MKT Labs

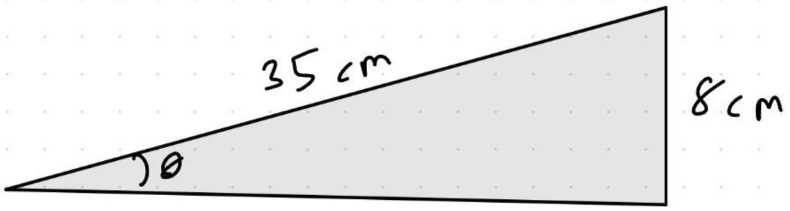
TOPL

Totally Ordinary Parking Lot Presented by:
Tom Croux, Muhammad Al-Lami, Mohib Amin and Krish Patel

Purpose Statement

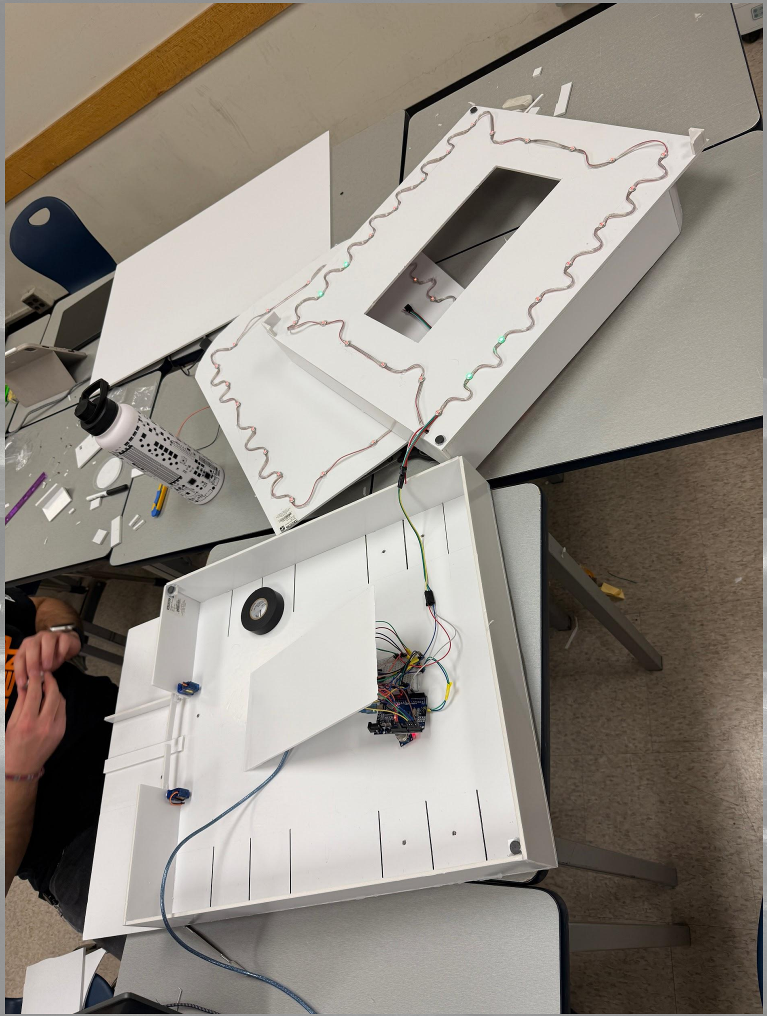
- Optimizing a Sustainable Society Using AI and Automation
- Solution: TOPL, a smart AI powered parking lot

