#include <Stepper.h>

const int stepsPerRevolution = 2048;

const int NUM\_SPOTS = 4;

const int FARE\_PER\_MIN = 2;

Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

const int spotSensors[NUM\_SPOTS] = {A0, A1, A2};

const int entrySensor = A3;

const int obstacleSensor = 2;

bool spotState[NUM\_SPOTS] = {false};

unsigned long entryTime[NUM\_SPOTS] = {0};

int lastAvailable = -1;

void setup() {

  Serial.begin(9600);

  for (int i = 0; i < NUM\_SPOTS; i++) pinMode(spotSensors[i], INPUT\_PULLUP);

  pinMode(entrySensor, INPUT\_PULLUP);

  pinMode(obstacleSensor, INPUT\_PULLUP);

  myStepper.setSpeed(10);

  Serial.println("Smart Parking Initialized");

}

void loop() {

  int available = updateSpots();

  if (available != lastAvailable) {

    Serial.println(available > 0 ? String("Available spots: ") + available : "Parking Lot FULL");

    lastAvailable = available;

  }

  if (digitalRead(entrySensor) == LOW) {

    Serial.println("Vehicle at entry...");

    if (available > 0) {

      openBarrier();

      delay(3000);

      closeBarrierWithCheck();

    } else {

      Serial.println("Access Denied: No spots.");

    }

    delay(1500);

  }

  delay(300);

}

int updateSpots() {

  int available = 0;

  for (int i = 0; i < NUM\_SPOTS; i++) {

    bool current = (digitalRead(spotSensors[i]) == LOW);

    if (current != spotState[i]) {

      unsigned long now = millis();

      if (current) {

        entryTime[i] = now;

        Serial.println("Vehicle in Spot " + String(i + 1));

      } else {

        int mins = max(1, (int)((now - entryTime[i]) / 60000));

        int fare = mins \* FARE\_PER\_MIN;

        Serial.println("Spot " + String(i + 1) + " Vacated. Time: " + mins + " mins. Fare: ₹" + fare);

      }

      spotState[i] = current;

    }

    if (!spotState[i]) available++;

  }

  return available;

}

void openBarrier() {

  Serial.println("Opening barrier...");

  myStepper.step(stepsPerRevolution / 2);

}

void closeBarrierWithCheck() {

  Serial.println("Closing barrier...");

  int stepsLeft = stepsPerRevolution / 2;

  const int chunk = 50;

  while (stepsLeft > 0) {

    if (digitalRead(obstacleSensor) == LOW) {

      Serial.println("Obstacle detected! Waiting...");

      while (digitalRead(obstacleSensor) == LOW) delay(100);

      Serial.println("Obstacle cleared.");

    }

    int steps = min(chunk, stepsLeft);

    myStepper.step(-steps);

    stepsLeft -= steps;

    delay(30);

  }

  Serial.println("Barrier Closed.");

}