Project Objectives:

The primary objectives of the project are:

Real-Time Occupancy Monitoring:

Implement IoT occupancy sensors in each restroom stall to provide real-time data on stall availability.
Objectively measure and display the occupancy status of each stall to help users quickly find vacant facilities.
Cleanliness Assessment:

Deploy IoT sensors (ammonia, H2S, turbidity) to monitor cleanliness parameters within the restroom environment.
Continuously assess cleanliness levels and provide users with data-driven information about restroom hygiene.
Maintenance Needs Detection:

Utilize cleanliness sensors to detect maintenance needs such as overflowing sinks, toilet malfunctions, or depleted soap dispensers.
Automatically generate maintenance requests or alerts to restroom management for timely interventions.
Data Analytics and Reporting:

Collect and analyze historical data from IoT sensors to derive insights and trends related to restroom occupancy, cleanliness, and maintenance needs.
Generate reports and visualizations to help facility managers make informed decisions for restroom management and maintenance.
User Accessibility:

Develop a user-friendly web-based platform or mobile app that allows the public to access real-time restroom information easily.
Ensure the platform is accessible to individuals with disabilities and provides multi-language support, if applicable.
Cost Efficiency:

Optimize the project's cost-effectiveness by selecting appropriate sensors and technologies that balance functionality with budget constraints.
User Engagement:

Develop features to engage users, such as gamification elements or rewards for providing feedback or maintaining cleanliness.
IoT Sensor Setup:
The IoT sensor setup involves the deployment of occupancy sensors (e.g., PIR sensors), cleanliness sensors (e.g., ammonia, H2S, turbidity sensors), microcontrollers (e.g., Raspberry

Pi, Arduino), and communication modules (e.g., Wi-Fi, Bluetooth) in each restroom stall. The sensors continuously collect data and transmit it to a central server for further processing.

Mobile App Development:
The mobile app provides users with real-time restroom information. It should include features such as:

Displaying the occupancy status of restroom stalls.
Presenting cleanliness scores based on sensor readings.
Allowing users to report issues or provide feedback.
Multilingual support and accessibility features.
Raspberry Pi Integration:
Raspberry Pi serves as a microcontroller and data processing unit. It interfaces with occupancy and cleanliness sensors, processes the data, and transmits it to the central server. The Raspberry Pi also hosts the web-based platform accessible by users.

Code Implementation:
Code implementation involves developing Python scripts for IoT devices, including the Raspberry Pi, to read sensor data, process it, and transmit it to the central server. Additionally, backend code for the central server (using Flask, for example) receives and processes data, updates the database, and serves data to the mobile app.

For a detailed code implementation, refer to the Python scripts provided in the previous responses for IoT devices and the Flask-based web application.

Real-Time Restroom Information System Benefits:
Enhanced User Experience:

Users can quickly identify and access vacant restroom stalls, reducing wait times.
Real-time cleanliness information improves user satisfaction and ensures a hygienic restroom experience.
Efficient Restroom Management:

Maintenance needs are detected promptly, allowing for timely interventions and improved restroom facilities.
Data analytics provide insights into usage patterns, enabling more efficient cleaning and maintenance schedules.
User Engagement and Feedback:

Gamification elements and user feedback mechanisms encourage users to contribute to maintaining cleanliness.
Facility managers can use feedback to address issues promptly, improving overall restroom conditions.
Cost Optimization:

Data-driven insights help in optimizing cleaning and maintenance schedules, reducing operational costs.
Selecting cost-effective sensors and technologies contributes to overall project cost efficiency.
By combining real-time data, user engagement features, and efficient management strategies, the restroom information system contributes to a more user-friendly and well-maintained restroom environment.

Mobile App Interface Overview:

Home Screen:

Display the main screen showing a clean and intuitive layout.
Include a header with the app name and logo.
Provide a summary of restroom information, e.g., the total number of stalls and cleanliness score.
Stall Occupancy Section:

Use visual indicators for each stall, such as color-coded icons or a simple list.
Green for vacant, red for occupied.
Include a numerical count of available stalls.
Cleanliness Section:

Display a cleanliness score prominently.
Use visual elements like stars, smiley faces, or a numerical scale.
Provide additional details on cleanliness parameters if needed.
Maintenance Alerts:

Include a section for maintenance alerts.
Show notifications for issues like overflowing sinks, malfunctioning toilets, or low supplies.
Allow users to report issues with a simple button.
User Feedback and Reporting:

Include a section for user feedback.
Allow users to provide feedback on cleanliness or report issues.
Use a simple form or rating system.
Language and Accessibility Options:

Provide options for users to choose their preferred language.
Include accessibility features such as text-to-speech or high-contrast mode.
Navigation and Menus:

Use a bottom navigation bar or a side menu for easy navigation.

Include tabs for different sections like Occupancy, Cleanliness, Maintenance, and Feedback.

Augmenting User Experience:

1. Mitigated Waiting Periods:
   - Swift identification of unoccupied restroom stalls through real-time occupancy data aids users in minimizing wait times, particularly during periods of high traffic, elevating overall user contentment.

