

|              |                             |
|--------------|-----------------------------|
| Title        | Project Report              |
| Team ID      | PNT2022TMID19615            |
| Project Name | Smart Solution for Railways |
| Date         | 19/11/2022                  |

# **Project Report**

1. **INTRODUCTION**
  - 1.1 Project Overview
  - 1.2 Purpose
2. **LITERATURE SURVEY**
  - 2.1 Existing problem
  - 2.2 References
  - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
  - 3.1 Empathy Map Canvas
  - 3.2 Ideation & Brainstorming
  - 3.3 Proposed Solution
  - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
  - 4.1 Functional requirement
  - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
  - 5.1 Data Flow Diagrams
  - 5.2 Solution & Technical Architecture
  - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
  - 6.1 Sprint Planning & Estimation
  - 6.2 Sprint Delivery Schedule
  - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
  - 7.1 Feature 1
  - 7.2 Feature 2
  - 7.3 Database Schema (if Applicable)
8. **TESTING**
  - 8.1 Test Cases
  - 8.2 User Acceptance Testing
9. **RESULTS**
  - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX** Source Code

GitHub & Project Demo Link

# **SMART SOLUTION FOR RAILWAYS**

## **1.INTRODUCTION:**

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

### **1.1 PROJECT OVERVIEW:**

- 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.

### **1.2 PURPOSE :**

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

## 2.LITERATURE SURVEY

### 2.1 EXISTING PROBLEM:

The seats of their choice are not available to passengers. They must bring a physical ticket with them, which could be misplaced. Quick-moving passengers might not have enough time to wait for the train for an extended period of time. They can decide whether to wait or use another form of transportation by knowing the train's current location.

### 2.2 REFERENCES

| S.No | TITLE   | PROPOSED WORK  | TOOLS USED/<br>ALGORITHM                           | TECHN<br>OLOGY     | ADVANTAGES/<br>DISADVANTAGE<br>S  |
|------|---|--|--|--------------------|---|
| 1.   | smart railway systems of layer applications based on internet of things | Railway networks across the world are getting busier and busier; trains travel at higher speeds and carry more passengers or heavier axle loads than ever before. Accordingly, the railway industry calls for new information technologies (ITs) to meet its development. Railway systems have already relied on ITs almost as much as they rely on physical assets, and this dependence is growing as these systems face burgeoning demands . | big data, sensors, data mining,intelligent systems | Internet of Things | This chapter will discuss the framework and for a smart railway based on the Internet of Things and big data, we present the architecture of a smart railway, which is divided into four layers perception and action layer, transfer layer, data engine layer, application layer, and discuss the advanced technologies in each layer. |

| S.No | TITLE  | PROPOSED WORK   | TOOLS USED/ ALGORITHM | TECHNOLOGY         | ADVANTAGES/ DISADVANTAGES  |
|------|--|---|-----------------------|--------------------|--|
| 2.   | A Novel Approach for Big Data Classification and Transportation in Rail Networks | A new framework into future data-driven railway condition monitoring systems (RCM). For this proposed an edge processing unit that includes two main parts: a data classification model that classifies Internet of Things (IoT) data into maintenance-critical data (MCD) and maintenance-non-critical data (MNCD) and a data transmission . | CBM, RCM, MCD,MNCD    | Internet of Things | The development of condition based monitoring CBN systems in the railway industry has received the highest investment policy will deal with of big data problem in the future because these have velocity, and volume . RCM will be strongly reliant on data received from heterogeneous IOT devices . |

| S.No | TITLE   | PROPOSED WORK   | TOOLS USED/ ALGORITHM                      | TECHNOLOGY         | ADVANTAGES/ DISADVANTAGES   |
|------|---|---|--|--------------------|---|
| 3.   | Remote sensor networks for condition monitoring: An application on railway industry | In recent years, the range of sensing technologies has expanded rapidly, whereas sensor devices have become cheaperThis has prompted to a fast extension in condition checking of frameworks, structures, vehicles, and hardware utilizing sensors. Key components are the current advances in systems administration | Remote sensor systems (WSNs) , LPWAN,RFID. | Internet of Things | This is indispensable for the advancement, redesigning, and extension of railroad systems. This venture studies the remote sensors arrange innovation for checking in the railroad business for dissecting frameworks, structures, vehicles, and apparatus. |

| S.No | TITLE                                  | PROPOSED WORK   | TOOLS USED/ ALGORITHM                                 | TECHNOLOGY         | ADVANTAGES/ DISADVANTAGE  |
|------|--|---|---|--------------------|---|
| 4.   | 5G key technologies for smart railways | Railway communications has attracted significant attention from both academia and industries due to the booming development of railways, especially high-speed railways (HSRs). To be in line with the vision of future smart rail communications, the rail transport industry needs to develop innovative communication network architectures and key technologies | SDN , SD-WAN ,5G edge, digital and hybrid multi cloud | Internet of Things | we have identified significant 5G-based key technologies for HSRs, such as spatial modulation, fast channel estimation, cell-free massive multiple-input-multiple-output (MIMO), mmWave, efficient beamforming, wireless backhaul, reliable low latency communications, and enhanced handover strategies. |

| S.No | TITLE  | PROPOSED WORK  | TOOLS USED/ ALGORITHM          | TECHNOLOGY         | ADVANTAGES/ DISADVANTAGES   |
|------|--|--|--------------------------------|--------------------|---|
| 5.   | OTFS-TSMA for Massive Internet of Things in High-Speed Railway | Massive internet of things (mIoT) could play an important role in the future smart high-speed railway (HSR), where grant-free multiple access technologies are required. Recently, tandem spreading multiple access (TSMA) has been raised for mIoT without mobility which achieves high connectivity and reliability. | MIOT , TSMA , OTFS Transceiver | Internet of Things | the four typical smart railways services, including railway safety-critical service, passenger oriented service , decision making smart HSR to enable environment sensing of IOT service in greater ways. |

