PHASE-5

1. Problem Statement:

Define the problem: Develop a smart public restroom system using IoT to optimize restroom operations and improve user experience.

2. Design Thinking Process:

Empathize: Understand the needs and pain points of restroom users.

Define: Clearly define the objectives and goals of the smart restroom.

Ideate: Brainstorm ideas and potential IoT solutions.

Prototype: Create a concept or mockup of the smart restroom system.

Test: Gather feedback and iterate on the design.

3. Phases of Development:

Hardware: Develop or procure IoT devices (sensors, actuators, cameras, etc.).

Software: Write code for data collection, processing, and control.

Integration: Connect and test all components in a real-world setting.

Deployment: Deploy the smart restroom system in public locations.

4. Dataset Used:

Collect data from sensors (occupancy, temperature, humidity, etc.).

Use historical restroom usage data for analysis and forecasting.

5. Data Processing Steps:

Data collection from IoT sensors.

Data cleaning and preprocessing.

Feature engineering (e.g., deriving usage patterns).

Data aggregation and storage.

6. Model Training Process:

Develop a time series forecasting model using historical data.

Use algorithms like ARIMA, LSTM, or Prophet for forecasting.

Train the model on the historical dataset.

7. Time Series Forecasting Algorithm:

Example using Python for forecasting:

python

```
from fbprophet import Prophet


# Prepare data in a DataFrame with 'ds' (timestamp) and 'y' (usage) columns
# Assuming 'df' is your dataset


model = Prophet()
model.fit(df)


# Create a future dataframe for forecasting
future = model.make_future_dataframe(periods=30)  # 30 days into the future


# Make predictions
forecast = model.predict(future)
```

8. Evaluation Metrics:

Metrics for time series forecasting, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and more.

Evaluate the model's accuracy and adjust as needed.

CODE:

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from statsmodels.tsa.arima_model import ARIMA

# Load time series occupancy data
occupancy_data = pd.read_csv('occupancy.csv')

# Fit an ARIMA model
model = ARIMA(occupancy_data, order=(5,1,0))
model_fit = model.fit(disp=0)

# Make predictions
forecast = model_fit.forecast(steps=24)  # Forecast 24 hours ahead

# Evaluation Metrics (Sample code)
from sklearn.metrics import mean_absolute_error,
mean_squared_error

# Actual occupancy data for the next 24 hours
actual_occupancy = [0, 1, 2, ...]
```

```
mae = mean_absolute_error(actual_occupancy, forecast)
mse = mean_squared_error(actual_occupancy, forecast)

print(f"Mean Absolute Error: {mae}")
print(f"Mean Squared Error: {mse}")
```