**Phase-4**

**Smart Public Restroom**

**Introduction :**

The smart public Restroom Website is the development of a web-based platform aimed at efficiently tracking and managing in public places, contributing to a sustainable future.

**Server Side code**

const express = require('express');

const path = require('path');

const mongoose = require('mongoose');

const app = express();

const port = 3000;

app.use(express.static(path.join(\_\_dirname, 'client', 'public')));

app.get('/', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'client', 'public', 'index.html'));

});

mongoose.connect('mongodb://localhost:27017/bathroomDB', {

useNewUrlParser: true,

useUnifiedTopology: true,

});

app.listen(port, () => {

console.log(`Server is running on )

import React from 'react';

const App = () => {

return (

<div>

{/\* Your React components here \*/}

</div>

);

};

export default App;

const mongoose = require('mongoose');

const bathroomSchema = new mongoose.Schema({

name: String,

status: String,

});

const Bathroom = mongoose.model('Bathroom', bathroomSchema);

module.exports = Bathroom;

**WebPage code**

<!DOCTYPE html>

<html>

<head>

<title>Bathroom Availability</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f5f5f5;

text-align: center;

margin: 0;

padding: 0;

}

h1 {

background-color: #333;

color: #fff;

padding: 20px 0;

margin: 0;

}

.container {

max-width: 800px;

margin: 0 auto;

padding: 20px;

background-color: #fff;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

.status {

display: flex;

justify-content: space-between;

align-items: center;

margin: 10px 0;

}

.label {

font-size: 18px;

color: #333;

}

.availability {

font-size: 18px;

font-weight: bold;

color: green; /\* Default color for 'Available' \*/

}

.info-box {

background-color: #f9f9f9;

padding: 20px;

border: 1px solid #ddd;

border-radius: 5px;

margin-top: 20px;

}

.feedback-button {

background-color: #007BFF;

color: #fff;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

margin-top: 20px;

}

.feedback-button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<h1>Bathroom Availability</h1>

<div class="container">

<div class="status">

<div class="label">Bathroom 1:</div>

<div class="availability" id="bathroom1Status">Loading...</div>

</div>

<div class="status">

<div class="label">Bathroom 2:</div>

<div class="availability" id="bathroom2Status">Loading...</div>

</div>

<div class="status">

<div class="label">Bathroom 3:</div>

<div class="availability" id="bathroom3Status">Loading...</div>

</div>

<!-- Additional content -->

<div class="info-box">

<h2>Information</h2>

<p>Welcome to our bathroom availability service. You can check the status of our three bathrooms in real-time. If a bathroom is marked as 'Occupied,' please be patient as it is currently in use. Thank you for using our service!</p>

</div>

<!-- Feedback button -->

<button class="feedback-button" onclick="openFeedbackForm()">Provide Feedback</button>

</div>

<!-- Firebase setup -->

<script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-app.js"></script>

<script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-database.js"></script>

<script>

var firebaseConfig = {

apiKey: "YOUR\_API\_KEY",

authDomain: "YOUR\_AUTH\_DOMAIN",

databaseURL: "YOUR\_DATABASE\_URL",

projectId: "YOUR\_PROJECT\_ID",

storageBucket: "YOUR\_STORAGE\_BUCKET",

messagingSenderId: "YOUR\_MESSAGING\_SENDER\_ID",

appId: "YOUR\_APP\_ID"

};

// Initialize Firebase

firebase.initializeApp(firebaseConfig);

// Get a reference to the Firebase Realtime Database

var db = firebase.database();

// Function to update the bathroom status

function updateBathroomStatus() {

// Reference to the 'bathrooms' node in your Firebase database

var bathroomsRef = db.ref('bathrooms');

bathroomsRef.on('value', function(snapshot) {

var bathroomData = snapshot.val();

if (bathroomData) {

document.getElementById('bathroom1Status').textContent = bathroomData.bathroom1;

document.getElementById('bathroom2Status').textContent = bathroomData.bathroom2;

document.getElementById('bathroom3Status').textContent = bathroomData.bathroom3;

}

});