Design Choice

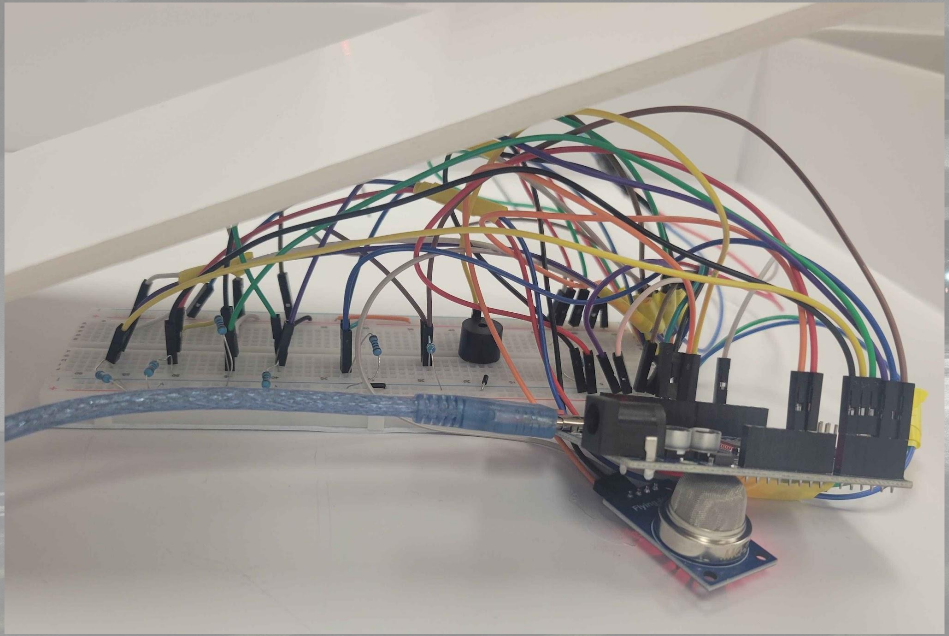


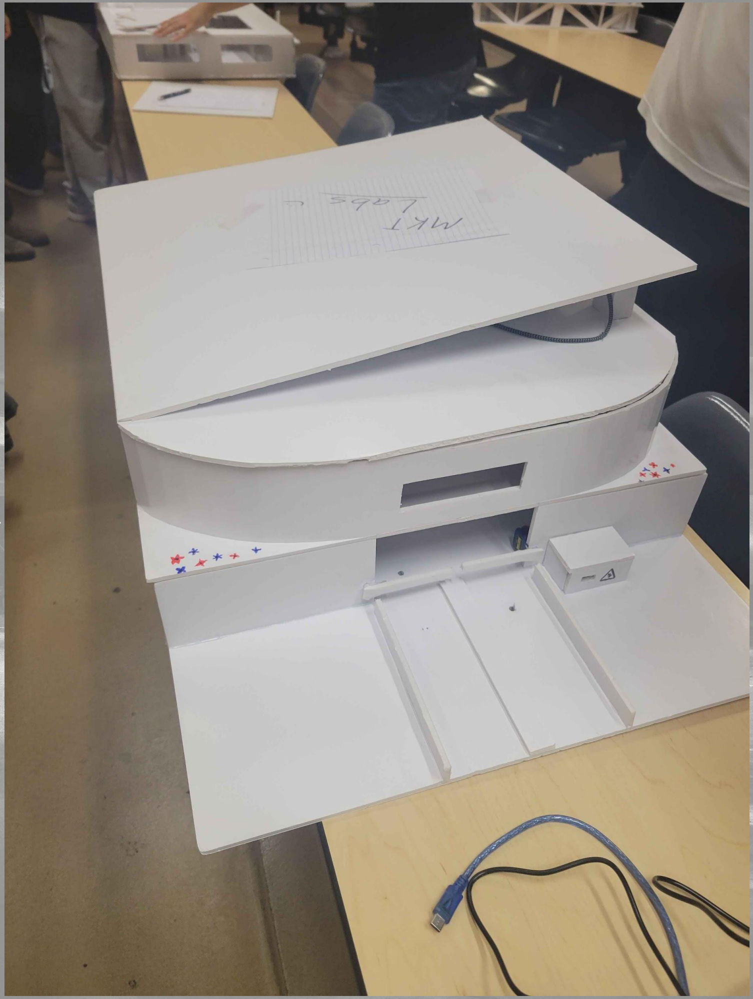
$$\theta = \sin^{-1}\left(\frac{8}{35}\right)$$

