

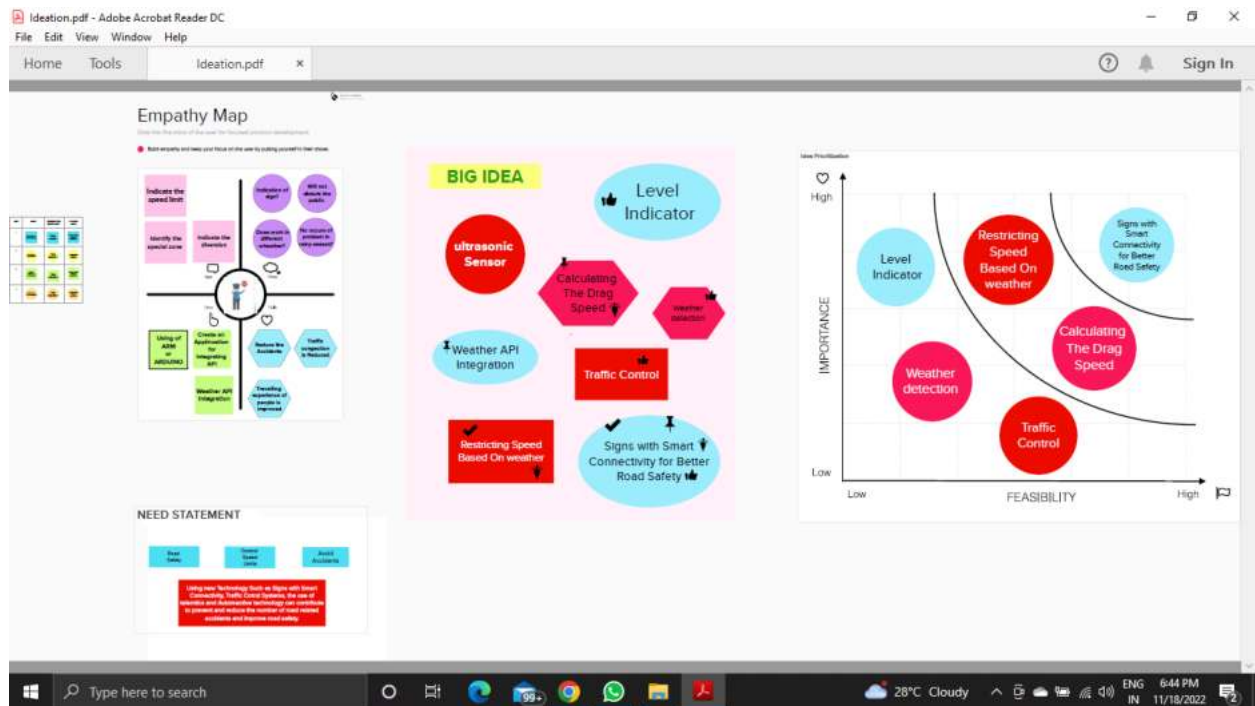
Signs with Smart Connectivity for Better Road Safety

Team ID	PNT2022TMID11067
Team Leader	Sakthivel (811519106119)
Team Member	Kamalesh (811519106063)
Team Member	Parthiban (811519106099)
Team Member	Rakhul Raghav (811519106110)

PROJECT OBJECTIVE:

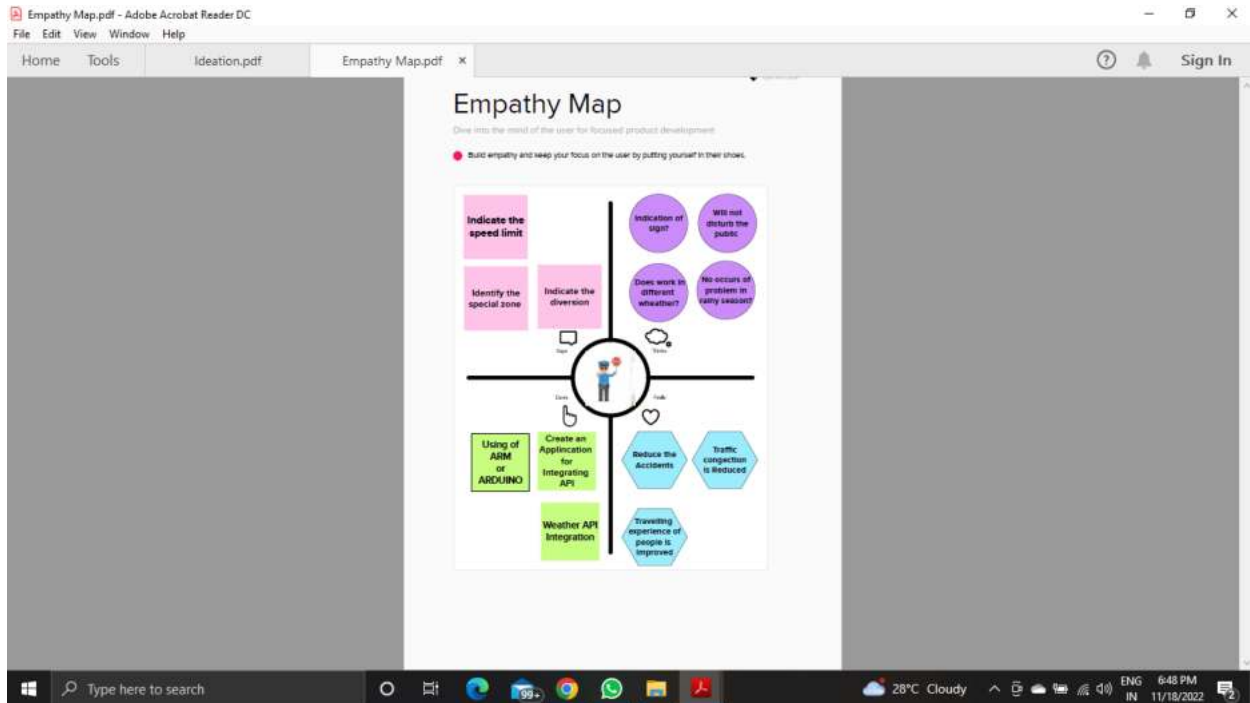
To replace the static signboards, smart connected sign boards are used. These smart connected signboards get the weather data from a web app using weather API and update automatically. Based on the weather changes the speed limit displayed may increase or decrease. Based on the traffic and fatal situations the diversion signs are displayed. Schools, hospitals, warning signs are also displayed accordingly. We make use of random values in python for sensor data as physical hardware is not used.

IDEATION:



EMPATHY MAP:

Signs with Smart Connectivity for Better Road Safety



LITERATURE SURVEY:

Kailas Shinde Published on “International Conference on Communication & Information Processing (ICCIP) 2021”. In modernizing countries, accidents are the main cause of death. If we tend to notice the dangerous roads in the world, they are all made up of mountain roads and winding roads. Intensity of the additional death zone unit in curvilinear curves. On mountain roads, there will be narrow roads, with sharp turns. In such forms, the dynamics of a vehicle cannot see the cars coming back from the alternative perspective. As a result of this inconvenience, thousands of people lose their lives every year. While we tend to talk about mountain roads here, another aspect could be due to the drop. The answer to this inconvenience is to warn the driver regarding the vehicle turning from another angle. One of the answers is provided in this article. We warn the driver by inserting an inaudible sensor element into one aspect of the road ahead of the turn and keeping the junction light on the other side of the turn, so if the vehicle is moving coming from the bend, element detection can detect the light weight of the car rectifier and the bright red junction in other respects. Looking at the red junction rectifier, light drivers will become wary and may impede the vehicle's speed. However, if a related accident occurs, we will save the victim's life by providing medical assistance like a shot. This will increase the chances of survival for the victims. Srimantini Bhattacharya Published on “Interdisciplinary Research in Technology and Management (IRTM) 2022”. This article explores the advancements of the Internet of Things (IoT) and machine learning in the field of road traffic safety and accident prevention with a status quo assessment of various techniques applied to implementing the system. Intelligent road safety system. In this review, the focus is on driver behaviour, vehicle health (two-wheelers, fourwheelers), road health, and theft-related issues using frequency recognition. radio number (RFID). It can be seen that with the help of IoT, the safety system can be updated in real time, which can help to create a highly efficient, intelligent and intelligent road safety system. Artificial intelligence (AI) is applied to further enhance the technology that detects driver behaviour, such as drowsiness, using real-time camera feeds or high-resolution images high. In addition, the role of AI in detecting the condition of roads and bridges in preventing road accidents is also discussed. Although the paper gives a good overview of the application of IoT and machine learning in intelligent road traffic safety system, some limitations are still highlighted. Ashish Sharma Published on “IEEE India Council International Subsections Conference (INDISCON) 2020”. There are more cars on the road today. As a result, managing traffic bottlenecks and accidents on

the roads around the world is difficult for society. The performance of the total road safety management system can be greatly enhanced by artificial intelligence (AI) technologies like machine learning (ML) algorithms. Numerous practical applications of AI are employed to make any system intelligent. The Smart Road Traffic Management System (SRTMS) recognises the impact that unplanned modifications have on traffic safety with ease. The SRTMS not only notifies the appropriate authorities but also recognises risky driving behaviours. Real-time monitoring of human activity is made possible via the Internet of Things (IoT). Sensors are a frequent component of Internet of Things (IoT) devices and nodes and are used to recognise and respond to electrical and other signals. The most popular technology today for automating transactions, or the sharing or exchange of information between IoT devices or nodes, is Blockchain (BC). Information sharing on the network is made possible by BC technology in a decentralised, secure, persistent, anonymous, appropriate, and reliable way. Blockchain aims to coordinate communication between nodes without the help of a third party or intermediary organisation thanks to consensus algorithms and smart contracts. AI has the potential to develop robots that are both intelligent and capable of making decisions, much like human minds. In order to address traffic congestion, road accidents, and information dissemination to all stakeholders, this article suggests the SRTMS paradigm. This recommended model. W. H. D. Fernando Published on "International Research Conference on Smart Computing and Systems Engineering (SCSE) 2021". Sensors are a frequent component of Internet of Things (IoT) devices and nodes and are used to recognise and respond to electrical and other signals. The most popular technology today for automating transactions, or the sharing or exchange of information between IoT devices or nodes, is Blockchain (BC). Information sharing on the network is made possible by BC technology in a decentralised, secure, persistent, anonymous, appropriate, and reliable way. Blockchain aims to coordinate communication between nodes without the help of a third party or intermediary organisation thanks to consensus algorithms and smart contracts. AI has the potential to develop robots that are both intelligent and capable of making decisions, much like human minds. In order to address traffic congestion, road accidents, and information dissemination to all stakeholders, this article suggests the SRTMS paradigm. This recommended model in intelligent transportation systems, but challenging procedure. To address the significant problems they encounter, numerous initiatives have been made. Using a method that first detects a traffic sign's bounding box, the goal of this work is to address the detection and recognition of road traffic signs. When a traffic sign is noticed, it will then be quickly recognised and put to use. The YOLOv4 network was used in this study because safe driving requires the real-time monitoring of traffic signs. On our dataset, which was composed of 43 distinct traffic sign classes identified through manual annotations, YOLOv4 was assessed. Averaging 84.7% accuracy, it was able to recognise objects. A simple yet successful model for the real-time detection and identification of traffic is presented in the paper, adding value overall. Sharmila Published on "International Conference on Innovative Trends in Information Technology (ICITIIT) 2022". Road safety is the biggest issue today, with thousands killed and injured on the roads due to fatigue and drunk driving. To prevent and reduce these traffic accidents, simple sensors are used in vehicles to perform various functions, such as horn control and cruise control, to monitor and control the speed of the vehicle in various places (such as bridges, overpasses, roads and schools). The vehicle is checked at traffic lights and when the signal is red, the vehicle stops automatically. The RF transmitter includes four buttons, such as no horn, speed control, green signal and no parking. This RF transmitter is located on the signal board, which sends the signal to the RF receiver connected to the Node MCU. The LCD displays messages by pressing the desired button on the transmitter. M. Divyaprabha presented at "IEEE Computational Intelligence and International Conference on Computational Research (ICCIC) 2018". Transport systems must take road safety into account. Road safety problems are the cause of many accidents. Road accidents are becoming more common in India due to the rapid increase in the number of cars. One of the main factors contributing to road safety problems is the growth of traffic congestion which is increasing at a rate of 7-10% per year, compared to 12% per year for vehicles. Globally, traffic accidents cause 50 million injuries and approximately 1 million deaths each year. 70,000 people die in India every year, so it needs to be addressed to save lives. This document is based on, road safety issues. The comparison identified the root cause of, major accidents. Xu Yiming presented at "IEEE

International Smart Cloud Conference (SmartCloud) 2019". The World Health Organization has identified, traffic accidents as a social and public health problem. To solve this problem, Intelligent Transport System has been launched which is completely based on advanced ICT. However, the increase in road deaths requires the use of technology to predict future interactions and behaviors. To find solutions to detect human activity and road interactions in real time, this paper explores modern technologies such as computer vision and artificial intelligence. Vasireddy Satish Published on "International Conference on Electronics, Communication and Aerospace Technology (ICECA) 2021". Population growth in countries like India is very fast, which leads to an increase in urban traffic. The World Health Organization (WHO) has published its global accident report for 2021 due to accidents. According to reports, between 50% and 60% of accidents result in personal death. Many reasons point to the cause of the accident. In one case, accidental deaths occurred mainly among two-wheeler riders, resulting from head injuries. Another case proved it due to driver drowsiness. Alcohol consumption is also one of the main causes of accidents that can lead to the death of the driver. Research by the Highway Traffic Safety Administration shows that the majority of traffic deaths among young people are due to reckless driving or driving under the influence. In this work, an advanced failure management method is proposed to prevent accidents. The Internet of Things (IoT)-based system is designed to start the bike or scooter only when the rider is wearing a helmet and has passed a breathalyzer test. Another method has been added, in which the alert system is activated when the driver is speeding. Sensors and microcontrollers are used for the proposed work. Ali Raza Published on "IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) 2017". This position paper suggests a way to reduce traffic accidents caused by mobile phone use. It features the use of recent advances in Internet of Things technology and the results of the famous Milgram experiment of 1963, which examined the struggle between submission to authority and personal awareness. In some countries, high-risk driving causes more than 75% of deaths due to driver behaviour. This article considers the use of cell phones to cause distraction as one of the main risks. Measures taken by road safety agencies include the use of high-tech cameras that can record the use of mobile devices. Cost constraints prevent agencies from providing full coverage for these devices. The position paper argues that the automotive and mobility industries should play a role in supporting road safety agencies. In addition, a new approach to smart road safety using IoT, geolocation and geolocation is introduced, which discourages the use of mobile phones and lays the foundation for future research. This article highlights recent research and technological developments that contribute to some aspects of the solution. Finally, the approach proposes new standards established between vehicle and mobile device manufacturers to enable smarter solutions. Dasari Vishal Published on "International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) 2017". Roads are an integral part of human civilization. They are the nerve system of any country, therefore they are placed on narrow slopes and hills, which poses a great threat to human life. Since roads play a vital role in our daily lives, they can be intelligently designed to provide us with enhanced capabilities. The architecture of the Internet of Things includes the ability to make things more connected and efficient. This paper synchronizes the concept of the Internet of Things with how to make it smart. The paper discusses the use of IoT technology to reduce the risk of traffic collisions with the onset of smart cities. Because every vehicle is equipped with the Internet of Things and connected to the Internet, we have effective technology to guide emergency vehicles on the road in the shortest possible time. The IoT system combines simple and cost-effective antenna technology with a fully automated Internet platform. Aichi Prefectural Published on "IEEE World Forum on Internet of Things (WF-IoT) 2014". In Japan, comprehensive measures to reduce traffic accidents have reduced traffic casualties for 12 years. Efforts include improving vehicle safety performance in both the passive and preventive safety domains. In terms of passive safety, airbag systems, seatbelts and vehicle crash safety have made significant savings. Preventive safety can play a more important role in further reducing traffic accidents. In recent years, driving support safety systems that adapt to the physical and mental conditions of drivers have been attracting attention in order to reduce traffic accidents. Therefore, it is necessary to clarify the effect of

Signs with Smart Connectivity for Better Road Safety

reducing the mind-body adaptive safety function and promote market penetration. Statistical analysis of traffic accidents is expected to assess the effects of traffic accidents on mind-body adaptive safety functions. To overcome this challenge, this study introduced an Internet survey by sending questionnaires to respondents. From the analysis of the collected responses, the main mental and physical condition of the driver seems to be impatience and distraction. As a first step, this study focused on driver distractions that can lead to serious traffic accidents. We used pattern recognition to measure the accuracy of driver distraction detection. The reduction effect of driver distraction in traffic accidents was estimated using the reduction ratios of both ASV (Advanced Safety Vehicle) and Intelligent Transportation Systems.

PROJECT DESIGN PHASE-I

PROBLEM SOLUTION FIT:

Prepare Solution Fit.pdf - Adobe Acrobat Reader DC

File Edit View Window Help

Home Tools Ideation.pdf Empathy Map.pdf Prepare Solution Fit...

Problem-Solution fit canvas 2.0

IoT Based Signs With Smart Connectivity For Better Road Safety

1. CUSTOMER SEGMENT(S) Who is your customer? • Department Of Transportation • Rider	6. CUSTOMER CONSTRAINTS What constraints prevent your customers from taking actions or limit their choices of solutions? The effect of the network on the experiments was a significant and unexpected element. Considering the quantity of sensors, this IoT-based system was successful in simulating a large-scale smart agricultural setting.	5. AVAILABLE SOLUTIONS Which solutions are available to the customers when they face the problem/need to get the job done? What have they tried in the past? What pros & cons do these solutions have? Static signs with specified instructions are posted along highways as possible solutions.
2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? The Smartboard Connectivity has a variety of responsibilities, including maintaining accurate temperature sensor readings and notifying the board of the customer's car's speed.	9. PROBLEM ROOT CAUSE What is the real reason that this problem exists? What is the back story behind the need to do this job? Without an internet connection, the speed restriction would not be affected by weather sensor readings. Certain people could accidentally hit the accident indicator button, which could lead to some problems.	7. BEHAVIOUR What does your customer do to address the problem/sadjet the job done? The IoT cloud acts as a teacher, periodically informing the smartboard about the state of the roads.
3. TRIGGERS What triggers customers to act? i.e. seeing their neighbor installing Unfavorable weather is present. The car ought to be moving at threshold speed. Values should be presented on the smart board to alert customers utilising the sensors.	10. YOUR SOLUTION 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. Smart connected sign boards are our replacement for static signboards. These intelligent connected sign boards update automatically and obtain the speed restrictions from a web application utilising weather API. The speed may rise or fall depending on weather changes. The display of the diversion signs depends on the flow of traffic and potential fatalities. The appropriate guide, warning, and service signs are also posted at hospitals and restaurants. With the use of buttons, many operating modes can be chosen.	8. CHANNELS OF BEHAVIOUR What kind of actions do customers take online? Customers can contact departments directly via email or messaging. (Patrol Officers in the Area) What kind of actions do customers take offline? Travelers may use the smartboard signs to check the state of the road from anywhere, and one of their biggest challenges is to follow directions.
4. EMOTIONS: BEFORE / AFTER How do customers feel when they face a problem or a job and afterward? After choosing an operation mode with the assistance of smartboard connectivity, clients will feel better and then follow the directions on the smartboard.		

PROPOSED SOLUTION TEMPLATE :

Signs with Smart Connectivity for Better Road Safety

FILE TOOLS VIEW Proposed Solution [Read-Only] - Word (Product Activation Failed)

Project Design Phase-I Proposed Solution Template

Date	30 September 2022
Team ID	PNT2022TMID11067
Project Name	Signs with smart connectivity for better road safety
Maximum Marks	Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	employing IOT to stop traffic accidents from happening.
2.	Idea / Solution description	By creating smart signs utilising IOT in place of conventional signs posted on the side of the road. Smart signs are constructed with IOT and LED.

