46     "Password": pass,
47     "Age": age,
48     "Mobile Number": mno,
49     "E-Mail Id": email,
50     "Aadhar Number":adrno,
51 }
52 return msg;
```

REGISTRATION:

Enter all the credentials and hit submit. The credentials will be stored in Cloudant DB and login page will appear.

The image displays a mobile application interface for user registration and its underlying Node-RED logic.

Registration Form (Left):

- Header:** An icon of a smartphone with a Wi-Fi signal and a bus.
- Title:** REGISTRATION
- Fields:**
 - User Name: Sanjay
 - Password: ...
 - Age: 20
 - Mobile Number: 9975431190
 - E-Mail ID: rcsk.appu@gmail.com
 - Aadhaar Number: 211670679781
- Action:** Submit button

Node-RED Flow (Right):

- Trigger:** when Button1 .Click
- Do Block:**
 - call Notifier1 .ShowAlert** with notice "Success"
 - set Web1 .Url to** join "https://node-red-zcqx-d-2022-11-12.eu-de.mybluemix..."
 - make a list** containing: TextBox2 .Text, PasswordTextBox1 .Text, TextBox3 .Text, TextBox4 .Text, TextBox5 .Text, and TextBox6 .Text
 - call Web1 .Get**
 - open another screen** screenName Screen2

Cloudant DB, stored values after registering:

← → ↻ 🏠 https://2cc2642a-2e7c-4dbe-9c71-e1bb7d555a76-bluemix.cloudant.com/dashboard.html#database/new_reg/_all_docs 🔊 ⭐ ⚙️ ⌘ 📖 🔔

↔️

new_reg

⋮

Document ID

⚙️ Options

{ } JSON

📖

🔔

All Documents +

Query

Permissions

Changes

Design Documents +

☐

Table Metadata { } JSON

Create Document

	_id	_rev	User Name	Password	Age	Mobile Number	E-Mail Id	Aadhar Number
<input type="checkbox"/>	2203e2fbb6c...	1-7935573d...	vikranth	8yt28o3	33	7384527289	foronline@go...	6283639254...
<input type="checkbox"/>	2886cb6d01...	1-a7f1a4ffba...	raman	3yurvdjidug3...	45	8363845284	raman56@go...	6154725492...
<input type="checkbox"/>	31fcf24cbfd6...	1-b114dc2f5...	agilan	6789	32	7896543210	agee@gmail....	2534619845...
<input type="checkbox"/>	51de4c65c9c...	1-80b396e4...	IBM	55321	30	9976553219	ibm@gmail.c...	2145789634...
<input type="checkbox"/>	6d8b341f5f1...	1-14bca82cd...	Sanjay	123	20	9975431190	rcsk.appu@g...	2116706797...
<input type="checkbox"/>	774a67508b...	1-7bf7e06a8...	singh	4567	60	6783564709	fireghost@joy...	6124537284...
<input type="checkbox"/>	884d49a149...	1-193e92ac1...	INDIAN	God	15	7658493240	abishek@gm...	2345679954...
<input type="checkbox"/>	90d1c88f8b8...	1-72754ca50...	shyam	928364vxj	22	7351836490	iamfire@slot....	6154826490...
<input type="checkbox"/>	90d1c88f8b8...	1-1c8cc13dc...	krish	4fyv4o	27	9835382862	krishhsirk@g...	6145395759...
<input type="checkbox"/>	90d1c88f8b8...	1-eaac19070...	amit	e8urv	23	7364924392	rockfan@gom...	6134749362...
<input type="checkbox"/>	939e9a19bf8...	1-0d41d08b...	leon	398y5hfjfy	56	7036488538	leeat445@go...	6145482538...
<input type="checkbox"/>	a5b729f96ca...	1-4b81ac023...	ram	1234	34	8654239172	migisoh347@...	2876590138...
<input type="checkbox"/>	a5b729f96ca...	1-bdecdbdd5...	sam	0987	44	8936294990	giraffeinpark...	6381538274...

Showing 8 columns. ☒ Show all columns.

Showing document 1 - 17. Documents per page: 20 < >

Log Out

After LOGIN, Booking page will appear after successful Login.