$$\theta \approx 23^{\circ}$$



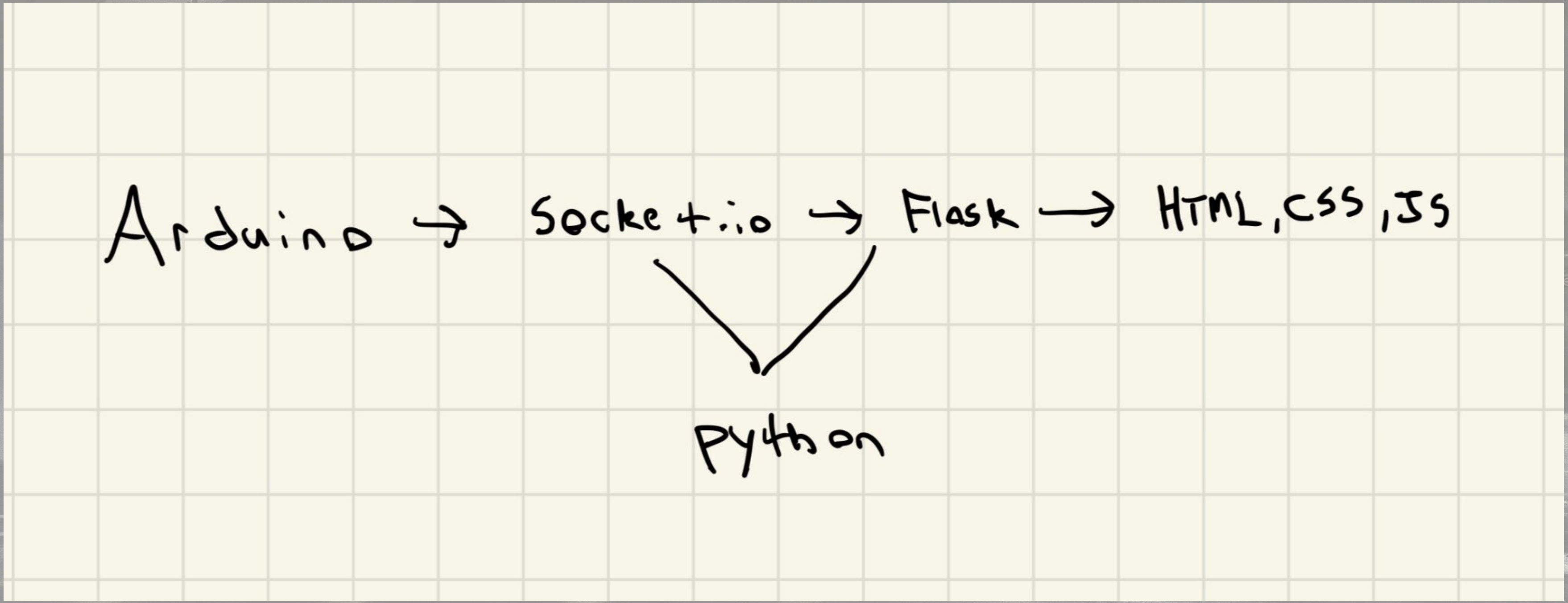
Design Choice





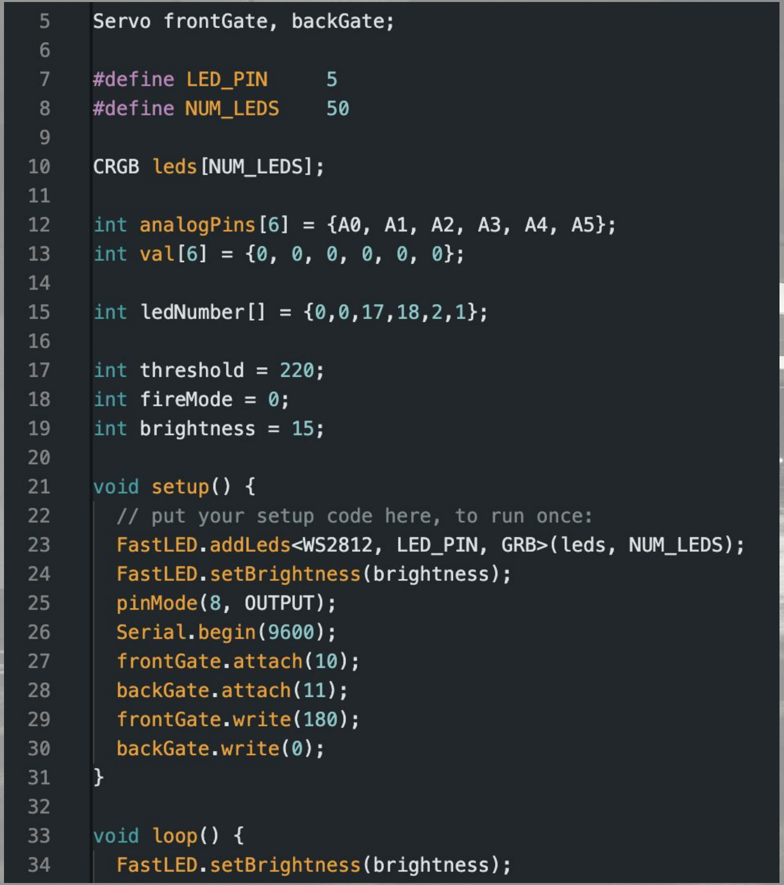
Code Overview

Logic

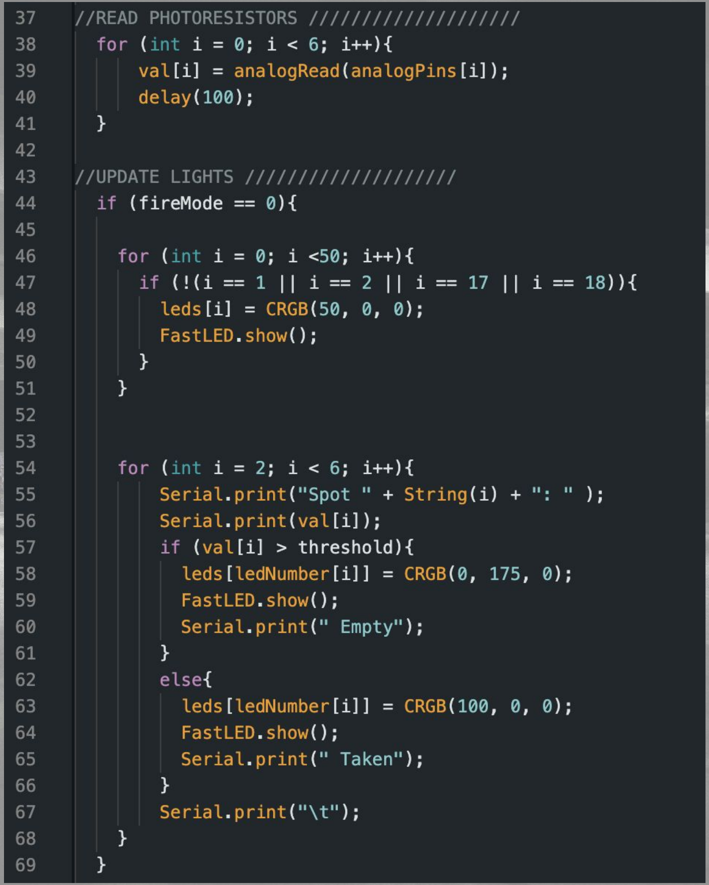


Code Overview

Arduino



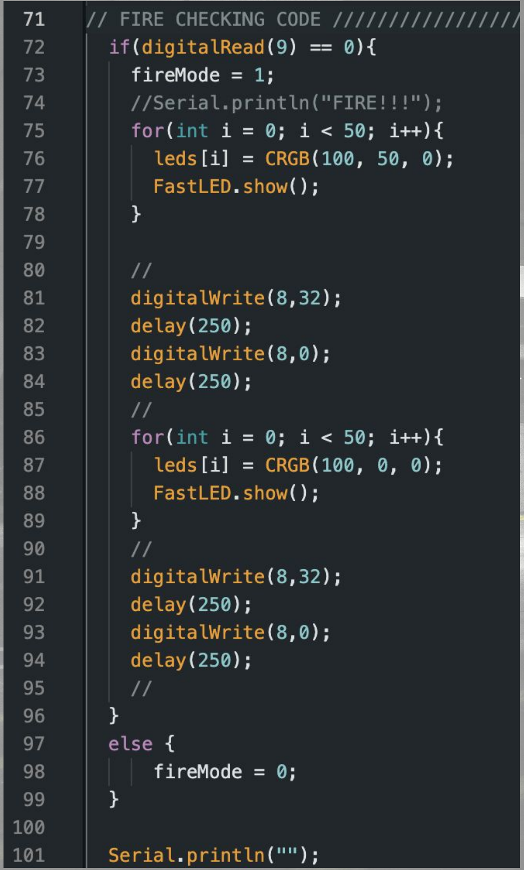
```
Servo frontGate, backGate;

#define LED_PIN     5
#define NUM_LEDS    50

CRGB leds[NUM_LEDS];

int analogPins[6] = {A0, A1, A2, A3, A4, A5};
int val[6] = {0, 0, 0, 0, 0, 0};

int ledNumber[] = {0,0,17,18,2,1};

int threshold = 220;
int fireMode = 0;
int brightness = 15;

void setup() {
  // put your setup code here, to run once:
  FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS);
  FastLED.setBrightness(brightness);
  pinMode(8, OUTPUT);
  Serial.begin(9600);
  frontGate.attach(10);
  backGate.attach(11);
  frontGate.write(180);
  backGate.write(0);
}

void loop() {
  FastLED.setBrightness(brightness);
```