2. Elevated Sanitary Conditions:
   - Real-time cleanliness monitoring furnishes users with precise insights into restroom hygiene levels, enabling informed choices for selecting cleaner facilities and fostering a more hygienic experience.

3. User Interaction:
   - Incorporating features like user feedback and reporting empowers users to actively participate in upholding restroom cleanliness.
   - Introducing elements of gamification, rewards, or incentives for feedback provision engages users in the collaborative management of restroom facilities.

4. Inclusivity:
   - Offering language options and accessibility features ensures diverse user accessibility, guaranteeing that restroom information caters to a broad spectrum of individuals.

5. Anticipatory Maintenance:
   - The system's ability to detect maintenance requirements in real-time facilitates proactive scheduling of repairs or replenishments, curtailing disruptions to users.

6. Holistic Satisfaction:
   - A well-maintained restroom with efficient occupancy oversight contributes to an optimistic overall user experience.

Enhancing Restroom Administration:

1. Streamlined Cleaning Regimen:
   - Utilizing data analytics from historical sensor data assists facility managers in comprehending usage patterns, enabling the optimization of cleaning schedules with a focus on high-impact areas.

2. Preemptive Maintenance:
   - Timely alerts on potential maintenance needs empower maintenance staff to swiftly address issues, preempting major malfunctions and enhancing overall restroom functionality.

3. Optimized Resource Deployment
  - Real-time occupancy and cleanliness data inform resource deployment decisions, allowing cleaning staff to concentrate efforts where usage is high or immediate cleanliness concerns exist.

4. Financial Efficiency:
  - Strategic optimization of cleaning and maintenance schedules, informed by data-driven insights, contributes to fiscal efficiency.
  - Selecting suitable sensors and technologies ensures a judicious balance between functionality and budgetary constraints.

5. Continuous Advancement:
  - User feedback mechanisms and reporting channels offer valuable insights into user preferences and concerns.
  - Facility managers can leverage this feedback for ongoing enhancements, addressing issues and refining restroom facilities.

6. Adherence and Reporting:
  - The system provides facility managers with tools to monitor and report on restroom conditions, supporting compliance with hygiene standards and regulations.

7.Informed Decision Making:
  - Analyzing historical data empowers facility managers to make judicious decisions regarding restroom management, resource allocation, and ongoing improvements.

Arudino Code:

```
#include<cvzone.h>

SerialData serialData(2,1);
int valsRec[2];
const int MAINTAIN_OUT = 6;
const int OCCUPIED_OUT = 4;
const int OCCUPIED_IN = A3;
const int MAINTAIN_IN = A4;

void setup() {
  // put your setup code here, to run once
  serialData.begin();
  pinMode(MAINTAIN_OUT,OUTPUT);
  pinMode(OCCUPIED_OUT,OUTPUT);
  pinMode(MAINTAIN_IN,INPUT);
  pinMode(OCCUPIED_IN,INPUT);
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  serialData.Send([maintain,occupied]);
  int maintain = analogRead(MAINTAIN_IN);
  int occupied = analogRead(OCCUPIED_IN);

  serialData.Get(valsRec);

  if(valsRec[0]==0){
    digitalWrite(MAINTAIN_OUT,LOW);
  }
  else{
    digitalWrite(MAINTAIN_OUT,HIGH);
  }

  if(valsRec[1]==0){
    digitalWrite(OCCUPIED_OUT,LOW);
  }
  else{
    digitalWrite(OCCUPIED_OUT,HIGH);
  }
}
```

Python Code:

```
from cvzone.SerialModule import SerialObject

# Initialize the Arduino SerialObject to establish connection to arduino with optional parameters
# baudRate = 9600, digits = 1, max_retries = 5
board = SerialObject(portNo='COM7', baudRate=9600, digits=1, max_retries=5)

#Initialize boolean variables to indicate occupancy and maintenance
#This boolean variable is sent to arduino in order to make sensors indicate occupancy and
maintenance
occupancyFlag = 0
maintainFlag = 0

while True:
    #Read sensor values from arduino which is sent as an array
    sensorValues = board.getData()
    ammoniaLevels = sensorValues[0]
    occupancyState = sensorValues[1]
```

```
#if the stall is occupied and the toilet has to be cleaned
if ammoniaLevels > 200 and occupancyState < 100:
    occupancyFlag = 1
    maintainFlag = 1

#if the stall is not occupied and the toilet has to be cleaned
elif ammoniaLevels < 200 and occupancyState < 100:
    occupancyFlag = 0
    maintainFlag = 1

#if the stall is occupied and the toilet is clean
elif ammoniaLevels > 200 and occupancyState > 100:
    occupancyFlag = 1
    maintainFlag = 0

#if the stall is not occupied and the toilet is not clean
else:
    occupancyFlag = 0
    maintainFlag = 0

#Sending boolean variables to arduino indicate through sensors
board.sendData([occupancyFlag, maintainFlag])
```

Circuit diagram: