

SMART PUBLIC RESTROOM

DEVELOPMENT PART-2

The development of a smart public restroom involves the integration of technology and design to enhance user experience, improve cleanliness, and optimize maintenance. Here are the key steps and components involved in creating a smart public restroom:

Intelligent Waste Management and Sustainability

Waste Sorting and Recycling:

The Smart Public Restroom goes beyond cleanliness; it's committed to sustainability. Smart waste bins equipped with sensors and cameras help users sort their trash correctly. If someone tries to dispose of recyclables in the general waste bin, the system can provide real-time feedback, guiding them to the appropriate bin and contributing to a more sustainable waste management system.

Energy Efficiency:

Energy-efficient systems are incorporated into the restroom's design. Smart lighting adjusts its intensity based on occupancy, natural light, and time of day, reducing energy consumption. Water-saving fixtures and sensors ensure that water is used sparingly. Additionally, solar panels on the roof generate clean energy to power the restroom.

Water Conservation:

The Smart Public Restroom features water-saving fixtures like low-flow toilets and sensor-based faucets. It also uses a greywater recycling system to treat and reuse water for flushing and cleaning. This not only conserves water but also promotes responsible water usage.

Green Roof and Vertical Gardens:

To enhance the aesthetic appeal of the restroom and contribute to environmental sustainability, a green roof covered in vegetation and vertical gardens are integrated into the design. This greenery serves as natural insulation, absorbs rainwater, and improves air quality.

Data Analytics for Sustainability:

It expertise in data analytics is leveraged to monitor and analyze the restroom's resource usage, waste generation, and environmental impact. This data helps in making informed decisions to further optimize resource management and sustainability efforts.

User Engagement:

The restroom has interactive screens that educate users on sustainability, water conservation, and waste reduction. Users can track their contributions to environmental conservation and receive rewards for responsible usage. These rewards may include discounts at local eco-friendly businesses or virtual badges for an eco-conscious lifestyle.

Community Involvement:

The Smart Public Restroom promotes community involvement in sustainability. It hosts events, workshops, and cleanup drives in collaboration with local environmental organizations. These events encourage citizens to take an active role in maintaining a clean and green public space.

Regular Maintenance and Self-Reporting:

IoT sensors in the restroom continuously monitor equipment health and cleanliness. If any issues are detected, the system automatically generates maintenance requests, ensuring that the restroom remains in perfect condition at all times.

Feedback Mechanism:

Users can provide feedback through a mobile app or touchscreen kiosks within the restroom. This feedback is processed through AI to improve service quality and make real-time adjustments based on user suggestions.

Security and Privacy:

The commitment to security and privacy ensures that user data is protected, and surveillance systems are only used for the intended purposes of restroom management and public safety.

The Smart Public Restroom not only provides a clean and safe environment for the public but also actively contributes to the sustainability and environmental consciousness of the community. Part 2 of this concept focuses on waste management, sustainability, data analytics, user engagement, and community involvement, all of which are essential components of a modern, intelligent public restroom.

HTML CODE FOR WASTE SORTING AND RECYCLING:

To create an HTML user interface for waste sorting and recycling, you can design a web page that allows users to select the type of waste they are disposing of (e.g., recyclable or non-recyclable). Here’s a simple HTML code example for a waste sorting and recycling interface:

```
<!DOCTYPE html>

<html>

<head>
```

```
<title>Waste Sorting and Recycling</title>
```

```
<style>
```

```
  body {
```

```
    font-family: Arial, sans-serif;
```

```
    text-align: center;
```

```
  }
```

```
  h1 {
```

```
    color: #333;
```

```
  }
```

```
  .container {
```

```
    display: flex;
```

```
    justify-content: center;
```

```
    align-items: center;
```

```
    height: 80vh;
```

```
  }
```

```
  .button-container {
```

```
    display: flex;
```

```
    justify-content: center;
```

```
    margin: 20px;
```

```
  }
```

```
  .button {
```

```
    padding: 10px 20px;
```

```
    font-size: 16px;
```

```
    background-color: #4CAF50; /* Green button color */
```

```
    color: white;
```

```
    border: none;
```

```
    border-radius: 5px;
```

```
    cursor: pointer;
```

```
  }
```

```
  .button.red {
```

```
        background-color: #FF5733; /* Red button color */
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Waste Sorting and Recycling</h1>
        <p>Select the type of waste you are disposing of:</p>
        <div class="button-container">
            <button class="button" id="recyclableBtn">Recyclable</button>
            <button class="button red" id="nonRecyclableBtn">Non-Recyclable</button>
        </div>
    </div>

    <script>
        // JavaScript code can be added here to handle user interactions and data processing.
        Document.getElementById("recyclableBtn").addEventListener("click", function() {
            alert("You have selected Recyclable waste. Please dispose of it in the Recyclable
bin.");

            // You can add further logic here to communicate with hardware or a backend
            system.

        });

        document.getElementById("nonRecyclableBtn").addEventListener("click",
function() {

            alert("You have selected Non-Recyclable waste. Please dispose of it in the Non-
Recyclable bin.");

            // You can add further logic here to communicate with hardware or a backend
            system.

        });
    </script>
</body>
</html>
```

In this HTML code:

We create a simple webpage with a title, heading, and buttons for selecting recyclable or non-recyclable waste.