3.	Novelty / Uniqueness	Temperature, humidity, and wind speed are displayed on the smart signs, which are visible from a distance thanks to the usage of LEDs. These details are obtained through a weather monitoring app, which also provides information about surrounding facilities like hospitals, schools, and other establishments so that users may adjust their speeding in accordance with that data.
4.	Social Impact / Customer Satisfaction	By setting a speed limit for the user, there is a good likelihood of lowering the accidents. These have a noticeable impact on the road safety department.
5.	Business Model (Revenue Model)	It is a terrific initiative to raise awareness among the populace for the government to carry out things for commoners. This opens the door for a safer environment since the government can set aside a distinct budget for it.
6.	Scalability of the Solution	It is a terrific initiative to raise awareness among the populace for the government to carry out things for commoners. This opens the door for a safer environment since the government can set aside a distinct budget for it..

End of document

Windows taskbar showing search bar, task view, and various application icons (Edge, File Explorer, WhatsApp, etc.). System tray shows weather (28°C Cloudy), language (ENG IN), and time (7:05 PM 11/18/2022).

SYSTEM ARCHITECTURE:

FILE TOOLS VIEW Solution Architecture - Word (Product Activation Failed)

```

graph LR
    A[Problem Statement] --> B[Idea/Solution description]
    B --> C[Novelty/ Uniqueness]
    C --> D[Social Impact/ Customer Satisfaction]
    D --> E[Business Model]
    E --> F[Scalability of the Solution]
  
```

```

graph LR
    subgraph Cloud_Services [Cloud Services]
        direction LR
        IoT[IoT Device] --> Python[Python Code random data]
        Python --> Watson[IBM Watson IoT Platform]
        Watson --> NodeRED[Node-RED]
        NodeRED --> WebUI[Web UI]
        WebUI --> User((User))
        Openweather[OpenweatherMap] --> Watson
    end
  
```

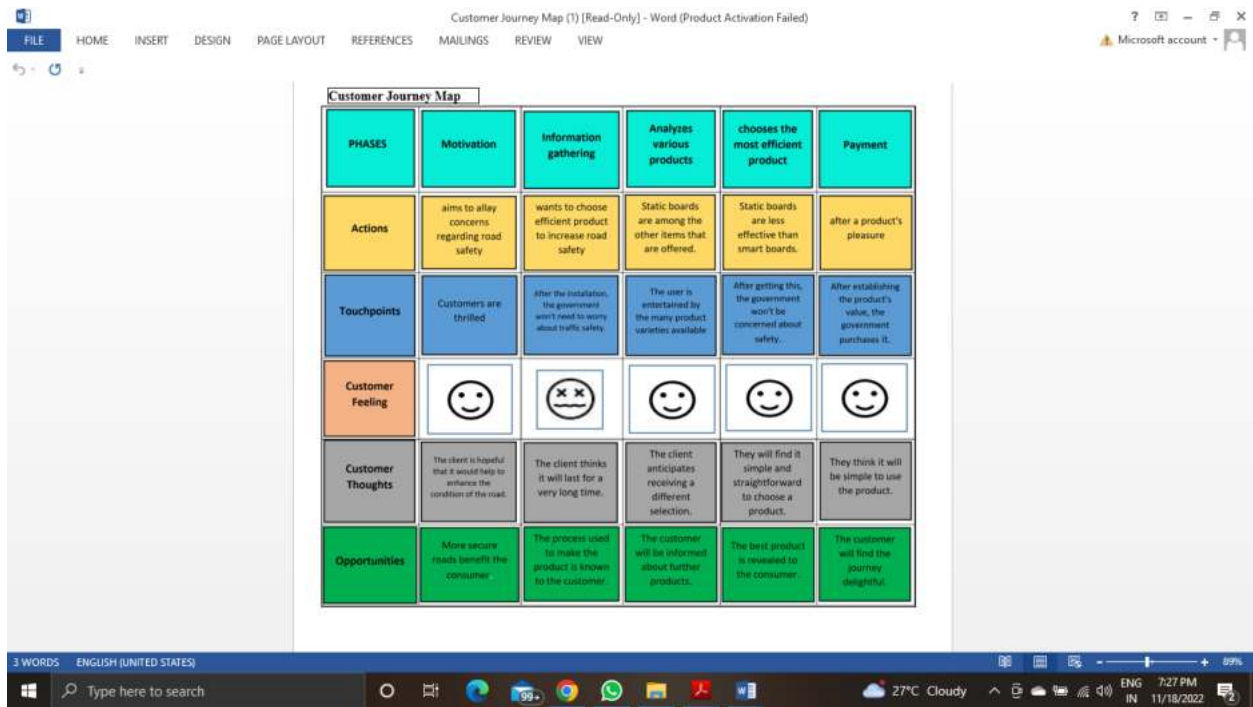
End of document

Windows taskbar showing search bar, task view, and various application icons (Edge, File Explorer, WhatsApp, etc.). System tray shows weather (27°C Cloudy), language (ENG IN), and time (7:19 PM 11/18/2022).

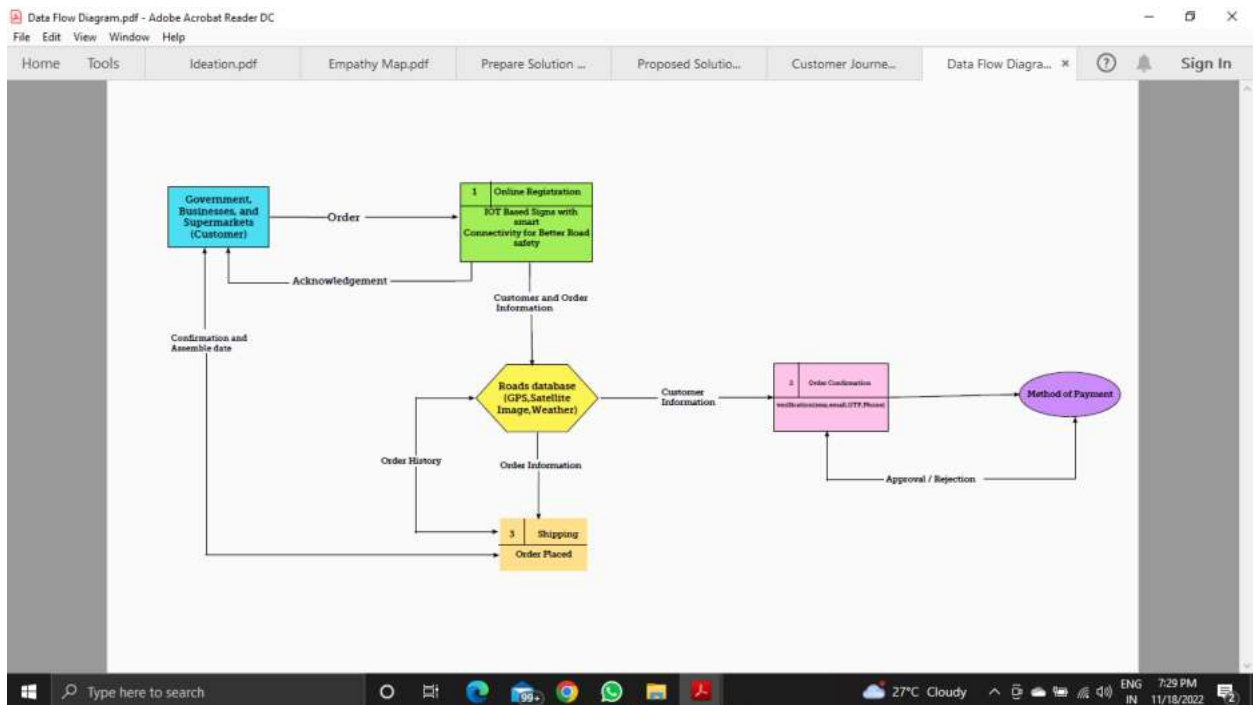
PROJECT DESIGN PHASE-II

CUSTOMER JOURNEY MAP

Signs with Smart Connectivity for Better Road Safety



DATA FLOW DIAGRAM:



FUNCTIONAL REQUIREMENT:

Signs with Smart Connectivity for Better Road Safety

Functional Requirement - Word (Product Activation Failed)

FILE TOOLS VIEW

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	12 October 2022
Team ID	PNT2022TMD11067
Project Name	Signs With Smart Connectivity for Better Road Safety
Maximum Marks	4 Marks

Functional Requirements:
Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Requirements	Smart linked sign boards that meet all criteria will replace static signboards.
FR-2	User Registration	Manual Sign-Up using a Website or Gmail
FR-3	User Confirmation	OTP authentication through phone confirmation Email confirmation
FR-4	Payments options	Transfer via Bank
FR-5	Product Delivery and installation	The length of the road will decide the installation fee.
FR-6	Product Feedback	Using a website and Gmail

Non-functional Requirements:
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Have clear product instructions and a product that is self-explanatory and easy to use.
NFR-2	Security	Cloud data must be contained within the network, eventually collapsing. Avoiding decisions in real time should be avoided, and the board should be regularly observed.
NFR-3	Reliability	Hardware is thoroughly tested.
NFR-4	Performance	The smart board must improve the user experience and give accurate output.
NFR-5	Availability	Depending on the needs of the consumer, all of the functions that the user need will be offered.
NFR-6	Scalability	The device is designed to improve traffic safety and should cover the entire highway system.

End of document

TECHNICAL ARCHITECTURE :

Technical Architecture Diagram - Word (Product Activation Failed)

FILE HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW

Technical Architecture Diagram:

```
graph LR; Field[Field] --> IoT[IoT Device]; IoT --> Python[Python Code (random data)]; Python --> Watson[IBM Watson IoT Platform]; Watson --> NodeRED[Node-RED]; NodeRED --> WebUI[Web UI]; WebUI --> User((User)); User --> PassengerUser[Passenger User]; OpenWeatherMap[OpenweatherMap] --> IoT;
```

Architecture and data flow of the IOT Based Signs With Smart Connectivity for Better Road Safety

22 WORDS ENGLISH (UNITED STATES)

PROJECT PLANNING PHASE:

Signs with Smart Connectivity for Better Road Safety

FILE TOOLS VIEW

Milestone and Activity List - Word (Product Activation Failed)

🔍 - 📄 ✕

Project Planning Phase(Milestoneand Activity List)

Date	09 November 2022
Team ID	PNT2022TH011067
Project Name	Signs with Smart Connectivity for Better Road Safety
Maximum Marks	4 Marks

MILESTONE AND ACTIVITY LIST (4 Marks)

Sprint	Functional Requirement (Rqts)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	SR	US00-1	Implement the software which is required to capture DS	2	High	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-1	Checking the condition with conditions	US00-1	Formulating the circuits and experimenting	2	High	Subhrovo Parthiban Kamalash Rakhal Raghav

Sprint-2	Software	US00-2	SRF Station Set -Roads Road Integration	2	High	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-2	Software	US00-2	Test the document workflow	2	High	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-3	Application Development	US00-3	Using RPT App- Incentivize as an App	2	High	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-3	Testing	US00-3	Testing the application	2	High	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-4	WEB UI	US00-4	User interface with the software	2	High	Subhrovo Parthiban Kamalash Rakhal Raghav

End of document

SPRINT PLANING:

FILE TOOLS VIEW

Sprint Delivery Plan (Read-Only) - Word (Product Activation Failed)

🔍 - 📄 ✕

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	9 November 2022
Team ID	HW2022TH00000000
Project Name	Signs with Smart Connectivity for Better Road Safety
Maximum Marks	4 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint scheme

Sprint	Functional Requirement (Rqts)	User Story / Task	Story Points	Priority	Team Members
--------	-------------------------------	-------------------	--------------	----------	--------------

Sprint-1	Resources Initialization	Create and initialize accounts in various public APIs like Open Weather Map API	1	LOW	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-1	Local Server/Software Run	Write a python program that makes local given the inputs like weather and location	1	MEDIUM	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-2	Push the server/software to cloud	Push the code from Sprint-1 to cloud so it can be accessed from anywhere	2	MEDIUM	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-3	Hardware Initialization	Integrate the hardware to be able to access the cloud functions and provide inputs to the same	2	HIGH	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-4	UI/UX Optimization & Debugging	Optimize all the shortcomings and provide better user experience	2	LOW	Subhrovo Parthiban Kamalash Rakhal Raghav

Project Tracker, Velocity & Burndown Chart: (4 Marks)

FILE TOOLS VIEW

Sprint Delivery Plan (Read-Only) - Word (Product Activation Failed)

🔍 - 📄 ✕

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	9 November 2022
Team ID	HW2022TH00000000
Project Name	Signs with Smart Connectivity for Better Road Safety
Maximum Marks	4 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint scheme

Sprint	Functional Requirement (Rqts)	User Story / Task	Story Points	Priority	Team Members
--------	-------------------------------	-------------------	--------------	----------	--------------

Sprint-1	Resources Initialization	Create and initialize accounts in various public APIs like Open Weather Map API	1	LOW	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-1	Local Server/Software Run	Write a python program that makes local given the inputs like weather and location	1	MEDIUM	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-2	Push the server/software to cloud	Push the code from Sprint-1 to cloud so it can be accessed from anywhere	2	MEDIUM	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-3	Hardware Initialization	Integrate the hardware to be able to access the cloud functions and provide inputs to the same	2	HIGH	Subhrovo Parthiban Kamalash Rakhal Raghav
Sprint-4	UI/UX Optimization & Debugging	Optimize all the shortcomings and provide better user experience	2	LOW	Subhrovo Parthiban Kamalash Rakhal Raghav

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Signs with Smart Connectivity for Better Road Safety

FILE TOOLS VIEW Sprint Delivery Plan [Read-Only] - Word (Product Activation Failed)

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	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	07 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \text{Sprint Duration} / \text{Velocity} = 20 / 6 = 3.33$$

**B
u
r
n
d
o
w
n
C
h
a
r
t
:**

FILE TOOLS VIEW Sprint Delivery Plan [Read-Only] - Word (Product Activation Failed)

Balance Work

End of document

PREREQUISITES:

IBM CLOUD SERVICES:

Signs with Smart Connectivity for Better Road Safety

Cloud Services - Word (Product Activation Failed)

FILE HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW

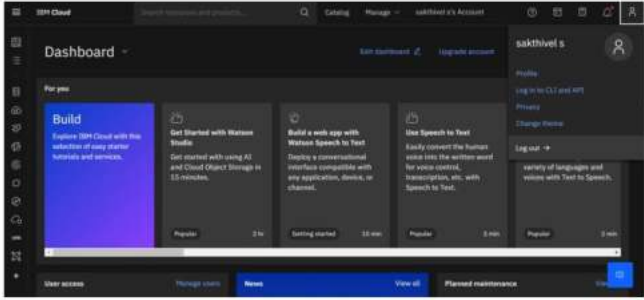
Microsoft account

Prerequisites

IBM Cloud Services

Date	09 November 2022
Team ID	PNT2022TMID11067
Project Name	Project - Signs with smart connectivity for Better road safety
Maximum Marks	4 Marks

Signs with smart connectivity for Better road safety



The screenshot shows the IBM Cloud dashboard with a dark theme. The top navigation bar includes 'Dashboard', 'Settings', 'Manage', and 'sakhilvel's Account'. The main content area is titled 'Dashboard' and features a 'For you' section with several cards: 'Build' (Explore IBM Cloud with this selection of easy starter templates and services), 'Get Started with Watson Studio' (Get started with using AI and Cloud Object Storage in 15 minutes), 'Build a web app with Watson Speech to Text' (Deploy a conversational interface compatible with any application, device, or channel), and 'Use Speech to Text' (Easily convert the human voice into the written word for voice control, transcription, etc., with Speech to Text). The bottom of the dashboard shows 'User access', 'Manage users', 'News', 'View all', and 'Planned maintenance'.

3 OF 35 WORDS ENGLISH (UNITED STATES)

Type here to search

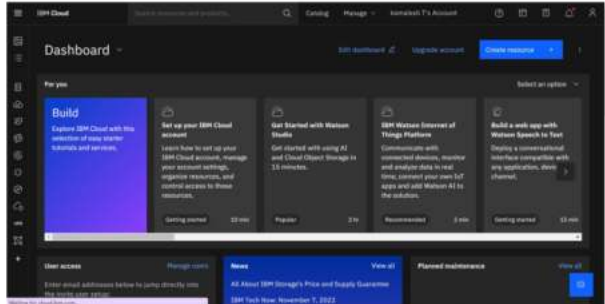
26°C Partly cloudy

ENG IN 8:03 PM 11/18/2022

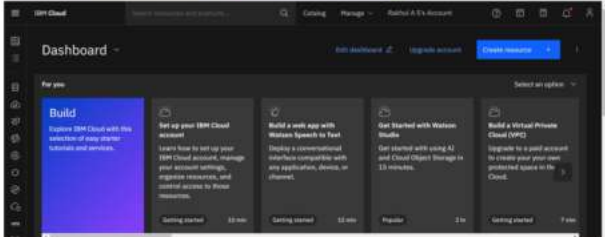
Cloud Services - Word (Product Activation Failed)

FILE HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW

Microsoft account



The screenshot shows the IBM Cloud dashboard with a dark theme. The top navigation bar includes 'Dashboard', 'Settings', 'Manage', and 'sakhilvel's Account'. The main content area is titled 'Dashboard' and features a 'For you' section with several cards: 'Build' (Explore IBM Cloud with this selection of easy starter templates and services), 'Set up your IBM Cloud account' (Learn how to set up your IBM Cloud account, manage your account settings, integrate resources, and control access to those resources), 'Get started with Watson Studio' (Get started with using AI and Cloud Object Storage in 15 minutes), 'IBM Watson Internet of Things Platform' (Communicate with connected devices, monitor and analyze data in real time, control your own IoT apps and add Watson AI to the solution), and 'Build a web app with Watson Speech to Text' (Deploy a conversational interface compatible with any application, device, or channel). The bottom of the dashboard shows 'User access', 'Manage users', 'News', 'View all', and 'Planned maintenance'.



The screenshot shows the IBM Cloud dashboard with a dark theme. The top navigation bar includes 'Dashboard', 'Settings', 'Manage', and 'sakhilvel's Account'. The main content area is titled 'Dashboard' and features a 'For you' section with several cards: 'Build' (Explore IBM Cloud with this selection of easy starter templates and services), 'Set up your IBM Cloud account' (Learn how to set up your IBM Cloud account, manage your account settings, integrate resources, and control access to those resources), 'Build a web app with Watson Speech to Text' (Deploy a conversational interface compatible with any application, device, or channel), 'Get started with Watson Studio' (Get started with using AI and Cloud Object Storage in 15 minutes), and 'Build a Watson Private Cloud (WPC)' (Upgrade to a paid account to create your own private cloud space in the Cloud). The bottom of the dashboard shows 'User access', 'Manage users', 'News', 'View all', and 'Planned maintenance'.

