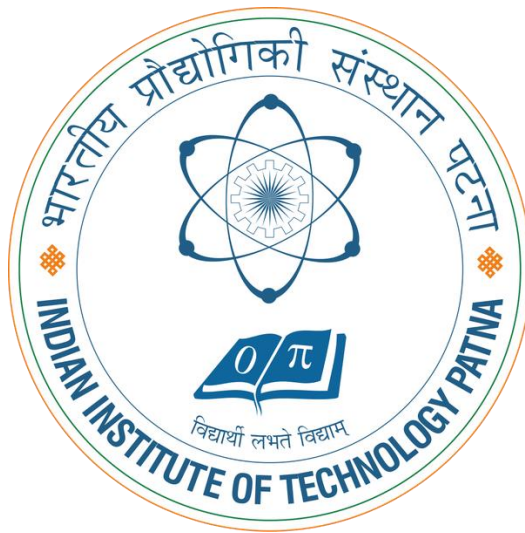


# INDIAN INSTITUTE OF TECHNOLOGY PATNA



## Workshop Report On

### “ Fundamental Of Internet Of Things”

#### Workshop Cordinator

Name: Yashwant Singh Gour  
Branch: IOT Department Of IT  
College: Samrat Ashok Technological Institute  
Vidisha  
Enrollment No: 0108IO221068

**Dr. R.K. BEHRA SIR**

B.Tech: NIT Rourkela

M.tech & P.hD: IITK

PD: University of Pretoria, South Africa

Associate Professor - Dept. Of EE IITP

Visiting Scholar - The Center for Energy Systems,  
Tennessee Technological University, Cookeville, USA

Project Associate - IITK

Trainee Engineer -Hindustan Aeronautics Limited ( HAL)

## **Abstract:**

"Fundamentals of IoT" training and workshop program at the Indian Institute of Technology, Patna, offered an immersive exploration into the transformative realm of IoT technology. Under the expert guidance of Dr. RK Behera, an esteemed Associate Professor in the Department of Electrical Engineering, participants delved into the captivating world of IoT, discovering its potential to revolutionize daily lives and industries.

The program fostered collaborative learning as students from diverse backgrounds came together to tackle real-world challenges using advanced technologies such as Raspberry Pi and sensors. Mentored by Mr. NILABH ANAND from Texas Instruments, participants engaged in hands-on projects, applying IoT principles to solve intricate problems.

Dr. RK Behera conducted an illuminating workshop, providing a comprehensive understanding of the IoT landscape, from its origins to devices, gateways, edge computing, internet connectivity, cloud storage, analytics, and visualization. Witnessing impactful projects developed by Dr. Behera and his team inspired attendees, showcasing the practical applications of IoT across industries.

The learning experience extended beyond IoT, encompassing a workshop on web development guided by Mr. Varinder, enabling participants to create innovative web applications using Django. Additionally, Mr. Priyesh Saini, a distinguished Research Scholar at IIT Patna, shared valuable insights into data analysis using machine learning techniques, empowering attendees to extract meaningful insights from complex datasets.

This report encapsulates the knowledge, skills, and inspiration gained during the workshop, highlighting the practical applications of IoT, web development, and data analysis techniques. Participants left equipped with a deep understanding of these technologies, ready to make meaningful contributions to the rapidly evolving digital landscape.

## **Content:**

### **1. Analysing Data Using Machine Learning**

(By Priyesh Saini Sir Research scholar @ IITP)

### **2. Introduction To IOT And Deep Discussion On Key Concept**

(By Dr. RK Behra Associate Professor @ IITP)

### **3. Raspberry Pi Workshop**

(By Mr. Nilabh Anand Sir @Texas Instruments)

### **4. Breif Introduction Of Web Devlopment Using Django Framework**

(By Mr. Varinder sir)

### **5. Workshop Post Session**

(By Mr. Yesveer Singh Yadav Technical Lead @IndiaMart)

# Objectives of the Workshop:

- 1. Understanding IoT Fundamentals:** The workshop aimed to provide participants with a solid understanding of the fundamental concepts of IoT, including its origins, components, and the intricate network of devices, gateways, and cloud infrastructure.
- 2. Hands-On Experience:** Participants were given hands-on experience with cutting-edge technologies such as Raspberry Pi and sensors. The objective was to enable attendees to apply theoretical knowledge in practical settings, fostering a deeper understanding of IoT implementation.
- 3. Problem-Solving Skills:** Through collaborative group projects, participants were encouraged to apply IoT principles to solve real-world challenges. This objective aimed to enhance problem-solving skills and promote innovative thinking among the attendees.
- 4. Comprehensive Knowledge:** The workshop aimed to cover the entire IoT landscape, including topics like edge computing, internet connectivity, cloud storage, data analytics, and visualization. Participants were expected to gain comprehensive knowledge of IoT technologies and their integration into various domains.
- 5. Inspiration and Motivation:** By showcasing impactful projects developed by experts in the field, the workshop aimed to inspire participants. Witnessing the practical applications of IoT in industries served as motivation, encouraging attendees to explore innovative solutions in their future endeavors.
- 6. Web Development Skills:** In addition to IoT, the workshop included a web development session using Django. The objective was to equip participants with practical skills in creating web applications, expanding their technical expertise beyond IoT.
- 7. Data Analysis Proficiency:** Participants had the opportunity to learn about data analysis techniques using machine learning. The objective was to empower attendees with the ability to extract meaningful insights from complex datasets, enhancing their analytical skills.
- 8. Networking and Collaboration:** The workshop provided a platform for participants to collaborate with talented students from diverse backgrounds. Networking opportunities encouraged the exchange of ideas, fostering a collaborative learning environment among attendees.
- 9. Empowering Future Contributions:** Ultimately, the workshop aimed to empower participants with the knowledge, skills, and motivation needed to contribute meaningfully to the fields of IoT, web development, and data analysis. Participants were encouraged to apply their learnings in real-world scenarios, driving innovation and progress in their respective domains.

# **#1 Workshop Report**

## **Analyzing Data Using Machine Learning**

Date: 06/07/2023

Presenter: **PRIYESH SAINI SIR**

### **1. Introduction:**

The workshop focused on analyzing data using machine learning techniques, with a particular emphasis on the DeepAR machine learning algorithm. Additionally, participants were introduced to Google Colab, a collaborative platform for coding, and discussed various machine learning problems, including supervised regression and classification, as well as unsupervised clustering. This report provides an overview of the workshop topics and key takeaways.

### **2. Google Colab:**

The workshop began by introducing Google Colab, an online platform for writing and executing Python code. Participants were instructed to open Google Colab and upload their code and data files, including a CSV file for styling purposes.

### **3. Machine Learning Problems:**

The workshop covered three main types of machine learning problems: regression, classification, and clustering.

#### **1. Regression:**

Regression involves predicting a continuous variable based on input features. It is useful for scenarios where the target variable is numeric and varies over a continuous range.

#### **2. Classification:**

Classification is employed when the goal is to assign data instances into predefined classes or categories. Examples discussed during the workshop included disease detection through classification and email classification between spam and important messages.

#### **3. Clustering:**

Clustering is an unsupervised learning problem that involves grouping similar data instances together based on their characteristics. The workshop did not provide specific details about the data used for clustering analysis.