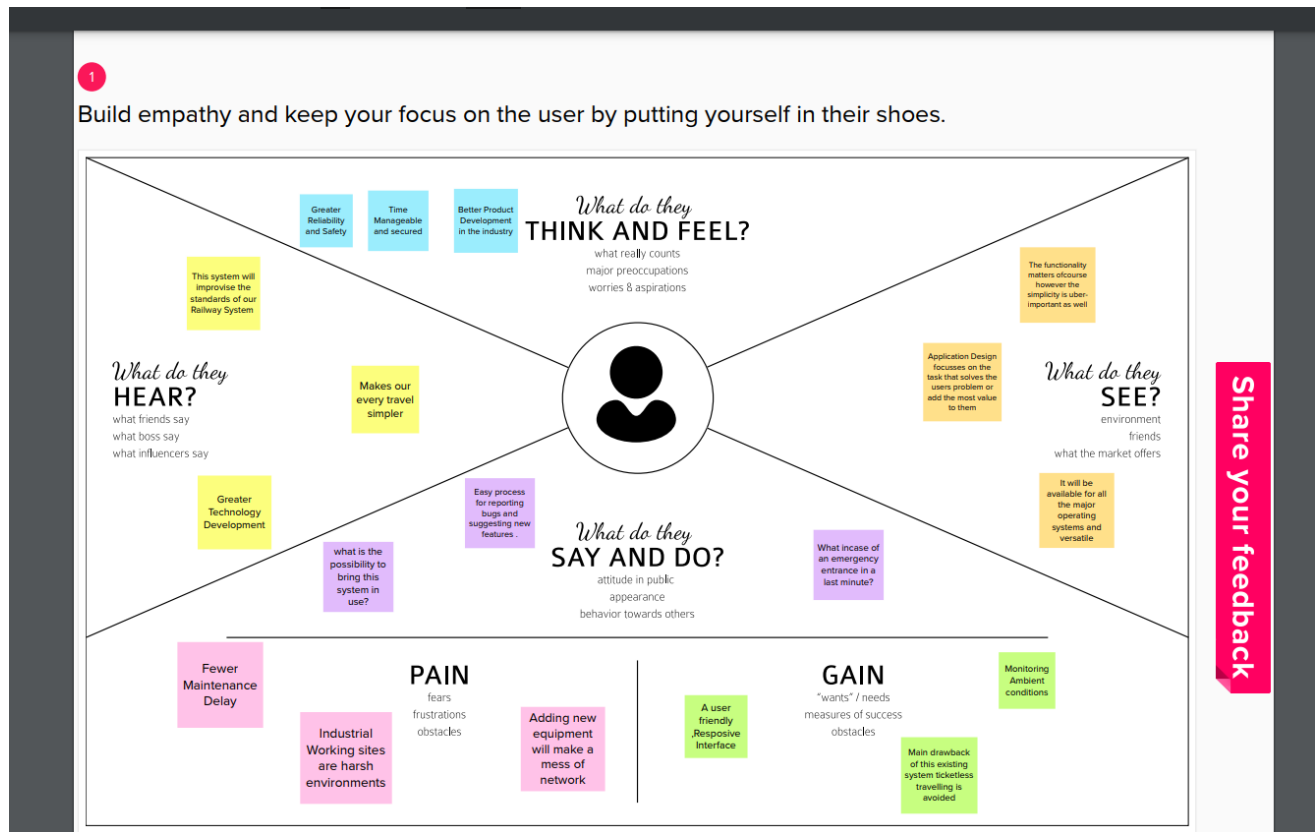
### 2.3.PROBLEM STATEMENT DEFINITION:

| QUESTION                                     | DESCRIPTION  |
|--|--|
| Who does the problem affect?                 | Voyagers, travelers & tourists.  |
| What are the boundaries of the problem?      | Mechanism for purchasing train tickets that generates a special QR code for each ticket.<br>GPS tracking of a train's present location.  |
| What is the issue?                           | The seats of their choice are not available to passengers. They must bring a physical ticket with them, which could be misplaced.<br>Quick-moving passengers might not have enough time to wait for the train for an extended period of time. They can decide whether to wait or use another form of transportation by knowing the train's current location. |
| When does the issue occur?                   | All the time.  |
| Where is the issue occurring?                | Train and in Railway Stations currently available train reservation systems.   |
| Why is it important that we fix the problem? | Railway is one of the most common modes of transport. Improving the user experience is very important. An efficient way to check the tickets is of top priority.   |



### 3.IDEATION & PROPOSED SOLUTION

#### 3.1.Empathy Map Canvas



#### 3.2 Ideation & Brainstorming:





### 3.3 Proposed Solution :

| S.No. | Parameter                                       | Description   |
|-------|---|---|
| 1.    | <b>Problem Statement (Problem to be solved)</b> | *Smart Solutions for railways is designed to reduced the work load of the user and also the use of paper and also provides the live location of the train.  |
| 2.    | <b>Idea / Solution description</b>              | *A webpage is designed in which the user can book tickets and will be provided with a QR code which will be shown to the ticket collector and the ticket collector will be scanning the QR code to get the passenger details. |

|    |  |  |
|----|--|--|
| 3. | <b>Novelty / Uniqueness</b>                  | *A QR code will be provided by the webpage to the user which will reduce the paper work.   |
| 4. | <b>Social Impact / Customer Satisfaction</b> | *The booking tickets is made easy to use and it is also reliable and no need to go to station for booking tickets and the transaction process is also made easy. |

|           |                                       |   |
|-----------|---------------------------------------|---|
| <b>5.</b> | <b>Business Model (Revenue Model)</b> | *With this solution - By using this application, the customer can schedule their destination, view availability of the seat, view interactive seat map and select their seat for their convenience.   |
| <b>6.</b> | <b>Scalability of the Solution</b>    | <ol style="list-style-type: none"> <li>1. No need of taking print out.</li> <li>2. Counter ticket has to be handled with care, but SMS on mobile is more than enough.</li> <li>3. You are becoming environment friendly and contributing for greener planet by ignoring printout.</li> <li>4. No need of taking out wallet and showing your ticket to TTR, just tell your name to TTR that you are passenger with a valid proof.</li> <li>5. While booking counter ticket you had to carry cash and while booking E- ticket you are paying through online directly from bank which makes work more easy for you.</li> </ol> |

### 3.4 Problem Solution fit:

|                        |  |  |   |                           |
|------------------------|--|--|---|---------------------------|
| Define CS, fit into CC | <b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span><br>Who is your customer ? <ul style="list-style-type: none"> <li>❖ People who travel via train passengers</li> <li>❖ who wants to book ticket remotely and also require list of trains scheduled</li> </ul>  | <b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span><br>What constraints prevent your customers from taking action or limit their choices of solutions? <ul style="list-style-type: none"> <li>❖ Some customers may struggle to use this application because they may not know how to use these kind of applications</li> <li>❖ Unaware of such alternative way and still depend on old traditional method</li> </ul>            | <b>5. AVAILABLE SOLUTIONS</b> <span>AS</span><br>Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? <ul style="list-style-type: none"> <li>❖ Customers can book tickets through internet instead of standing in long queue at ticket counter</li> <li>❖ Solution requires internet and handheld devices throughout the journey</li> </ul> | Explore AS, differentiate |
|                        | <b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span><br>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <ul style="list-style-type: none"> <li>❖ To book train ticket through internet</li> <li>❖ To track and update live location of train using GPS</li> </ul> | <b>9. PROBLEM ROOT CAUSE</b> <span>RC</span><br>What is the real reason that this problem exists? What is the back story behind the need to do this job? <ul style="list-style-type: none"> <li>❖ Old method is time consuming due to long queue chances missing the train even some passengers may travel without ticket, this factors affects the revenue of railways</li> <li>❖ Modernize the railway system</li> </ul> | <b>7. BEHAVIOUR</b> <span>BE</span><br>What does your customer do to address the problem and get the job done? <ul style="list-style-type: none"> <li>❖ Customer need to create account and login into the application</li> <li>❖ Select boarding and departing station and complete the payment process</li> </ul>   |                           |

|                               |   |  |  |
|-------------------------------|---|--|--|
| Identify system triggers & EM | <b>3. TRIGGERS</b> <span>TR</span><br>What triggers customers to act? <p>People wish to avoid Long queue at ticket counter to reduce time consumption and avoid mental pressure</p>   | <b>10. YOUR SOLUTION</b> <span>SL</span><br>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. <ul style="list-style-type: none"> <li>❖ Book train tickets through internet</li> <li>❖ Tickets can be generated in QR code format</li> <li>❖ User friendly interface is implemented</li> <li>❖ Live location of the train and list of trains available to reach the destination can be updated</li> </ul> | <b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span><br><b>8.1 ONLINE</b><br>What kind of actions do customers take online? <ul style="list-style-type: none"> <li>❖ User need login into the application and booking ticket</li> <li>❖ Make use of other features such as live location of the train</li> </ul><br><b>8.2 OFFLINE</b><br>What kind of actions do customers take offline?<br>Station master may validate the ticket through QR code scanner |
|                               | <b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span><br>How do customers feel when they face a problem or a job and afterwards?<br>Customer may worry about missing train due long queue(before)> instant booking gives them confidence |  |  |