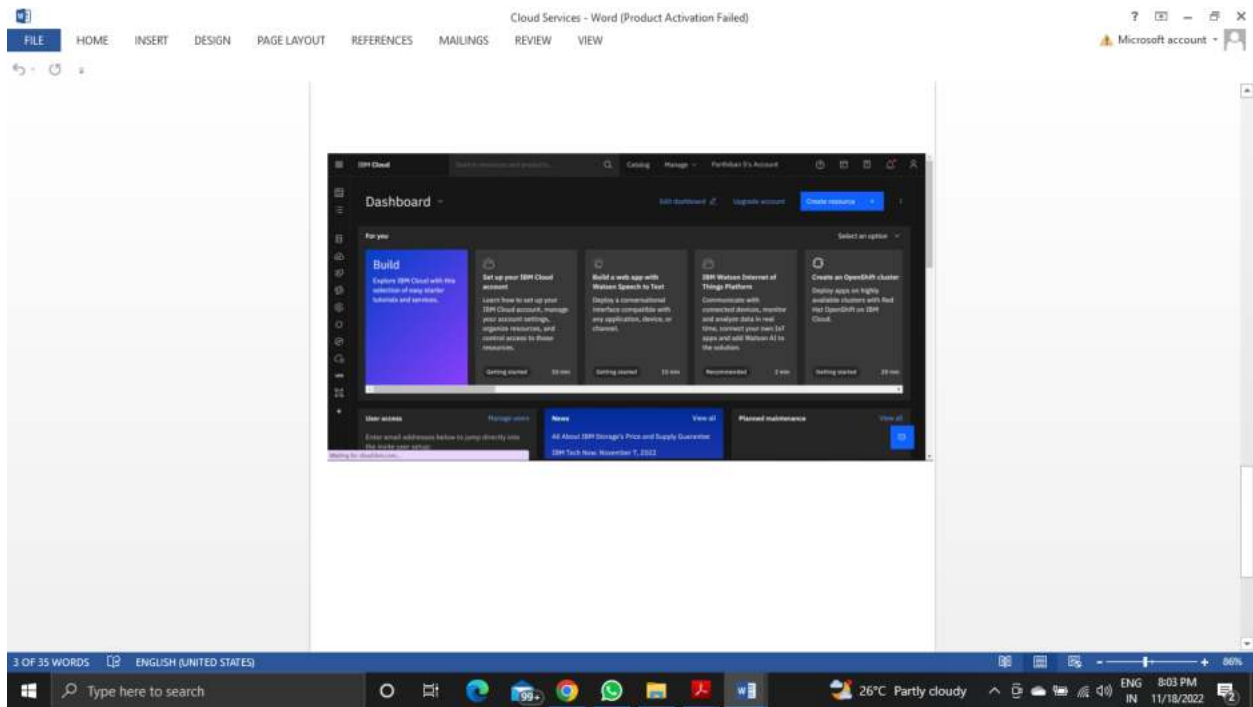
3 OF 35 WORDS ENGLISH (UNITED STATES)

Type here to search

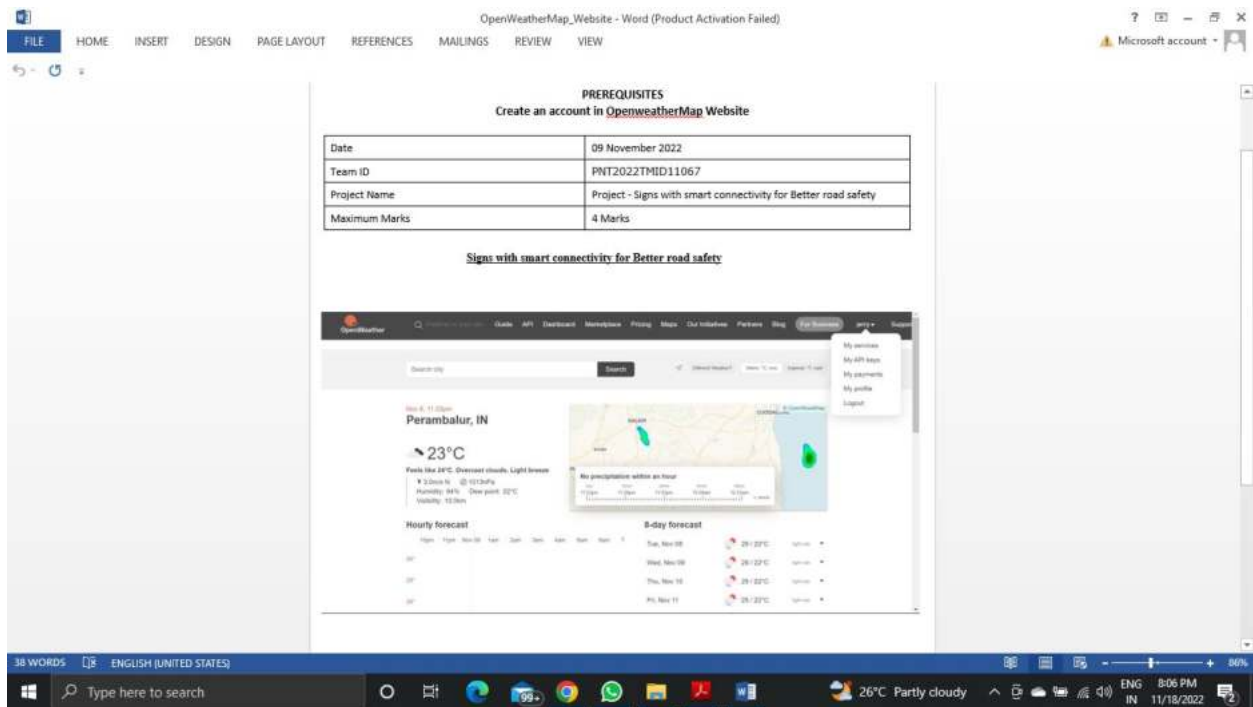
26°C Partly cloudy

ENG IN 8:03 PM 11/18/2022

Signs with Smart Connectivity for Better Road Safety

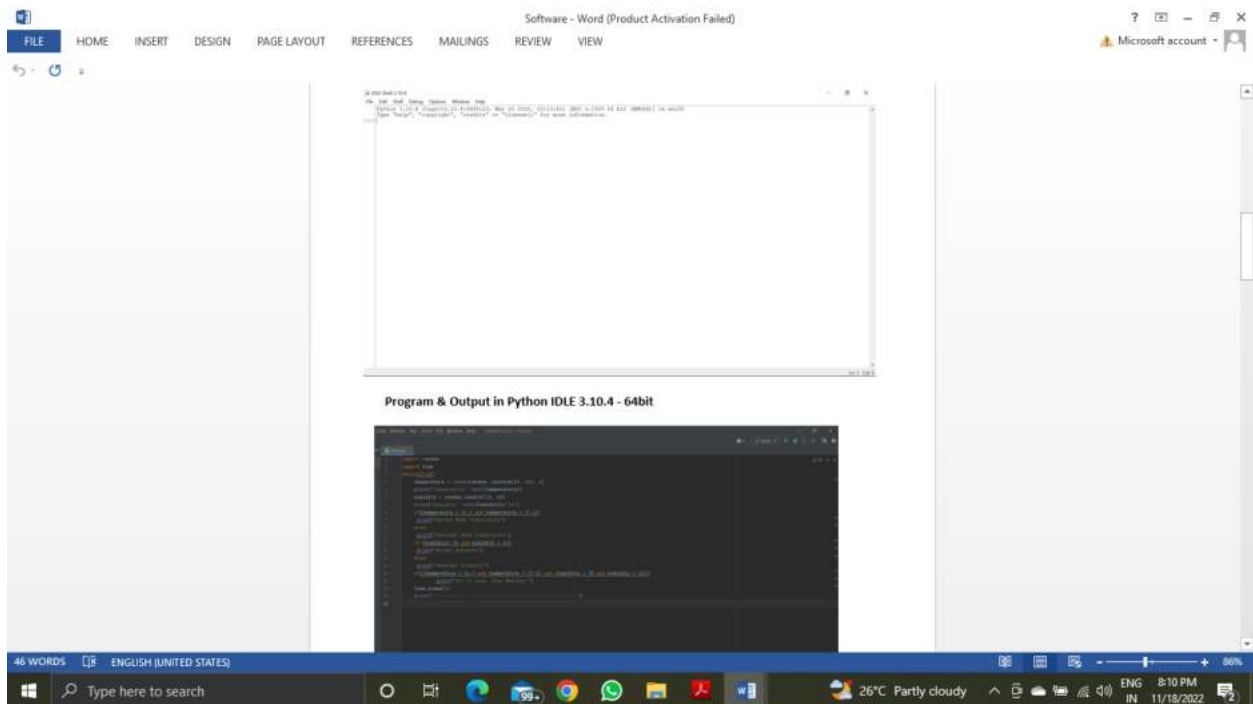
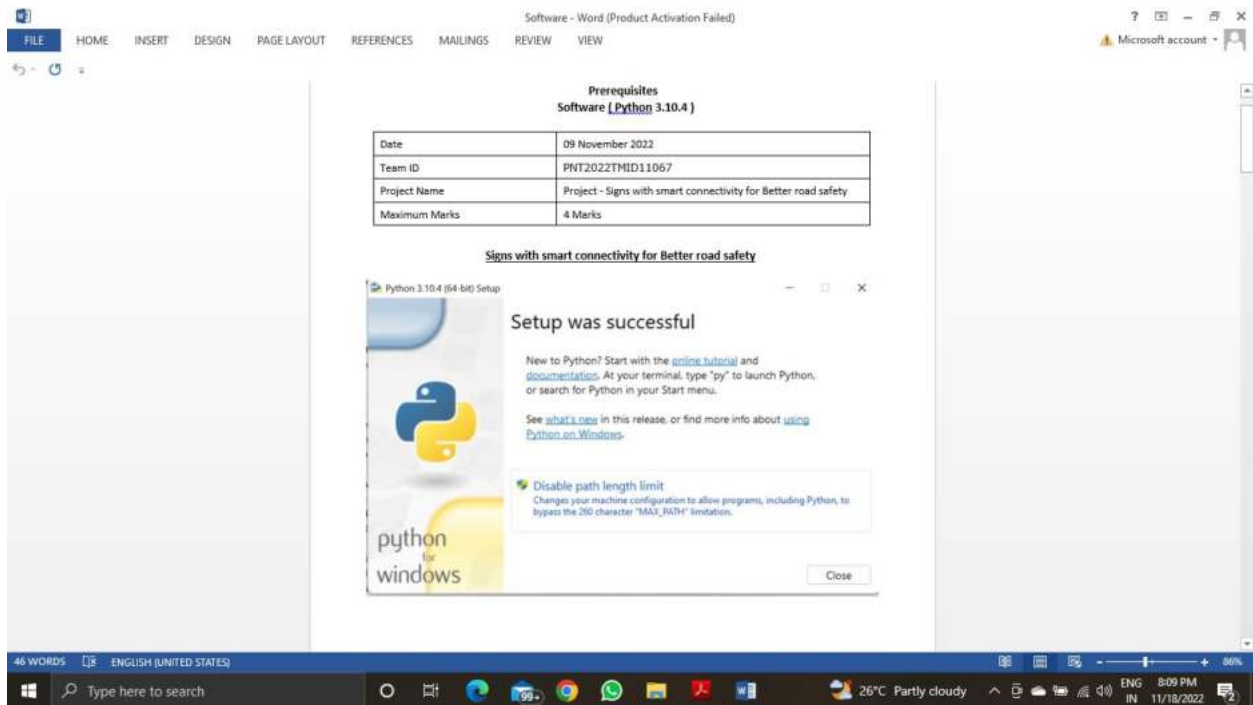


OPENWEATHERMAP WEBSITE :

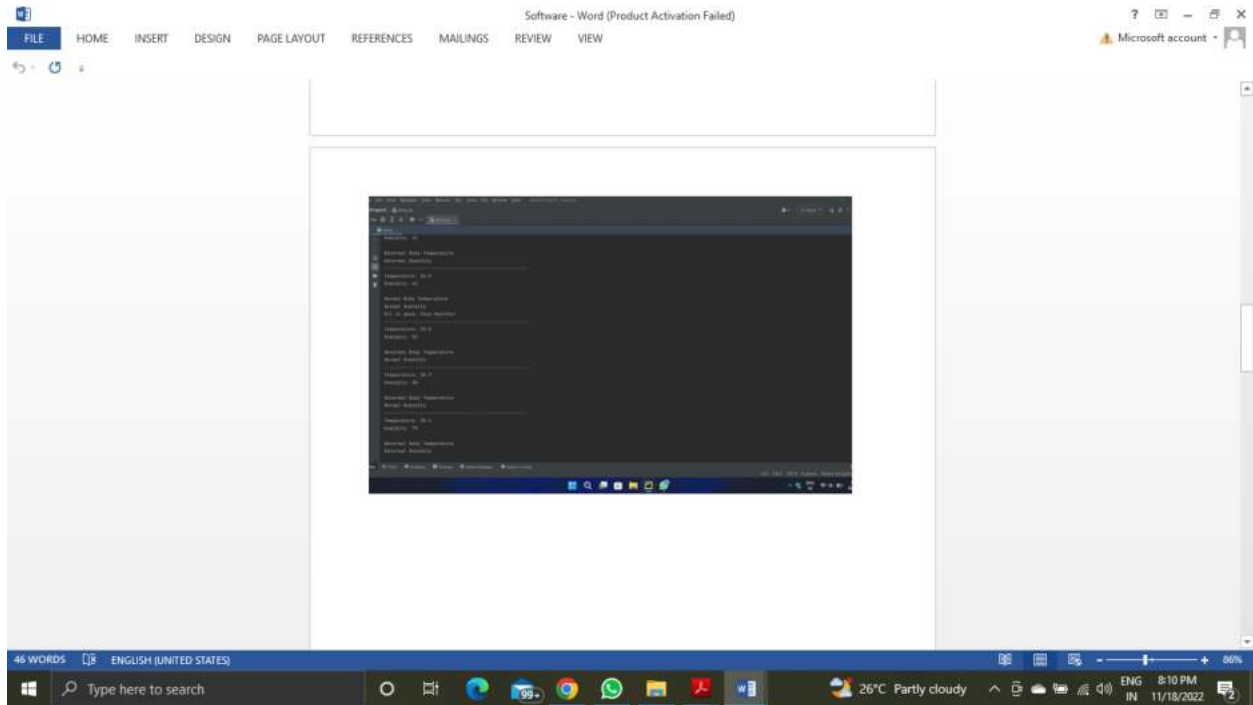


SOFTWARE:

Signs with Smart Connectivity for Better Road Safety

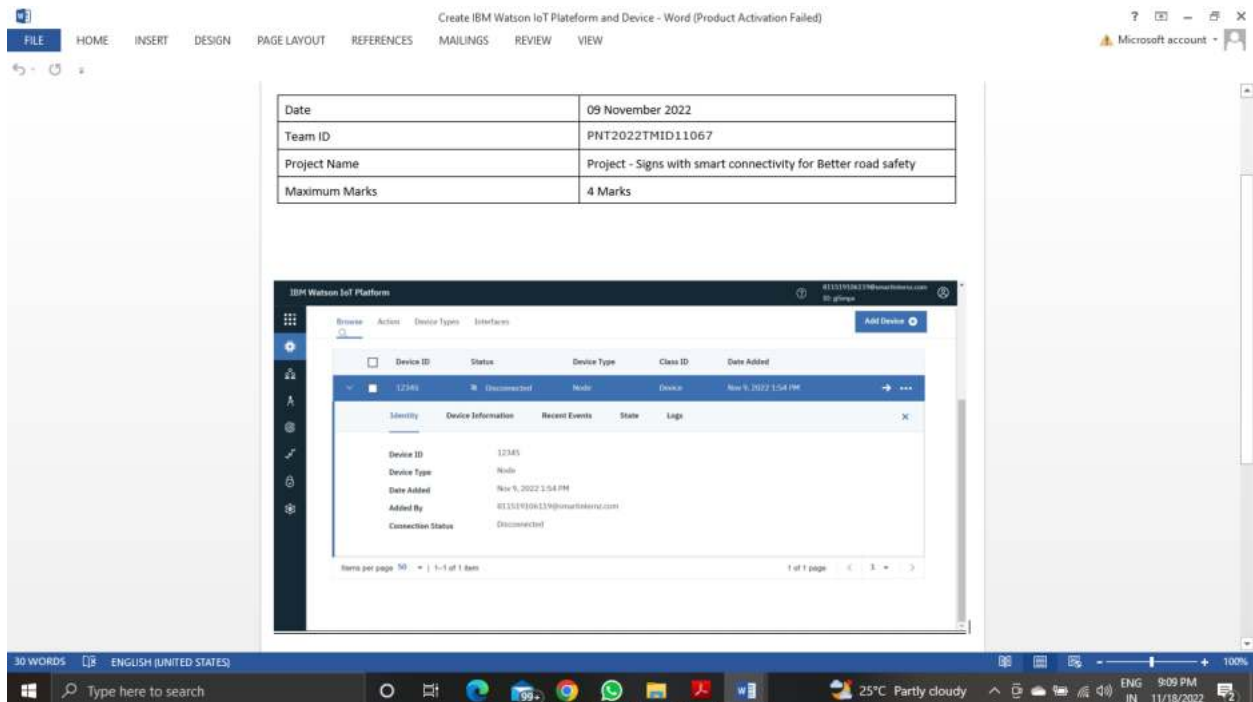


Signs with Smart Connectivity for Better Road Safety



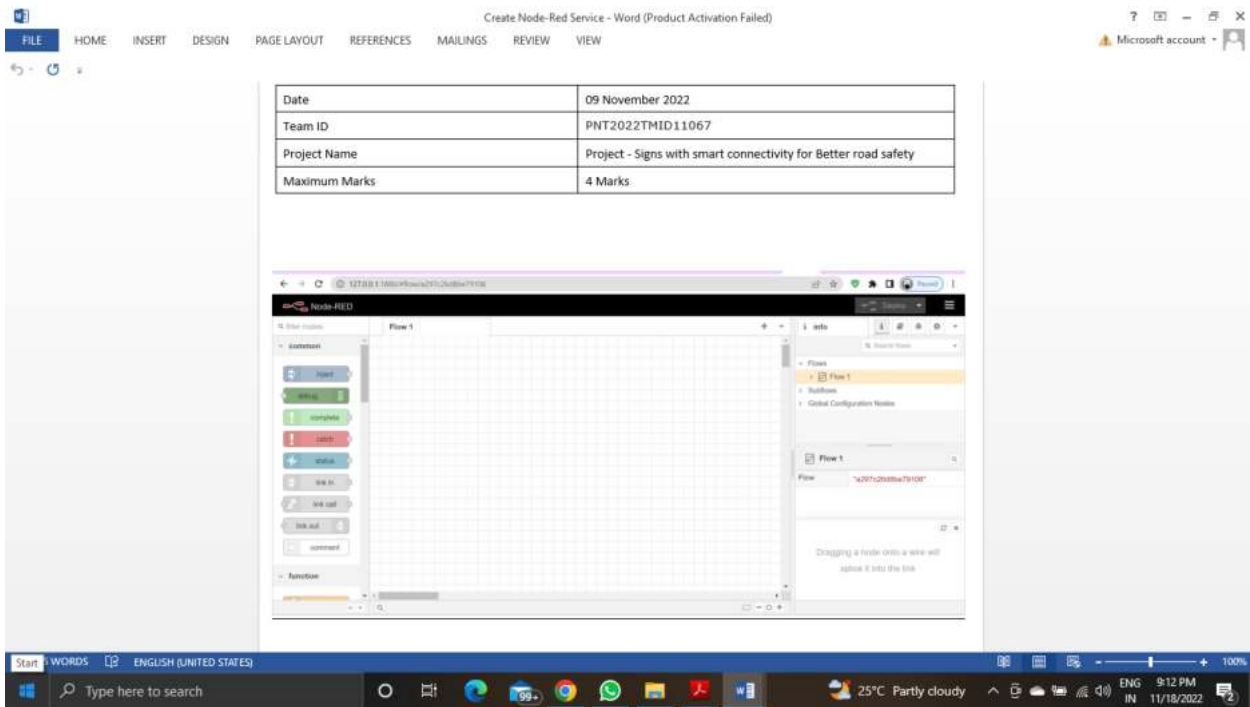
CREATE AND CONFIGURE IBM CLOUD SERVICE:

CREATE IBM WATSON IoT PLATFORM AND DEVICE:



CREATE NODE-RED SERVICE :

Signs with Smart Connectivity for Better Road Safety

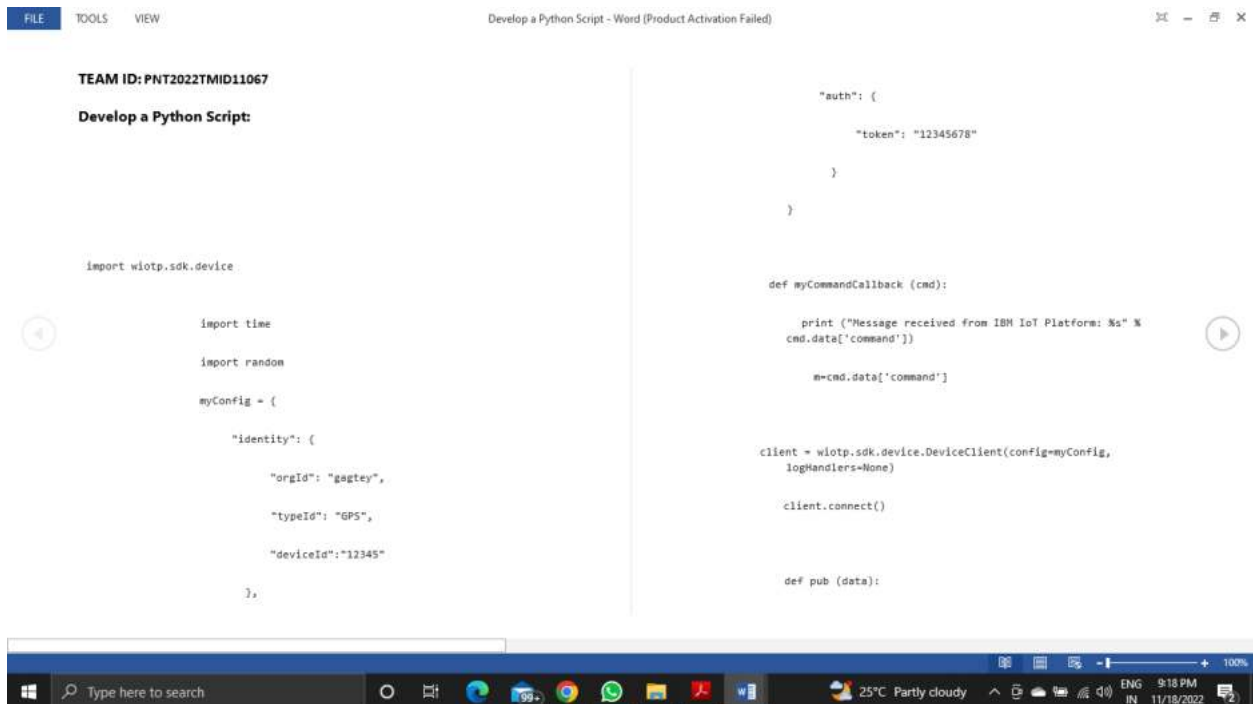


Date	09 November 2022
Team ID	PNT2022TMID11067
Project Name	Project - Signs with smart connectivity for Better road safety
Maximum Marks	4 Marks

The screenshot shows the Node-RED web interface in a browser. It features a left sidebar with various nodes like 'inject', 'http in', 'http out', 'websocket in', and 'websocket out'. The main workspace is empty, and the right sidebar shows the 'Flows' panel with 'Flow 1' selected.

