SMART SOLUTION FOR RAILWAYS

DOMAIN NAME: INTERNET OF THINGS

TEAM ID : PNT2022TMID38188

TEAM MEMBERS:

TEAM LEADER :	DEVIBALA V
TEAM MEMBER 1:	SHARMILA B
TEAM MEMBER 2:	HEMA S
TEAM MEMBER 3:	SANTHALAKSHMI L
TEAM MEMBER 4:	KEERTHIMATHI M

AGENDA

- Introduction
- Purpose
- Problem Statement
- Technical Architecture
- Process
- Future scope

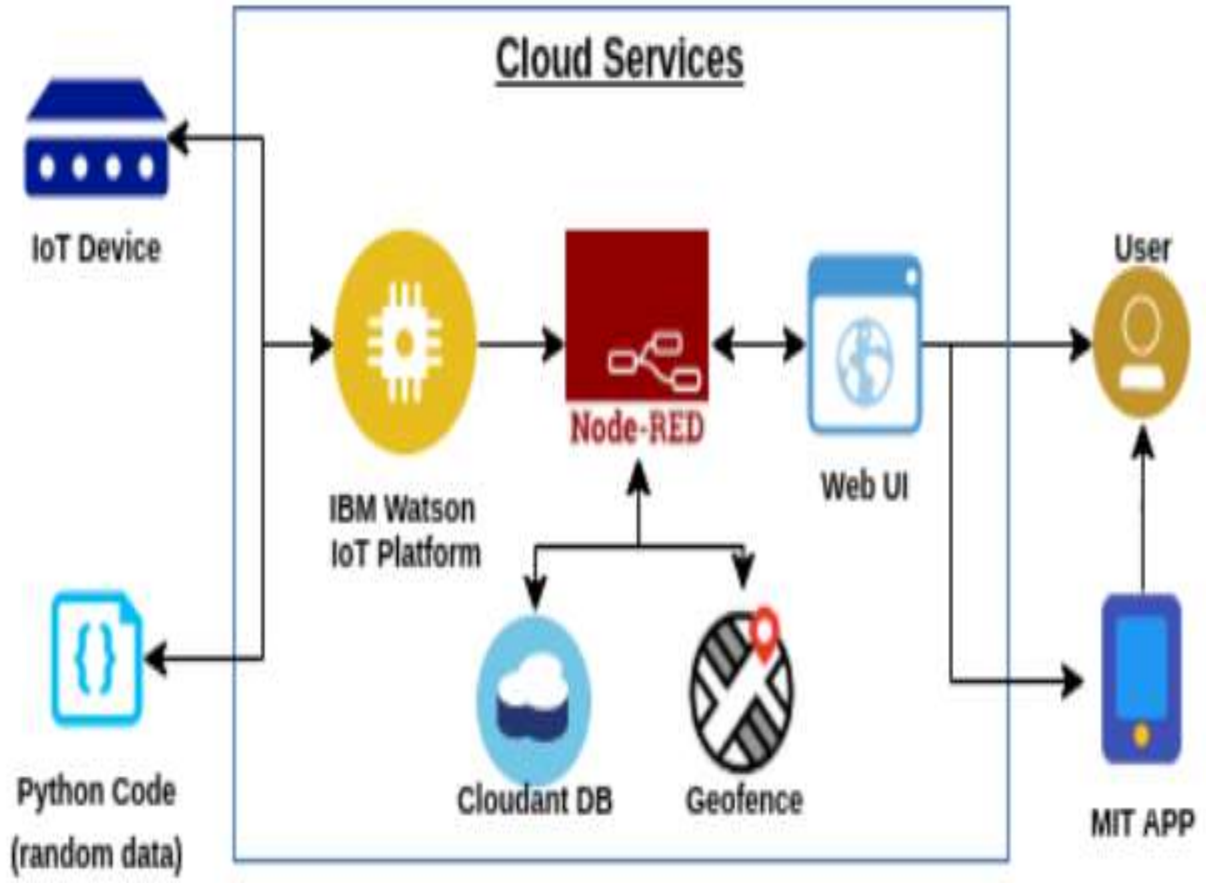
INTRODUCTION

- The developed countries has been implemented smart train using internet of things(IOT)
- IOT based on the system. Using IDL python program . IOT can improve the efficiency of rail travel, and customer experience all those who use it. Smart sensor can used to track important assets, manager passenger flow, and enable predictive maintenance.
- Smart solution of railways is designed to reduce the work load of user.
- In the given project we will developing a web page with help of user find train detail and GPS module present in the train to track it.

PURPOSE

- A web page is designed for the public where can book Tickets by sending the available seats.
- Ticket booking system by using QR code scanning. A GPS module is present in the train to track it. The live status of the journey is update in the web app continuously.

CLOUD SERVICES



PERFORMANCE METRICS

- The user will find the available of the seat.
- Tickets booking system by QR code scanning. A GPS module is present in the train to track. The live status of the journey is updated in the web app continuously. The ticket collectors can scan the QR code to identify the personal details.
- Easy to use.