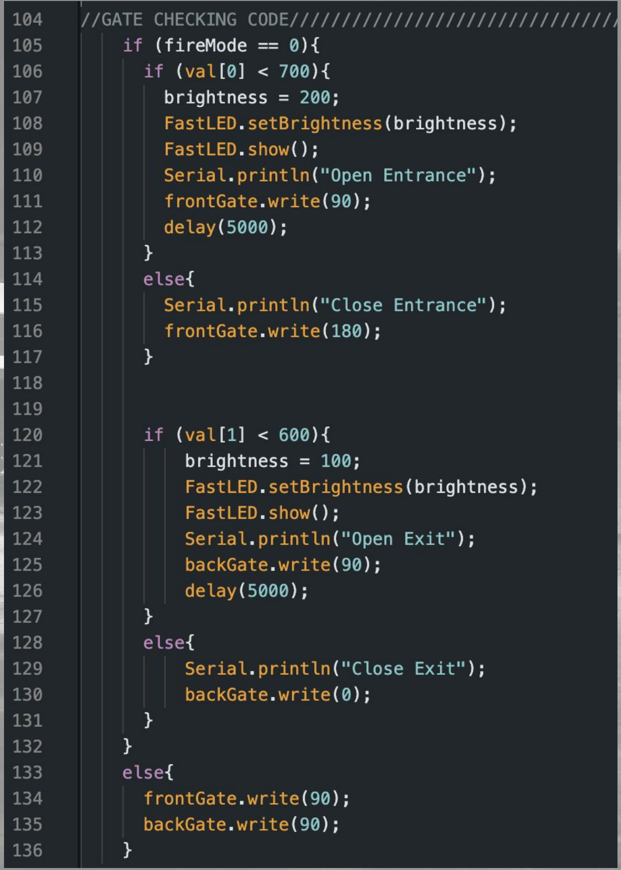
```
//READ PHOTORESISTORS ////////////////////
  for (int i = 0; i < 6; i++){
      val[i] = analogRead(analogPins[i]);
      delay(100);
  }

//UPDATE LIGHTS ////////////////////
  if (fireMode == 0){

    for (int i = 0; i <50; i++){
      if (!(i == 1 || i == 2 || i == 17 || i == 18)){
        leds[i] = CRGB(50, 0, 0);
        FastLED.show();
      }
    }


    for (int i = 2; i < 6; i++){
        Serial.print("Spot " + String(i) + ": " );
        Serial.print(val[i]);
        if (val[i] > threshold){
          leds[ledNumber[i]] = CRGB(0, 175, 0);
          FastLED.show();
          Serial.print(" Empty");
        }
        else{
          leds[ledNumber[i]] = CRGB(100, 0, 0);
          FastLED.show();
          Serial.print(" Taken");
        }
        Serial.print("\t");
    }
  }
```

Code Overview

Arduino



```
// FIRE CHECKING CODE /////////////////
  if(digitalRead(9) == 0){
    fireMode = 1;
    //Serial.println("FIRE!!!");
    for(int i = 0; i < 50; i++){
      leds[i] = CRGB(100, 50, 0);
      FastLED.show();
    }

    //
    digitalWrite(8,32);
    delay(250);
    digitalWrite(8,0);
    delay(250);
    //
    for(int i = 0; i < 50; i++){
      leds[i] = CRGB(100, 0, 0);
      FastLED.show();
    }
    //
    digitalWrite(8,32);
    delay(250);
    digitalWrite(8,0);
    delay(250);
    //
  }
  else {
      fireMode = 0;
  }

  Serial.println("");
```