PYTHON SCRIPT:

DEVELOP A PYTHON SCRIPT:



TEAM ID: PNT2022TMID11067

Develop a Python Script:

```
import wiotp.sdk.device

import time
import random

myConfig = {
    "identity": {
        "orgId": "gagtey",
        "typeId": "GPS",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}

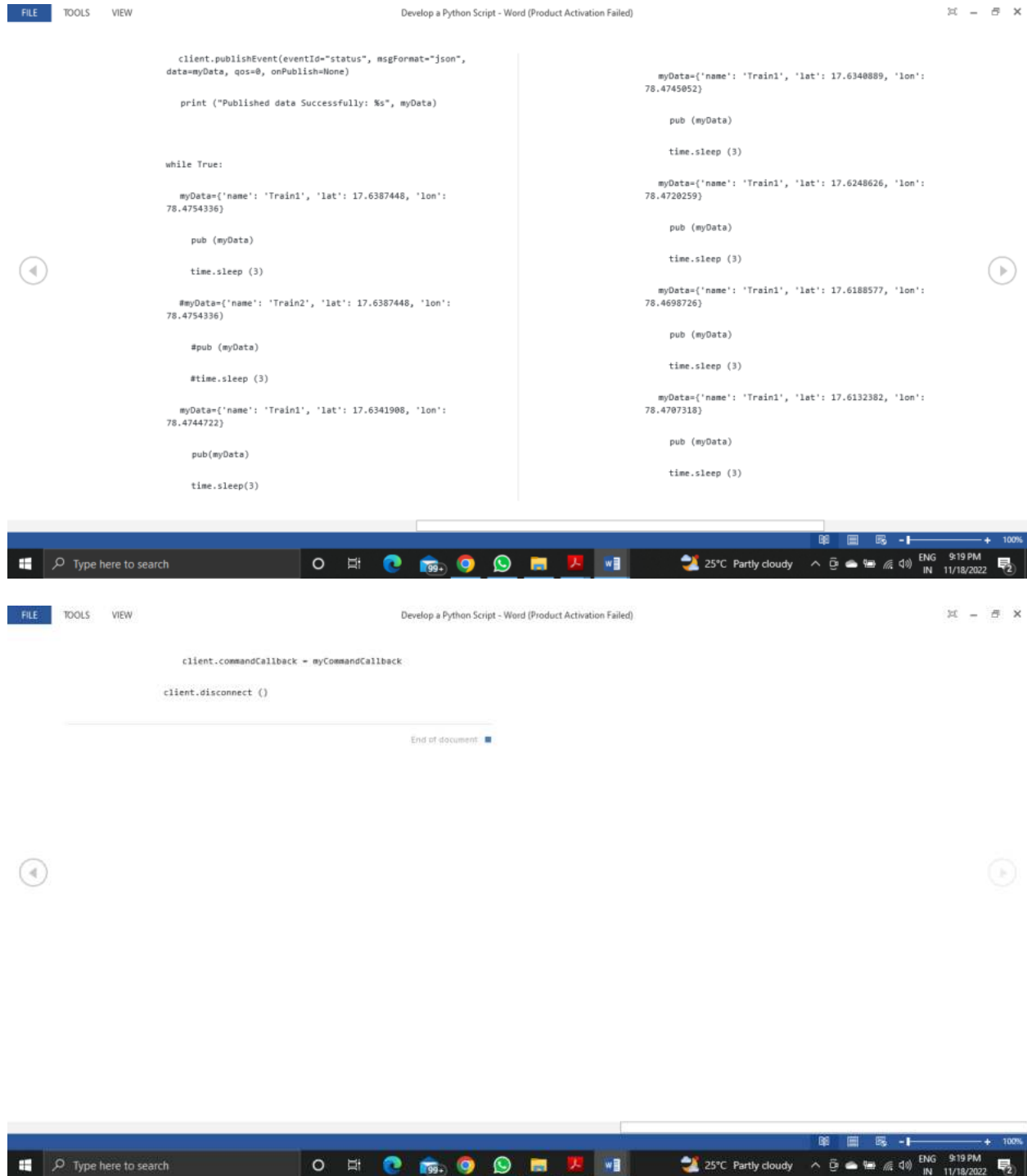
def myCommandCallback(cmd):
    print ("Message received from IBM IoT Platform: %s" %
          cmd.data['command'])
    m=cmd.data["command"]

client = wiotp.sdk.device.DeviceClient(config=myConfig,
                                       logHandlers=None)

client.connect()

def pub (data):
```

Signs with Smart Connectivity for Better Road Safety



```
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': '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':
78.4707310}

pub (myData)

time.sleep (3)
```

```
client.commandCallback = myCommandCallback

client.disconnect ()
```

End of document

PYTHON SCRIPT PNT2022TMID11067:

Signs with Smart Connectivity for Better Road Safety

The image displays a Python script in a code editor, titled "Python Scripts PNT2022TMD11067 - Word (Product Activation Failed)". The script is divided into two main sections: initialization and a data publishing loop.

Initialization Section:

```
import wiotp.sdk.device

import time
import random

myConfig = {
    "identity": {
        "orgId": "gagtey",
        "typeId": "GPS",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}

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, qos=0, onPublish=None)
```

Data Publishing Loop Section:

```
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': 17.6341988, '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': 78.4728259}
    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': 78.4787318}
    pub (myData)
    time.sleep (3)
    client.commandCallback = myCommandCallback
    client.disconnect ()
```

The script is presented in a split-view format, with the left pane showing the full script and the right pane showing a zoomed-in view of the initialization and callback functions. The code editor interface includes a menu bar (FILE, TOOLS, VIEW) and a status bar at the bottom showing system information like temperature (25°C) and time (9:21 PM).

DEVELOP A WEB APPLICATION USING NODE-RED SERVICE:

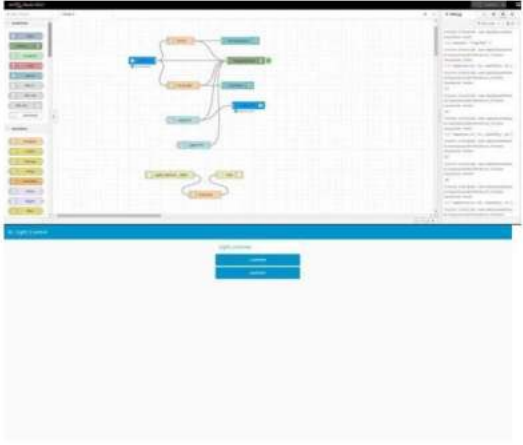
Signs with Smart Connectivity for Better Road Safety

FILE TOOLS VIEW Develop the web application using node red - Word (Product Activation Failed)


DEVELOP THE WEB APPLICATION USING NODE RED

Date	18 November 2022
Team ID	PNT2022TMD11067
Project Name	Project - Signs with smart connectivity for Better road safety
Maximum Marks	4 Marks

Create Node Red Flow To Send Date to Device:



FILE TOOLS VIEW Develop the web application using node red - Word (Product Activation Failed)



End of document

USE DASHBOARD NODES FOR CREATING UI (WEB APP):


Signs with Smart Connectivity for Better Road Safety

FILE TOOLS VIEW Use dashboard nodes for creating UI (WEB APP) - Word (Product Activation Failed)

USE DASHBOARD NODES FOR CREATING UI (WEB APP)

Date	18 November 2022
Team ID	PNT2022TMD11067
Project Name	Project - Signs with smart connectivity for Better road safety
Maximum Marks	4 Marks

UI Nodes Installation:



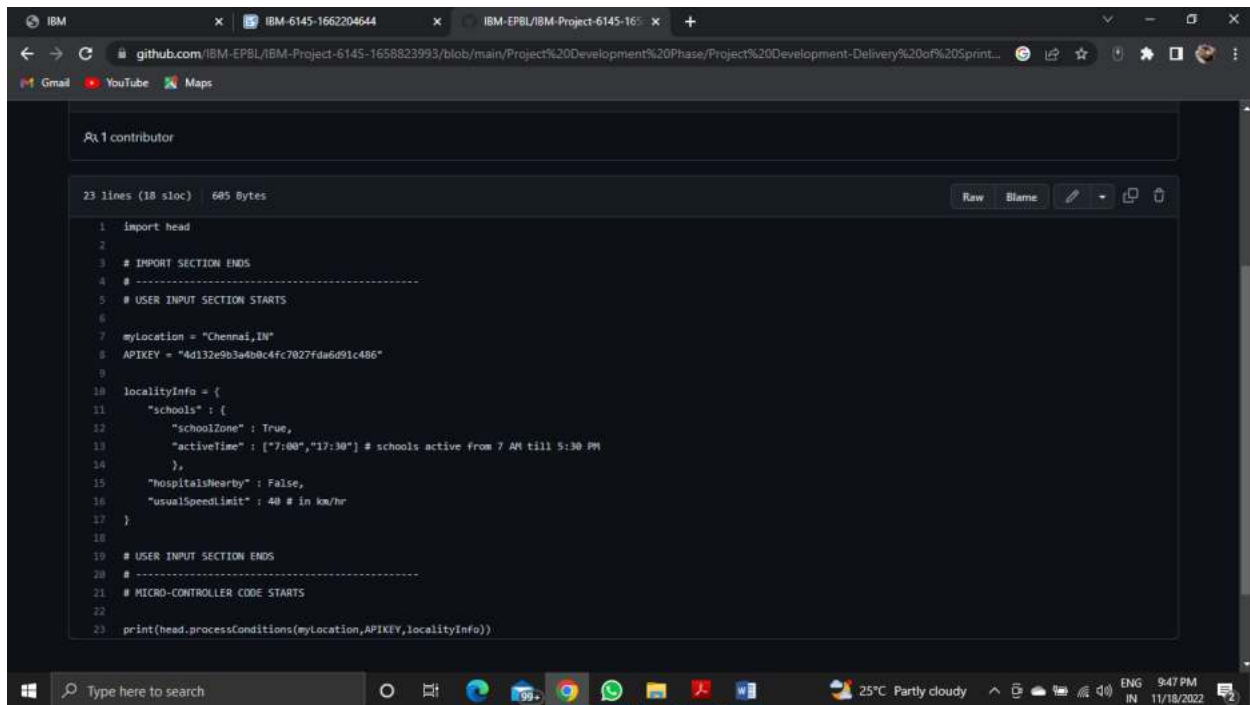
End of document

PROJECT DEVELOPMENT PHASE: PROJECT DEVELOPMENT -DELIVERY OF SPRINT 1: brain.txt:

```
29 lines (25 sloc) 1.08 KB
1 import weather
2 from datetime import datetime as dt
3
4 # IMPORT SECTION ENDS
5 #
6 # UTILITY LOGIC SECTION STARTS
7 def processConditions(myLocation,APIKEY,localityInfo):
8     weatherData = weather.get(myLocation,APIKEY)
9
10    finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherData else localityInfo["usualSpeedLimit"]/2
11    finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpeed/2
12
13    if(localityInfo["hospitalsNearby"]):
14        # hospital zone
15        doNotHonk = True
16    else:
17        if(localityInfo["schools"])[["schoolZone"]]==False:
18            # neither school nor hospital zone
19            doNotHonk = False
20        else:
21            # school zone
22            now = [dt.now().hour,dt.now().minute]
23            activeTime = [list(map(int,_.split(":"))) for _ in localityInfo["schools"][["activeTime"]]]
24            doNotHonk = activeTime[0][0]<now[0]<activeTime[1][0] and activeTime[0][1]<now[1]<activeTime[1][1]
25
26    return({
27        "Speed" : finalSpeed,
28        "doNotHonk" : doNotHonk
29    })
```

main.txt:

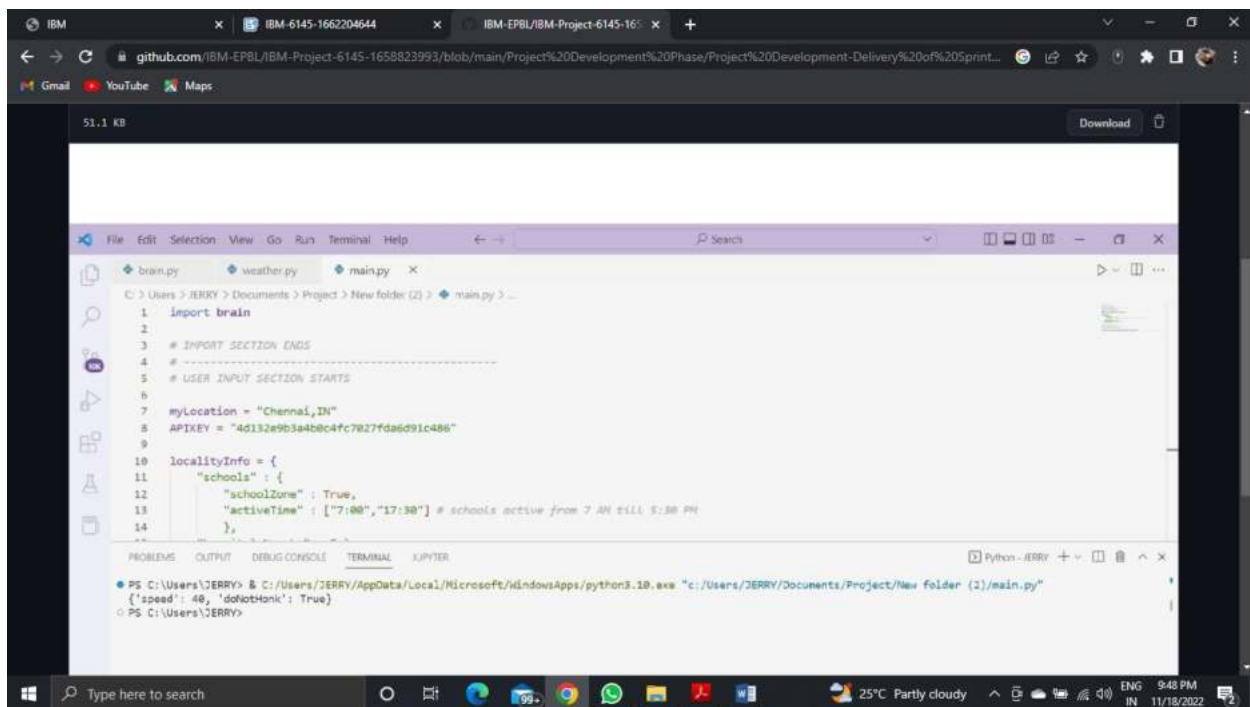
Signs with Smart Connectivity for Better Road Safety



The screenshot shows a web browser displaying a GitHub repository page for a project titled "IBM-EPBL/IBM-Project-6145-1658823993". The repository is a Python script named "main.py" with 23 lines of code. The code is a Python script that uses the "head" library to process conditions for smart road signs. It defines a "myLocation" variable as "Chennai, IN" and an "APIKEY" variable as "4d132e9b3a4b0c4fc7027fda6d91c486". It then defines a "localityInfo" dictionary with keys for "schools", "schoolZone", "activeTime", "hospitalsNearby", and "usualSpeedLimit". The "schools" key is a dictionary with "schoolZone" set to True and "activeTime" set to ["7:00", "17:30"], with a comment "# schools active from 7 AM till 5:30 PM". The "schoolZone" key is set to True, "activeTime" is set to ["7:00", "17:30"], "hospitalsNearby" is set to False, and "usualSpeedLimit" is set to 40 with a comment "# in km/hr". The script then prints the result of "head.processConditions(myLocation, APIKEY, localityInfo)".

```
1 import head
2
3 # IMPORT SECTION ENDS
4 # -----
5 # USER INPUT SECTION STARTS
6
7 myLocation = "Chennai, IN"
8 APIKEY = "4d132e9b3a4b0c4fc7027fda6d91c486"
9
10 localityInfo = {
11     "schools": {
12         "schoolZone": True,
13         "activeTime": ["7:00", "17:30"] # schools active from 7 AM till 5:30 PM
14     },
15     "hospitalsNearby": False,
16     "usualSpeedLimit": 40 # in km/hr
17 }
18
19 # USER INPUT SECTION ENDS
20 # -----
21 # MICRO-CONTROLLER CODE STARTS
22
23 print(head.processConditions(myLocation, APIKEY, localityInfo))
```

output.txt:



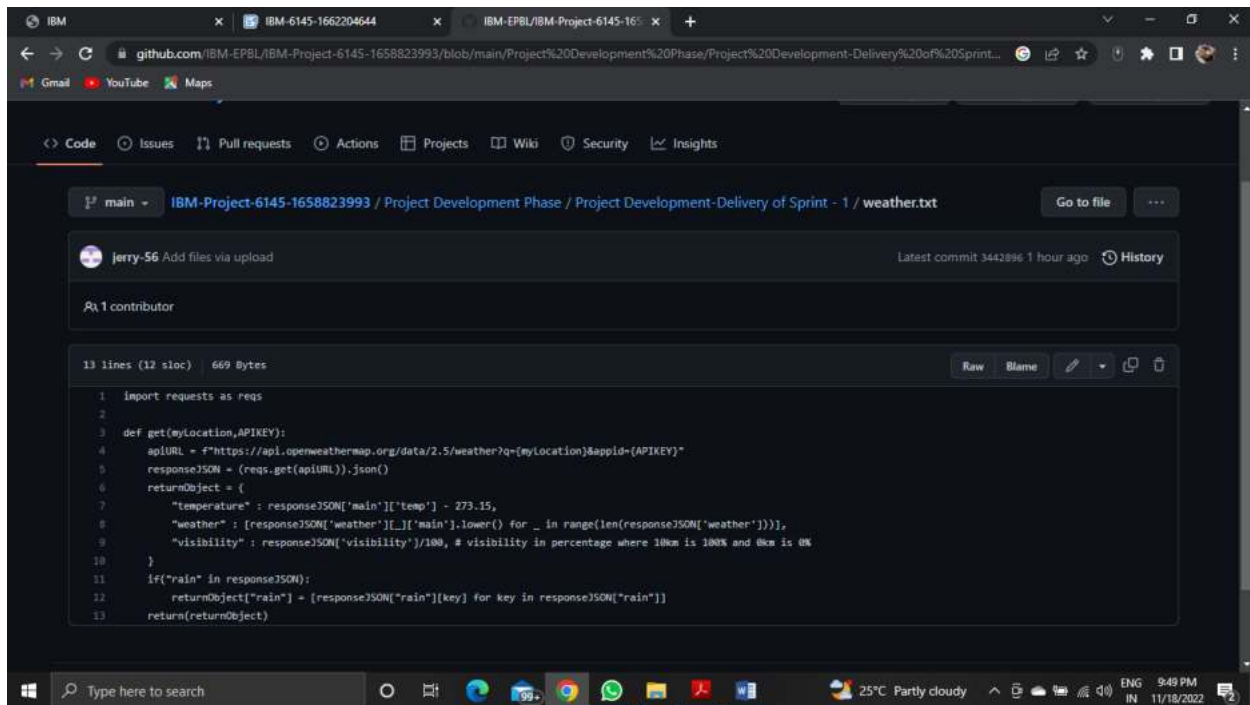
The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor in the center. The file explorer shows a folder named "main.py" with a file named "main.py". The code editor shows the same Python script as the previous screenshot. The terminal at the bottom shows the command "python3 main.py" being executed, and the output is displayed as a JSON object: {"speed": 40, "dotdotdot": True}.

```
1 import brain
2
3 # IMPORT SECTION ENDS
4 # -----
5 # USER INPUT SECTION STARTS
6
7 myLocation = "Chennai, IN"
8 APIKEY = "4d132e9b3a4b0c4fc7027fda6d91c486"
9
10 localityInfo = {
11     "schools": {
12         "schoolZone": True,
13         "activeTime": ["7:00", "17:30"] # schools active from 7 AM till 5:30 PM
14     },
15     "hospitalsNearby": False,
16     "usualSpeedLimit": 40 # in km/hr
17 }
18
19 # USER INPUT SECTION ENDS
20 # -----
21 # MICRO-CONTROLLER CODE STARTS
22
23 print(head.processConditions(myLocation, APIKEY, localityInfo))
```

```
PS C:\Users\JERRY> python3 main.py
{"speed": 40, "dotdotdot": True}
```

weather.txt:

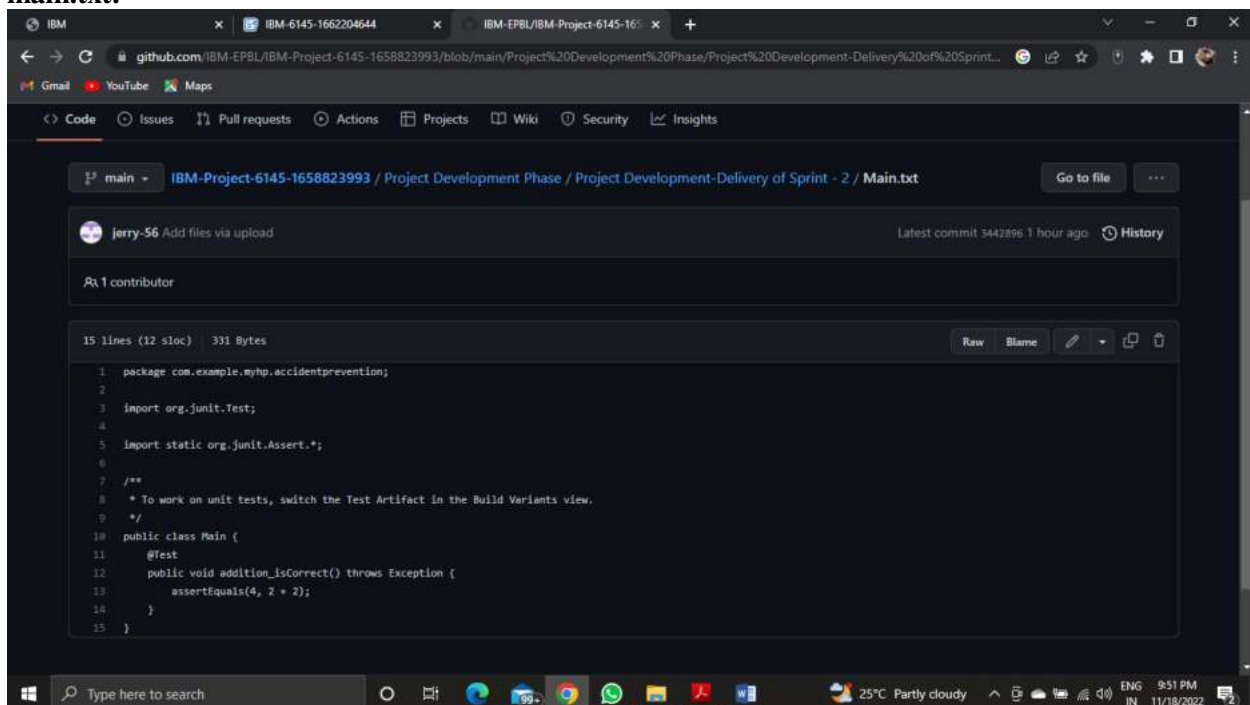
Signs with Smart Connectivity for Better Road Safety



The screenshot shows a web browser displaying a GitHub repository page. The repository is named 'IBM-EPBL/IBM-Project-6145-1658823993' and the file being viewed is 'weather.txt'. The file is 13 lines long and 669 bytes. The code is written in Python and uses the 'requests' library to fetch weather data from the OpenWeatherMap API. The code includes a function 'get(myLocation, APIKEY)' that returns a dictionary with 'temperature', 'weather', and 'visibility' keys. The 'weather' key is a list of strings, and the 'visibility' key is a float representing visibility in km. The code also includes a check for rain in the response and returns a dictionary with 'rain' and 'temperature' keys.

```
1 import requests as reqs
2
3 def get(myLocation, APIKEY):
4     apiURL = f"https://api.openweathermap.org/data/2.5/weather?q={myLocation}&appid={APIKEY}"
5     responseJSON = (reqs.get(apiURL)).json()
6     responseObject = {
7         "temperature": responseJSON["main"]["temp"] - 273.15,
8         "weather": [responseJSON["weather"][_]["main"].lower() for _ in range(len(responseJSON["weather"]))],
9         "visibility": responseJSON["visibility"]/100, # visibility in percentage where 10km is 100% and 0km is 0%
10    }
11    if("rain" in responseJSON):
12        responseObject["rain"] = [responseJSON["rain"][key] for key in responseJSON["rain"]]
13    return(responseObject)
```

PROJECT DEVELOPMENT -DELIVERY OF SPRINT 2: main.txt:

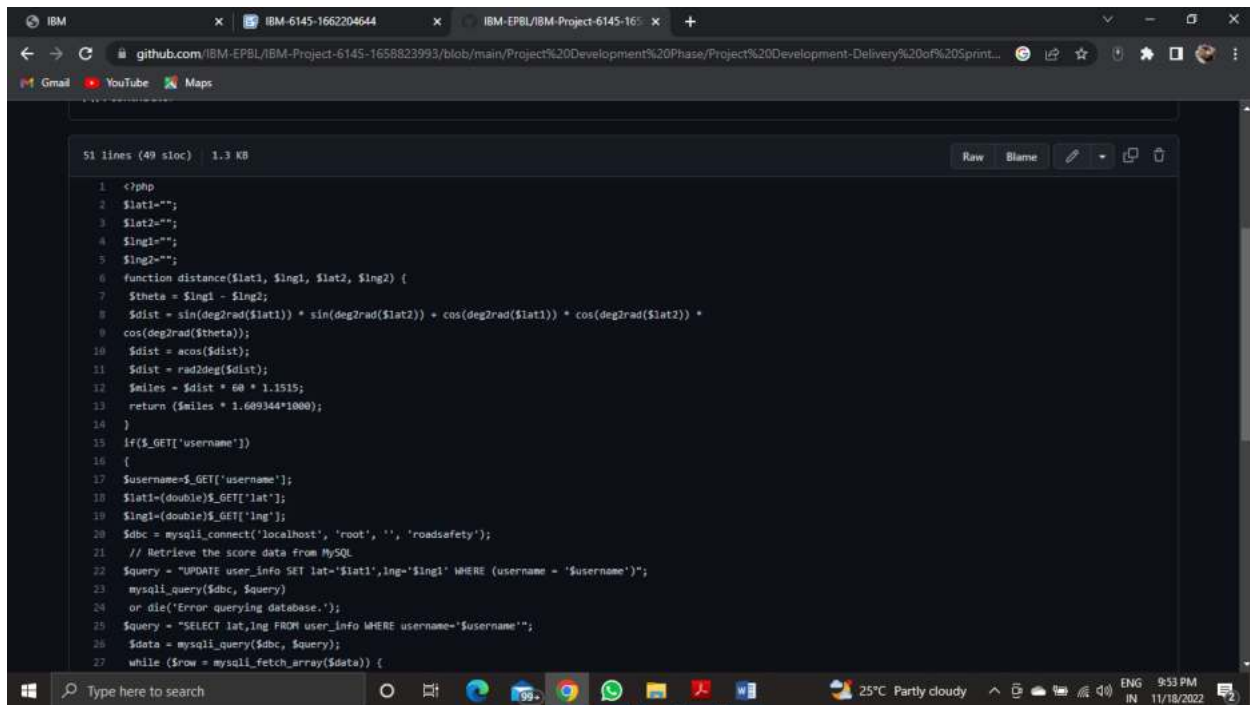


The screenshot shows a web browser displaying a GitHub repository page. The repository is named 'IBM-EPBL/IBM-Project-6145-1658823993' and the file being viewed is 'Main.txt'. The file is 15 lines long and 331 bytes. The code is written in Java and includes a package declaration, imports for 'org.junit.Test' and 'org.junit.Assert.*', and a class 'Main' with a method 'addition_isCorrect()' that throws an exception if the assertion fails.

```
1 package com.example.myhp.accidentprevention;
2
3 import org.junit.Test;
4
5 import static org.junit.Assert.*;
6
7 /**
8  * To work on unit tests, switch the Test Artifact in the Build Variants view.
9  */
10 public class Main {
11     @Test
12     public void addition_isCorrect() throws Exception {
13         assertEquals(4, 2 + 2);
14     }
15 }
```

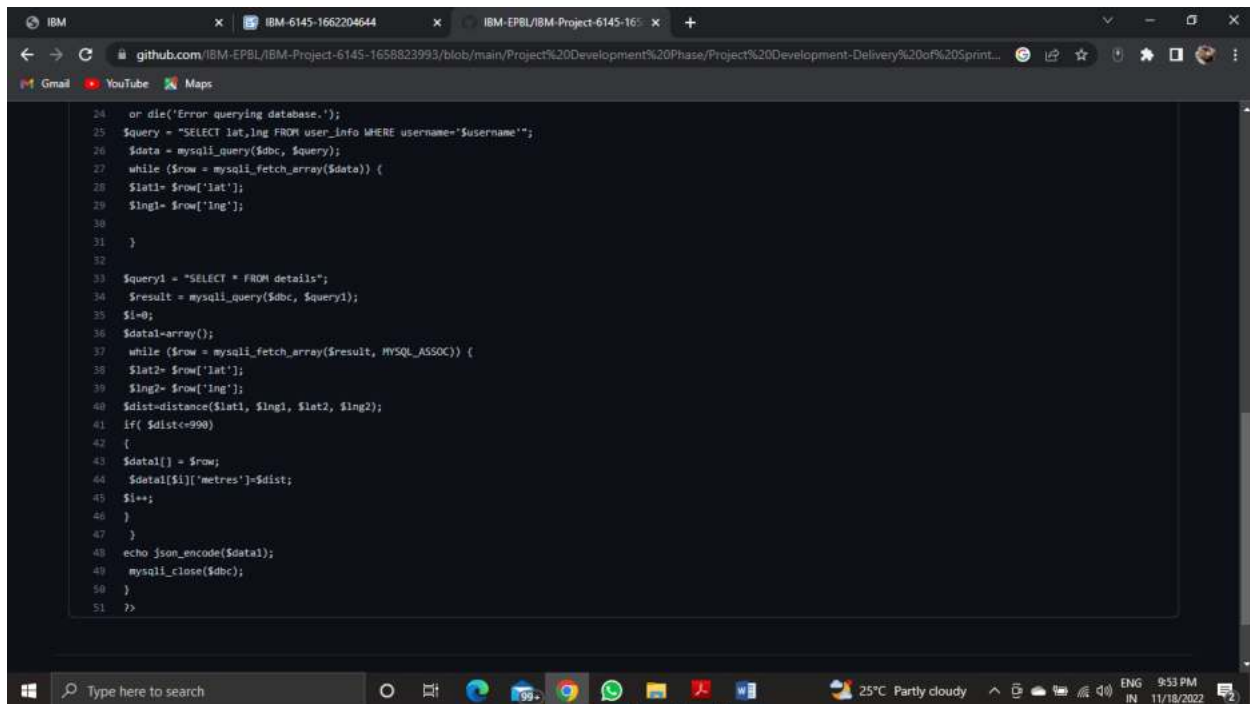
safety.txt:

Signs with Smart Connectivity for Better Road Safety



The screenshot shows a web browser displaying a GitHub repository for an IBM EPBL project. The repository is titled "IBM-EPBL/IBM-Project-6145-1658823993/blob/main/Project%20Development%20Phase/Project%20Development-Delivery%20of%20Sprint...". The code is a PHP script with 51 lines (49 sloc) and a size of 1.3 KB. The code defines a function to calculate distance between two points, retrieves user information from a MySQL database, and calculates the distance between the user's location and a set of points.

```
1 <?php
2 $lat1="";
3 $lat2="";
4 $lng1="";
5 $lng2="";
6 function distance($lat1, $lng1, $lat2, $lng2) {
7     $theta = $lng1 - $lng2;
8     $dist = sin(deg2rad($lat1)) * sin(deg2rad($lat2)) + cos(deg2rad($lat1)) * cos(deg2rad($lat2)) *
9     cos(deg2rad($theta));
10    $dist = acos($dist);
11    $dist = rad2deg($dist);
12    $miles = $dist * 60 * 1.1515;
13    return ($miles * 1.609344*1000);
14 }
15 if($_GET['username'])
16 {
17     $username=$_GET['username'];
18     $lat1=(double)$_GET['lat'];
19     $lng1=(double)$_GET['lng'];
20     $dbc = mysqli_connect('localhost', 'root', '', 'roadsafety');
21     // Retrieve the score data from MySQL
22     $query = "UPDATE user_info SET lat='$lat1',lng='$lng1' WHERE (username = '$username')";
23     mysqli_query($dbc, $query)
24     or die('Error querying database.');
```



The screenshot shows the continuation of the PHP code from the previous screenshot. The code continues to calculate the distance between the user's location and a set of points, and then outputs the results in JSON format.

```
25 $query = "SELECT lat,lng FROM user_info WHERE username='$username'";
26 $data = mysqli_query($dbc, $query);
27 while ($row = mysqli_fetch_array($data)) {
28     $lat1= $row['lat'];
29     $lng1= $row['lng'];
30 }
31 }
32
33 $query1 = "SELECT * FROM details";
34 $result = mysqli_query($dbc, $query1);
35 $i=0;
36 $data1=array();
37 while ($row = mysqli_fetch_array($result, MYSQL_ASSOC)) {
38     $lat2= $row['lat'];
39     $lng2= $row['lng'];
40     $dist=distance($lat1, $lng1, $lat2, $lng2);
41     if( $dist<=990)
42     {
43         $data1[] = $row;
44         $data1[$i]['metres']=$dist;
45         $i++;
46     }
47 }
48 echo json_encode($data1);
49 mysqli_close($dbc);
50 }
51 >>
```

build gradle.txt:

The image displays two screenshots of a web browser showing a GitHub repository for the project 'IBM-EPBL/IBM-Project-6145-1658823993'. The first screenshot shows the 'build.gradle' file for the 'Flutter SDK' plugin, which defines local properties, Flutter SDK version, and applies the 'com.android.application' and 'com.google.gms.google-services' plugins. The second screenshot shows the 'build.gradle' file for the 'Flutter SDK' plugin, which defines the application ID, minSdkVersion, targetSdkVersion, multiDexEnabled, versionCode, versionName, buildConfigField, buildTypes (release), flutter (source), dependencies, and multiDex version.

65 lines (65 sloc) 2.15 KB

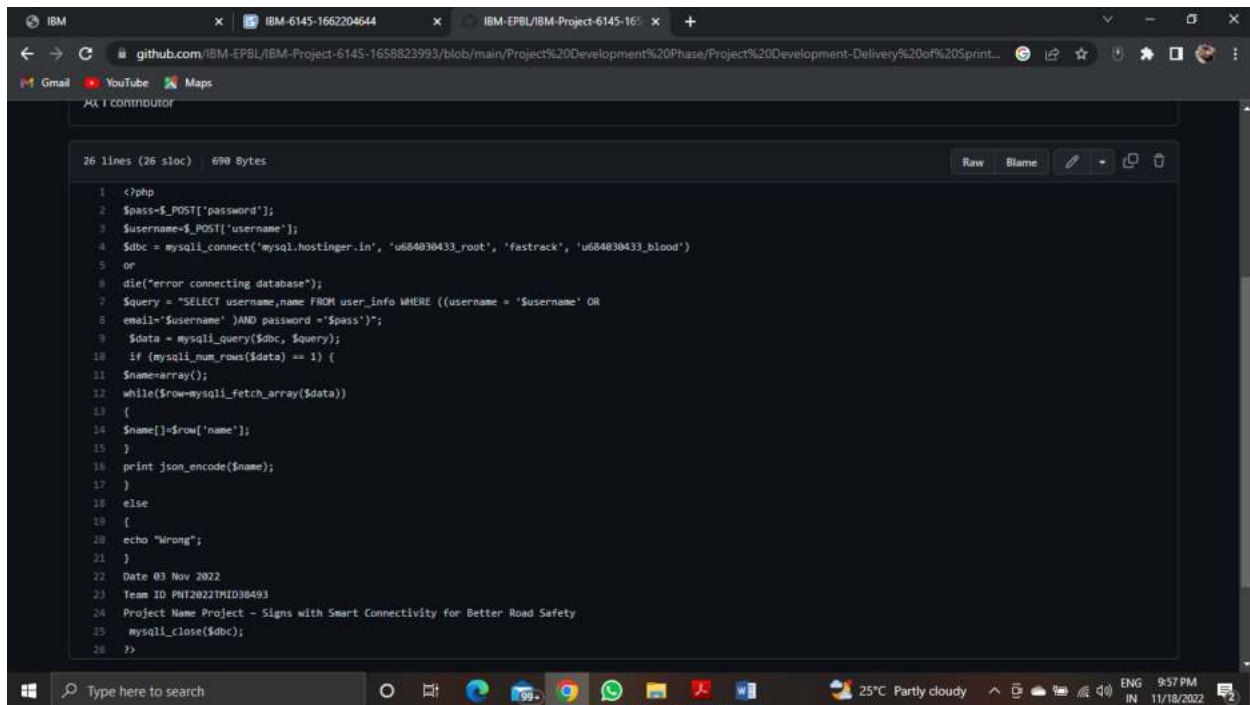
```
1 def localProperties = new Properties()
2 def localPropertiesFile = rootProject.file('local.properties')
3 if (localPropertiesFile.exists()) {
4     localPropertiesFile.withReader('UTF-8') { reader ->
5         localProperties.load(reader)
6     }
7 }
8 def flutterRoot = localProperties.getProperty('flutter.sdk')
9 if (flutterRoot == null) {
10     throw new GradleException("Flutter SDK not found. Define location with flutter.sdk in the
11     local.properties file.")
12 }
13 def flutterVersionCode = localProperties.getProperty('flutter.versionCode')
14 if (flutterVersionCode == null) {
15     flutterVersionCode = '1'
16 }
17 def flutterVersionName = localProperties.getProperty('flutter.versionName')
18 if (flutterVersionName == null) {
19     flutterVersionName = '1.0'
20 }
21 apply plugin: 'com.android.application'
22 apply plugin: 'com.google.gms.google-services'
23 apply plugin: 'kotlin-android'
24 apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"
25 android {
26     compileSdkVersion 28
27     sourceSets {
28         main.java.srcDirs += 'src/main/kotlin'
29     }
30     defaultConfig {
31         // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
32         applicationId "com.example.roads"
33         minSdkVersion 18
34         targetSdkVersion 28
35         multiDexEnabled true
36         versionCode flutterVersionCode.toInteger()
37         versionName flutterVersionName
38     }
39     buildConfigField 'String', 'MONDERPUSH_CLIENT_ID',
40         '"Idfce26a84bd50e2b2117ae3a65d9f3d88821cb"'
41     buildConfigField 'String', 'MONDERPUSH_CLIENT_SECRET',
42         '"q489ca2368fb19536c8e5efc378d18e9802a5034bf14e13e4b4e92286d39c5a0"'
43     buildConfigField 'String', 'MONDERPUSH_SENDER_ID', '"1098284096327"'
44 }
45 buildTypes {
46     release {
47         // TODO: Add your own signing config for the release build.
48         // Signing with the debug keys for now, so 'flutter run --release' works.
49         signingConfig signingConfigs.debug
50     }
51 }
52 flutter {
53     source '../..'
54 }
55 dependencies {
56     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
57     implementation platform('com.google.firebase:firebase-bom:26.1.0')
58     implementation 'com.google.firebase:firebase-analytics'
59     def multidex_version = "2.0.1"
60     implementation 'androidx.multidex:multidex:$multidex_version'
61 }
```