CODE AND SOLUTION

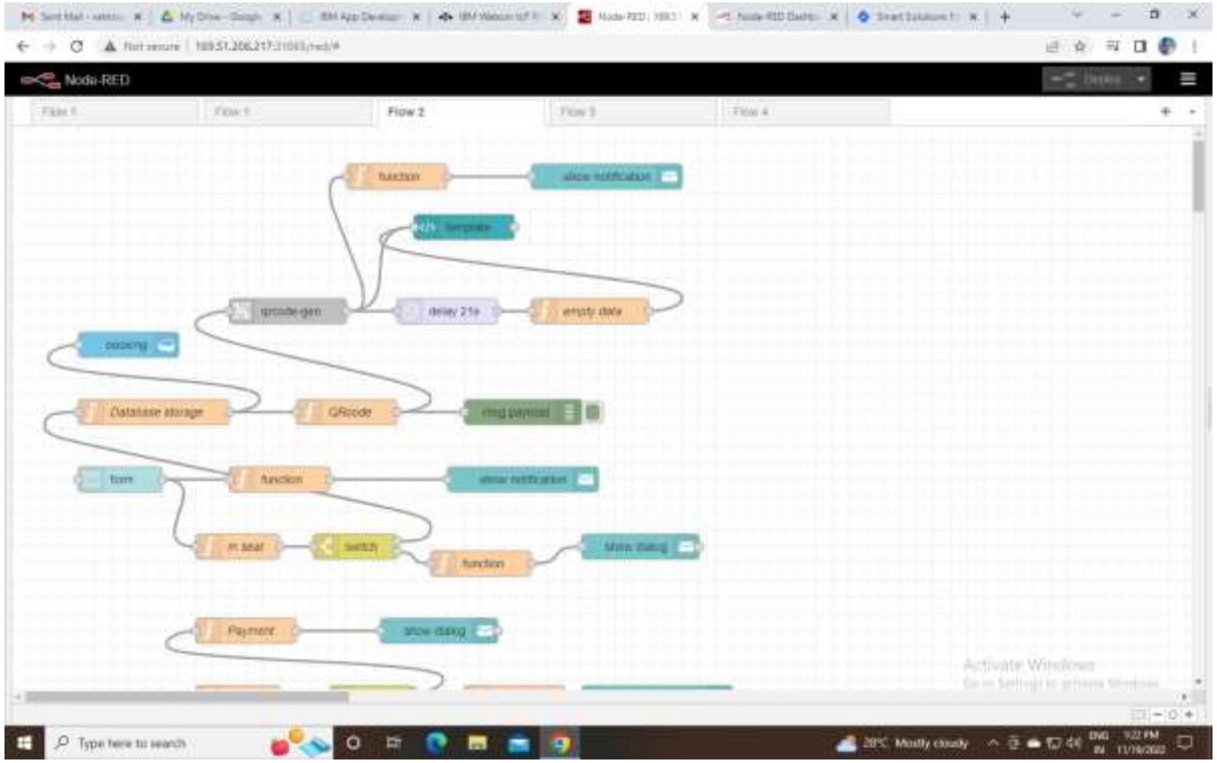
- QR code

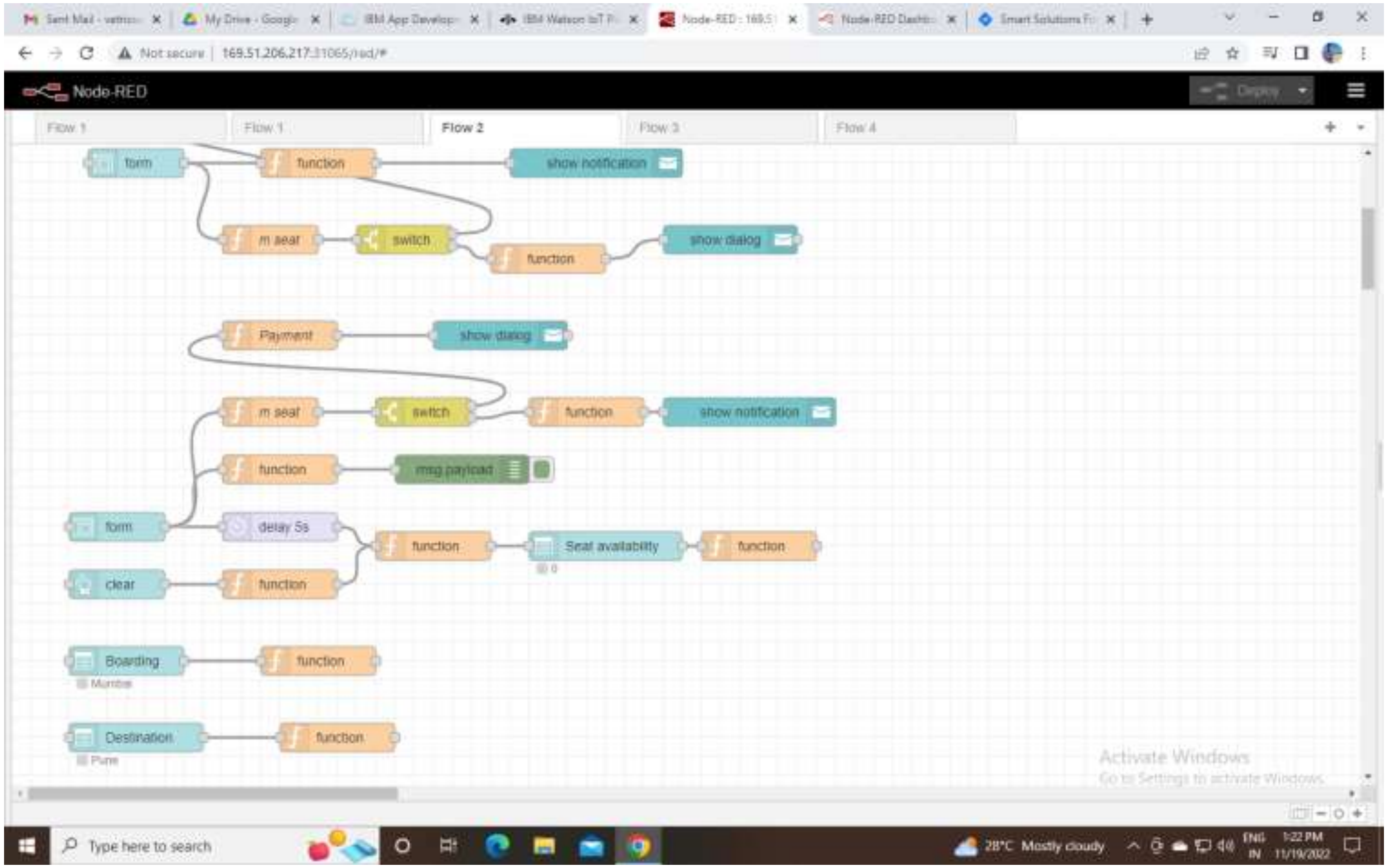
```
import cv2 as cv import numpy as np import time
import pyzbar.pyzbar as pyzbar
from ibmcloudant.cloudant_v1 import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator import wiotp.sdk.device

authenticator=BasicAuthenticator('apikey-v2- 2ji0x00sov1b6clf61hctelp07os2c41mauy6mk7a3ot',
'6866a033c311b4968d996ca9fa217206')
service=CloudantV1(authenticator=authenticator) service.set_service_url('https://apikey-v2-
2ji0x00sov1b6clf61hctelp07os2c41mauy6mk7a3ot:6866a033c311b4968d996ca9fa 217206@53e4077b-d008-4545-8ea1-1d70926b1b71-
bluemix.cloudantnosqldb.appdomain.cloud')

cap = cv.VideoCapture(0) font = cv.FONT_HERSHEY_PLAIN if not cap.isOpened():
print("Cannot open camera") exit()

myConfig = {
"identity" :{
"orgId":"u3neop",
"typeId":"qrcode", "deviceId":"1234567"
},
"auth":{
"token":"1234567890"
}
}
def myCommandCallback(cmd):
print("Message received fromIBM IoT Platform: %s" % cmd.data['command'])
m=cmd.data['command']
```

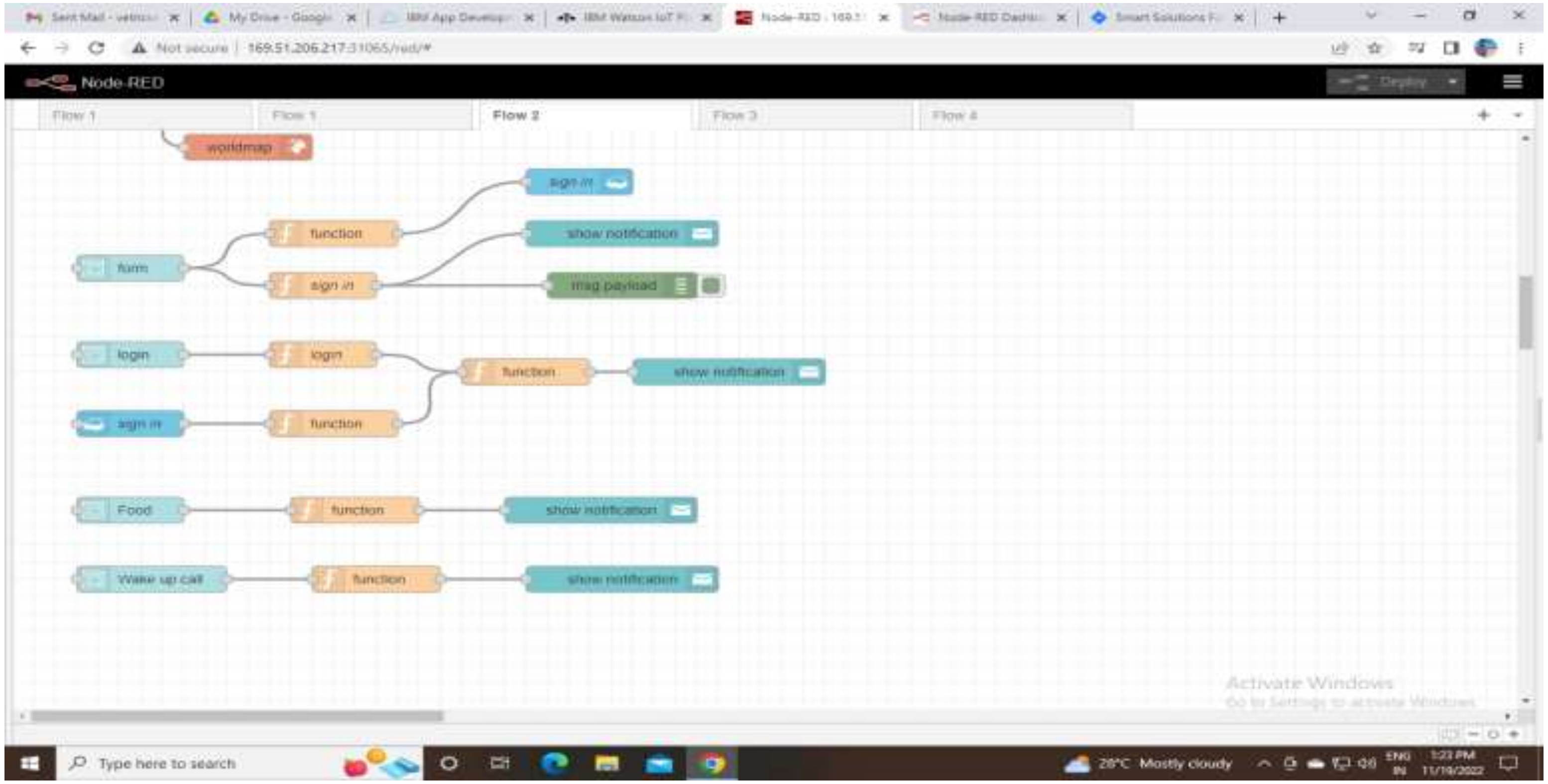
- client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
- client.connect()
-
- def pub(data):
- client.publishEvent(eventId = "status", msgFormat="json", data=response, qos=0, onPublish=None)
- print("Published data Successfully: %s",response) print("\n")
-
- while True:
-

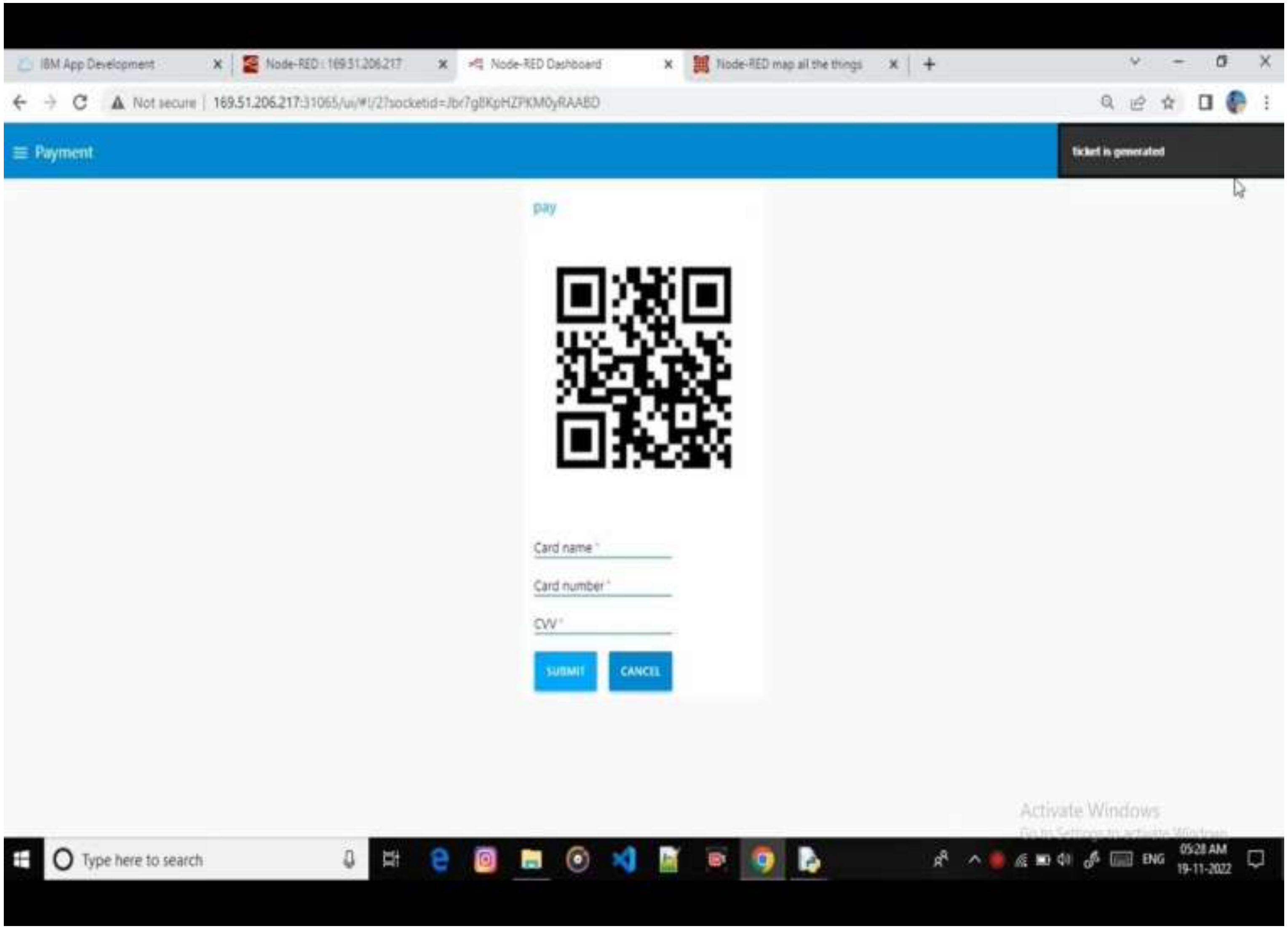
  ret, frame=cap.read()
- decodedObjects = pyzbar.decode(frame) if not ret:
- print("Can't receive frame (stream end?). Exiting ...") break
- for obj in decodedObjects: a=obj.data.decode('UTF-8') cv.putText(frame, "Ticket", (50,50),font,2,
- (255 ,0, 0),3)
-
- try:
- response=service.get_document(db='bookingdetails', doc_id = a
-) .get_result() print(response) print("\n\n") pub(response) time.sleep(5)
- except Exception as e: response={'Error':'Not a Valid Ticket'} pub(response)
- print("Not a Valid Ticket") print("\n\n")
- time.sleep(5)
-

