SMART PARKING

# Hardware Components:

1. Ultrasonic Distance Sensors
2. Arduino board (e.g., Arduino Uno or ESP8266)
3. IoT Module (e.g., ESP8266 or ESP32)
4. LED indicators
5. Breadboard and jumper wires

# Software Components:

1. Arduino IDE
2. HTML, CSS, JavaScript for the web interface
3. MQTT for communication between Arduino and IoT module
4. A server or cloud service to host the web interface

**Python Code for a Simulated Smart Parking System**:

python Copy code

class ParkingLot:

def init(self, total\_spots): self.total\_spots = total\_spots self.available\_spots = total\_spots

def occupy\_spot(self):

if self.available\_spots > 0: self.available\_spots -= 1 return True

else:

return False

def vacate\_spot(self):

if self.available\_spots < self.total\_spots: self.available\_spots += 1

return True else:

return False

def get\_status(self):

return f"Total spots: {self.total\_spots}, Available spots:

{self.available\_spots}"

def main():

parking\_lot = ParkingLot(total\_spots=10)

while True:

print("\nSmart Parking System") print("1. Park a vehicle") print("2. Remove a vehicle") print("3. Check parking status") print("4. Exit")

choice = input("Enter your choice: ") if choice == '1':

if parking\_lot.occupy\_spot():

print("Vehicle parked successfully.") else:

print("Parking lot is full.") elif choice == '2':

if parking\_lot.vacate\_spot(): print("Vehicle removed successfully.")

else:

print("Parking lot is already empty.") elif choice == '3':

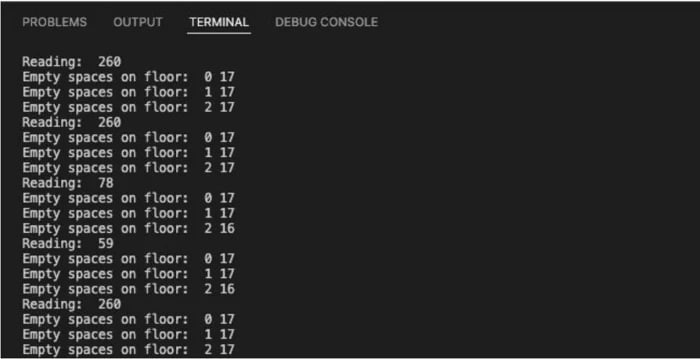
print(parking\_lot.get\_status()) elif choice == '4':

break else:

print("Invalid choice. Please try again.")

if name == "main": main()

**Debug Log Output:**



**OUTPUT-spot is empty OUTPUT-spot is full**