Type here to search

25°C Partly cloudy

ENG 9:55 PM 11/18/2022

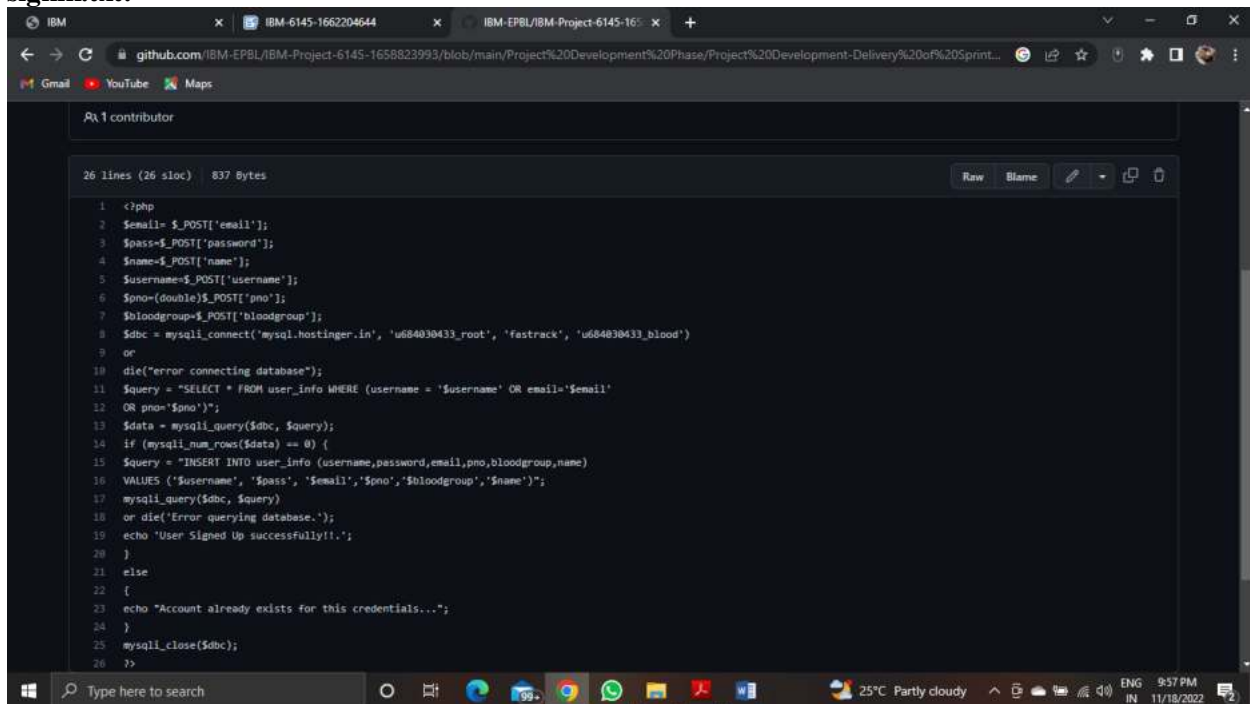
Signs with Smart Connectivity for Better Road Safety



The screenshot shows a web browser displaying a GitHub repository page. The repository is named "signin.txt" and is part of a project titled "Signs with Smart Connectivity for Better Road Safety". The file "signin.txt" is open, showing a PHP script for user login. The script connects to a MySQL database, checks the username and password, and returns a JSON response if the user is found. The script is 26 lines long and 690 bytes in size.

```
1 <?php
2 $pass=$_POST['password'];
3 $username=$_POST['username'];
4 $dbc = mysqli_connect('mysql.hostinger.in', 'u684830433_root', 'fastrack', 'u684830433_blood')
5 or
6 die("error connecting database");
7 $query = "SELECT username,name FROM user_info WHERE ((username = '$username' OR
8 email='$username' )AND password = '$pass')";
9 $data = mysqli_query($dbc, $query);
10 if (mysqli_num_rows($data) == 1) {
11 $name=array();
12 while($row=mysqli_fetch_array($data))
13 {
14 $name[]=$row['name'];
15 }
16 print json_encode($name);
17 }
18 else
19 {
20 echo "Wrong";
21 }
22 Date 03 Nov 2022
23 Team ID PNT2022TMD30493
24 Project Name Project - Signs with Smart Connectivity for Better Road Safety
25 mysqli_close($dbc);
26 >}
```

signin.txt:

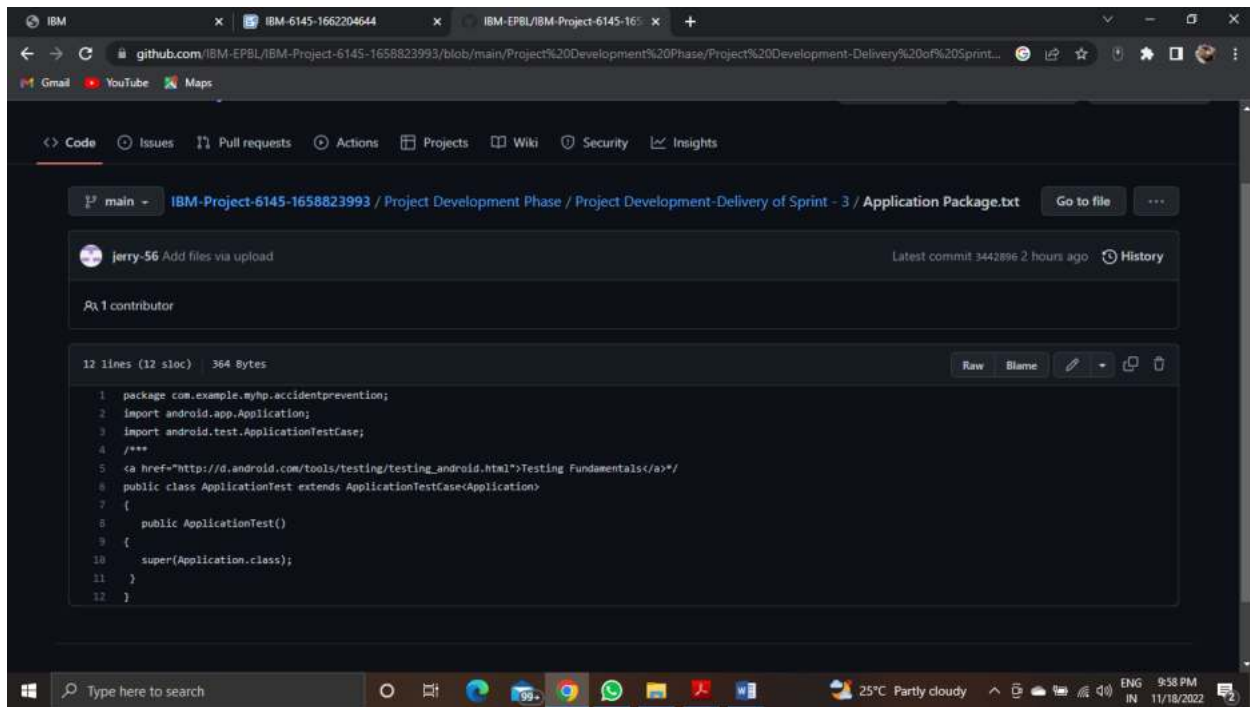


The screenshot shows a web browser displaying a GitHub repository page. The repository is named "signin.txt" and is part of a project titled "Signs with Smart Connectivity for Better Road Safety". The file "signin.txt" is open, showing a PHP script for user registration. The script connects to a MySQL database, checks if the user already exists, and inserts the user information if it does not. The script is 26 lines long and 837 bytes in size.

```
1 <?php
2 $email= $_POST['email'];
3 $pass=$_POST['password'];
4 $name=$_POST['name'];
5 $username=$_POST['username'];
6 $pno=(double)$_POST['pno'];
7 $bloodgroup=$_POST['bloodgroup'];
8 $dbc = mysqli_connect('mysql.hostinger.in', 'u684830433_root', 'fastrack', 'u684830433_blood')
9 or
10 die("error connecting database");
11 $query = "SELECT * FROM user_info WHERE (username = '$username' OR email='$email'
12 OR pno='$pno')";
13 $data = mysqli_query($dbc, $query);
14 if (mysqli_num_rows($data) == 0) {
15 $query = "INSERT INTO user_info (username,password,email,pno,bloodgroup,name)
16 VALUES ('$username', '$pass', '$email','$pno','$bloodgroup','$name')";
17 mysqli_query($dbc, $query)
18 or die("Error querying database.");
19 echo "User Signed Up successfully!!.";
20 }
21 else
22 {
23 echo "Account already exists for this credentials...";
24 }
25 mysqli_close($dbc);
26 >}
```

PROJECT DEVELOPMENT -DELIVERY OF SPRINT 3: APPLICATION PACKAGE.TXT:

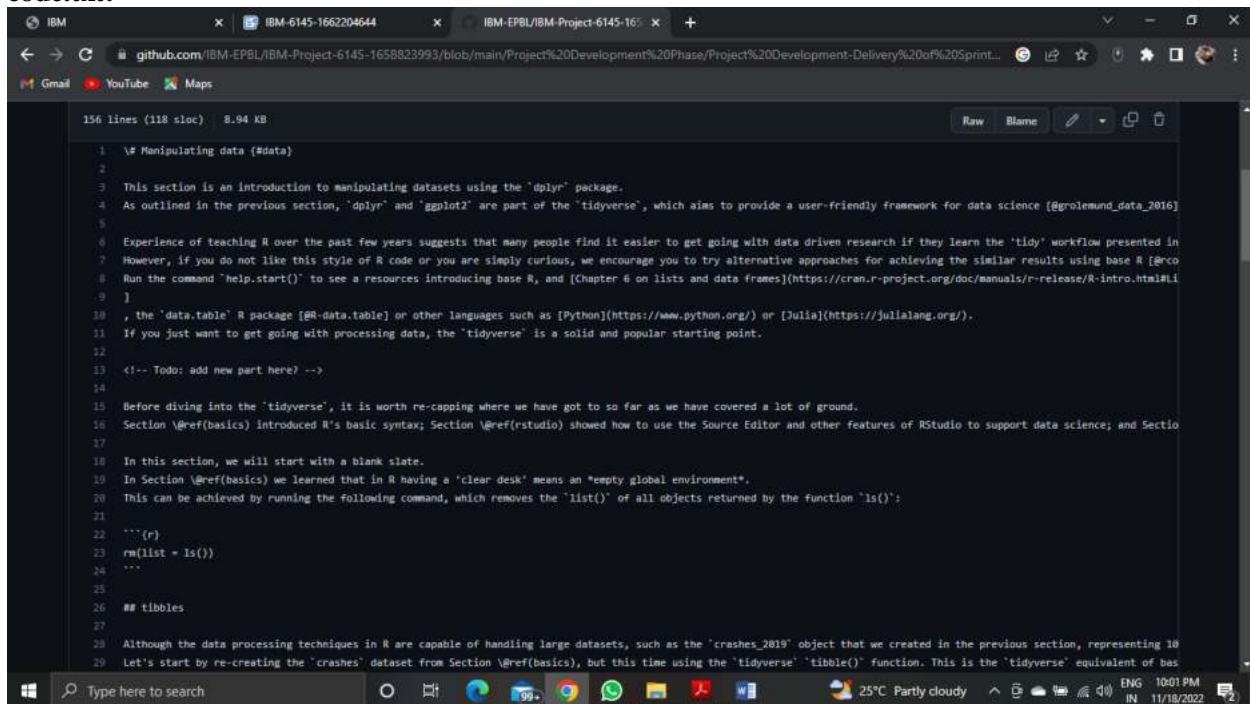
Signs with Smart Connectivity for Better Road Safety



The screenshot shows a web browser window displaying a GitHub repository. The repository is named "IBM-EPBL/IBM-Project-6145-1658823993" and the file being viewed is "Application Package.txt". The file is 12 lines long and 364 bytes. The code is in Java and defines a class "ApplicationTest" that extends "ApplicationTestCase". The code includes package declarations, imports, and a main method.

```
1 package com.example.myapplication;
2 import android.app.Application;
3 import android.test.ApplicationTestCase;
4 /**
5  * @href="http://d.android.com/tools/testing/testing_android.html">Testing Fundamentals</a>*/
6  public class ApplicationTest extends ApplicationTestCase<Application>
7  {
8      public ApplicationTest()
9      {
10         super(Application.class);
11     }
12 }
```

code.txt:



The screenshot shows a web browser window displaying a GitHub repository. The repository is named "IBM-EPBL/IBM-Project-6145-1658823993" and the file being viewed is "code.txt". The file is 156 lines long and 8.94 KB. The file contains a mix of text and R code. The text describes the 'tidyverse' package and the 'dplyr' package. The R code includes comments and a function definition.

```
1 \# Manipulating data (#data)
2
3 This section is an introduction to manipulating datasets using the 'dplyr' package.
4 As outlined in the previous section, 'dplyr' and 'ggplot2' are part of the 'tidyverse', which aims to provide a user-friendly framework for data science (@golemund_data_2016)
5
6 Experience of teaching R over the past few years suggests that many people find it easier to get going with data driven research if they learn the 'tidy' workflow presented in
7 However, if you do not like this style of R code or you are simply curious, we encourage you to try alternative approaches for achieving the similar results using base R (@rco
8 Run the command 'help.start()' to see a resources introducing base R, and (Chapter 6 on lists and data frames)(https://cran.r-project.org/doc/manuals/r-release/R-intro.html#li
9 ]
10 , the 'data.table' R package [R-data.table] or other languages such as [Python](https://www.python.org/) or [Julia](https://julialang.org/).
11 If you just want to get going with processing data, the 'tidyverse' is a solid and popular starting point.
12
13 <!-- TODO: add new part here? -->
14
15 Before diving into the 'tidyverse', it is worth re-capping where we have got to so far as we have covered a lot of ground.
16 Section \@ref(basics) introduced R's basic syntax; Section \@ref(rstudio) showed how to use the Source Editor and other features of RStudio to support data science; and Sectio
17
18 In this section, we will start with a blank slate.
19 In Section \@ref(basics) we learned that in R having a 'clean desk' means an "empty global environment".
20 This can be achieved by running the following command, which removes the 'list()' of all objects returned by the function 'ls()':
21
22 ```{r}
23 rm(list = ls())
24 ```
25
26 ## tibbles
27
28 Although the data processing techniques in R are capable of handling large datasets, such as the 'crashes_2019' object that we created in the previous section, representing 18
29 Let's start by re-creating the 'crashes' dataset from Section \@ref(basics), but this time using the 'tidyverse' 'tibble()' function. This is the 'tidyverse' equivalent of bas
```