## 1. Continuous and Discrete Variables:

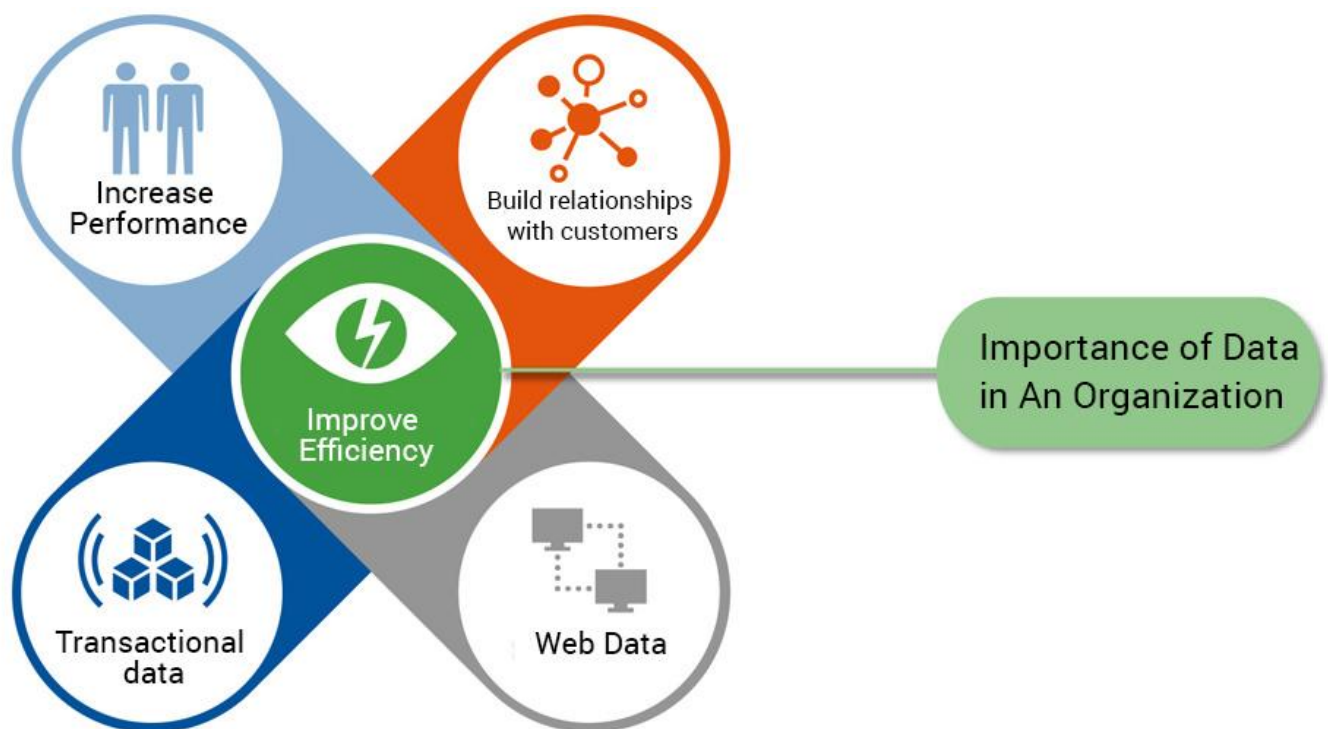
The workshop highlighted the distinction between continuous and discrete variables. Continuous variables can take any value within a range, while discrete variables only have specific values. Understanding this distinction is crucial for selecting appropriate machine learning algorithms and preprocessing techniques.

## 2. Reinforcement Learning:

The concept of reinforcement learning was briefly introduced during the workshop. It was explained that in reinforcement learning, positive feedback is given when the model produces good results, while negative feedback is provided for suboptimal outcomes. This analogy was likened to dog training with treats, where desired behavior is rewarded.

## 3. Importance of Data:

The workshop emphasized the significance of data in the machine learning process. Participants were encouraged to conduct early analysis of their data and explore Python libraries available in Google Colab for data manipulation and visualization.

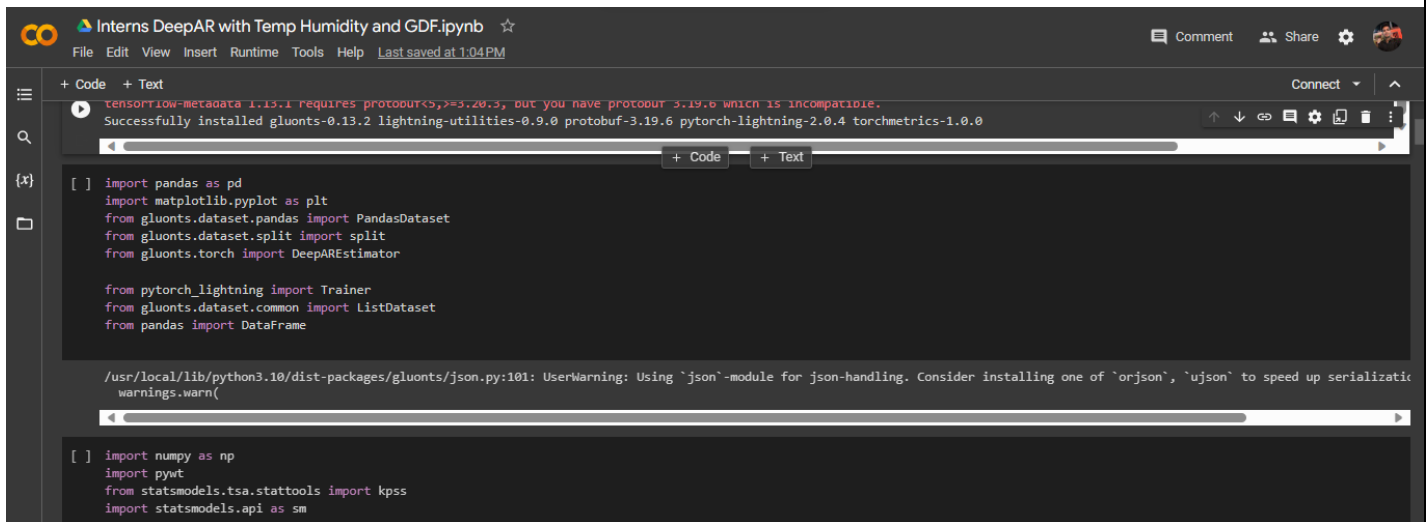


## 4. Key Steps in Data Analysis:

The following steps were highlighted during the workshop as important for data analysis:

### 1. Importing Libraries:

Participants were instructed to import the necessary Python libraries for data analysis and machine learning tasks.



```
tensorflow==2.10.0 requires protobuf<3.20.3, but you have protobuf 3.19.6 which is incompatible.
Successfully installed gluonts-0.13.2 lightning-utilities-0.9.0 protobuf-3.19.6 pytorch-lightning-2.0.4 torchmetrics-1.0.0

[ ] import pandas as pd
import matplotlib.pyplot as plt
from gluonts.dataset.pandas import PandasDataset
from gluonts.dataset.split import split
from gluonts.torch import DeepAREstimator

from pytorch_lightning import Trainer
from gluonts.dataset.common import ListDataset
from pandas import DataFrame

/usr/local/lib/python3.10/dist-packages/gluonts/json.py:101: UserWarning: Using `json`-module for json-handling. Consider installing one of `orjson`, `ujson` to speed up serialization.
warnings.warn(

[ ] import numpy as np
import pywt
from statsmodels.tsa.stattools import kpss
import statsmodels.api as sm
```

## 2. Loading Data:

Data files were loaded using the "pd.read\_csv()" function, and the resulting dataset was stored in a data frame (df).

## 3. Data Exploration:

The data frame (df) was inspected to understand the data types of different columns using the "df.dtypes" command.

## 4. Handling Time Series Data:

For time series data, participants were advised to convert the date-time column to a pandas datetime object using "pd.to\_datetime()".

## 5. Dealing with Missing Values:

The workshop emphasized the importance of addressing missing values in the data to ensure proper functioning of the models. Techniques for handling missing data were not discussed in detail.

## 6. Data Distribution:

Understanding the data distribution was deemed important, and participants were encouraged to explore Bayesian estimators to assess the distribution.

## 7. Feature Importance:

Determining the relevance of input variables to the problem at hand was discussed. Participants were encouraged to explore correlation analysis and visualize it using tools such as heat maps.

## 8. Outliers:

Outliers, which represent abnormal or unusual data points, were briefly discussed. It was mentioned that outliers could be the result of rare events or corrupted measurements. However, certain exceptions, such as spikes in electricity consumption during festivals like Diwali, were considered practical and could be utilized in model development.

## 5. Deep Analysis:

The workshop introduced the installation of the "gluonts[torch]" library for deep analysis of time series data. However, the specific details and purpose of deep analysis were not elaborated upon.

## 6. Stationary and Non-stationary Time Series:

The workshop touched upon the concept of stationary and non-stationary time series. Stationary time series have relatively constant maximum and minimum values, while non-stationary time series exhibit trends and variations. Techniques for transforming non-stationary data into stationary data were briefly mentioned, involving operations such as differencing and converting columns to series.