## 4 REQUIREMENT ANALYSIS

### Functional requirement:

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task)   |
|--------|-------------------------------|--|
| FR-1   | <b>User Registration</b>      | ✚ Registration through Gmail<br>✚ Registration through phone number  |
| FR-2   | <b>User Confirmation</b>      | ✚ Confirmation via Email<br>✚ Confirmation via OTP   |
| FR-3   | <b>User Authentication</b>    | ✚ Authentication using password<br>✚ Authentication using phone number OTP<br>✚ Authentication Using Aadhar number & biometric |
| FR-4   | <b>User Details</b>           | ✚ Enter the personal details<br>✚ Enter payment details<br>✚ Enter source & destination to travel                              |
| FR-5   | <b>Priority</b>               | ✚ Administrator can have access to the database<br>✚ Ticket booking process & payment has priority                             |
| FR-6   | <b>Features of Admin</b>      | ✚ The Admin is a super user that have to access to everything.<br>✚ He/her can Analysis the data and process it                |

### Non Functional planning:

Following are the non-functional requirements of the proposed solution.

| FR No. | Non-Functional Requirement | Description  |
|--------|----------------------------|--|
| NFR-1  | <b>Usability</b>           | <ul style="list-style-type: none"><li>✚ User friendly UI is required for both website &amp; mobiles</li><li>✚ Tickets can be booked instantly</li><li>✚ List of train to reach the destination is listed</li></ul>   |
| NFR-2  | <b>Security</b>            | <ul style="list-style-type: none"><li>✚ The system provides password protection and end to end encryption technology</li><li>✚ Tickets cannot be duplicated</li><li>✚ User data &amp; privacy is protected in secured database</li></ul>   |
| NFR-3  | <b>Reliability</b>         | <ul style="list-style-type: none"><li>✚ More reliable since the user can access application from anywhere and any time</li><li>✚ Tickets can be booked without any waiting period</li><li>✚ In case customer missed the train, list of next arriving trains to the destination can be sorted out</li></ul> |
| NFR-4  | <b>Performance</b>         | <ul style="list-style-type: none"><li>✚ The application can handle huge number of users at a time so no server issues occur</li><li>✚ Increases revenue through the fast &amp; secure way of ticket booking</li></ul>  |
| NFR-5  | <b>Availability</b>        | <ul style="list-style-type: none"><li>✚ Application can be built for specific operating systems &amp; access the website using any kind of browsers</li><li>✚ User can log in at anytime and anywhere to book tickets without reaching the ticket counter</li></ul>  |
| NFR-6  | <b>Scalability</b>         | <ul style="list-style-type: none"><li>✚ Lite version of website can be developed for less data consumption</li><li>✚ simplify the complicated process of booking tickets with few steps</li></ul>  |

## 5 PROJECT DESIGN

### 5.2 Data Flow Diagrams

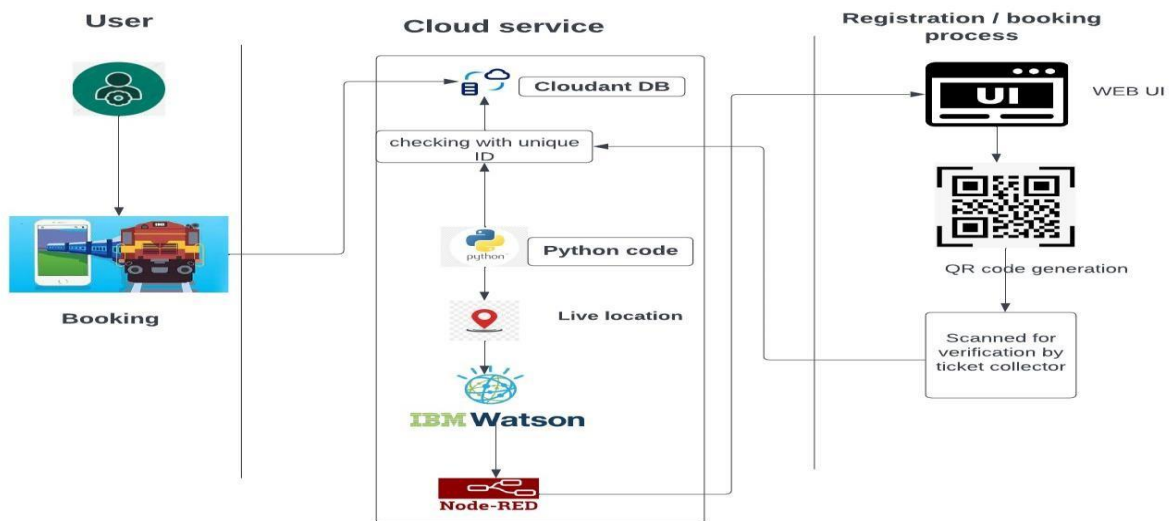
#### Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example:



### 5.3 Solution & Technical Architecture



## Technical Architecture:

**Table-1 : Components & Technologies:**

| S.No | Component       | Description  | Technology            |
|------|-----------------|--|-----------------------|
| 1.   | Web UI          | User can login and book their ticket through the website based on the availability of the seats. | HTML, CSS, JavaScript |
| 2.   | Cloud Services  | Requirements filled by the passenger is stored in the cloud database.                            | Python                |
| 3.   | GPS Tracking    | Live Location details shared through the code to share the location in the website               | IBM Watson Service    |
| 4.   | External API-1  | Used for rail schedule, ticketing and travel documents generation, cancellation.                 | Sabre API             |
| 5.   | External API-2  | Used for combining carriers and ticket types, Multilanguage & currency support.                  | Trainline B2B API     |
| 6.   | Data Processing | Ticket is verified with the unique ID generated with the cloudant DB                             | Python, IBM cloud     |

**Table-2: Application Characteristics:**

| S.No | Characteristics          | Description   | Technology                        |
|------|--------------------------|---|-----------------------------------|
| 1.   | Open-Source Frameworks   | CSS, Backend framework,   | Python, IBM cloudant DB           |
| 2.   | Security Implementations | Data entered are encrypted, Continuous Location Tracking  | Python, Cloud service             |
| 3.   | Scalable Architecture    | The scanner and the codes written are highly scalable where any implementation can be done anytime needed   | Python                            |
| 4.   | Availability             | Any time available system. The ticket can be verified by the ticket collector from anywhere.  | IBM Load Balancer                 |
| 5.   | Performance              | Though the details are get stored in the cloud the system crash will not affect the data. The data can be retrieved from anywhere with a scanner. And the GPS states the exact location of the train. | Distributed Services, GPS Tracker |