Signs with Smart Connectivity for Better Road Safety

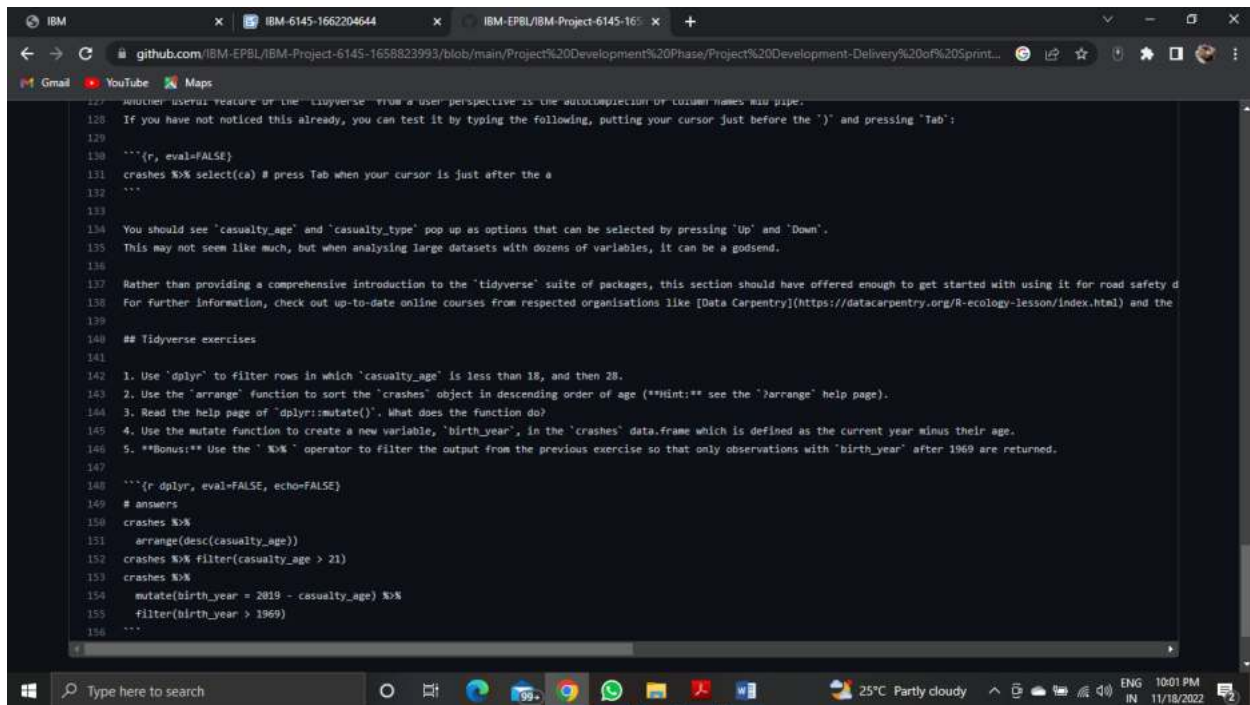
```
IBM x IBM-6145-1662204644 x IBM-EPBL/IBM-Project-6145-1662204644 x +
github.com/IBM-EPBL/IBM-Project-6145-1662204644/blob/main/Project%20Development%20Phase/Project%20Development-Delivery%20of%20Sprint...
Gmail YouTube Maps

26 ## tibbles
27
28 Although the data processing techniques in R are capable of handling large datasets, such as the 'crashes_2019' object that we created in the previous section, representing 10
29 Let's start by re-creating the 'crashes' dataset from Section \@ref(basics), but this time using the 'tidyverse' 'tibble()' function. This is the 'tidyverse' equivalent of bas
30 'tibble' objects can be created, after loading the 'tidyverse', as follows:
31
32 ```{r, message=FALSE}
33 library(tidyverse)
34 crashes = tibble(
35   casualty_type = c("pedestrian", "cyclist", "cat"),
36   casualty_age = seq(from = 20, to = 60, by = 20),
37   vehicle_type = c("car", "bus", "tank"),
38   dark = c(TRUE, FALSE, TRUE)
39 )
40 ```
41
42 In the previous code chunk, we passed four vector objects as "named arguments" to the 'tibble' function, resulting in columns such as 'casualty_type'.
43 A 'tibble' is just a fancy way of representing 'data.frame' objects, preferred by 'tidyverse' users and optimised for data science.
44 It has a few sensible defaults and advantages compared with the 'data.frame', one of which can be seen by printing a 'tibble':
45
46 ```{r}
47 class(crashes)
48 crashes
49 ```
50
51 Note the '<chr>', '<dbl>' or '<lgl>' text below each column, providing a quick indication of the class of each variable - this is not provided when using 'data.frame'.
52
53 ## filter() and select() rows and columns
54
55 In the previous section, we briefly introduced the package 'dplyr', which provides an alternative to base R for manipulating objects. 'dplyr' provides different, and some woul
56 'dplyr' operations for subsetting rows (with the function 'filter()') and columns (with the function 'select()') are demonstrated below. Here we can also see the use of the p
```

```
IBM x IBM-6145-1662204644 x IBM-EPBL/IBM-Project-6145-1662204644 x +
github.com/IBM-EPBL/IBM-Project-6145-1662204644/blob/main/Project%20Development%20Phase/Project%20Development-Delivery%20of%20Sprint...
Gmail YouTube Maps

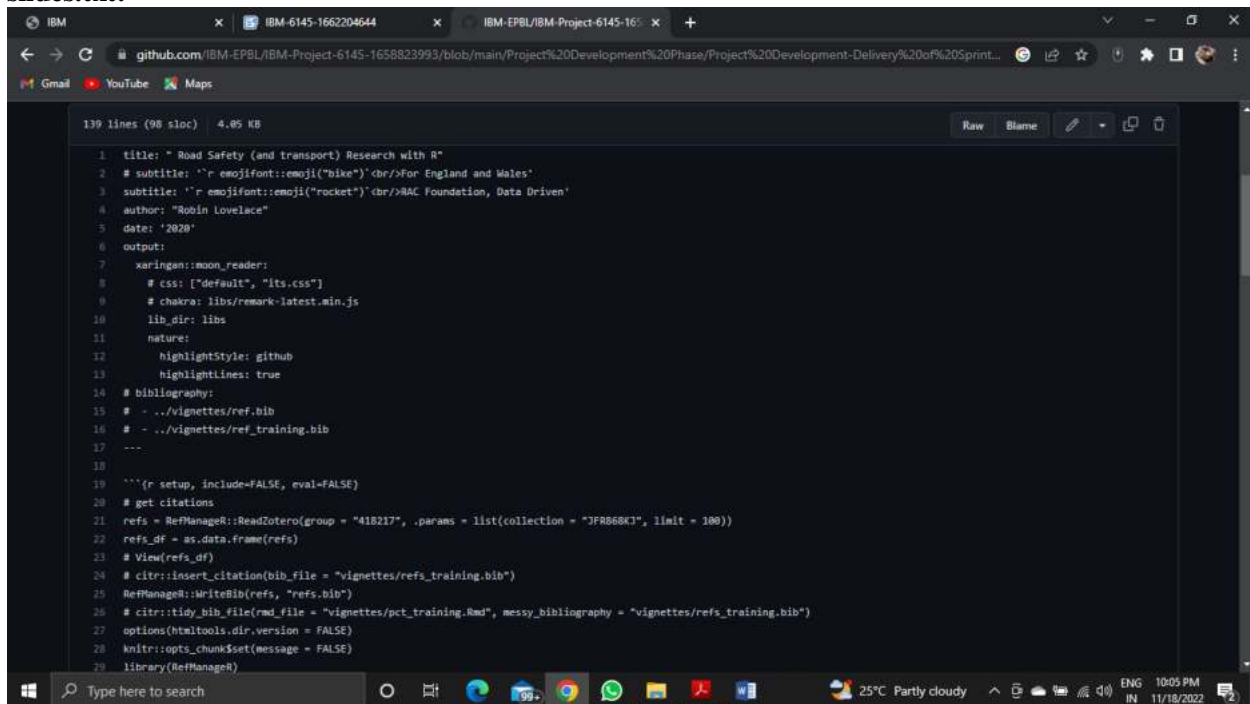
53 dim(crashes)
54 dim(crashes_darkness)
55 ?contains # get help on contains() to help guess the output of the next line
56 dim(crashes_a)
57 dim(crashes_darkness_a)
58 ```
59
60 Look at the help pages associated with 'filter()', 'select()' and the related function 'slice()' as follows and try running the examples that you will find at the bottom of th
61
62 ```{r, eval=FALSE}
63 %>% filter
64 %>% select
65 %>% slice
66 ```
67
68 ## Ordering and selecting the 'top n'
69
70 Other useful pipe-friendly functions are 'arrange()' and 'top_n()'. 'arrange()' can be used to sort data. Within the 'arrange()' function, optional arguments can be used to de
71 We can use these functions to arrange datasets and take the top most 'n' values, as follows:
72
73 ```{r}
74 crashes %>%
75   arrange(vehicle_type)
76 crashes %>%
77   top_n(n = 1, wt = casualty_age)
78 ```
79
80 <!-- ## Long and wide data -->
81
82 ## Summarise
```

Signs with Smart Connectivity for Better Road Safety



```
122 another useful feature of the 'tidyverse' from a user perspective is the auto-completion of column names with pipes.  
123 If you have not noticed this already, you can test it by typing the following, putting your cursor just before the ')' and pressing 'Tab':  
124  
125 ```{r, eval=FALSE}  
126 crashes %>% select(ca) # press Tab when your cursor is just after the a  
127 ```  
128  
129 You should see 'casualty_age' and 'casualty_type' pop up as options that can be selected by pressing 'Up' and 'Down'.  
130 This may not seem like much, but when analysing large datasets with dozens of variables, it can be a godsend.  
131  
132 Rather than providing a comprehensive introduction to the 'tidyverse' suite of packages, this section should have offered enough to get started with using it for road safety d  
133 For further information, check out up-to-date online courses from respected organisations like [Data Carpentry](https://datacarpentry.org/R-ecology-lesson/index.html) and the  
134  
135 ## Tidyverse exercises  
136  
137 1. Use 'dplyr' to filter rows in which 'casualty_age' is less than 18, and then 28.  
138 2. Use the 'arrange' function to sort the 'crashes' object in descending order of age (**Hint:** see the 'arrange' help page).  
139 3. Read the help page of 'dplyr::mutate()'. What does the function do?  
140 4. Use the mutate function to create a new variable, 'birth_year', in the 'crashes' data.frame which is defined as the current year minus their age.  
141 5. **Bonus:** Use the '%>%' operator to filter the output from the previous exercise so that only observations with 'birth_year' after 1969 are returned.  
142  
143 ```{r dplyr, eval=FALSE, echo=FALSE}  
144 # answers  
145 crashes %>%  
146   arrange(desc(casualty_age))  
147 crashes %>% filter(casualty_age > 21)  
148 crashes %>%  
149   mutate(birth_year = 2019 - casualty_age) %>%  
150   filter(birth_year > 1969)  
151 ```
```

slides.txt:



```
139 lines (98 sloc) 4.05 KB  
Raw Blame  
1 title: " Road Safety (and transport) Research with R"  
2 # subtitle: "r emoji::emoji('bike')<br/>For England and Wales"  
3 subtitle: "r emoji::emoji('rocket')<br/>RAC Foundation, Data Driven"  
4 author: "Robin Lovelace"  
5 date: "2020"  
6 output:  
7   xaringan::moon_reader:  
8     # css: ["default", "its.css"]  
9     # chakra: libs/remark-latest.min.js  
10    lib_dir: libs  
11    nature:  
12      highlightStyle: github  
13      highlightLines: true  
14    # bibliography:  
15    # - ../vignettes/ref.bib  
16    # - ../vignettes/ref_training.bib  
17    ---  
18  
19    ```{r setup, include=FALSE, eval=FALSE}  
20    # get citations  
21    refs = RefManager::ReadZotero(group = "418217", .params = list(collection = "JFR868KJ", limit = 100))  
22    refs_df = as.data.frame(refs)  
23    View(refs_df)  
24    # citr::insert_citation(bib_file = "vignettes/refs_training.bib")  
25    RefManager::WriteBib(refs, "refs.bib")  
26    # citr::tidy_bib_file(rmd_file = "vignettes/pct_training.Rmd", messy_bibliography = "vignettes/refs_training.bib")  
27    options(htmltools.dir.version = FALSE)  
28    knitr::opts_chunk$set(message = FALSE)  
29    library(RefManager)
```

Signs with Smart Connectivity for Better Road Safety

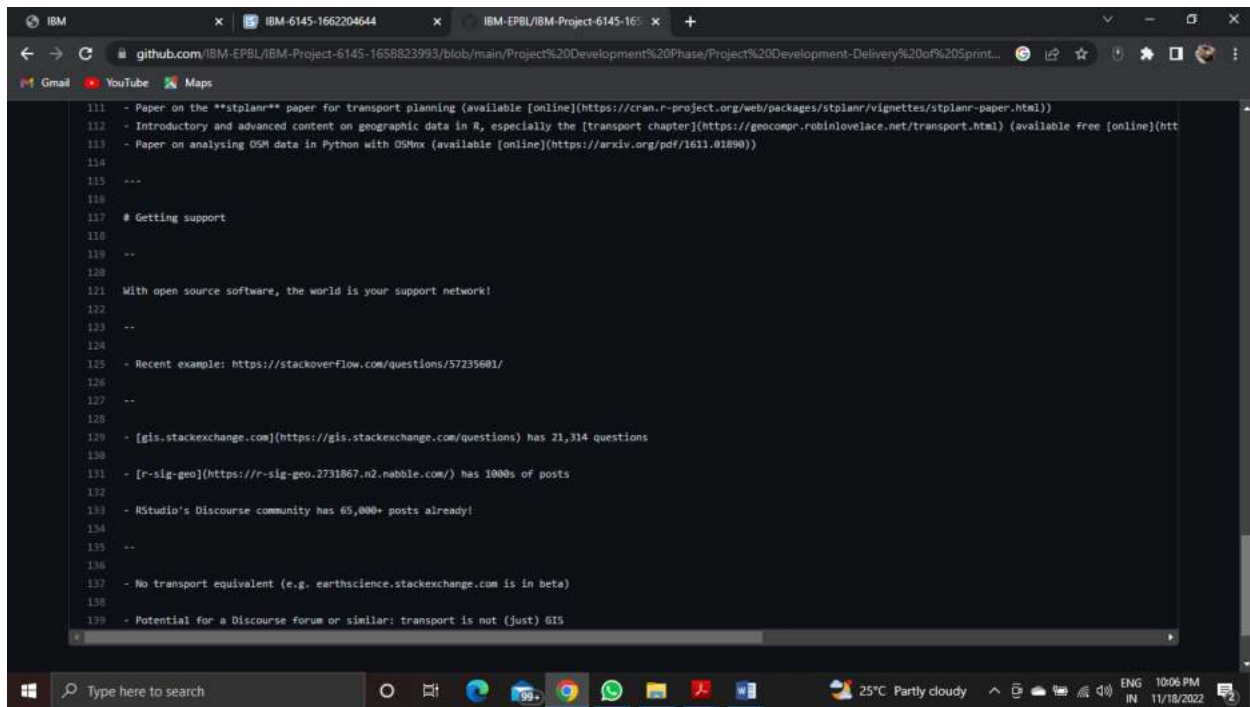
```
IBM  x  IBM-6145-1662204644  x  IBM-EPBL/IBM-Project-6145-16:  x  +
github.com/IBM-EPBL/IBM-Project-6145-1658823993/blob/main/Project%20Development%20Phase/Project%20Development-Delivery%20of%20Sprint...
Gmail  YouTube  Maps

29  library(R/Manager)
30  BibOptions(check.entries = FALSE,
31             bib.style = "authoryear",
32             cite.style = "alphabetic",
33             style = "markdown",
34             first.inits = FALSE,
35             hyperlink = FALSE,
36             dashed = FALSE)
37  my_bib = refs
38  ""
39
40
41  ""{r, eval=FALSE, echo=FALSE, engine="bash"}
42  # publish results online
43  cp -Rv code/rrrr-slides* ~/saferactive/site/static/slides/
44  cp -Rv code/libs ~/saferactive/site/static/slides/
45  cd ~/saferactive/site
46  git add -A
47  git status
48  git commit -am 'Update slides'
49  git push
50  cd -
51  ""
52
53  # Slide/links
54
55  https://itsleeds.github.io/rrrr/
56
57  https://bookdown.org/
58
59  https://www.pct.bike/
60
```

```
IBM  x  IBM-6145-1662204644  x  IBM-EPBL/IBM-Project-6145-16:  x  +
github.com/IBM-EPBL/IBM-Project-6145-1658823993/blob/main/Project%20Development%20Phase/Project%20Development-Delivery%20of%20Sprint...
Gmail  YouTube  Maps

65  # Coding
66
67  Ideal:
68
69  ""{r, eval=FALSE}
70  od_test$perc_cycle = round(od_test$bicycle / od_test$all) * 100
71  l = od_to_sf(od_test, od_date_centroids)
72  r = stplanr::route(l = l, route_fun = journey)
73  rnet = overline(r, "bicycle")
74  ""
75
76  --
77
78  
79
80  Reality
81
82  ---
83
84
85  ## Transport software - which do you use?
86
87  ""{r, echo=FALSE, message=FALSE, warning=FALSE}
88  u = "https://github.com/ITSLeeds/TDS/raw/master/transport-software.csv"
89  tms = readr::read_csv(u)[1:5]
90  tms = dplyr::arrange(tms, dplyr::desc(Citations))
91  knitr::kable(tms, booktabs = TRUE, caption = "Sample of transport modelling software in use by practitioners. Note: citation counts based on searches for company/developer nam
92  ""
93
94  ---
95
96  ## Data sources and the different
```

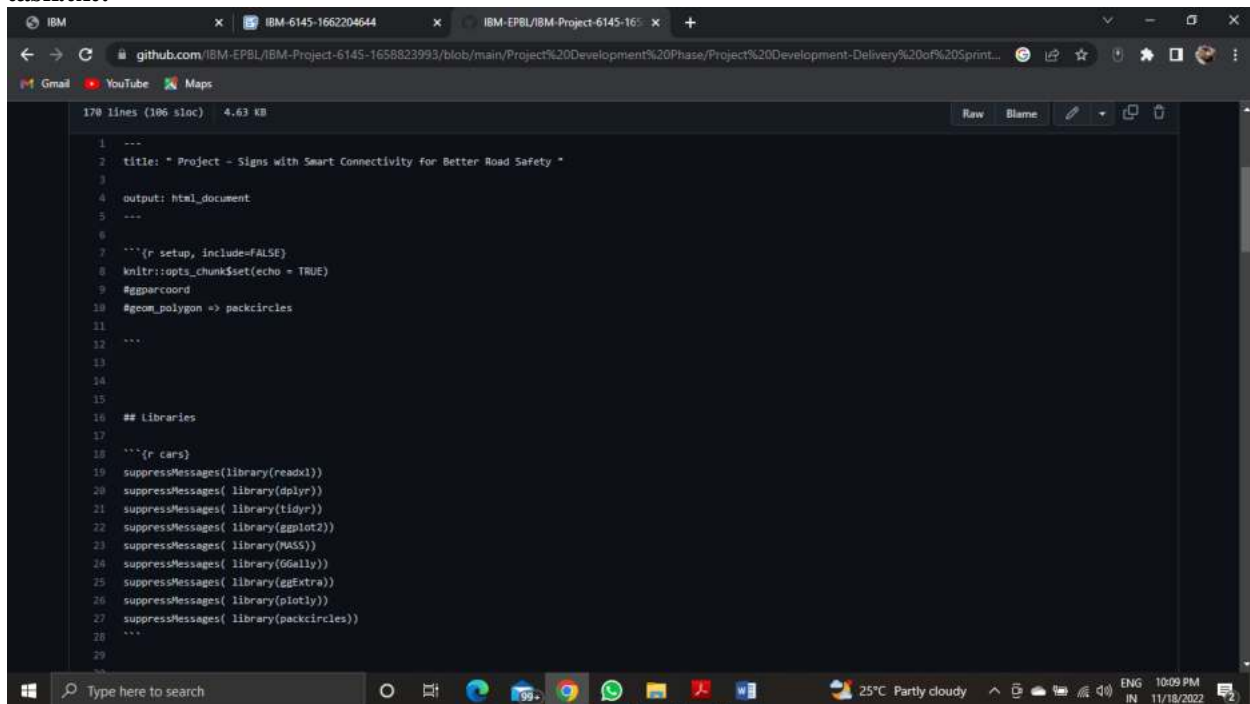

Signs with Smart Connectivity for Better Road Safety



The screenshot shows a web browser window displaying a GitHub repository page. The URL is github.com/IBM-EPBL/IBM-Project-6145-1658823993/blob/main/Project%20Development%20Phase/Project%20Development-Delivery%20of%20Sprint.... The page shows a README file with the following content:

```
111 - Paper on the stplanr paper for transport planning (available [online](https://cran.r-project.org/web/packages/stplanr/vignettes/stplanr-paper.html))
112 - Introductory and advanced content on geographic data in R, especially the [transport chapter](https://geocompr.robinlovelace.net/transport.html) (available free [online](ht
113 - Paper on analysing OSM data in Python with OSMnx (available [online](https://arxiv.org/pdf/1611.01890))
114
115 ---
116
117 # Getting support
118
119 --
120
121 With open source software, the world is your support network!
122
123 --
124
125 - Recent example: https://stackoverflow.com/questions/57235601/
126
127 --
128
129 - [gis.stackexchange.com](https://gis.stackexchange.com/questions) has 21,314 questions
130
131 - [r-sig-geo](https://r-sig-geo.2731867.n2.nabble.com/) has 1000s of posts
132
133 - RStudio's Discourse community has 65,000+ posts already!
134
135 --
136
137 - No transport equivalent (e.g. earthscience.stackexchange.com is in beta)
138
139 - Potential for a Discourse forum or similar: transport is not (just) GIS
```

task.txt:



The screenshot shows a web browser window displaying a GitHub repository page. The URL is github.com/IBM-EPBL/IBM-Project-6145-1658823993/blob/main/Project%20Development%20Phase/Project%20Development-Delivery%20of%20Sprint.... The page shows a task.txt file with the following content:

```
170 lines (106 sloc) 4.63 KB
Raw Blame

1 ---
2 title: " Project - Signs with Smart Connectivity for Better Road Safety "
3
4 output: html_document
5 ---
6
7 ```{r setup, include=FALSE}
8 knitr::opts_chunk$set(echo = TRUE)
9 #ggparcoord
10 #geom_polygon => packcircles
11
12 ---
13
14
15
16 ## Libraries
17
18 ```{r cars}
19 suppressMessages(library(readxl))
20 suppressMessages( library(dplyr))
21 suppressMessages( library(tidyrr))
22 suppressMessages( library(geomplot2))
23 suppressMessages( library(MASS))
24 suppressMessages( library(ggally))
25 suppressMessages( library(ggExtra))
26 suppressMessages( library(plotly))
27 suppressMessages( library(packcircles))
28
29 ---
```