Categorical Variables:

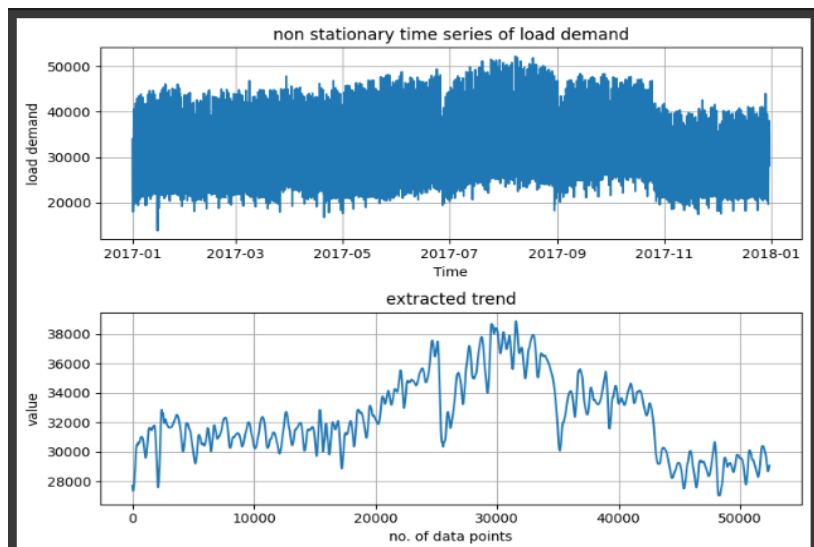
The workshop briefly mentioned categorical variables, which have a limited number of values or categories. An example provided was the gender variable, which can have values like Male, Female, or transgender.

Positional Encoding:

Positional encoding, using trigonometry, was mentioned as a technique to incorporate time-related information into the model.

Attention Mechanism:

The workshop introduced the concept of attention as a crucial component in machine learning algorithms. However, details about its implementation and significance were not discussed.



## 7. DeepAR Algorithm:

The DeepAR algorithm, introduced as a machine learning algorithm, was presented as an important topic. However, specific details about its functioning, advantages, and limitations were not provided.



```
Interns DeepAR with Temp Humidity and GDF.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at 1:04PM

Files
- sample_data

+ Code + Text

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->fsspec[http]>2021.06.0->py...)
Installing collected packages: protobuf, lightning-utilities, gluonts, torchmetrics, pytorch-lightning
Attempting uninstall: protobuf
Found existing installation: protobuf 3.20.3
Uninstalling protobuf-3.20.3:
Successfully uninstalled protobuf-3.20.3
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependencies:
tensorflow 2.12.0 requires protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.21.5,<5.0.dev,>=3.20.3, but you have protobuf 3.19.6 which is incompatible.
tensorflow-datasets 4.9.2 requires protobuf>=3.20, but you have protobuf 3.19.6 which is incompatible.
tensorflow-metadata 1.13.1 requires protobuf<5,>=3.20.3, but you have protobuf 3.19.6 which is incompatible.
Successfully installed gluonts-0.13.2 lightning-utilities-0.9.0 protobuf-3.19.6 pytorch-lightning-2.0.4 torchmetrics-1.0.0

[ ] import pandas as pd
import matplotlib.pyplot as plt
from gluonts.dataset.pandas import PandasDataset
from gluonts.dataset.split import split
from gluonts.torch import DeepAREstimator

from pytorch_lightning import Trainer
from gluonts.dataset.common import ListDataset
from pandas import DataFrame

/usr/local/lib/python3.10/dist-packages/gluonts/json.py:101: UserWarning: Using `json`-module for json-handling. consider installing one of `orjson`, `ujson`
warnings.warn(

[ ] import numpy as np
import pywt
from statsmodels.tsa.stattools import kpss
import statsmodels.api as sm

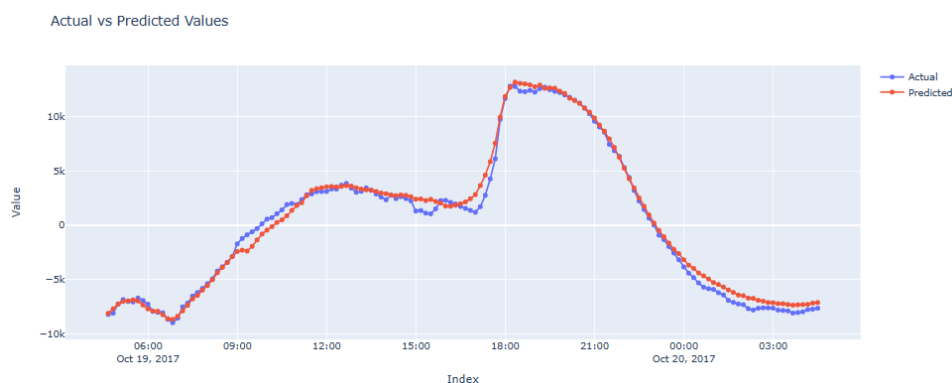
[ ] # Step 1: Load the data into a pandas DataFrame
df=pd.read_csv(r'/content/Tetuan City power consumption.csv',parse_dates=True)
data = df[['Actual', 'Temperature', 'Humidity', 'Season', 'Difference', 'Cloud', 'DeepAR', 'DeepAR-Seasonal']]
```

## 8. Preparing Data for Training:

The workshop covered the preparation of data for training the machine learning model. The data was divided into training and test sets, with recommended ratios of 80:20 for normal data and 50:50 for large datasets. The purpose of the test set was to assess the model's performance and its ability to learn patterns rather than rectifying the data.

## 9. Overfitting and Underfitting:

Overfitting and underfitting, two common issues in machine learning, were briefly mentioned. Participants were advised to conduct further research on these topics to understand the challenges associated with them.



## 10. Training and Evaluation:

The workshop introduced the concept of training epochs, which refer to the number of times the model is trained. It was recommended to keep the number of training epochs less than or equal to 5 to ensure efficiency. The training process was not described in detail.

## **11. Forecasting and Evaluation:**

The workshop demonstrated how to use the trained model to make predictions on the test dataset. Graphs were generated to compare the actual values with the predicted values. The average error between the actual and predicted values was calculated using metrics such as MAPE (Mean Absolute Percentage Error) and SMAPE (Symmetric Mean Absolute Percentage Error), which were considered more reliable than other metrics such as MAE, MSE, RMSE, MedAE, and R2 Score.

## **12. Mathematical Models vs. Data-driven Models:**

The workshop briefly mentioned that mathematical models make assumptions that may not capture the randomness of the data accurately. Data-driven models, trained on actual data, were suggested as a more appropriate choice for modeling.

## **13. Conclusion:**

The workshop provided an introduction to data analysis using machine learning techniques, focusing on the DeepAR algorithm and Google Colab as the coding platform. The report summarizes the main topics covered during the workshop, including regression, classification, clustering, data preprocessing, time series analysis, and model evaluation. Participants were encouraged to conduct further research and explore additional resources to deepen their understanding of the discussed concepts

.

# Workshop Report

## Introduction to IoT and Deep Discussion on Key Concepts

Date: 05/07/2023

Presenter: **Dr. R.K. BEHRA SIR**

B.Tech: NIT Rourkela

M.tech & P.hD: IITK

PD: University of Pretoria, South Africa

Associate Professor - Dept. Of EE IITP

Visiting Scholar - The Center for Energy Systems,  
Tennessee Technological University, Cookeville, USA

Project Associate - IITK

Trainee Engineer -Hindustan Aeronautics Limited ( HAL)