## 5.4 User Stories



## User Stories

Use the below template to list all the user stories for the product.

| User Type               | Functional Requirement (Epic) | User Story Number | User Story / Task  | Acceptance criteria   | Priority | Release  |
|-------------------------|-------------------------------|-------------------|--|---|----------|----------|
| Customer (Mobile user)  | Reserving ticket              | USN-1             | As a user, I can reserve for the ticket by entering my email, password, and confirming my password.  | I can access my account / dashboard                           | High     | Sprint-1 |
|                         |                               | USN-2             | As a user, I will receive confirmation email once I have registered for the account  | I can receive confirmation email & click confirm              | High     | Sprint-1 |
|                         |                               | USN-3             | As a user, I can register for the account and enter the details for reserving the ticket and get information about Train and Receive Alerts. | I can register & access the dashboard with Login Credentials. | Low      | Sprint-1 |
|                         | Dashboard                     | USN-4             | As a user, I can view the reserved ticket in the dashboard.  | I can access it using database                                | Medium   | Sprint-3 |
| Customer (Webuser)      | Reserving ticket              | USN-1             | As a user, I can register to creating account by entering email, password.   | I can access my account                                       | High     | Sprint-1 |
|                         |                               | USN-2             | As a user, I will receive confirmation email once I have registered for the web user.  | I can receive confirmation email.                             | High     | Sprint-1 |
|                         |                               | USN-3             | As a user, I can log into the website entering email & password.   | Only valid credentials must be acceptable.                    | High     | Sprint-1 |
|                         | Dashboard                     | USN-4             | As a user, I can register for the account and enter the details for reserving the ticket and get information about Train                     | I can register & access the dashboard with Login Credentials. | Low      | Sprint-1 |
| Customer Care Executive | Customer                      | USN-1             | Connects with the service by logging in and get alert through it.  | Can get connected with the server                             | Medium   | Sprint-1 |
| Administrator           | Admin                         | USN-1             | As a admin, He/She can monitor real time and send alerts.  | The admin can monitor the process by 24/7 hrs.                | High     | Sprint-1 |

## 6.PROJECT PLANNING & SCHEDULING

### 6.1. Sprint Planning & Estimation

| Sprint   | Functional Requirement (Epic) | User Story Number | User Story / Task   | Story Points | Priority | Team Members   |
|----------|-------------------------------|-------------------|---|--------------|----------|--|
| Sprint-1 | Monitor the Speed of Train    | USN-1             | The Railway must take care of passengers and peoples. In the train there are so many families we should secure them.  | 2            | High     | Pravin k<br>Sanjai Kumar S<br>Ramesh Prasath B<br>Sakthi Kumar M |
| Sprint-2 | Avoid From Accidents          | USN-2             | If any accident occurs their Technical team will take care of it and save the passengers.   | 1            | High     | Pravin k<br>Sanjai Kumar S<br>Ramesh Prasath B<br>Sakthi Kumar M |
| Sprint-3 | Detect the Motions            | USN-3             | We have monitor the motions and delays by 24/7 hrs. To avoid the accidents, and delays by using only sensors. The railway must take care of what are the necessary process to avoid the train accidents and delays. | 2            | Low      | Pravin k<br>Sanjai Kumar S<br>Ramesh Prasath B<br>Sakthi Kumar M |

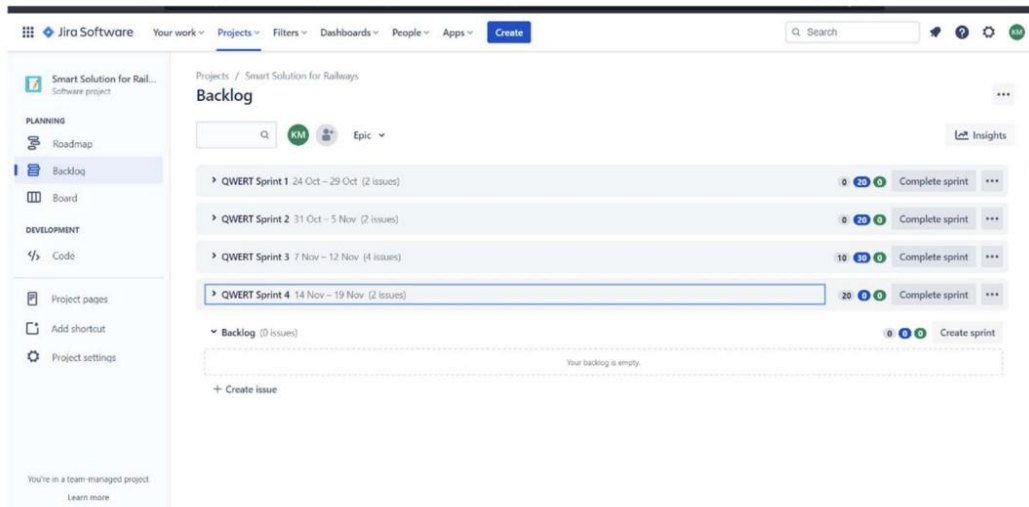
|               |  |                          |   |                     |                 |  |
|---------------|--|--------------------------|---|---------------------|-----------------|--|
| Sprint-4      | The model is trained and tested by sample dataset. | USN-4                    | The programmer design the model to detect the Train Details.  | 2                   | Medium          | Pravin k<br>Sanjai Kumar S<br>Ramesh Prasath B<br>Sakthi Kumar M |
| <b>Sprint</b> | <b>Functional Requirement (Epic)</b>               | <b>User Story Number</b> | <b>User Story / Task</b>  | <b>Story Points</b> | <b>Priority</b> | <b>Team Members</b>  |
| Sprint-5      | Warning message                                    | USN-5                    | Incase any accident or delay occur, the devicegive the alarm and alert message to concerned department within a minute. | 1                   | High            | Pravin k<br>Sanjai Kumar S<br>Ramesh Prasath B<br>Sakthi Kumar M |

