To develop a smart public restroom information platform and mobile app that displays real-time restroom availability and cleanliness data using IoT and web development technologies Using (HTML, CSS, JavaScript), you can follow these steps:

1. Hardware Setup:

   - Install IoT devices in each restroom to collect data. For simplicity, let's assume each device has a button for availability and a slider for cleanliness.

   - Use an IoT platform like Arduino or Raspberry Pi to connect the devices to the internet and send data to a server.

2. Server-side Development:

   - Set up a server using a backend technology like Node.js.

   - Create API endpoints to receive data from IoT devices.

   - Store the received data in a database for further processing.

3. Web Application Development:

   - Design the user interface using HTML and CSS.

   - Create a web page that displays the real-time restroom information.

   - Use JavaScript to fetch data from the server's API endpoints and update the web page dynamically.

   - Update the UI elements based on the availability and cleanliness data received.

Here's an example code snippet for the web application:

```html
<!DOCTYPE html>
<html>
<head>
  <title>Restroom Information Platform</title>
  <style>
    /* CSS styles for the web page */
  </style>
```

```
</head>
<body>
  <h1>Restroom Information</h1>
  <div id="availability"></div>
  <div id="cleanliness"></div>

  <script>
    // JavaScript code to fetch and update data
    // Fetch availability data from the server
    fetch('/api/availability')
      .then(response => response.json())
      .then(data => {
        const availabilityElement = document.getElementById('availability');
        availabilityElement.textContent = `Availability: ${data.available ? 'Available' : 'Occupied'}`;
      });

    // Fetch cleanliness data from the server
    fetch('/api/cleanliness')
      .then(response => response.json())
      .then(data => {
        const cleanlinessElement = document.getElementById('cleanliness');
        cleanlinessElement.textContent = `Cleanliness: ${data.level}`;
      });
  </script>
</body>
</html>
```

4. Mobile App Development:

- Use a mobile app development framework like React Native or Flutter.
- Design the mobile app interface with screens to display real-time restroom information.
- Utilize APIs provided by the server to fetch data and update the app's UI.
- Implement features like real-time updates and user authentication.

Remember that this is a simplified example, and you would need to adapt it to your specific requirements. Additionally, you would need to handle user authentication, implement data persistence, and consider security measures.

For completeness, you would also need to implement the server-side code to receive data from IoT devices, store it in the database, and provide API endpoints for fetching the data. This would involve using a backend framework (e.g., Express for Node.js) and a database system (e.g., MongoDB).

Overall, this project requires a combination of IoT, backend development, web development, and mobile app development skills.