}

// Function to open the feedback form

function openFeedbackForm() {

document.getElementById("feedback-form").style.display = "block";

}

// Periodically update the bathroom status

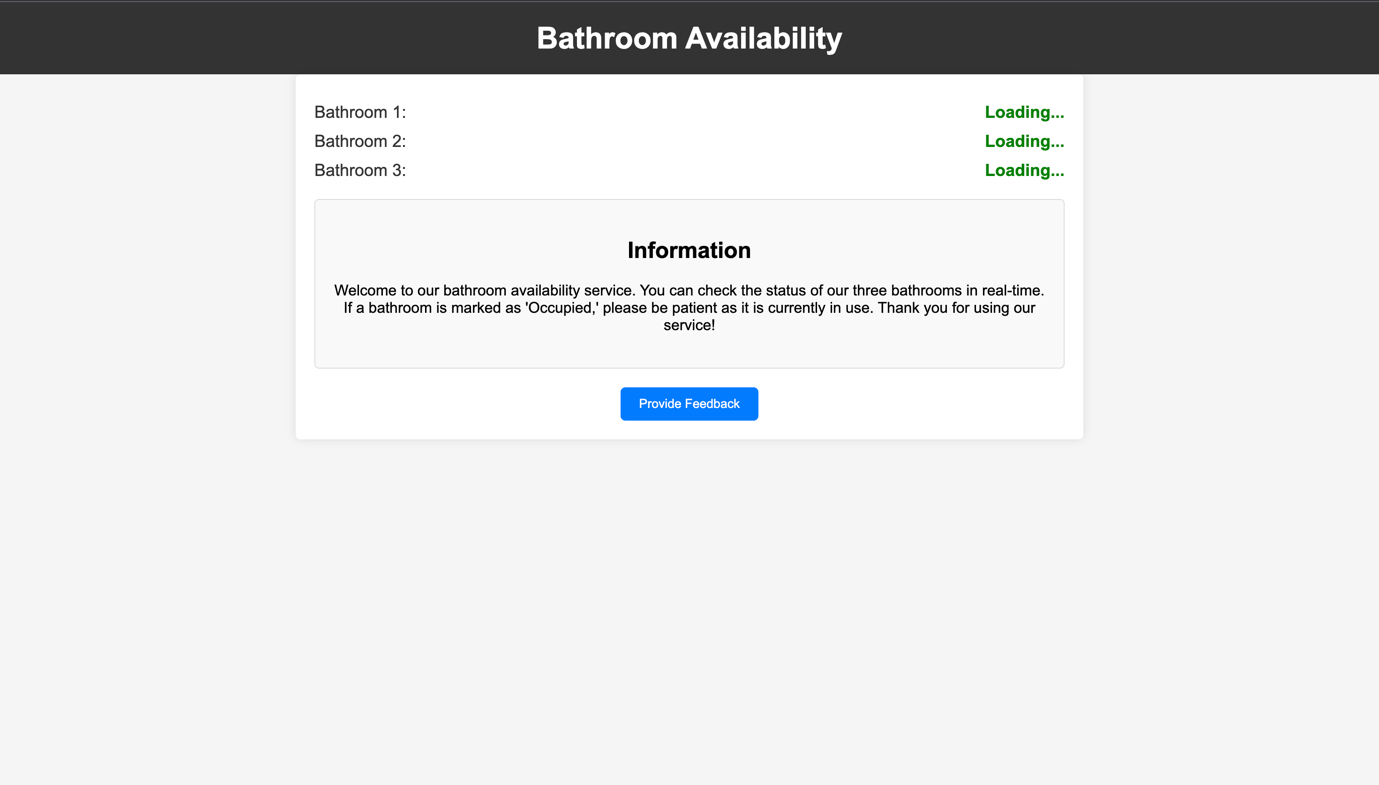
updateBathroomStatus(); // Initial update

</script>

</body>

</html>

**Webpage**

****

**Concustion:**

In conclusion above smart public Restroom Website is the development of a web-based platform aimed at efficiently tracking and managing in public places, contributing to a sustainable future.