IBM SMART INTERNZ – PROJECT BASED LEARNING 2022

SMART SOLUTION FOR RAILWAYS

TEAM ID : PNT2022TMID14902

TEAM MEMBERS: YALINI P, NIVETHA K, JANARANJANI S, VIVEKA R

PROJECT OBJECTIVE:

Smart Solutions for railways is designed to reduced the work load of the user and also the use of paper.

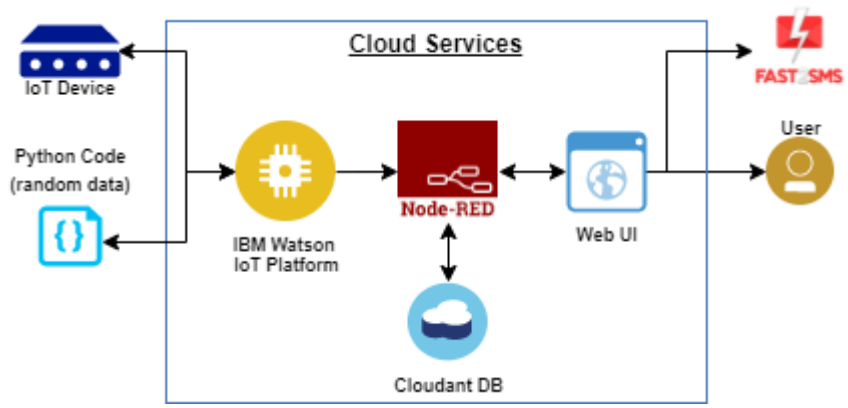
PROJECT FEATURES:

- A Web page is designed for the public where they can book tickets by seeing the available seats.
- After booking the train, the person will get a QR code which has to be shown to the Ticket Collector while boarding the train.
- The ticket collectors can scan the QR code to identify the personal details.
- A GPS module is present in the train to track it. The live status of the journey is updated in the Web app continuously
- All the booking details of the customers will be stored in the database with a unique ID and they can be retrieved back when the Ticket Collector scans the QR Code

TOOLS REQUIRED:

- Python
- IBM Cloud
- Node- RED,
- IoT Platform
- MIT App Inventor
- IBM Cloudant DB

TECHNICAL ARCHITECTURE:

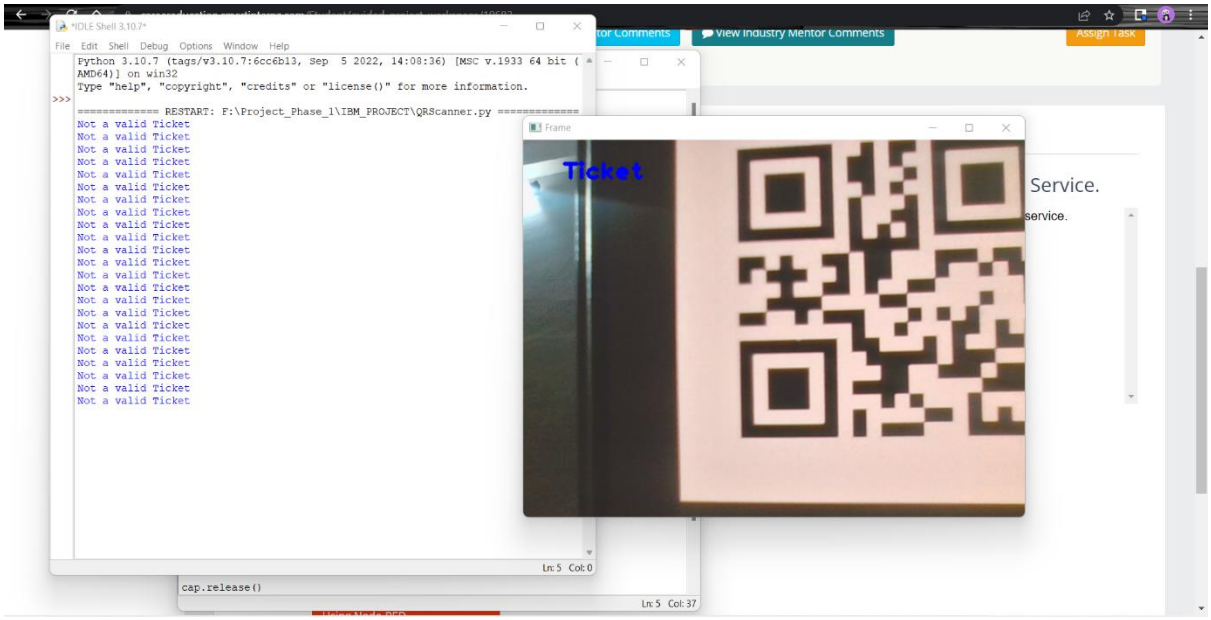


1. **Python code for smart railways:**

```
import cv2
import numpy as np
import time
import pyzbar.pyzbar as pyzbar
#from ibmcloudant.cloudant_v1 import Cloudant_v1
from ibmcloudant.cloudant_v1 import CloudantV1

from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator

authenticator = BasicAuthenticator('apikey-v2-2x3slc1z0cwbin5pcaaaclzqi2g9p2gd5j4got
service = CloudantV1(authenticator=authenticator)
service.set_service_url('https://apikey-v2-amoudkk04osw9s8ofdgy4ywsguhm184fbqjo7qgbb

cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_PLAIN

while True:
    _,frame = cap.read()
    decodedObjects = pyzbar.decode(frame)
    for obj in decodedObjects:
        #print("Data",obj.data)
        a = obj.data.decode('UTF-8')
        cv2.putText(frame,"Ticket",(50,50),font,2,(255,0,0),3)
        #print(a)
        try:
            response = service.get_document(
                db='booking',
                doc_id = a
                ).get_result()
            print(response)
            time.sleep(5)
        except Exception as e:
            print("Not a valid Ticket")
            time.sleep(5)
    cv2.imshow("Frame",frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
client.disconnect()
```