The buttons have event listeners in JavaScript to display a message when clicked. You can replace the alert messages with actual actions or commands to communicate with waste sorting hardware.

Remember to adapt the JavaScript code to interface with your specific hardware or backend system for waste sorting and recycling.

HTML CODE FOR ENERGY CONSUMPTION:

Creating an HTML interface for displaying energy consumption typically involves presenting data in a user-friendly way. Here's a basic HTML code example for an energy consumption display:

```
<!DOCTYPE html>
<html>
<head>
  <title>Energy Consumption Dashboard</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
    }
    h1 {
      color: #333;
    }
    .container {
      display: flex;
      justify-content: center;
      align-items: center;
```

```
        height: 80vh;
    }

    .data-container {
        border: 1px solid #ccc;
        padding: 20px;
        border-radius: 10px;
    }

    .data {
        font-size: 24px;
    }

    .unit {
        font-size: 16px;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Energy Consumption Dashboard</h1>
        <div class="data-container">
            <p class="data">Current Energy Consumption:</p>
            <p class="data"><span id="currentEnergy">0</span> kWh</p>
            <p class="unit">(Kilowatt-hours)</p>
        </div>
    </div>
    <script>
        const currentEnergyConsumption = 50; // kWh

        document.getElementById("currentEnergy").textContent =
currentEnergyConsumption;
    </script>
</body>
```

```
</html>
```

In this HTML code:

We create a simple webpage with a title, heading, and a container to display energy consumption data.

The JavaScript section is used to update the energy consumption data. In a real application, you would fetch this data from a sensor or a database and replace the simulated value with actual data.

Remember that this is a basic example, and you can customize the HTML and JavaScript code to match the specific needs and data sources of your energy consumption monitoring system.

HTML CODE FOR TOILET PAPER LEVEL DETECTION:

To create a basic HTML user interface for displaying toilet paper level, you can use a simple webpage that shows the current level of toilet paper. You would need to use JavaScript or a back-end system to obtain and update the toilet paper level information. Here's a basic HTML code example:

```
<!DOCTYPE html>

<html>

<head>

  <title>Toilet Paper Level</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      text-align: center;

    }

    h1 {

      color: #333;

    }

    .container {
```

```
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    height: 80vh;
}

.level-container {
    background-color: #eee;
    padding: 20px;
    border-radius: 10px;
}

.level {
    font-size: 24px;
}

.unit {
    font-size: 16px;
}

</style>

</head>

<body>

    <div class="container">

        <h1>Toilet Paper Level</h1>

        <div class="level-container">

            <p class="level">Current Toilet Paper Level:</p>

            <p class="level"><span id="toiletPaperLevel">0</span></p>

            <p class="unit">Sheets</p>

        </div>

    </div>

    <script>

)
```



```
const currentToiletPaperLevel = 50;

document.getElementById("toiletPaperLevel").textContent =
currentToiletPaperLevel;

</script>

</body>

</html>
```

In this HTML code:

We create a simple webpage with a title, heading, and a container to display the toilet paper level.

The JavaScript section is used to update the toilet paper level data. In a real application, you would fetch this data from a sensor or a database and replace the simulated value with actual data.

Customize the HTML and JavaScript code to match the specific needs and data sources of your toilet paper level detection system.

HTML CODE FOR WATER LEAK DETECTION:

Creating a user interface for water leak detection typically involves displaying real-time information about the status of the water leak sensor. Here's a basic HTML code example for a water leak detection interface:

```
<!DOCTYPE html>

<html>

<head>

  <title>Water Level Detection</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      text-align: center;

    }

    h1 {

      color: #333;

    }

    .container {
```

```

    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    height: 80vh;
}
.level-container {
    background-color: #eee;
    padding: 20px;
    border-radius: 10px;
}
.level {
    font-size: 24px;
}
.unit {
    font-size: 16px;
}
</style>
</head>
<body>
    <div class="container">
        <h1>Water Level Detection</h1>
        <div class="level-container">
            <p class="level">Current Water Level:</p>
            <p class="level"><span id="waterLevel">0</span></p>
            <p class="unit">cm</p>
        </div>
    </div>
    <script>
a database.

```

```
const currentWaterLevel = 30;
```

```
    document.getElementById("waterLevel").textContent = currentWaterLevel;
```

```
</script>
```

```
</body>
```

```
</html>
```