## 6.2. Sprint Delivery Schedule

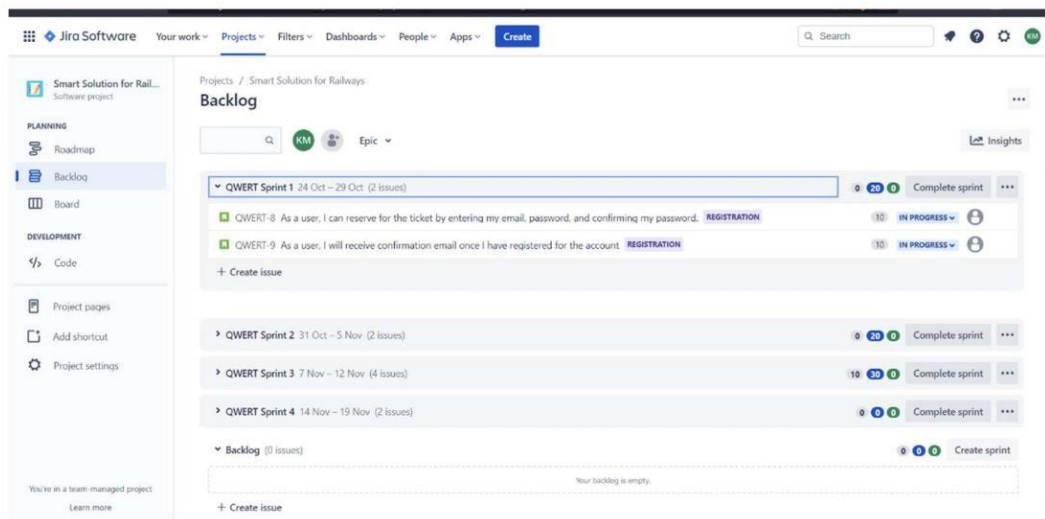
| Sprint   | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|----------|--------------------|----------|-------------------|---------------------------|---|------------------------------|
| Sprint-1 | 20                 | 6 Days   | 24 Oct 2022       | 29 Oct 2022               | 20  | 29 Oct 2022                  |
| Sprint-2 | 20                 | 6 Days   | 31 Oct 2022       | 05 Nov 2022               | 20  | 05 Nov 2022                  |
| Sprint-3 | 20                 | 6 Days   | 07 Nov 2022       | 12 Nov 2022               | 20  | 12 Nov 2022                  |
| Sprint-4 | 20                 | 6 Days   | 14 Nov 2022       | 19 Nov 2022               | 20  | 19 Nov 2022                  |
| Sprint-5 | 20                 | 6 Days   | 14 Nov 2022       | 19 Nov 2022               | 20  | 19 Nov 2022                  |

## 6.3. Reports from JIRA:

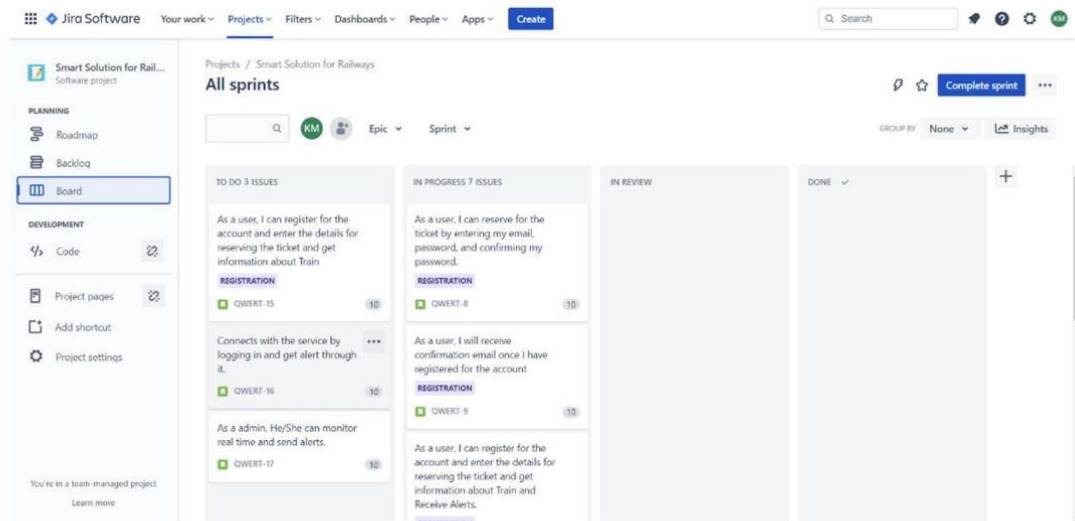
## 1. Image showing the backlogs create in the JIRA Software (PETA Spring1, PETA Spring2, PETA Spring3, PETA Spring4)



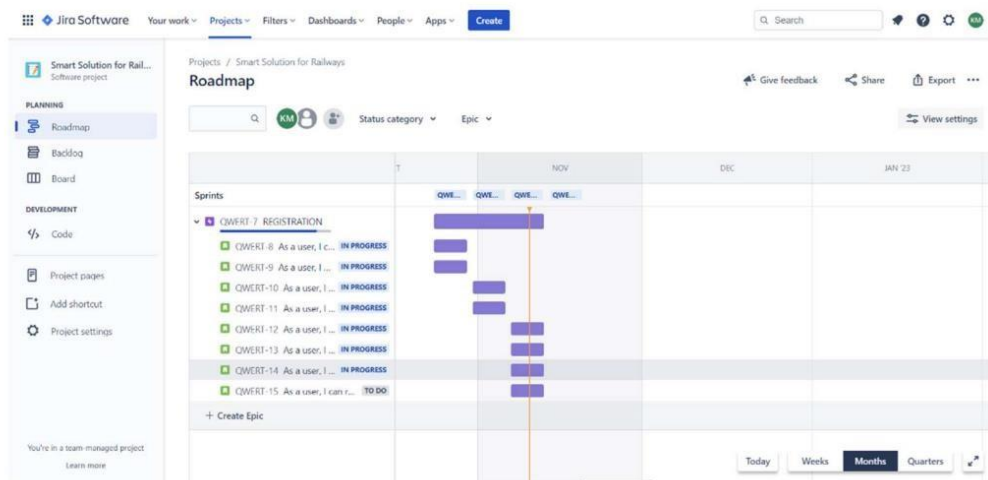
## 2. Image Showing the User Stories in the respective sprints.



### 3. Image showing the workspace progress in the sprint



### 4. Image Showing the RoadMap of Sprint



## 6.CODING & SOLUTIONING:

### Python py:

```
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "rail67",
        "typeId": "railways",
        "deviceId": "device"
    },
    "auth": {
        "token": "648790"
    }
}

def myCommandCallback (cmd):
    print ("Message received from IBM 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=myData,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': 'Train2', '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

```

**Scanner.py** from http import

```

client import cv2 import pyzbar
from pyzbar. pyzbar import
decode import time

```

```

from ibmcloudant.cloudant_v1 import CloudantV1 from
ibmcloudant import CouchDbSessionAuthenticator from
ibm_cloud_sdk_core.authenticators import BasicAuthenticator

```

```

authenticator = BasicAuthenticator('apikey-v2-1oj043bu90m78ng4h2j27w5nob2nvcma6xanc6bk0a7m',
'daf3c00c2cc182af425a5691a07f7b93') service =
CloudantV1(authenticator=authenticator)
service.set_service_url('https://apikey-v2-
1oj043bu90m78ng4h2j27w5nob2nvcma6xanc6bk0a7m:daf3c00c2cc182af425a5691a07f7b93@932393aa
-9f82-4144-9251-2c519fb30962-bluemix.cloudantnosqldb.appdomain.cloud')

```

```

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

```

```

while True:
    _, frame = cap.read()    decodedObjects = 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(a)    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()

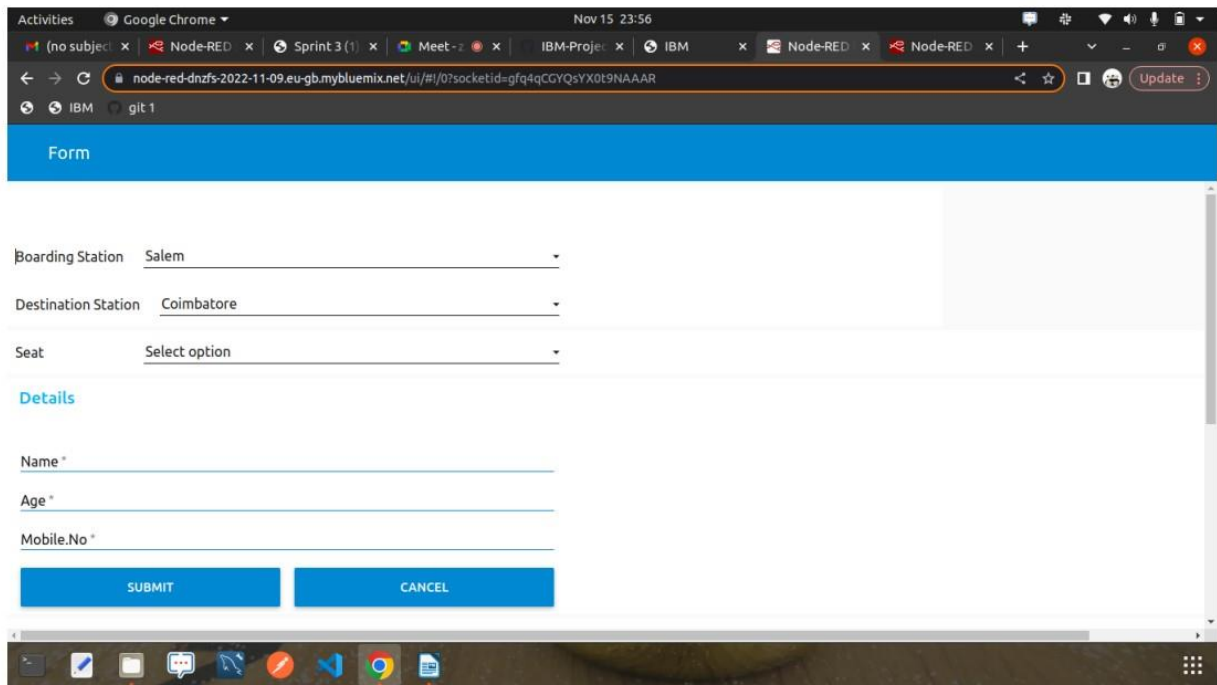
```



## 8 TESTING

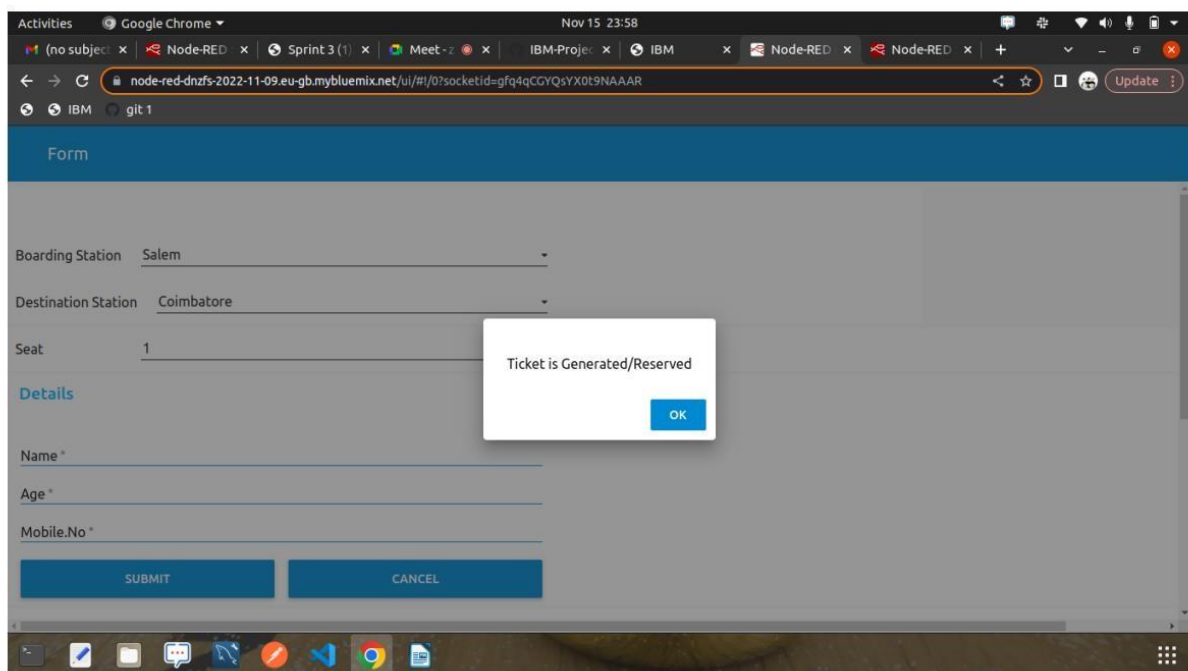
### 8.2 Test Cases

#### Checking the web UI



A screenshot of a Google Chrome browser window displaying a flight booking form. The browser's address bar shows the URL: `node-red-dnzs-2022-11-09.eu-gb.mybluemix.net/ui/#/0?socketid=gfq4qCGYQsYX0t9NAAAR`. The form has a blue header bar with the word "Form". Below the header, there are three dropdown menus: "Boarding Station" with "Salem" selected, "Destination Station" with "Coimbatore" selected, and "Seat" with "Select option" selected. Below these is a section titled "Details" in blue. It contains three input fields: "Name \*", "Age \*", and "Mobile.No \*". At the bottom of the form are two blue buttons: "SUBMIT" and "CANCEL". The browser's taskbar at the bottom shows various application icons.