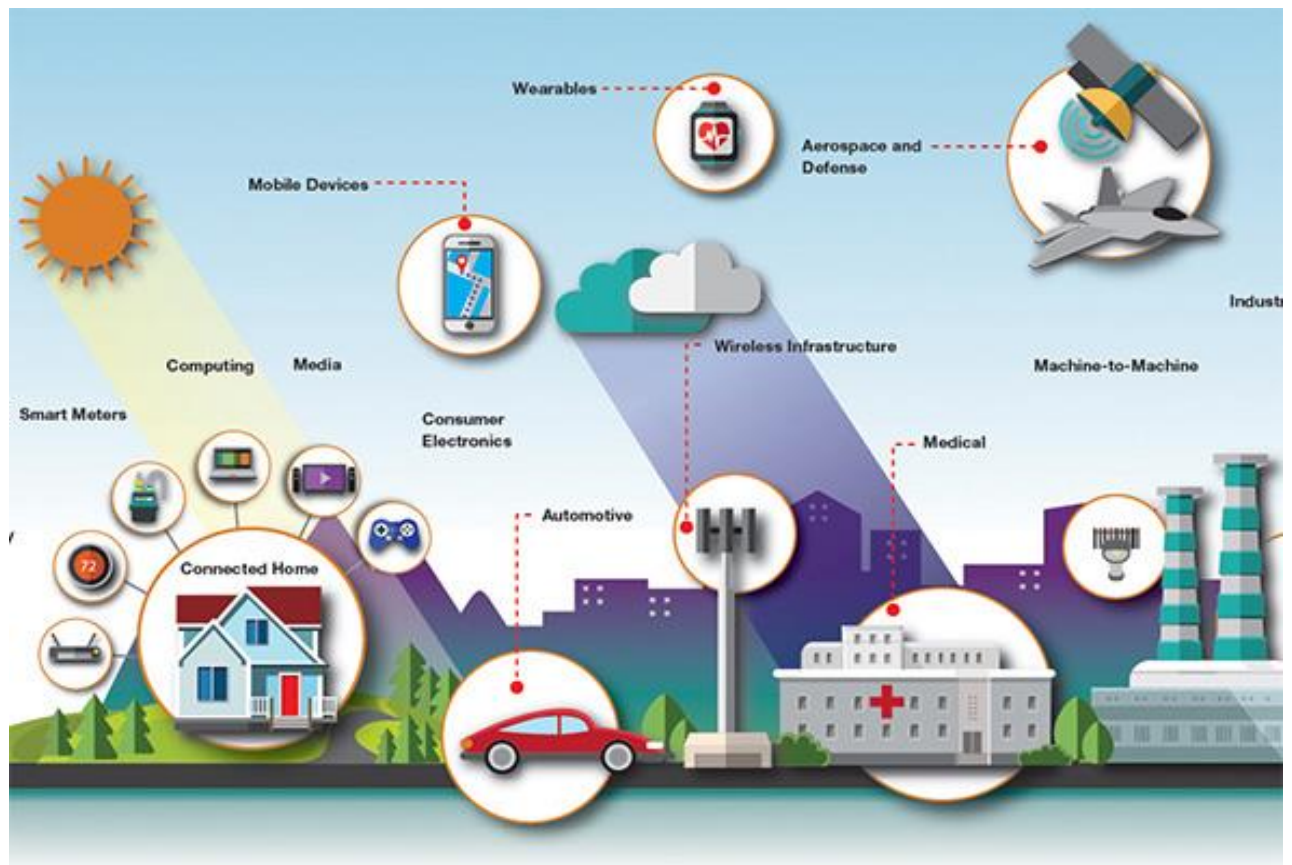
### 1. Story of IoT

The workshop conducted by Dr. R.K. Behra provided attendees with a comprehensive understanding of the Internet of Things (IoT) and its evolution over time. The session commenced with an enlightening discussion on the story of IoT, tracing its origins and development. Participants were introduced to the concept of connecting physical objects to the internet, enabling them to exchange data and perform intelligent actions.



## 2. IOT Devices

Dr. R.K. Behra elaborated on IoT devices, which are the fundamental building blocks of the IoT ecosystem. The participants gained insights into various types of IoT devices, including sensors and embedded systems. The workshop emphasized the importance of these devices in collecting and transmitting data, enabling real-time monitoring and control.



### 3. Gateway

The session also delved into the concept of gateways in IoT. Attendees were introduced to the role of gateways in connecting IoT devices to the internet and facilitating communication between devices and the cloud. Dr. R.K. Behra highlighted the significance of gateways as intermediaries that ensure secure and reliable data transfer.



#### **4. Edge Devices**

The workshop provided a deep discussion on edge devices and their role in IoT. Participants learned about the significance of edge computing, where data processing and analytics take place closer to the data source, reducing latency and bandwidth requirements. Dr. R.K. Behra highlighted the benefits of edge devices in enabling real-time decision-making and reducing dependence on the cloud.

#### **5. Internal Communication**

Attendees were enlightened about the internal communication within an IoT system. The workshop explored different communication protocols used in IoT. Dr. R.K. Behra discussed the advantages and use cases of each protocol, enabling participants to understand how IoT devices exchange data efficiently and securely.

#### **6. Storage (Cloud)**

The session proceeded with an in-depth exploration of cloud storage in IoT. Participants gained a comprehensive understanding of how the cloud plays a vital role in storing and managing the vast amount of data generated by IoT devices. Dr. R.K. Behra highlighted the scalability, accessibility, and security features provided by cloud storage solutions.

#### **7. Analytics**

The workshop emphasized the importance of data analytics in extracting valuable insights from IoT-generated data. Dr. R.K. Behra discussed various analytics techniques and algorithms used to process and analyze IoT data. Participants learned how data analytics enables predictive maintenance, anomaly detection, and optimization in different IoT applications.

## 8. Visualization

The final topic covered in the workshop was data visualization. Dr. R.K. Behra highlighted the significance of presenting IoT data in a meaningful and intuitive manner. Participants were introduced to various visualization techniques, including dashboards, charts, and graphs, to effectively communicate insights derived from IoT data.

In conclusion, Dr. R.K. Behra's workshop provided attendees with a comprehensive understanding of IoT, covering its evolution, key components, and essential concepts. Participants gained insights into IoT devices, gateways, edge devices, internal communication protocols, cloud storage, data analytics, and data visualization. The workshop equipped participants with the knowledge necessary to explore and implement IoT solutions in various domains, thereby enhancing their understanding of the transformative power of IoT technology.

[Workshop Organizers/Company Name] expresses sincere gratitude to Dr. R.K. Behra for conducting an enlightening and informative session on IoT. The workshop was a resounding success, leaving participants inspired and equipped with valuable insights into the world of IoT.

# \*RASBERRY PI PROJECTS\*

## 1. Project: Remote Accessing Raspberry Pi by VNC

- **Setup Required:**

- Raspberry Pi board
- VNC Viewer software

- **Steps and Commands:**

1. Install VNC server on Raspberry Pi:

```

`sudo apt update`

`sudo apt install realvnc-vnc-server`

```

2. Enable VNC server:

```

`sudo raspi-config`

```

Go to Interfacing Options > VNC > Enable VNC

3. Install VNC Viewer on the remote device (computer, smartphone, etc.):

- Download VNC Viewer software from the RealVNC website and install it on the remote device.

4. Connect to Raspberry Pi from VNC Viewer:

- Open VNC Viewer and enter the IP address or hostname of the Raspberry Pi to establish a remote connection.

- **Explanation of the Project:**

The remote accessing Raspberry Pi project involved setting up and configuring the VNC (Virtual Network Computing) server on the Raspberry Pi. By enabling VNC, it allowed remote access to the Raspberry Pi's graphical desktop interface from a different device using VNC Viewer software.

- **Learnings from the Project:**

- Understanding remote access concepts and their importance in Raspberry Pi projects.
- Configuring the VNC server on the Raspberry Pi to enable remote desktop access.
- Establishing a secure connection between the remote device and the Raspberry Pi.



## 2. Project: Sinusoidal Wave Representation

- **Setup Required:**

- Raspberry Pi board
- Python
- Matplotlib library

- **STEPS**

**Step 1: Install the required Python libraries:**

- Run the following command to install matplotlib:

```
$ pip install matplotlib
```

**Step 2: Write the Python code:**

- Create a new Python file (e.g., `sinusoidal\_wave.py`) and open it in a text editor.
- Import the necessary libraries:

**python code**

```
import numpy as np
import matplotlib.pyplot as plt
#Generate the sinusoidal wave data:
amplitude = 1.0 # Adjust the amplitude as desired
frequency = 1.0 # Adjust the frequency as desired
duration = 2.0 # Adjust the duration of the wave in seconds
# Generate time values from 0 to the desired duration
t = np.linspace(0, duration, num=1000)

# Generate the sinusoidal wave using the amplitude, frequency, and time values
wave = amplitude * np.sin(2 * np.pi * frequency * t)

- Plot and display the wave:
```python
# Create the figure and axes
fig, ax = plt.subplots()

# Plot the wave
ax.plot(t, wave)

# Set the x-axis and y-axis labels
ax.set_xlabel('Time')
ax.set_ylabel('Amplitude')

# Show the plot
plt.show()
``
```

### Step 3: Run the code:

- Save the Python file and run it using the following command:

```
$ python sinusoidal_wave.py
```

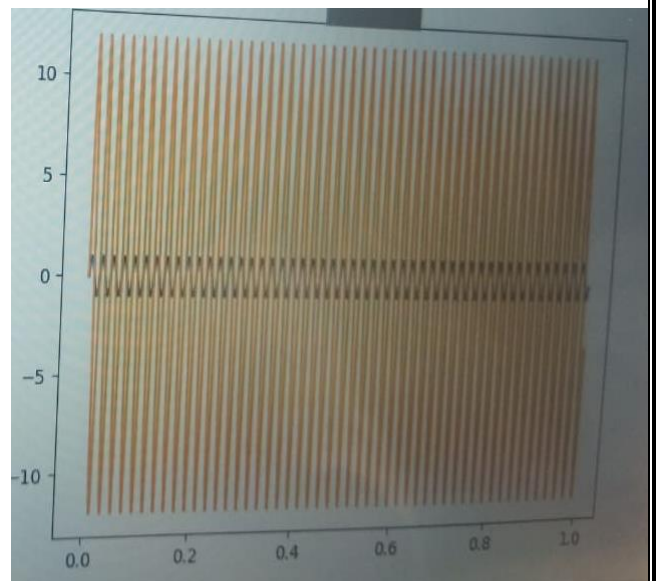
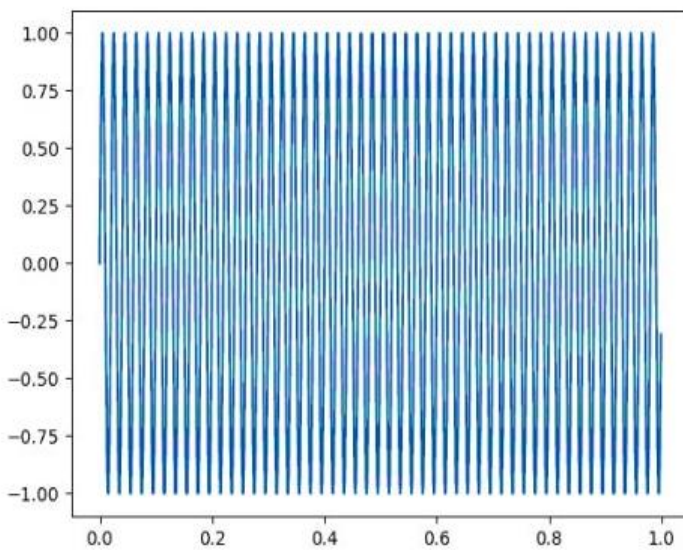
This will display a matplotlib figure showing the sinusoidal wave with the specified amplitude, frequency, and duration.

Adjust the amplitude and frequency values in the code to explore different waveforms and parameters. Have fun experimenting with your Raspberry Pi projects!

- **Explanation of the Project:**

The sinusoidal wave representation project involved generating and plotting a sinusoidal waveform using the Matplotlib library in Python. By varying the amplitude and frequency parameters, different waveforms could be created. The X-axis represented time, and the Y-axis represented the amplitude of the waveform.

- **Working Model:**



- **Learnings from the Project:**

- Understanding the concept of sinusoidal waves and their representation.
- Utilizing the Matplotlib library to plot graphs and visualize data.
- Exploring the relationship between amplitude, frequency, and time in sinusoidal waves.

### 3. Project: Blinking LED

- **Setup Required:**
  - Raspberry Pi board
  - Breadboard
  - LED
  - Jumper wires
- **Steps to connect and control the LED with Raspberry Pi:**

#### **Step 1: Connect the components:**

- Connect the anode (longer leg) of the LED to one leg of the resistor.
- Connect the other leg of the resistor to a GPIO pin on the Raspberry Pi.
- Connect the cathode (shorter leg) of the LED to the ground (GND) pin on the Raspberry Pi.

#### **Step 2: Set up the Python environment:**

- Make sure your Raspberry Pi is powered on and connected to the internet.
- Open a terminal or SSH into your Raspberry Pi.

#### **Step 3: Install the required Python libraries:**

- Run the following command to install the RPi.GPIO library:

```
'''  
  
pip install RPi.GPIO  
'''
```

#### **Step 4: Write the Python code:**

- Create a new Python file (e.g., `blinking\_led.py`) and open it in a text editor.
- Import the necessary libraries:

```
'''python  
import RPi.GPIO as GPIO  
import time  
'''
```

- Set up the GPIO mode and pin:

```
'''python  
GPIO.setmode(GPIO.BCM)  
LED_PIN = 18 # Change this to the GPIO pin you connected the LED to  
GPIO.setup(LED_PIN, GPIO.OUT)  
'''
```

- Implement the blinking logic:

```
'''python  
try:  
    while True:  
        GPIO.output(LED_PIN, GPIO.HIGH) # Turn the LED on
```

```
time.sleep(1) # Delay for 1 second
GPIO.output(LED_PIN, GPIO.LOW) # Turn the LED off
time.sleep(1) # Delay for 1 second
except KeyboardInterrupt:
    GPIO.cleanup() # Clean up the GPIO pins on keyboard interrupt
'''
```

### **Step 5: Run the code:**

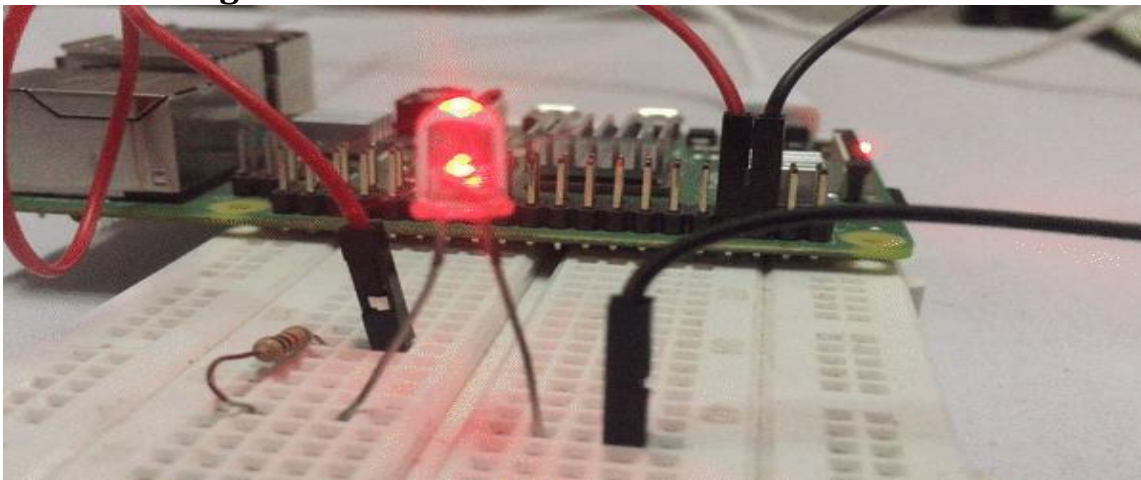
- Save the Python file and run it using the following command:

```
'''
python blinking_led.py
'''
```

- **Explanation of the Project:**

The blinking LED project involved connecting an LED to the Raspberry Pi and controlling its state using Python code. The GPIO (General Purpose Input/Output) pins of the Raspberry Pi were utilized to control the LED. By turning the GPIO pin ON and OFF at specific time intervals, the LED would blink continuously.

- **Working Model:**



- **Learnings from the Project:**

- Understanding GPIO pins and their functionality on the Raspberry Pi.
- Implementing basic digital output using Python and the RPi.GPIO library.
- Gaining hands-on experience with physical computing and connecting electronic components to the Raspberry Pi.

## 4. Project: FM Transmitter

- **Setup Required:**

- Raspberry Pi board
- Breadboard
- Jumper wires
- Antenna (20-40 cm wire)

**To set up an FM transmitter on your Raspberry Pi by pulling a GitHub repository, follow these steps:**

**Step 1: Prepare the Raspberry Pi:**

- Make sure your Raspberry Pi is powered on and connected to the internet.
- Open a terminal or SSH into your Raspberry Pi.

**Step 2: Install the necessary dependencies:**

- Run the following commands to install the required software packages:

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

```
$ sudo apt-get install gcc g++ make
```

**Step 3: Clone the FM transmitter repository:**

- Run the following command to clone the FM transmitter repository from GitHub:

```
$git clone https://github.com/AshrafAkon/fm_transmitter.git
```

**Step 4: Build the FM transmitter:**

- Change to the repository directory:

```
$cd fm_transmitter
```

- command to compile the program:

```
$ sudo make
```

**Step 5: Transmit audio using the FM transmitter:**

- Connect an antenna or a wire to the GPIO 4 (pin 7) of your Raspberry Pi. Ensure it is appropriately positioned for better signal transmission.

While cloning the program, there is an audio file known as `star_wars.wav` which is downloaded by default. We will use this audio file and a frequency of 103MHz (you

can choose any frequency as long as it is within the FM band) to see if the program works correctly. The command for launching the program is:

```
$ sudo ./fm_transmitter -f 103 -r star_wars.wav
```

syntax for the above command is:

```
$ sudo ./fm_transmitter [-f frequency] [-r] filename
```

Note: ``<frequency>`` represents the desired frequency in MHz (e.g., 101.1).

Now, the FM transmitter should start broadcasting the audio file on the specified frequency. You can use an FM receiver to tune in and listen to the transmitted audio.

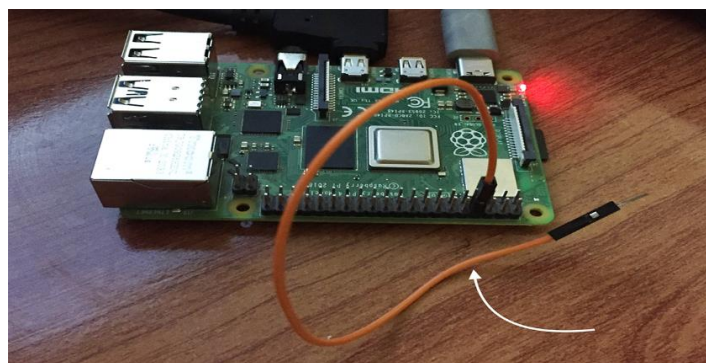
**\*Please note that broadcasting FM signals without proper licensing may violate regulations in your country. Make sure to comply with local laws and regulations regarding radio broadcasting.**

**Also, keep in mind that the range of the FM transmitter may vary depending on the power output, antenna, and local environmental factors. Ensure you are using the FM transmitter responsibly and within legal limits.**

- **Explanation of the Project:**

The FM transmitter project involved utilizing the Raspberry Pi to transmit FM signals. By modulating the GPIO pin of the Raspberry Pi at a specific frequency, an FM signal was generated. This signal could be received and played back on an FM radio receiver.

- **Working Model:**



- **Learnings from the Project:**

- Understanding the basics of FM modulation and transmission.
- Utilizing the GPIO pins of the Raspberry Pi to generate FM signals.
- Exploring the possibilities of wireless communication using the Raspberry Pi.



## 5. Project: Reverse safety car parking system



### 2. Setup Required for the Project:

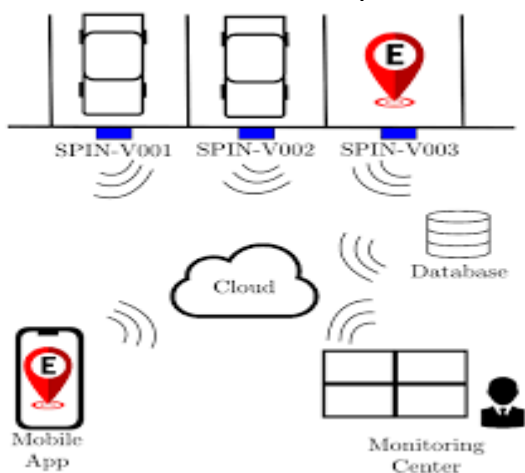
- Raspberry Pi Model 3 B
- Ultrasonic sensor
- Buzzer sensor
- Jumper wires
- Breadboard
- Power supply (e.g., USB power bank)
- Optional: LED for visual indication
- Optional: Case or enclosure for Raspberry Pi

### 3. Explanation of the Project:

The Reverse Safety Car Parking System is designed to maintain a safe distance between a car and an outer object while parking in reverse. It utilizes an ultrasonic sensor to measure the distance between the car and the obstacle behind it. When the distance falls below a predefined safe threshold, a buzzer is sounded as a warning signal to the driver.

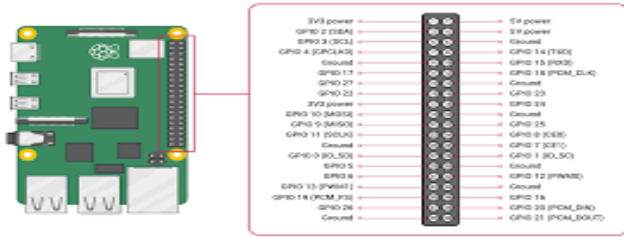
### 4. Working Model:

The setup consists of a Raspberry Pi connected to the ultrasonic sensor, buzzer, and optionally an LED for visual indication. The ultrasonic sensor measures the distance between the car and the obstacle by sending and receiving ultrasonic pulses. The Raspberry Pi reads the distance data and triggers the buzzer when the distance falls below the safe threshold. The LED can be used to provide a visual indication of the parking situation.



### 5. Learnings from the Project:

- Understanding and implementing sensor interfacing with Raspberry Pi.
- Programming GPIO (General Purpose Input/Output) pins of Raspberry Pi.



- Reading and processing sensor data using Python.
- Applying logical conditions to determine parking safety.
- Controlling output devices (buzzer, LED) based on sensor data.
- Gaining practical experience with hardware-software integration

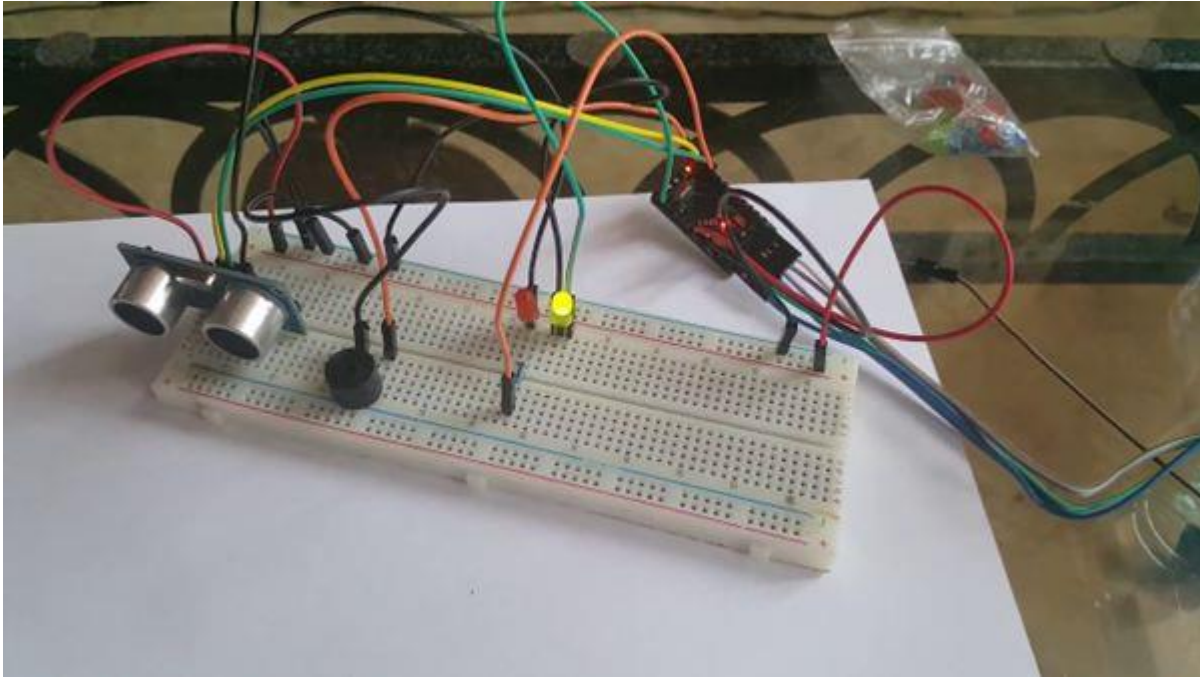
### Code used for the project :

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
trig_pin = 23
echo_pin = 24
buzzer_pin = 17
GPIO.setup(trig_pin, GPIO.OUT)
GPIO.setup(echo_pin, GPIO.IN)
GPIO.setup(buzzer_pin, GPIO.OUT)
def measure_distance():
    GPIO.output(trig_pin, GPIO.LOW)
    time.sleep(0.5)
    GPIO.output(trig_pin, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(trig_pin, GPIO.LOW)
    while GPIO.input(echo_pin) == 0:
        pulse_start = time.time()
    while GPIO.input(echo_pin) == 1:
        pulse_end = time.time()
    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration * 17150 # Speed of sound is approximately 343 meters/second
    distance = round(distance, 2)
    return distance

def activate_buzzer():
    GPIO.output(buzzer_pin, GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output(buzzer_pin, GPIO.LOW)
    time.sleep(0.5)
try:
```



```
while True:
    distance = measure_distance()
    print("Distance:", distance, "cm")
    if distance < 30: # Adjust this threshold according to your requirements
        activate_buzzer()
        time.sleep(0.1)
except KeyboardInterrupt:
    GPIO.cleanup()
```



- **Similar Projects of Interest for the Future:**

1. Home automation system using Raspberry Pi and sensors to control various devices.
2. Image recognition and object detection using Raspberry Pi and OpenCV.
3. Weather station project with Raspberry Pi to collect and analyze environmental data.
4. Voice-controlled assistant using Raspberry Pi and speech recognition software.

- **Acknowledgment:**

We would like to express our gratitude to our workshop instructors, who provided us with valuable guidance and support throughout the projects. Their expertise and knowledge significantly contributed to our learning experience. We would also like to thank the organizers of the workshop for providing us with the opportunity to explore and work on exciting Raspberry Pi projects.

