**Simulation code:**

import random

class Room:

def \_\_init\_\_(self, name):

self.name = name

self.temperature = 22

self.lights\_on = False

self.occupied = False

self.energy\_usage = 0.0

def update(self, occupied):

self.occupied = occupied

self.lights\_on = occupied

self.energy\_usage = self.calculate\_energy\_usage()

if occupied:

self.temperature = 22

else:

self.temperature = 20

def calculate\_energy\_usage(self):

base\_usage = 0.5 if self.lights\_on else 0.1

hvac\_usage = 0.3 if self.occupied else 0.1

return base\_usage + hvac\_usage

class SecuritySystem:

def \_\_init\_\_(self):

self.doors\_locked = True

def check\_doors(self, hour):

after\_hours = hour >= 18 or hour < 6

self.doors\_locked = after\_hours

return self.doors\_locked

class SmartBuilding:

def \_\_init\_\_(self):

self.rooms = {

"Lobby": Room("Lobby"),

"Office1": Room("Office1"),

"Office2": Room("Office2"),

"Conference": Room("Conference"),

}

self.security = SecuritySystem()

self.current\_hour = 0

def simulate\_step(self):

print(f'\n--- Building Status Update at {self.current\_hour}:00 ---')

total\_energy\_usage = 0

for room\_name, room in self.rooms.items():

occupied = random.choice([True, False]) if self.current\_hour >= 8 and self.current\_hour < 18 else False

room.update(occupied)

total\_energy\_usage += room.energy\_usage

print(

f"{room\_name} Occupied: {room.occupied}, "

f"Lights: {'On' if room.lights\_on else 'Off'}, "

f"Temp: {room.temperature}°C, "

f"{room.energy\_usage:.2f} kWh"

)

doors\_locked =self.security.check\_doors(self.current\_hour)

print(

f"Security: Doors {'Locked' if doors\_locked else 'Unlocked'}"

)

print(

f"Total Energy Consumption: {total\_energy\_usage:.2f} kWh"

)

self.current\_hour = (self.current\_hour + 1) % 24

# Run simulation

building = SmartBuilding()

for \_ in range(24): # Simulate 24 hours

building.simulate\_step()