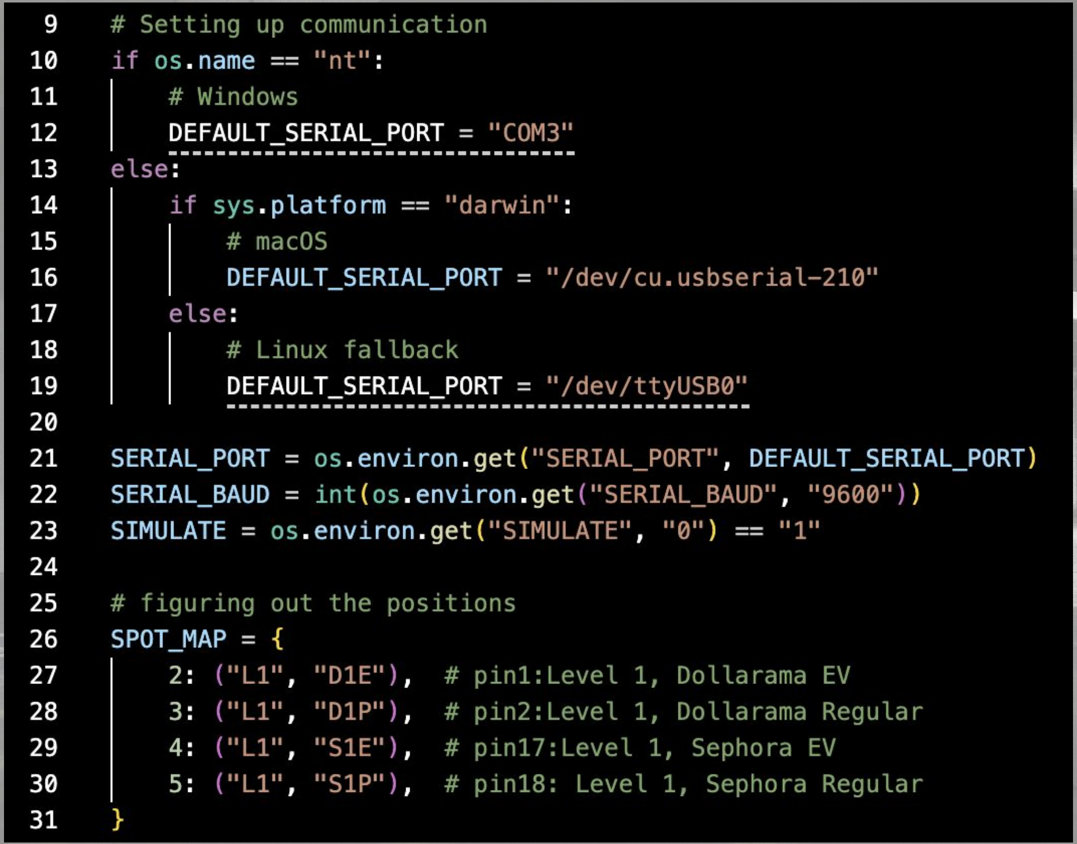
```
//GATE CHECKING CODE///////////////////////////////////
    if (fireMode == 0){
      if (val[0] < 700){
        brightness = 200;
        FastLED.setBrightness(brightness);
        FastLED.show();
        Serial.println("Open Entrance");
        frontGate.write(90);
        delay(5000);
      }
      else{
        Serial.println("Close Entrance");
        frontGate.write(180);
      }


      if (val[1] < 600){
          brightness = 100;
          FastLED.setBrightness(brightness);
          FastLED.show();
          Serial.println("Open Exit");
          backGate.write(90);
          delay(5000);
      }
      else{
          Serial.println("Close Exit");
          backGate.write(0);
      }
    }
    else{
      frontGate.write(90);
      backGate.write(90);
    }
```

Brightness Gets Reduced to Save on Power

“brightness = 100”

