**Phase 4 project**

**Project Title: SMART PUBLIC RESTROOM**

**Project ID:** proj\_223730\_Team\_1

**College code:** 6208

**College:** Gnanamani College of Technology

**Year:** IIIrd year

**Branch:** B.Tech-Information Technology

Team Members:

Boopathi.V(620821205009)

Pandiyaraj.S.L(620821205039)

Saranraj.P(620821205049)

Sathishkumar.V(620821205050)

Muralitharan.R(620821205301)

**SMART PUBLIC RESTROOM**

**Definition:**

A smart public restroom, also known as an intelligent or connected restroom, refers to a technologically advanced facility designed to enhance user experience, hygiene, and resource efficiency in public restroom environments. It leverages a range of Internet of Things (IoT) technologies and other smart features to provide a more convenient, safe, and environmentally-friendly experience for users.

**PHASE 4**

Creating a platform that displays real-time restroom availability and cleanliness data, along with mobile apps for iOS and Android, is a complex project. Here's a high-level overview of the steps involved:

1. Backend Development:

* Choose a backend technology stack (e.g., Node.js, Python, Ruby on Rails) to handle data processing and server-side logic.
* Set up a database (e.g., MySQL, PostgreSQL, MongoDB) to store restroom data, including availability and cleanliness status.

2. RESTful API:

* Develop a RESTful API that allows communication between the server and mobile apps. This API will be used to send and receive restroom data.

3. Real-time Data Handling:

* Implement technologies like WebSockets or Server-Sent Events to provide real-time updates on restroom availability and cleanliness.

4. Frontend Web App:

* Use HTML, CSS, and JavaScript (and possibly a frontend framework like React or Angular) to create a web-based platform where users can view restroom data in real-time.

5. Mobile App Development:

* For iOS, use Swift and for Android, use Java or Kotlin to develop mobile apps.
* Implement user-friendly interfaces that can access the RESTful API for real-time restroom data.

6. User Authentication:

* Implement user authentication for mobile apps and web platform, allowing users to log in and contribute data (e.g., marking a restroom as clean or dirty).

7. Data Collection:

* Provide a way for users to report restroom status.
* Use user-generated data to continuously update cleanliness and availability information.

8. Geolocation Services:

* Use GPS services to allow users to find nearby restrooms.

9. Reviews and Ratings:

* Implement a review and rating system for restrooms, so users can provide feedback.

10. Scalability and Load Balancing:

* Ensure that the system can handle a large number of users and restroom data updates by using load balancing and cloud services.

11. Privacy and Data Security:

* Implement strong security measures to protect user data and privacy.

12. Testing:

* Thoroughly test the web platform and mobile apps for functionality, usability, and performance.

13. Deployment:

* Deploy the platform and apps to web servers and app stores (Apple App Store and Google Play Store).

14. Marketing and User Engagement:

* Promote the platform and apps to attract users and encourage them to contribute restroom data.

15. Maintenance and Updates:

* Continuously monitor and maintain the system, addressing bugs and releasing updates as needed.

**HTML (index.html):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="styles.css">

<title>Restroom Status</title>

</head>

<body>

<header>

<h1>Restroom Status</h1>

</header>

<main>

<div id="restroom-data">

<!-- Real-time data will be displayed here -->

</div>

</main>

<footer>

<button id="refresh-button">Refresh</button>

</footer>

<script src="script.js"></script>

</body>

</html>

**CSS (styles.css):**

body {

font-family: Arial, sans-serif;

text-align: center;

margin: 0;

padding: 0;

}

header {

background-color: #0074d9;

color: white;

padding: 20px;

}

h1 {

font-size: 24px;

}

main {

padding: 20px;

}

button {

background-color: #0074d9;

color: white;

border: none;

padding: 10px 20px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

#restroom-data {

font-size: 20px;

}

**JavaScript (script.js):**

document.addEventListener("DOMContentLoaded", () => {

const restroomDataElement = document.getElementById("restroom-data");

const refreshButton = document.getElementById("refresh-button");

// Function to update the restroom data display

function updateRestroomData(data) {

restroomDataElement.textContent = `Availability: ${data.availability}, Cleanliness: ${data.cleanliness}`;

}

// Event listener for the Refresh button

refreshButton.addEventListener("click", () => {

// Simulate data update (replace with real-time data retrieval)

const updatedData = {

availability: "Available",

cleanliness: "Clean"

};

updateRestroomData(updatedData);

});

// Initial data display

const initialData = {

availability: "Available",

cleanliness: "Clean"

};

updateRestroomData(initialData);

});