Building an IoT Smart Public restroom system involves multiple steps, and I can provide you with an outline for Phase 3, Development Part 1.

Creating a complete smart public restroom project using Python code requires an extensive implementation involving various hardware components and sensors. However, I can provide you with a sample code snippet that demonstrates a basic functionality using a simulated occupancy sensor. Please keep in mind that this is a simplified example, and you would need to adapt and expand it based on your specific project requirements.

~~~
```python
import time


# Simulated occupancy sensor
class OccupancySensor:
    def __init__(self):
        self.occupied = False


    def is_occupied(self):
        # Simulating the occupancy status
        return self.occupied


    def update_occupancy(self, status):
        self.occupied = status


# Function to control restroom lighting
def control_lighting(occupied):
    if occupied:
        print("Restroom is occupied. Turning on the lights.")
        # Code to control the lighting system
    else:
        print("Restroom is vacant. Turning off the lights.")
~~~

```
        # Code to turn off the lights


# Main program
if __name__ == '__main__':
    # Initialize the occupancy sensor
    occupancy_sensor = OccupancySensor()

    try:
        while True:
            # Read the occupancy status
            occupied = occupancy_sensor.is_occupied()

            # Control the restroom lighting based on occupancy
            control_lighting(occupied)

            # Wait for a certain interval before checking occupancy again
            time.sleep(5)  # Adjust the interval as per your requirements

    except KeyboardInterrupt:
        print("Program stopped by the user.")
```

In the above code, we have a simulated occupancy sensor represented by the `OccupancySensor` class. The `is_occupied()` method returns the current occupancy status, and the `update_occupancy()` method can be used to update the occupancy status.

The `control_lighting()` function simulates controlling the restroom lighting based on the occupancy status. You can replace the print statements with actual code to control the lighting system.

In the main program, we continuously check the occupancy status using a while loop. We call the `is_occupied()` method of the occupancy sensor, and then pass the occupancy status to the `control_lighting()` function to control the lighting accordingly. The program repeats this process at a specified interval using the `time.sleep()` function.