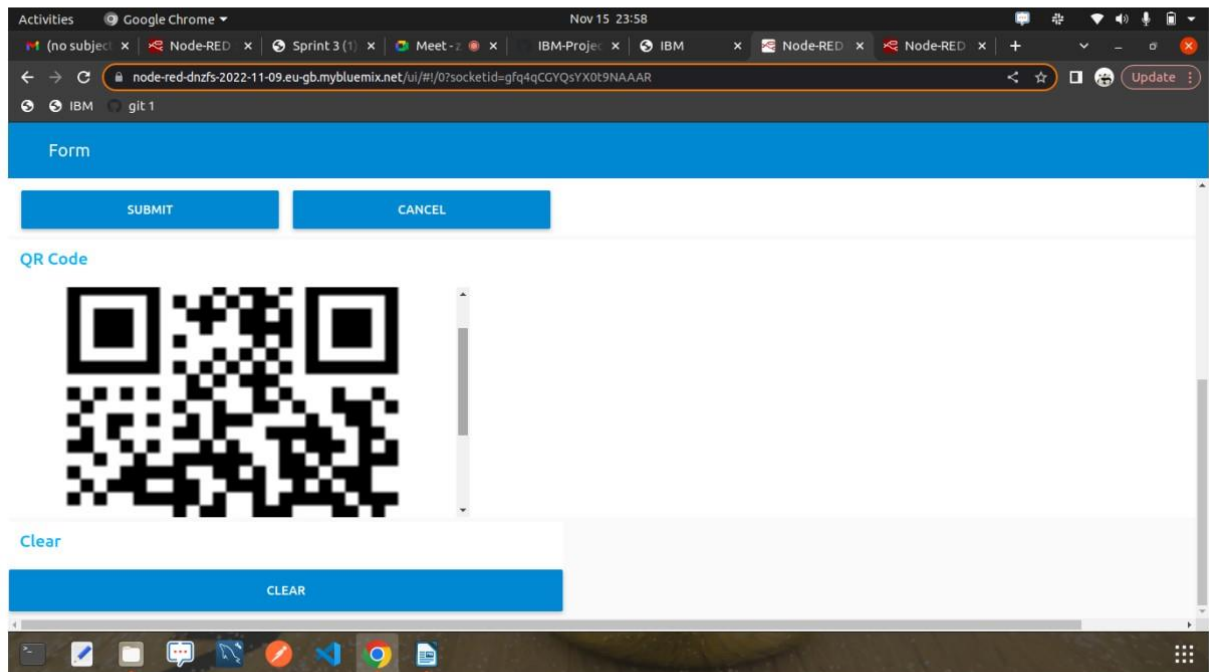
#### Testing for ticket registration



A screenshot of the same Google Chrome browser window as above, but with a confirmation dialog box overlaid in the center. The dialog box is white with a thin grey border and contains the text "Ticket is Generated/Reserved" in black. At the bottom right of the dialog box is a blue button with the text "OK" in white. The background form is dimmed. The browser's address bar and taskbar are visible, matching the previous screenshot.