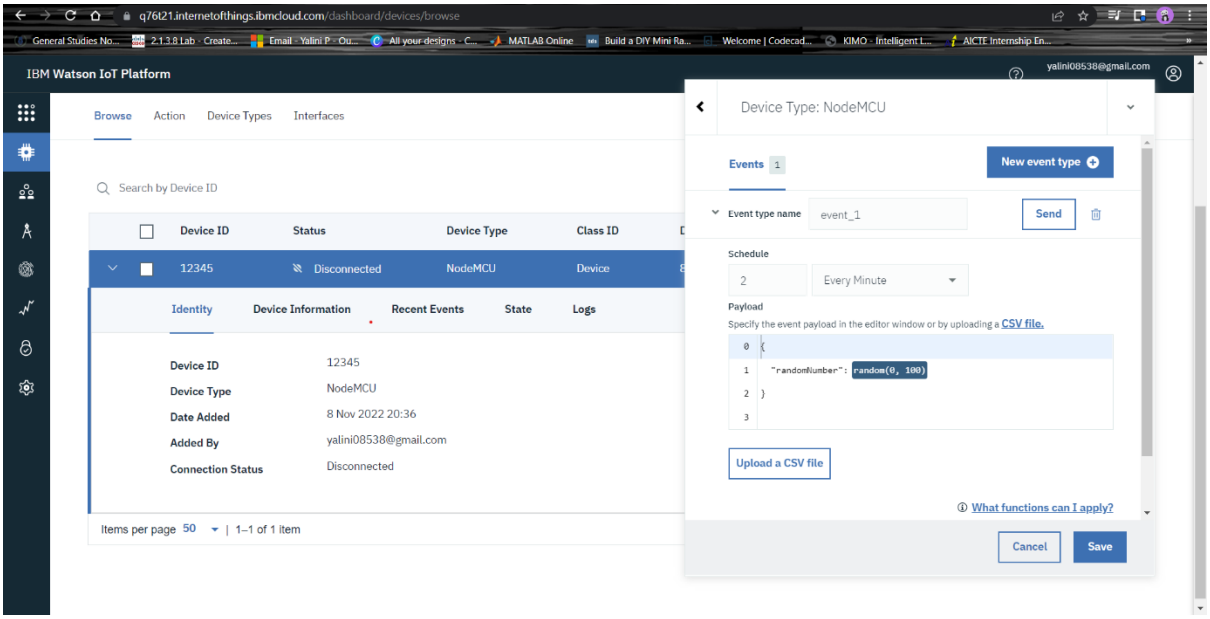
```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity":{
        "orgID":"i63nvt",
        "typeId":"GPS",
        "deviceId":"12345"
        },
    "auth":{
        "token":"12345678"
        }
}
def myCommandCallback(cmd):
    print("Message received fro, IBM IoT Platform: %s"%cmd.data['command'])
    m = cmd.data['command']
client = wiotp.sdk.device.DeviceClient(config,logHandlers=None)
client.connect()

def pub(data):
    client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,onPublish= None)
    print("Published data Successfully:%s",myData)
while True:
    myData = {'name':'Train1','lat':17.6387448,'lon':78.4754336}
    pub(myData)
    time.sleep(3)
    #myData = {'name':'Train1','lat':17.6387448,'lon':78.4754336}
    #pub(myData)
    #time.sleep(3)
    myData = {'name':'Train1','lat':17.6341908,'lon':78.4744722}
    pub(myData)
    time.sleep(3)
    myData = {'name':'Train1','lat':17.6340889,'lon':78.4745052}
    pub(myData)
    time.sleep(3)
    myData = {'name':'Train1','lat':17.6248626,'lon':78.4720259}
    pub(myData)
    time.sleep(3)
    myData = {'name':'Train1','lat':17.6188577,'lon':78.4698726}
    pub(myData)
    time.sleep(3)
    myData = {'name':'Train1','lat':17.6132382,'lon':78.4707318}
    pub(myData)
    time.sleep(3)
    client.commandCallback = myCommandCallback
client.disconnect()
```

OUTPUT:



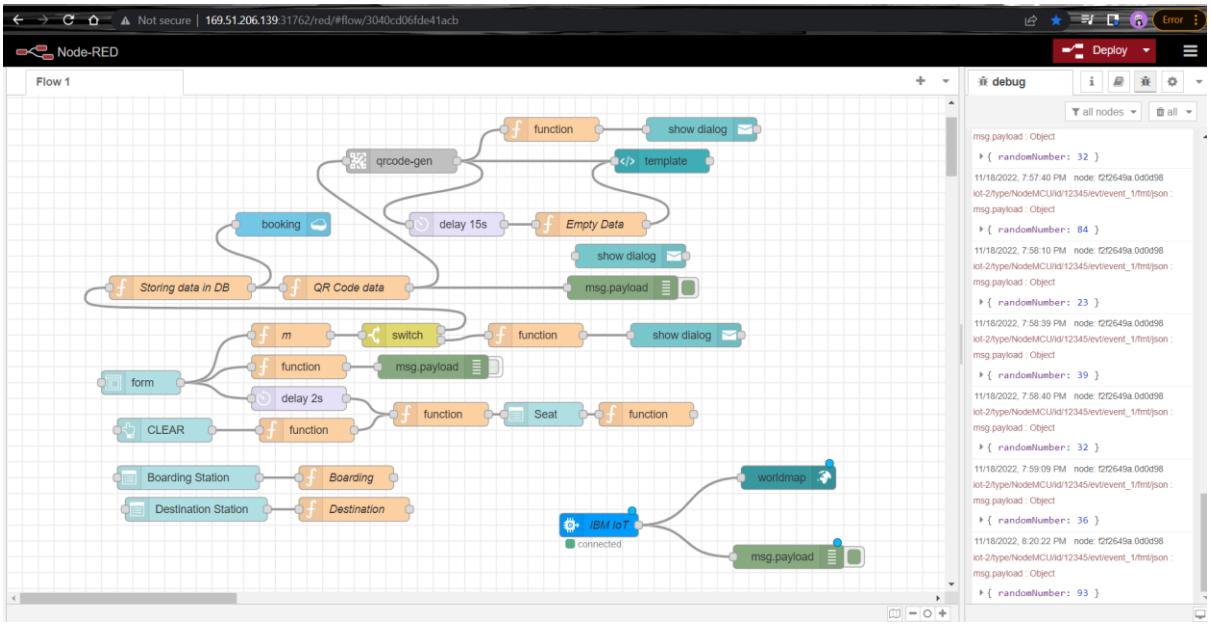
2. IBM WATSON IoT PLATFORM:

- A Node MCU device has been created through IBM Watson IoT platform , using Kubernetes.
- Device ID, Token, Authentication details are given and generated automatically in the Platform.

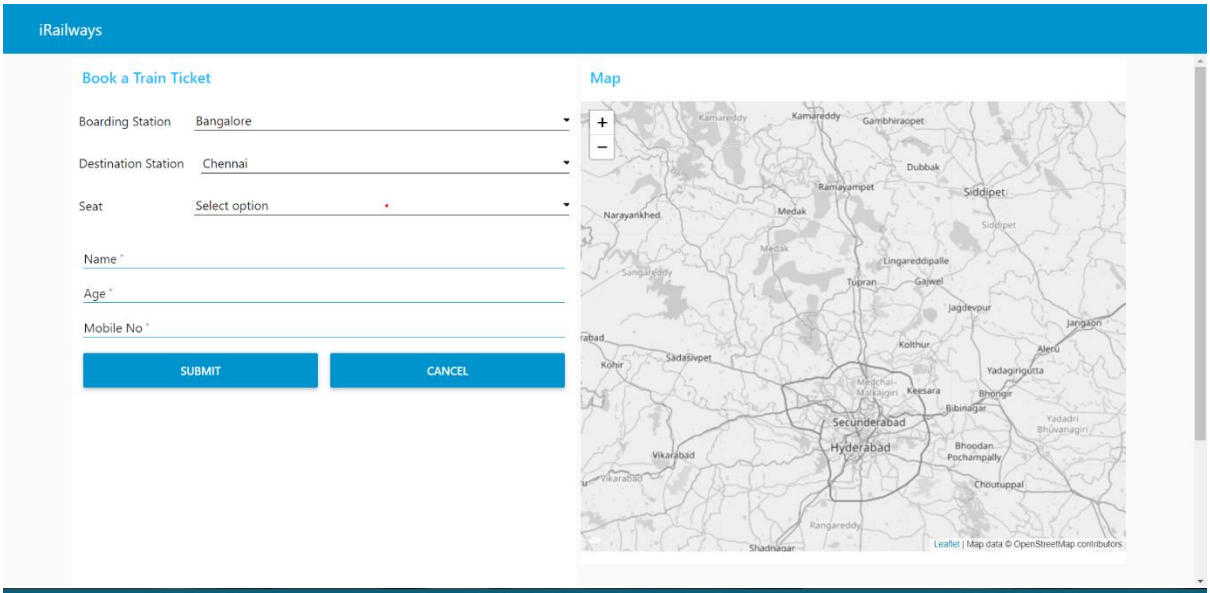


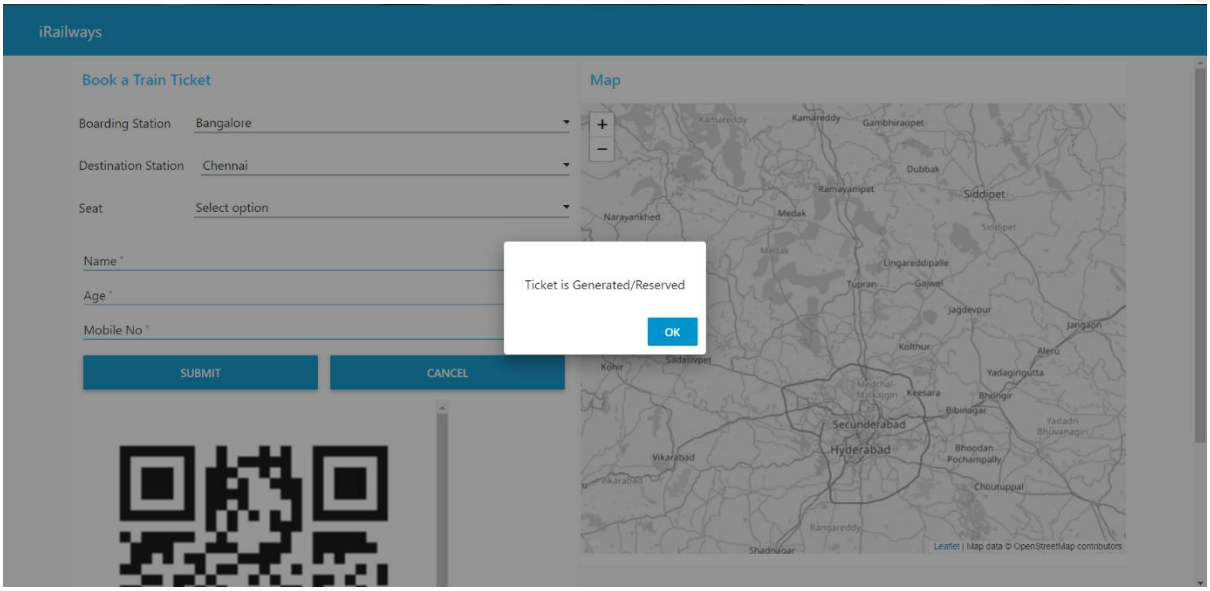