- cv.imshow("Frame" ,frame)
- if cv.waitKey(1) & 0xFF == ord('q'): break
- client.commandCallback = myCommandCallback cap.release()
- cv.destroyAllWindows() client.disconnect()




Node-RED
Deploy
Flow 1
Flow 1
Flow 2
Flow 3
Flow 4
form
function
show notification
m seat
switch
function
show dialog
Payment
show dialog
m seat
switch
function
show notification
function
msg.payload
form
delay 5s
function
Seat availability
function
clear
function
Boarding
Mumbai
function
Destination
Pune
function


Node-RED
Deploy
Flow 1
Flow 1
Flow 2
Flow 3
Flow 4
worldmap
sign in
function
show notification
form
sign in
msg.payload
login
login
function
show notification
sign in
function
Food
function
show notification
Wake up call
function
show notification
Type here to search
28°C Mostly cloudy


IBM App Development
Node-RED : 169.51.206.217
Node-RED Dashboard
Node-RED map all the things
Not secure | 169.51.206.217:31065/ui/#!/2?socketid=Jbr7g8KpHZPKM0yRAABD
Payment
ticket is generated
pay
Card name *
Card number *
CVV *
SUBMIT
CANCEL
Activate Windows
Type here to search
ENG
05:28 AM
19-11-2022

FUTURE SCOPE

- Innovation for superior passenger experience increase safety and security for passenger, staff and assets. Improve operational efficiency.

THANK YOU

- To All