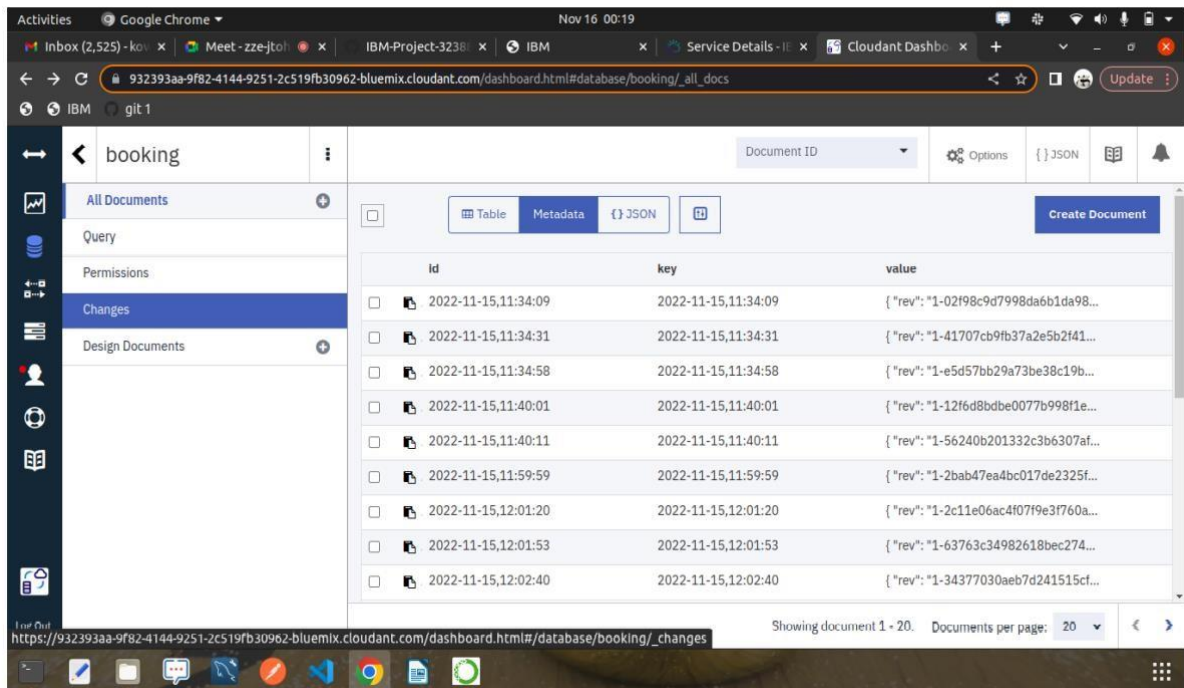
## 8.3 User Acceptance Testing

### Testing for QR code generation



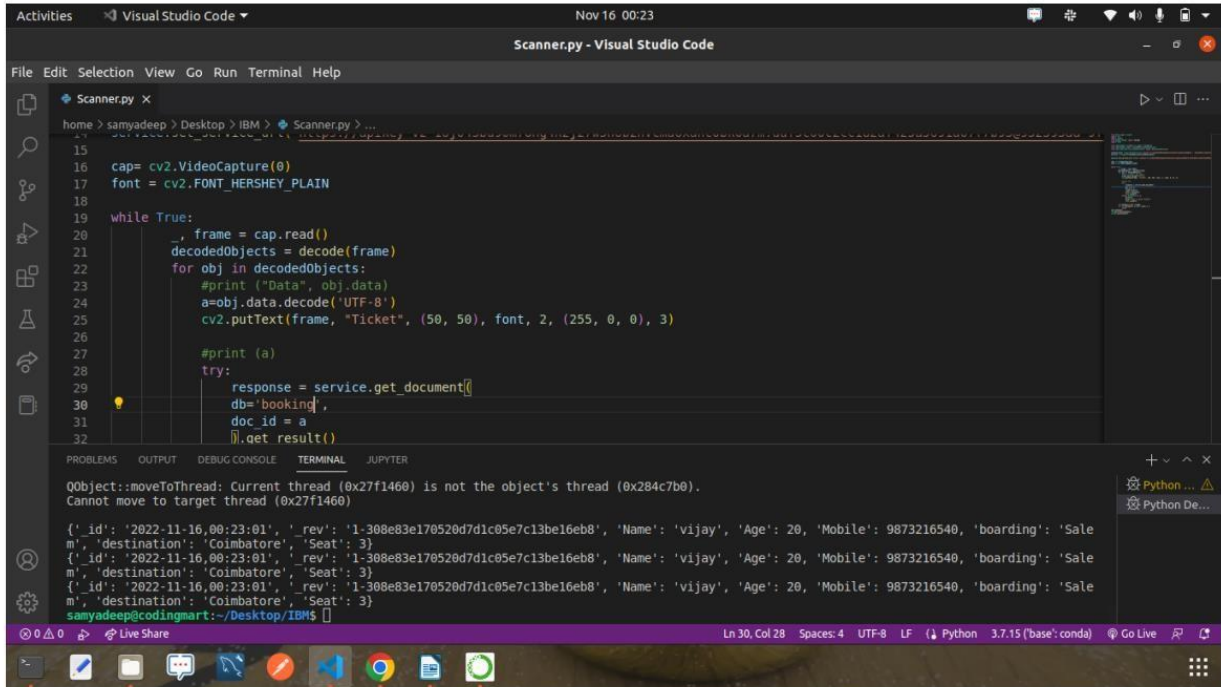
Generated successfully-test case passed

### Testing for Storing in database



## 9 RESULTS

### 9.2 Performance Metrics

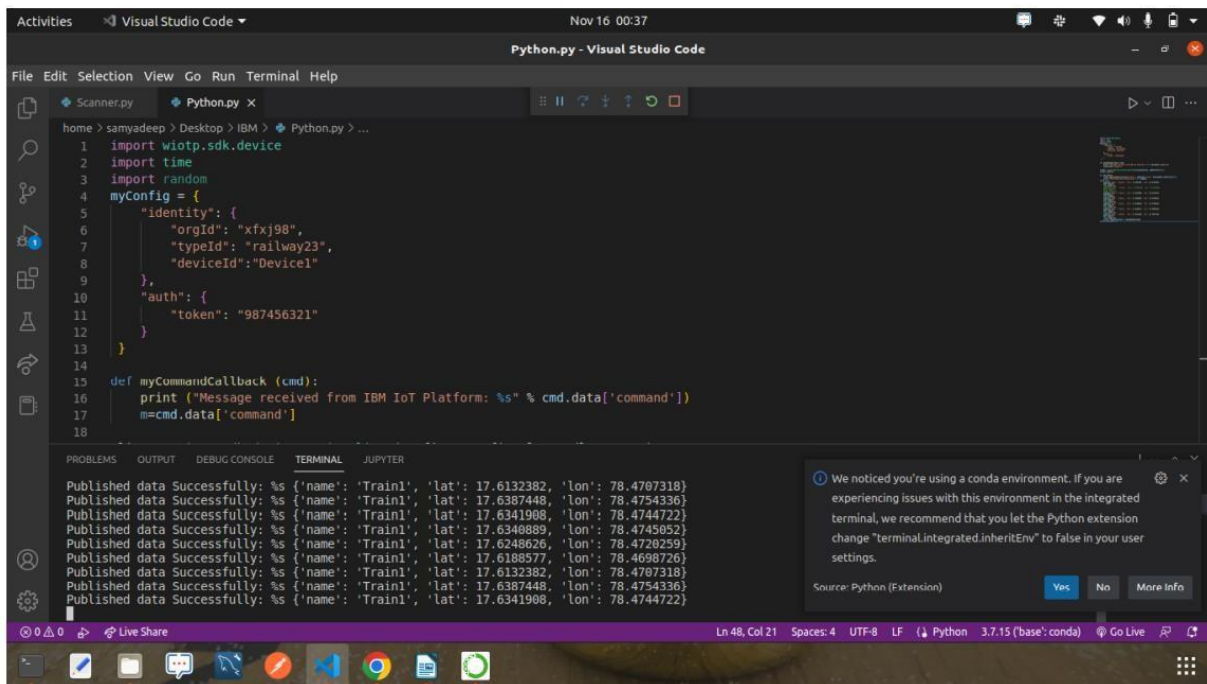


The screenshot shows the Visual Studio Code editor with the file `Scanner.py` open. The code is a Python script that uses OpenCV to capture video from a camera and processes the frames to extract data. The terminal output shows the script running successfully, displaying the extracted data for each frame.

```
15
16 cap = cv2.VideoCapture(0)
17 font = cv2.FONT_HERSHEY_PLAIN
18
19 while True:
20     frame = cap.read()
21     decodedObjects = decode(frame)
22     for obj in decodedObjects:
23         #print ("Data", obj.data)
24         a=obj.data.decode('UTF-8')
25         cv2.putText(frame, "Ticket", (50, 50), font, 2, (255, 0, 0), 3)
26
27         #print (a)
28         try:
29             response = service.get_document([
30                 db='booking',
31                 doc_id = a
32             ]).get_result()
```

Terminal Output:

```
QObject::moveToThread: Current thread (0x27f1460) is not the object's thread (0x284c7b0).
Cannot move to target thread (0x27f1460)
{'_id': '2022-11-16,00:23:01', 'rev': '1-308e83e170520d7d1c05e7c13be16eb8', 'Name': 'vijay', 'Age': 20, 'Mobile': 9873216540, 'boarding': 'Sale
m', 'destination': 'Coimbatore', 'Seat': 3}
{'_id': '2022-11-16,00:23:01', 'rev': '1-308e83e170520d7d1c05e7c13be16eb8', 'Name': 'vijay', 'Age': 20, 'Mobile': 9873216540, 'boarding': 'Sale
m', 'destination': 'Coimbatore', 'Seat': 3}
{'_id': '2022-11-16,00:23:01', 'rev': '1-308e83e170520d7d1c05e7c13be16eb8', 'Name': 'vijay', 'Age': 20, 'Mobile': 9873216540, 'boarding': 'Sale
m', 'destination': 'Coimbatore', 'Seat': 3}
samyadeep@codingmart:~/Desktop/IBM$
```



The screenshot shows the Visual Studio Code editor with the file `Python.py` open. The code is a Python script that uses the `wiotp.sdk.device` module to connect to the IBM IoT Platform and publish data. The terminal output shows the script running successfully, displaying the published data for each command.

```
1 import wiotp.sdk.device
2 import time
3 import random
4 myConfig = {
5     "identity": {
6         "orgId": "xfxj98",
7         "typeId": "railway23",
8         "deviceId": "Device1"
9     },
10     "auth": {
11         "token": "987456321"
12     }
13 }
14
15 def myCommandCallback(cmd):
16     print ("Message received from IBM IoT Platform: %s" % cmd.data['command'])
17     m=cmd.data['command']
18
```

Terminal Output:

```
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6348889, 'lon': 78.4745052}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720299}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6188577, 'lon': 78.4698726}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6132382, 'lon': 78.4707318}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6387448, 'lon': 78.4754336}
Published data Successfully: %s {'name': 'Train1', 'lat': 17.6341908, 'lon': 78.4744722}
```

## **10 CONCLUSION**

Using the Web application, a user books a ticket based on the availability of the seats by giving the general required information. Once a user clicks on the submit button, a QR code is generated with a Unique ID and the data is stored in the Cloudant DB with that Unique ID. Users can save the QR code for further process. In python code, a Ticket collector can scan the QR code and extract the information from the QR Code i.e., Unique ID. With that Unique ID, data is fetched from the Cloudant DB, if it is not found, then it displays Not a Valid Ticket. Also, the live location of the train will be published to IBM IoT platform using python code . The train location can be tracked from a Web Application.

## **11 FUTURE SCOPE**

Improving and increasing customer experience, vehicle tracking system. IOT is used along with AI which Provides enhanced features in finding out delays. Predicting delay and detecting the train arrival time so that help the passenger to act accordingly and keep tracking the location of the train and travel in easy and modern way. To maintain the tracks, repairs and services to avoid accident, safeguard of things, track the running status of the train in smart railway system and reach the destination place on time due to train delay.