3. **NODE RED SERVICE:**

Node red has been created for smart railways and IBM Watson Iot Platform has been connected to provide results.



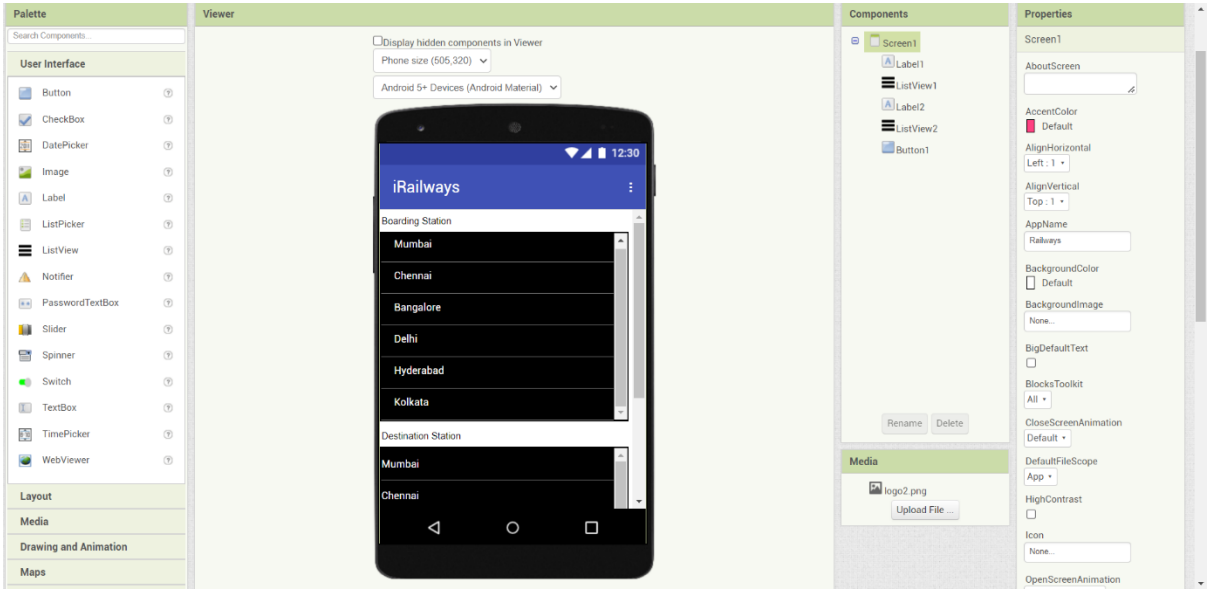
The web UI has been created in accordance with the node red flow.