Signs with Smart Connectivity for Better Road Safety

```
30
31
32 ## Preparing The Data For Analysis
33 ```{r}
34 df <- read_excel("mAll.xlsx")
35 #d13 <- read_excel("m_m2013.xlsx")
36 #d14 <- read_excel("m_m2014.xlsx")
37 #d15 <- read_excel("m_m2015.xlsx")
38 #d16 <- read_csv("m_m2016.csv", header = 1)
39 #d17 <- read_excel("m_m2017.xlsx")
40 #d18 <- read_excel("m_m2018.xlsx")
41
42
43 names(df)[9] <- "Vehicle and Safety Equipment"
44 names(df)[10] <- "Vehicle and Safety Equipment N"
45 names(df)[22] <- "Chassis and Body N"
46 names(df)[26] <- "Suspension Test N"
47 names(df)[36] <- "Incomplete Tests N"
48
49 df$reportYear <- as.factor(df$reportYear)
50 ```
51
52
53
54
55 ## Which part failed the most per report year?
56 ```{r, echo=FALSE}
57 #####DATA#####
58 cols <- c("VehicleMake", "Vehicle and Safety Equipment", "Lighting and Electrical", "Steering and Suspension", "Braking Equipment", "Wheels and Tyres", "Engine, Noise and Exha
59
60 m <- df %>% dplyr :: select(c("reportYear", cols)) %>% group_by(reportYear) %>% summarise_if(is.numeric, mean, na.rm = TRUE)
61 # m <- gather(m, reportYear, key=Part, value= Failures)
```

```
57 #####DATA#####
58 cols <- c("VehicleMake", "Vehicle and Safety Equipment", "Lighting and Electrical", "Steering and Suspension", "Braking Equipment", "Wheels and Tyres", "Engine, Noise and Exha
59
60 m <- df %>% dplyr :: select(c("reportYear", cols)) %>% group_by(reportYear) %>% summarise_if(is.numeric, mean, na.rm = TRUE)
61 m <- gather(m, reportYear, key=Part, value= Failures)
62
63 ```
64
65
66 ```{r}
67 #####PLOT#####
68 #ggplot(m, aes(x=factor(reportYear), y=, colour=supp, group=supp)) + geom_line()
69
70 library(MASS)
71 library(Ggally)
72 # Vector color
73 library(RColorBrewer)
74 palette <- brewer.pal(3, "Set1")
75 my_colors <- palette(as.numeric(as.factor(reportYear)))
76
77 #names(x) <- c("2013", "2014", "2015", "2016", "2017", "2018")
78 #p <- ggparcoord(m, columns=2:13, groupColumn = "reportYear")+geom_line(size=0.3)+theme_minimal() + geom_point()+
79 # xlab("Car Part")+ylab("Average failure rate")
80
81
82 ggplotly(ggplot(data=m, mapping = aes(x = reportYear, y = Failures, colour = Part, group=1))+
83   geom_point()+
84   geom_line()+xlab("Report Year")+ ylab("Average Number of Failures")
85 )
86 ```
87
```

Signs with Smart Connectivity for Better Road Safety

```
140 ```{r}
141 z <- df %>% group_by(VehicleMake) %>% summarise(tot=sum(Total),res = sum(PASS)/sum(Total)) %>% arrange(desc(tot)) %>% print(Inf())
142 ```
143
144
145 ```{r}
146
147 require(scales)
148 q <- z %>% arrange(desc(tot)) %>% slice(1:15)
149
150 ggplot(q)+
151   geom_col(mapping = aes(x = reorder(VehicleMake, -tot), y = tot, fill = "green"))+ xlab("Vehicle Make")+ylab("Number of Vehicles") + coord_flip() + theme(legend.position = "no
152   scale_y_continuous(labels = comma)
153
154   #marginal(g, type = "histgram", fill="transparent")
155   ```
156
157
158
159
160 ## Pass Percentage versus Number of Vehicles for a given VehicleMake
161 ```{r}
162 require(scales)
163 library(plotly)
164 p <- ggplot(q, aes(x = tot, y = res*100))+
165   geom_line(color = "red")+
166   geom_point(aes(text = VehicleMake))+xlab("Number of Vehicles") + ylab("Pass Percentage (%)") +
167   scale_x_continuous(labels = comma)
168
169 ggplotly(p, tooltip = "text")
170 ```
```

test center.txt:

```
95 lines (82 sloc) 6.24 KB
Raw Blame

1 ---
2 pagetitle: "Analysis based on test centres"
3 author: "Amol Nansware"
4 date: "06/12/2019"
5 always_allow_html: true
6 output:
7   html_document: default
8 ---
9
10 ```{r setup, include=FALSE}
11 knitr::opts_chunk$set(echo = TRUE)
12 ```
13
14 ```{r, warning=FALSE, echo=FALSE, include=FALSE}
15 packages <- c("plotly", "tidyverse")
16 newPackages <- packages[!(packages %in% installed.packages()[,"Package"])]
17 if(length(newPackages)) install.packages(newPackages)
18 library(tidyverse)
19 library(plotly)
20 ```
21
22 ```{r,echo=FALSE}
23 load("passfail.RData")
24 passfail <- passfail %>%
25   mutate(totalFails = Fail1 + ifelse(is.na(Fail2), 0, Fail2), Totalpass = Pass1 + ifelse(is.na(Pass2), 0, Pass2))
26 ```
27
28
```

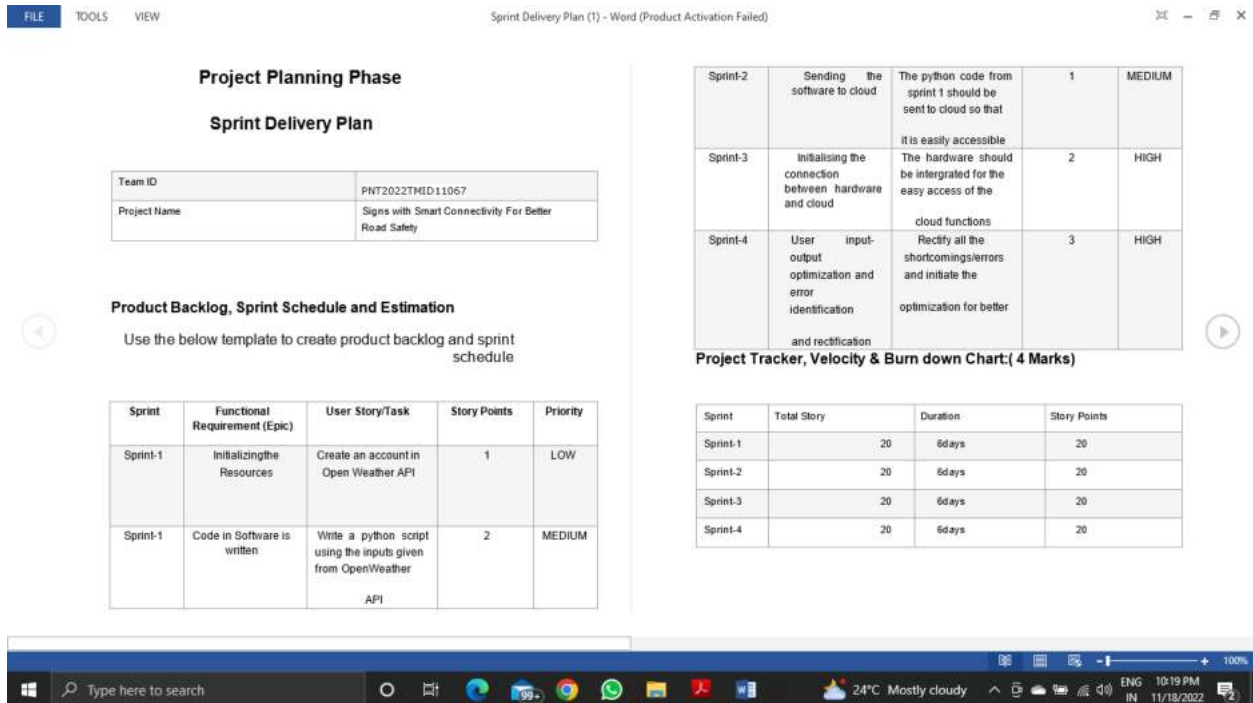
Signs with Smart Connectivity for Better Road Safety

```
23 load("passfail.RData")
24 passfail <- passfail %>%
25   mutate(totalFails = Fail1 + ifelse(is.na(Fail2), 0, Fail2), Totalpass = Pass1 + ifelse(is.na(Pass2), 0, Pass2))
26   ""
27
28
29   ""{r,echo=FALSE}
30 passfailGroup <- summarise(group_by(passfail, Centre), Pass1 = sum(Pass1), Fail1 = sum(Fail1), Total1 = sum(Total1), Pass2 = sum(Pass2, na.rm = T), Fail2 = sum(Fail2, na.rm =
31 passfailGroup <- mutate(passfailGroup, Pass1prop = Pass1/Total1, Pass2prop = Pass2/Total2, totalPassProp = (Totalpass / (Total1 + Total2)), totalFailsProp = (totalFails / (Tot
32   ""
33
34
35   ""{r,echo=FALSE}
36 passfailGroup$totalPassProp = round((passfailGroup$totalPassProp * 100), digits = 2)
37 passfailGroup$totalFailsProp = round((passfailGroup$totalFailsProp * 100), digits = 2)
38 passfailGroup1 <- passfailGroup[c(1, 8)]
39 passfailGroup1$Test <- "Pass"
40 names(passfailGroup1) <- c("Centre", "Count", "Test")
41 passfailGroup2 <- passfailGroup[c(1, 9)]
42 passfailGroup2$Test <- "Fail"
43 names(passfailGroup2) <- c("Centre", "Count", "Test")
44 passfailcount <- rbind(passfailGroup1, passfailGroup2)
45   ""
46
47
48   ## Analysis based on test centres
49   <br/>In this section we will analyse data from 2013 till 2018 about each test centre. As shown in the <a href = "https://github.com/NanawareMol/R-project_Road-safety/blob/mas
50 The bar chart shows the total number of tests that each centre performed and the total pass and fail counts as well as percentages. So, based on the test counts, the top 3 tes
51
52   ""{r,echo=FALSE, fig.width=9,fig.height=4}
53 p <- plot_ly(passfailGroup, x = -passfailGroup$Centre, y = -passfailGroup$Totalpass, type = 'bar', name = 'Pass', text = paste("Total tests = ", (passfailGroup$Totalpass+passf
```

```
68   ""{r,echo=FALSE, fig.width=9,fig.height=4}
69 #scatter plot for centre total pass per year
70 ggplot(data = passfail, aes(x = fct_reorder(Centre, -Totalpass), y = Totalpass, color = Year, size = Totalpass)) + geom_point(alpha = 0.5) +
71   theme(axis.text.x = element_text(size=9, angle=90, hjust = 0, vjust = 0.5), axis.ticks.x = element_blank(), panel.background = element_rect(fill = "white", colour = "lightb
72   panel.grid.minor = element_line(size = 0.5, linetype = 'solid', colour = "lightblue")) +
73   labs(x = "Test Centres", y = "Total pass count")
74   ""
75
76   ##<b> <br>Test performance for each test centre<br>
77 The graph gives the overall idea of the test performance based on pass rate and the year.
78 As per the graph we can say that for year 2013, 2015, 2016, 2017 and 2018, the pass rate is higher than 55%. And the highest and lowest performance found in Kilkenny and Monag
79   ""{r,echo=FALSE, fig.width=9,fig.height=4}
80 passfail$totalPassPercentage <- round((passfail$totalpass / (passfail$totalpass + passfail$totalFails)) * 100, digits = 2)
81 passfail$totalFailPercentage <- round((passfail$totalFails / (passfail$totalpass + passfail$totalFails)) * 100, digits = 2)
82 #scatter plot for centre pass percentage per year
83 ggplot(data = passfail, aes(x = fct_reorder(Centre, -totPassPercentage), y = totPassPercentage, color = Year, size = totPassPercentage)) + geom_point(alpha = 0.5) +
84   theme(axis.text.x = element_text(size=9, angle=90, hjust = 0, vjust = 0.5), axis.ticks.x = element_blank(), panel.background = element_rect(fill = "white", colour = "lightb
85   panel.grid.minor = element_line(size = 0.5, linetype = 'solid', colour = "lightblue")) +
86   labs(x = "Test Centres", y = "Total Pass %")<br>title = "Test centre pass% per year",
87   ""
88
89   ##<b> <br>Total pass count limits per year<br>
90 The box plot shows the total pass count against each year. With this we can fetch the details on maximum and minimum pass counts per year, the median pass count and the ousta
91   ""{r,echo=FALSE, fig.width=9,fig.height=4}
92 p <- plot_ly(passfail, x = passfail$Year, y = passfail$totalpass, color = -passfail$Year, type = "box", text = paste("Centre = ", passfail$Centre)) %>%
93   layout(title = "Yearly performance", yaxis = list(title = "Total Pass Count"), xaxis = list(title = "Year"))
94 p
95   ""
```

PROJECT DEVELOPMENT -DELIVERY OF SPRINT 4:

OUTPUT:



Signs with Smart Connectivity for Better Road Safety



sprint 4.txt:

```
89 lines (89 sloc) 2.76 KB
1 import wiotp.sdk.device
2 import time
3 import random
4 import ibmiotf.application
5 import ibmiotf.device
6 import requests, json
7 myConfig = {
8     #Configuration
9     "identity": {
10         "orgid": "gilemex",
11         "typeid": "NodeMCU",
12         "deviceId": "12345678"
13     },
14     #API Key
15     "auth": {
16         "token": "1234567890"
17     }
18 }
19 #Receiving callbacks from IBM IoT platform
20 def myCommandCallback(cmd):
21     print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
22     m=cmd.data['command']
23     client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
24     client.connect()
25 #OpenWeatherMap Credentials
26 BASE_URL = "https://api.openweathermap.org/data/2.5/weather?"
27 CITY = "TRIPOLY IN"
```

Signs with Smart Connectivity for Better Road Safety

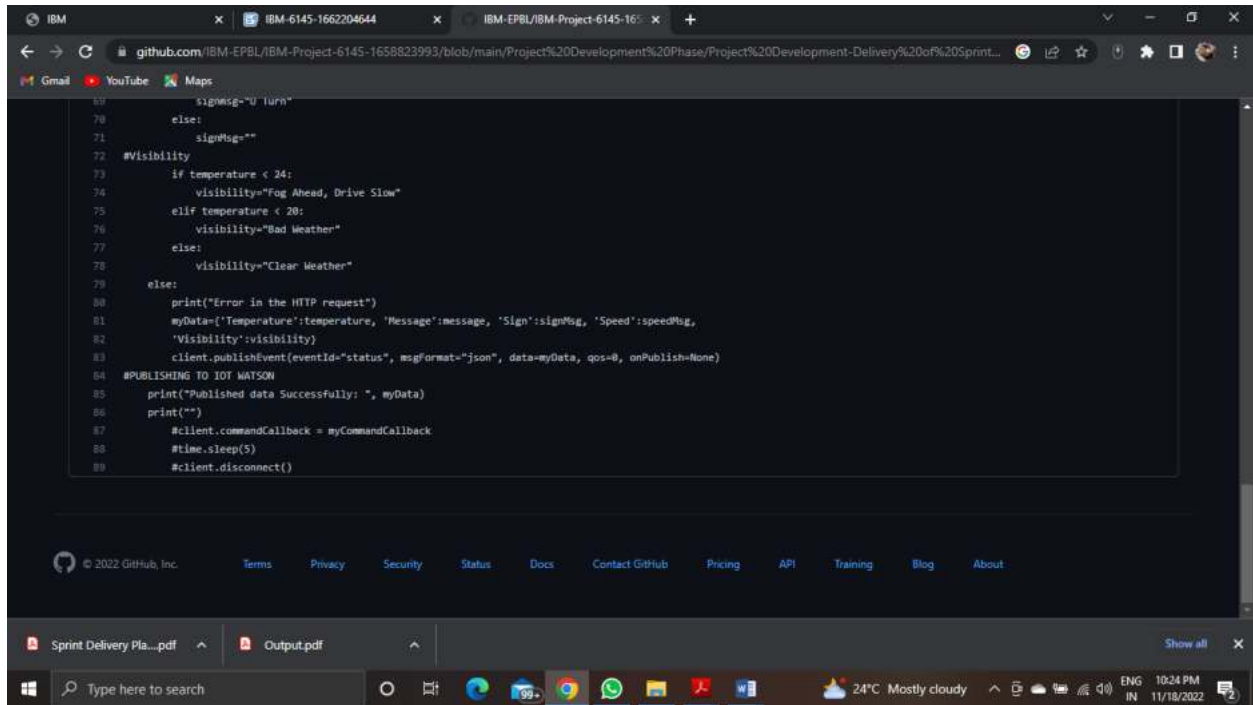
```
IBM  x  IBM-6145-1662204644  x  IBM-EPBL/IBM-Project-6145-1658823993  x  +
github.com/IBM-EPBL/IBM-Project-6145-1658823993/blob/main/Project%20Development%20Phase/Project%20Development-Delivery%20of%20Sprint...
Gmail  YouTube  Maps

21  print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
22  m=cmd.data['command']
23  client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
24  client.connect()
25  #OpenWeatherMap Credentials
26  BASE_URL = "https://api.openweathermap.org/data/2.5/weather?"
27  CITY = "TRICHY, IN"
28  URL = BASE_URL + "q=" + CITY + "&units=metric"&"&appid=" + "4d132e9b3e4b0c44c7027fda6d91c486"
29  while True:
30      response = requests.get(URL)
31      if response.status_code == 200:
32          data = response.json()
33          main = data['main']
34          temperature = main['temp']
35          humidity = main['humidity']
36          pressure = main['pressure']
37          report = data['visibility']
38  #message part
39  msg=random.randint(0,5)
40  if msg==1:
41      message="GO SLOW, SCHOOL ZONE AHEAD"
42  elif msg==2:
43      message="NEED HELP, POLICE STATION AHEAD"
44  elif msg==3:
45      message="EMERGENCY, HOSPITAL NEARBY"
46  elif msg==4:
47      message="DINE IN, RESTAURENT AVAILABLE"
48  elif msg==5:
```

```
IBM  x  IBM-6145-1662204644  x  IBM-EPBL/IBM-Project-6145-1658823993  x  +
github.com/IBM-EPBL/IBM-Project-6145-1658823993/blob/main/Project%20Development%20Phase/Project%20Development-Delivery%20of%20Sprint...
Gmail  YouTube  Maps

50      else:
51          message=""
52  #Speed Limit part
53  speed=random.randint(0,150)
54  if speed>100:
55      speedMsg=" Limit Exceeded"
56  elif speed>60 and speed<100:
57      speedMsg="Moderate"
58  else:
59      speedMsg="Slow"
60  #Diverision part
61  sign=random.randint(0,5)
62  if sign==1:
63      signMsg="Right Diverision"
64  elif sign==2:
65      signMsg="Speed Breaker"
66  elif sign==3:
67      signMsg="Left Diverision"
68  elif sign==4:
69      signMsg="U Turn"
70  else:
71      signMsg=""
72  #visibility
73  if temperature < 24:
74      visibility="Fog Ahead, Drive Slow"
75  elif temperature < 20:
76      visibility="Bad Weather"
77  else:
```


Signs with Smart Connectivity for Better Road Safety



The screenshot shows a web browser window displaying a GitHub repository for an IBM project. The repository name is 'IBM-EPBL/IBM-Project-6145-1658823993'. The code is written in Python and is part of a project for smart road signs. The code logic checks for temperature and visibility conditions to determine the appropriate sign message. It also includes error handling and a command callback function. The browser window shows several tabs, including 'IBM', 'IBM-6145-1662204644', and 'IBM-EPBL/IBM-Project-6145-1658823993'. The Windows taskbar at the bottom shows the search bar, task view button, and several application icons including Edge, Chrome, WhatsApp, and a file explorer. The system tray shows the date and time as 10:24 PM on 11/18/2022, along with weather information (24°C, Mostly cloudy).

```
67 signMsg="U turn"
68 else:
69     signMsg=""
70 #Visibility
71 if temperature < 24:
72     visibility="Fog Ahead, Drive Slow"
73 elif temperature < 20:
74     visibility="Bad Weather"
75 else:
76     visibility="Clear Weather"
77 else:
78     print("Error in the HTTP request")
79     myData={'Temperature':temperature, 'Message':message, 'Sign':signMsg, 'Speed':speedMsg,
80           'Visibility':visibility}
81     client.publishEvent(eventId="status", msgformat="json", data=myData, qos=0, onPublish=None)
82 #PUBLISHING TO IOT WATSON
83 print("Published data Successfully: ", myData)
84 print("")
85 #client.commandCallback = myCommandCallback
86 #time.sleep(5)
87 #client.disconnect()
```

<https://youtu.be/O8FK0HCEGzQ>

Signs with Smart Connectivity for Better Road Safety