# Workshop report

## Web development using django

### #1Workshop Report: Creating Virtual Environments and Django Project

Date: 4/07/2023

Workshop: Windows Environment Setup and Django Project Creation

#### Introduction:

The workshop focused on providing participants with hands-on experience in setting up virtual environments, activating them, and troubleshooting common errors. Additionally, the participants learned how to install Django and create a project using the 'start project' subcommand in Django admin. The following report summarizes the key points covered during the workshop.

#### Setup Required for the Project:

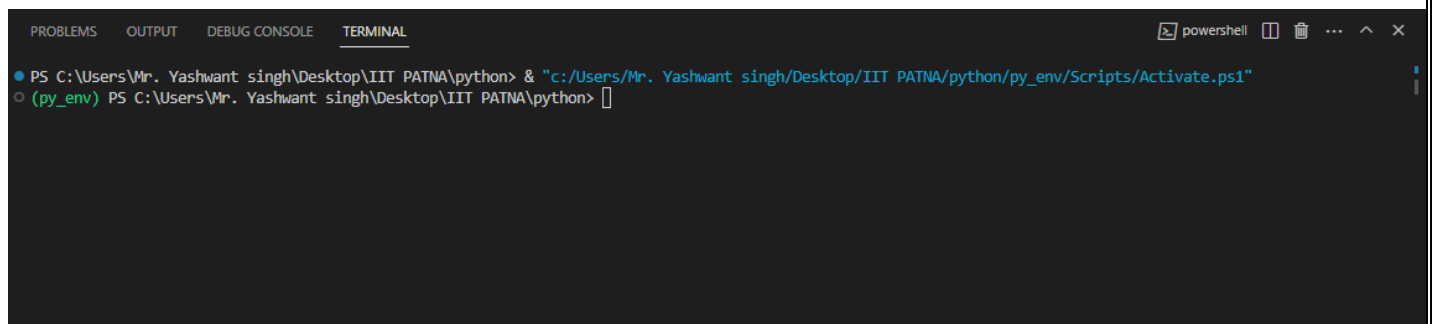
The following steps were undertaken to set up the project:

##### Step 1: Creating a Virtual Environment

- Open the command prompt or terminal.
- Navigate to the desired directory for the project.
- Execute the command: ``python -m venv myenv`` (where ``myenv`` is the name of the virtual environment).

##### Step 2: Activating the Virtual Environment

`$ .\myenv\Scripts\activate`



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\Mr. Yashwant singh\Desktop\IIT PATNA\python> & "c:/Users/Mr. Yashwant singh/Desktop/IIT PATNA/python/py_env/Scripts/Activate.ps1"
(py_env) PS C:\Users\Mr. Yashwant singh\Desktop\IIT PATNA\python>
```

### **Step 3: Installing Django**

- Execute the command:

```
$ pip install django
```

### **Step 4: Creating a Django Project**

- Execute the command:

```
$ django-admin startproject projectname (replace `projectname`)
```

### **Explanation of the Project:**

The project created during the workshop was a basic Django project. Django is a powerful Python web framework that follows the MVC (Model-View-Controller) architectural pattern. It provides a convenient way to build web applications by abstracting common tasks, such as handling URLs, managing databases, and handling forms.

### **Learnings from the Project:**

During the workshop, participants gained the following key learnings:

- Understanding the importance of virtual environments for isolating project dependencies.
- Familiarity with the process of creating and activating virtual environments.
- Knowledge of installing Django and starting a new project using the 'startproject' SUB command.
- Introduction to Django's project structure and configuration files.

### **Conclusion:**

The workshop on Windows environment setup and Django project creation was successful in familiarizing participants with the process of creating virtual environments, installing Django, and starting a new project. Participants gained valuable insights into web development using Django and expressed enthusiasm for exploring more advanced projects in the future.

## #2Workshop Report: Displaying a String and Creating a Form in Django

### Introduction:

The workshop focused on teaching participants how to display a string on a home webpage using Django and subsequently create a form to collect user input, submit it, and display the entered data on the same page. The following report outlines the step-by-step process along with the relevant code snippets and commands used during the workshop.

### 1. Displaying a String on the Home Webpage:

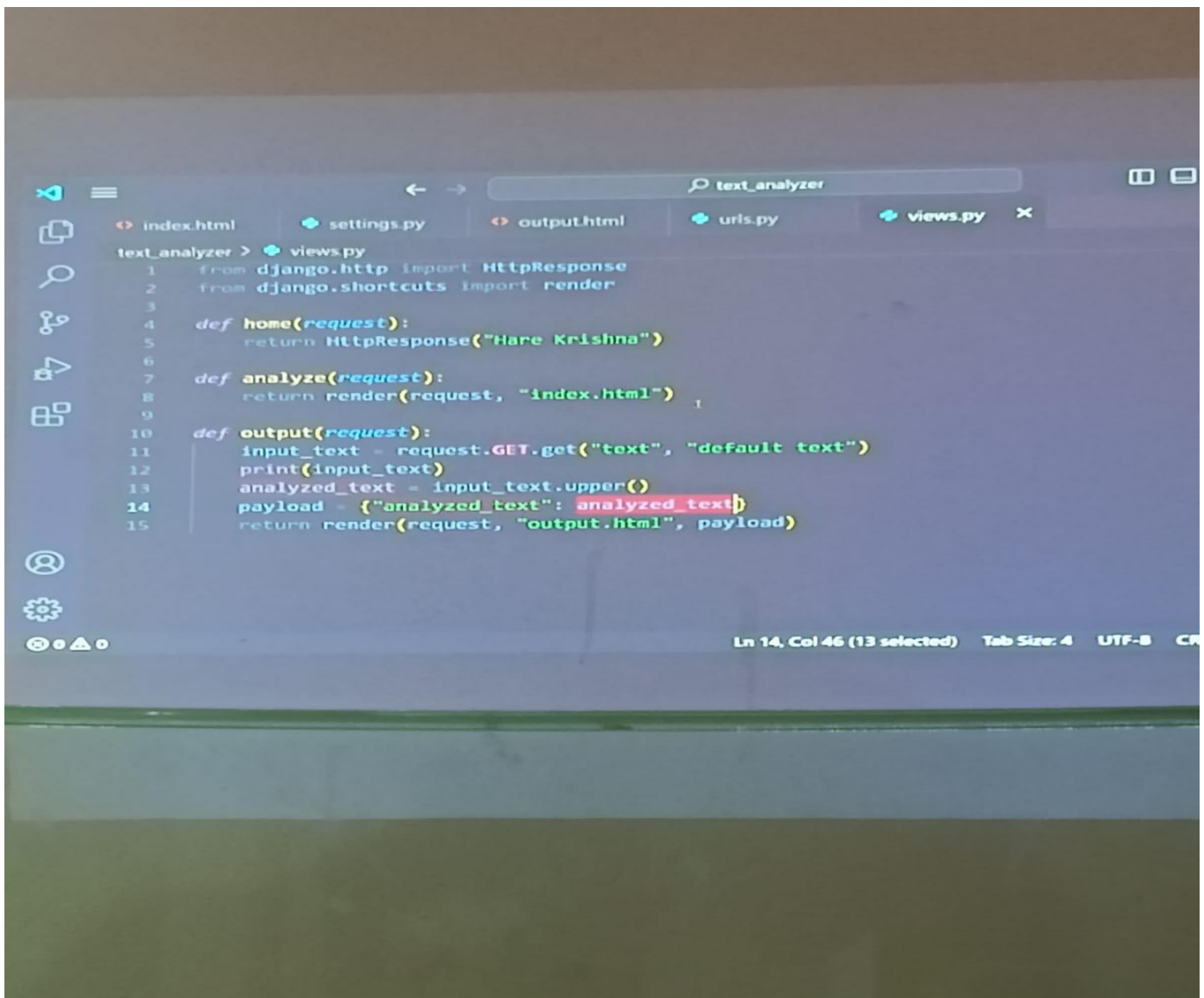
To display a string on the home webpage, the following steps were followed:

#### Step 1: Update the Views

- Open the `views.py` file inside the app directory.
- Import the necessary modules: `from django.shortcuts import render`
- Define a function-based view to render the home page:

```
```python
def home(request):
    context = {'message': 'Welcome to our website!'}
    return render(request, 'home.html', context)
```
```

```
1 from django.http import HttpResponse
2 from django.shortcuts import render
3
4 def home(request):
5     return render(request, "welcome.html")
6
7 def analyze(request):
8     return render(request, "index.html")
```

A screenshot of a code editor window titled 'text\_analyzer'. The editor shows the 'views.py' file with the following Python code:

```
1 from django.http import HttpResponse
2 from django.shortcuts import render
3
4 def home(request):
5     return HttpResponse("Hare Krishna")
6
7 def analyze(request):
8     return render(request, "index.html")
9
10 def output(request):
11     input_text = request.GET.get("text", "default text")
12     print(input_text)
13     analyzed_text = input_text.upper()
14     payload = {"analyzed_text": analyzed_text}
15     return render(request, "output.html", payload)
```

The code is syntax-highlighted. The editor interface includes a sidebar on the left with icons for Explorer, Search, Source Control, and Run and Debug. The top of the editor shows tabs for 'index.html', 'settings.py', 'output.html', 'urls.py', and 'views.py'. The status bar at the bottom indicates 'Ln 14, Col 46 (13 selected)', 'Tab Size: 4', and 'UTF-8'.

## Step 2: Create the Home Template

- Inside the app directory, create a `templates` folder (if not already present).
- Inside the `templates` folder, create a `home.html` file.
- Add the following code to `home.html` to display the string:

```
```html
<!DOCTYPE html>
<html>
<head>
  <title>Home</title>
</head>
<body>
  <h1>{{ message }}</h1>
</body>
</html>
```
```

```

index.html x base.html output.html urls.py 1 welcome.html
text_analyzer > templates > index.html > ...
5 {% endblock title %}
6
7 {% block body %}
8 <div class="container">
9     <form action="output" method="get">
10     <div class="mb-3">
11         <label for="exampleInputEmail1" class="form-label">
12         <textarea type="text" name="text" class="form-cont
13     </div>
14     <button type="submit" class="btn btn-success">Analyze
15 </form>
16 </div>

```

### Step 3: Map the URL

- Open the `urls.py` file inside the project directory.
- Import the necessary modules: `from appname.views import home`
- Add the following code to map the URL to the home view:

```
```python
```

```
from django.urls import path
```

```
urlpatterns = [
    path('', home, name='home'),
]
```

```

```

index.html base.html output.html urls.py 1 x welcome.html
text_analyzer > text_analyzer > urls.py > urlpatterns
15 2. Add a URL to urlpatterns: path('blog/', include('b
16 """
17 from django.contrib import admin
18 from django.urls import path
19 from . import veiws
20
21 urlpatterns = [
22     path('', veiws.home,name='home'),
23     path('analyze', veiws.analyze,name='analyze'),
24     path('output', veiws.output,name='output')
25 ]
26

```

## **Learnings from the Project:**

During the workshop, participants gained the following key learnings:

Understanding URL routing in Django and mapping views to URLs.

Creating view functions to handle HTTP requests and return appropriate responses.

Creating and using Django formclasses to collect user input and validate it.

Rendering form fields in HTML templates and handling form submission.

Displaying user input on the webpage.

## **Conclusion:**

The workshop successfully covered the process of displaying a string on the home webpage using Django and creating a form to collect user input, submit it, and display the entered data on the same page. Participants gained hands-on experience in building interactive webpages and learned the fundamentals of form handling in Django.

# **Workshop Report**

## Template Inheritance and Bootstrap Navigation Bar Implementation

Date- 07/07/2023

## **Introduction:**

In this workshop, we focused on Django template inheritance and improving a website using the Bootstrap framework. The main objective was to create a navigation bar with functionality using Django and enhance the overall appearance and user experience of a website.

## **Project Worked On:**

The project we worked on was a simple web application built with Django. It consisted of multiple pages and required the implementation of a navigation bar to improve the site's navigation and user interface.

## **Setup Required for the Project:**

To set up the project, follow these steps:

1. Install Python: Make sure Python is installed on your system. You can download the latest version from the official Python website.

2. Install Django: Open a terminal or command prompt and run the following command:

```
```\n\npip install django\n\n```\n
```

3. Create a Django project: Use the following command to create a new Django project:

```
```\n\ndjango-admin startproject project_name\n\n```\n
```

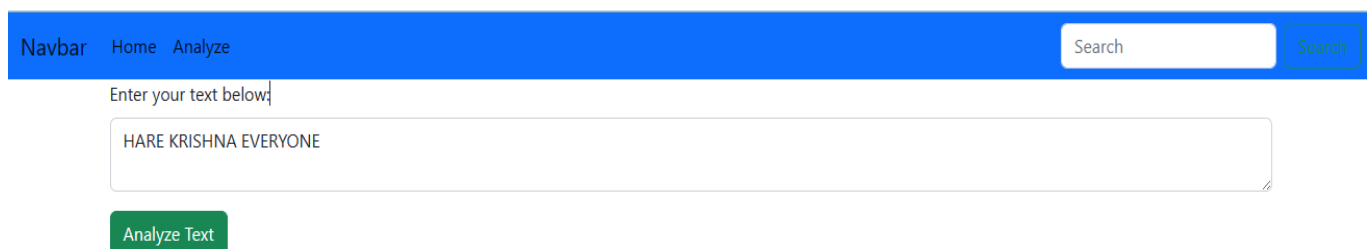
4. Create Django app: Inside the project directory, create a new Django app using the following command:

```
python manage.py startapp app_name
```

- **Explanation of the Project:**

The project involved building a basic Django web application. We utilized Django's template inheritance feature to create a base template that contains the common elements of all pages, such as the navigation bar. Each page then extends this base template and adds its unique content.

- **Working Model:**



The screenshot displays a web application interface. At the top, there is a blue navigation bar with links for 'Navbar', 'Home', and 'Analyze'. On the right side of the navigation bar, there is a search bar with the placeholder text 'Search' and a green 'Search' button. Below the navigation bar, there is a text input field with the placeholder text 'Enter your text below'. The input field contains the text 'HARE KRISHNA EVERYONE'. Below the input field, there is a green button labeled 'Analyze Text'.



## **Learnings from the Project:**

Throughout the workshop, participants gained the following key insights and learnings:

1. Django template inheritance: Understanding how to utilize template inheritance allows for efficient code reuse and easier maintenance of Django projects.
2. Bootstrap framework: Familiarity with Bootstrap and its components, such as the navigation bar, enables the creation of visually appealing and responsive web pages.
3. Django project structure: Participants learned about the recommended project structure in Django and how to organize their code effectively.

## **Similar Projects of Interest for Future Work:**

Based on the knowledge gained from this workshop, participants expressed interest in working on the following projects:

1. Building a blog application with user authentication and permissions.
2. Developing an e-commerce website with product listings, shopping cart functionality, and secure payment integration.
3. Creating a social media platform with user profiles, posts, likes, and comments.

## **Acknowledgement:**

We would like to express our gratitude to the instructors and organizers of the workshop for their valuable guidance and support. Special thanks to the contributors of Django and Bootstrap for providing excellent documentation and resources that aided in the completion of this project.

## **Conclusion:**

The Django template inheritance workshop was successful in enhancing participants' understanding of Django's template system and the implementation of a navigation bar using the Bootstrap framework. By combining these concepts, we were able to improve the overall design and functionality of a Django web application. Participants gained valuable insights and are now equipped with the knowledge to undertake more complex projects in the future.

## **Post-Session Reflection:**

It was an honor to have Mr. Yesveer Yadav, Technical Lead at IndiaMART, share his insights and expertise in the post-session segment of our workshop. His extensive knowledge and experience added immense value to our learning journey, leaving a lasting impact on all participants.

During this interactive session, Mr. Yesveer Yadav sir delved deep into the practical applications of IoT technologies in the business landscape. His real-world examples and case studies provided valuable context, showcasing how IoT innovations are driving transformation across industries. Through engaging discussions, he highlighted the challenges faced in the industry and explored innovative solutions, sparking curiosity and encouraging critical thinking among the attendees.

Sir's expertise not only illuminated the current trends in IoT but also shed light on the future prospects of this technology. His insights into the evolving IoT landscape, including emerging technologies and market demands, offered participants a glimpse into the exciting possibilities that lie ahead.

Furthermore, his emphasis on the importance of collaboration, continuous learning, and adaptability resonated deeply with all attendees. The session served as a reminder that the world of technology is ever-changing, and staying updated with the latest advancements is key to staying ahead in the competitive digital realm.

We express our heartfelt gratitude to Mr. Yadav for his time, wisdom, and engaging discourse. His session was not just an addition to our workshop; it was a cornerstone, inspiring us to approach our future endeavors with renewed enthusiasm and a broader perspective.

Thank you, Mr. Yesveer Yadav Sir, for being an integral part of our workshop and for guiding us towards a future illuminated by the endless possibilities of IoT technology. Your insights have undoubtedly set the stage for innovative thinking and transformative actions among all participants. Here's to embracing the future of technology with confidence and knowledge!