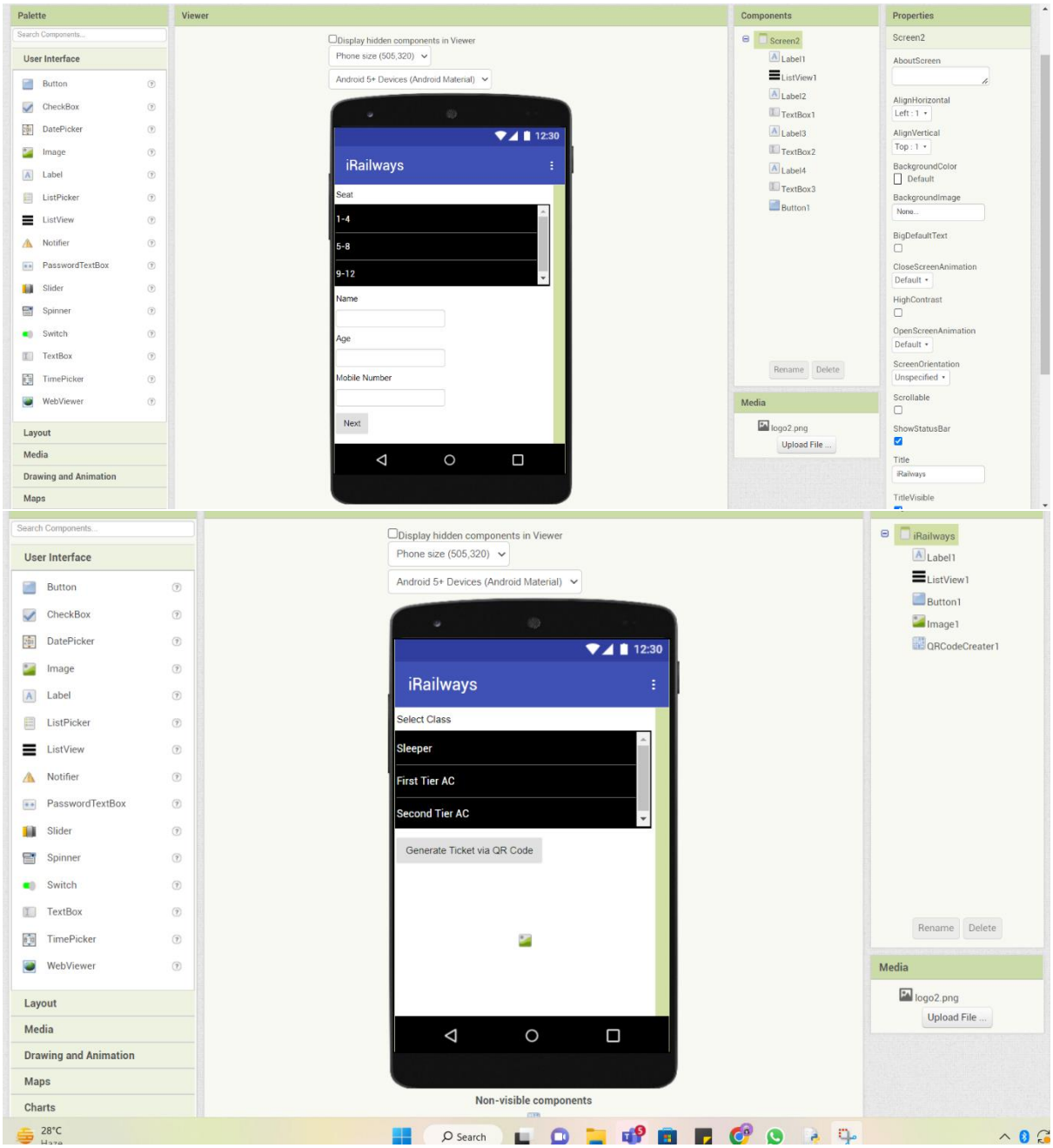


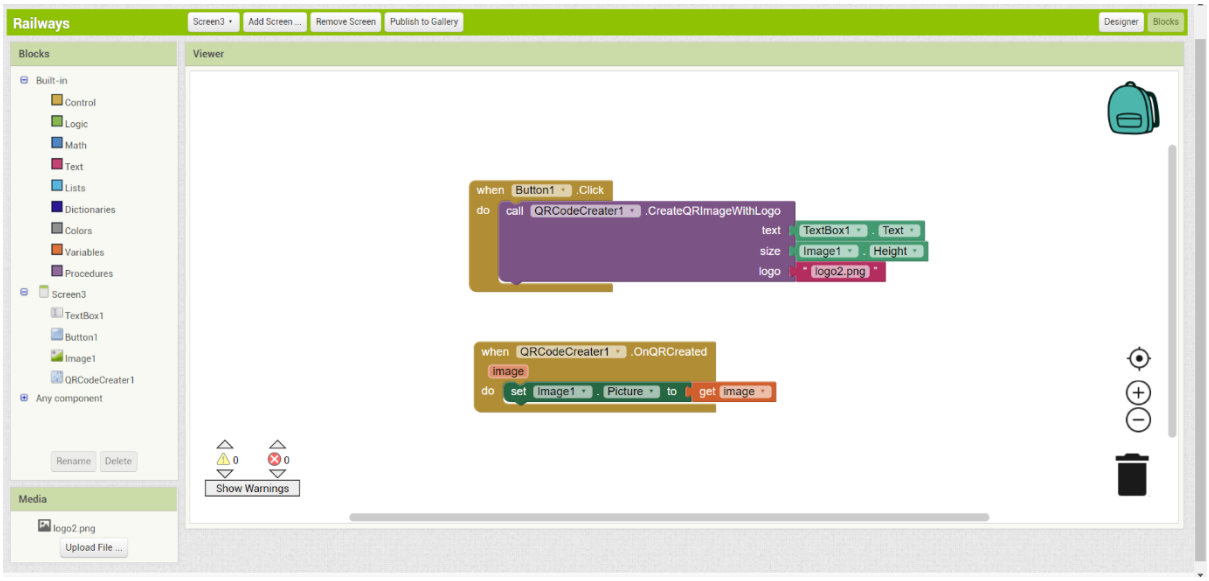


4. MIT APP INVENTOR:

AN application has been created via MIT App inventor.




Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Viewer
Display hidden components in Viewer
Phone size (505,320)
Android 5+ Devices (Android Material)
12:30
iRailways
Seat
1-4
5-8
9-12
Name
Age
Mobile Number
Next
Components
Screen2
Label1
ListView1
Label2
TextBox1
Label3
TextBox2
Label4
TextBox3
Button1
Rename
Delete
Media
logo2.png
Upload File ...
Properties
Screen2
AboutScreen
AlignHorizontal
Left : 1
AlignVertical
Top : 1
BackgroundColor
Default
BackgroundImage
None...
BigDefaultText
CloseScreenAnimation
Default
HighContrast
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable
ShowStatusBar
Title
iRailways
TitleVisible
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Charts
Display hidden components in Viewer
Phone size (505,320)
Android 5+ Devices (Android Material)
12:30
iRailways
Select Class
Sleeper
First Tier AC
Second Tier AC
Generate Ticket via QR Code
Non-visible components
iRailways
Label1
ListView1
Button1
Image1
QRCodeCreater1
Rename
Delete
Media
logo2.png
Upload File ...
28°C
Search



CONCLUSION:

Thus Smart solutions for Railways has been implemented through IBM services and MIT App Inventor. The solution could remain as easy handle one for users.