Code Overview

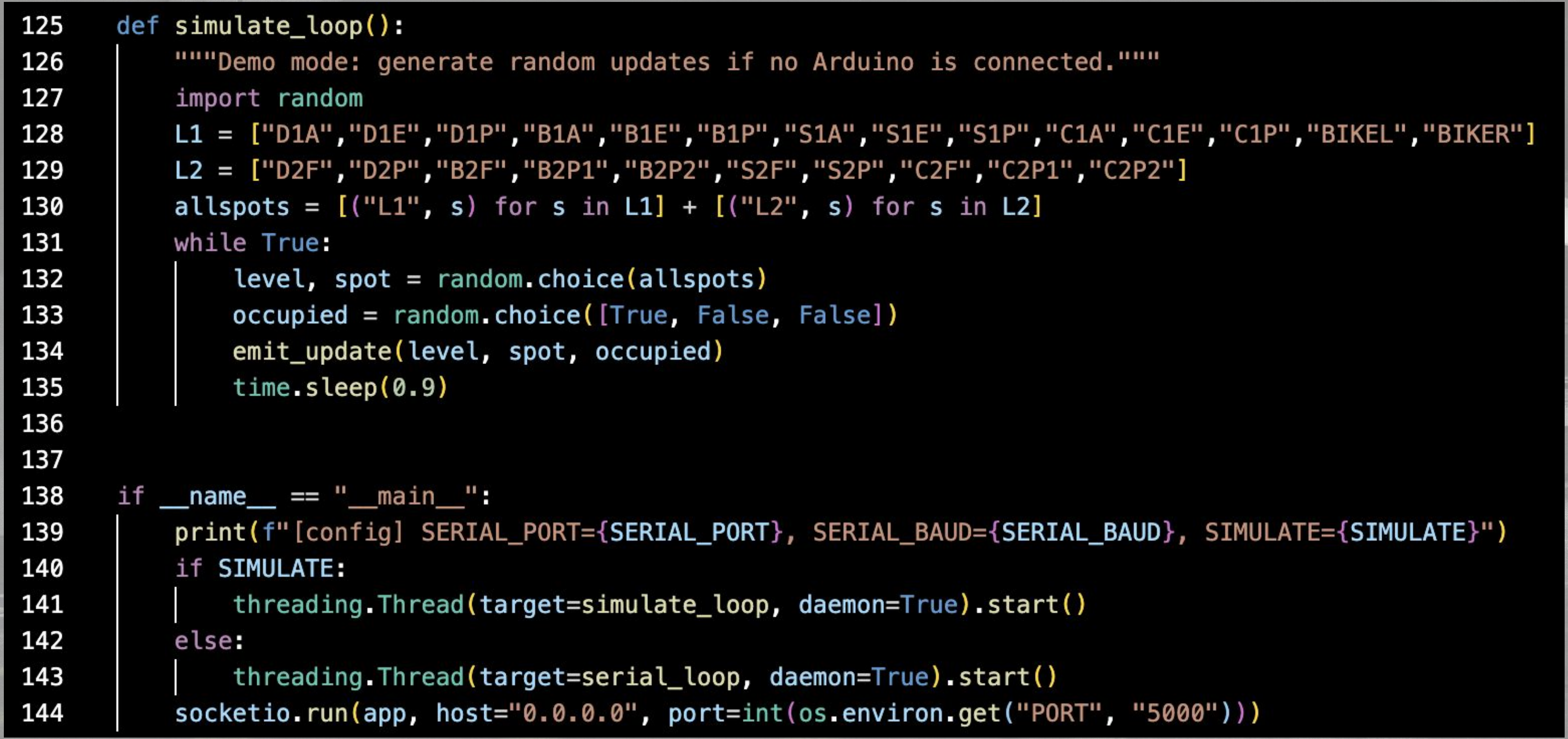
Python App



```python
# Setting up communication
if os.name == "nt":
    # Windows
    DEFAULT_SERIAL_PORT = "COM3"
else:
    if sys.platform == "darwin":
        # macOS
        DEFAULT_SERIAL_PORT = "/dev/cu.usbserial-210"
    else:
        # Linux fallback
        DEFAULT_SERIAL_PORT = "/dev/ttyUSB0"

SERIAL_PORT = os.environ.get("SERIAL_PORT", DEFAULT_SERIAL_PORT)
SERIAL_BAUD = int(os.environ.get("SERIAL_BAUD", "9600"))
SIMULATE = os.environ.get("SIMULATE", "0") == "1"

# figuring out the positions
SPOT_MAP = {
    2: ("L1", "D1E"),  # pin1:Level 1, Dollarama EV
    3: ("L1", "D1P"),  # pin2:Level 1, Dollarama Regular
    4: ("L1", "S1E"),  # pin17:Level 1, Sephora EV
    5: ("L1", "S1P"),  # pin18: Level 1, Sephora Regular
}
```

Code Overview

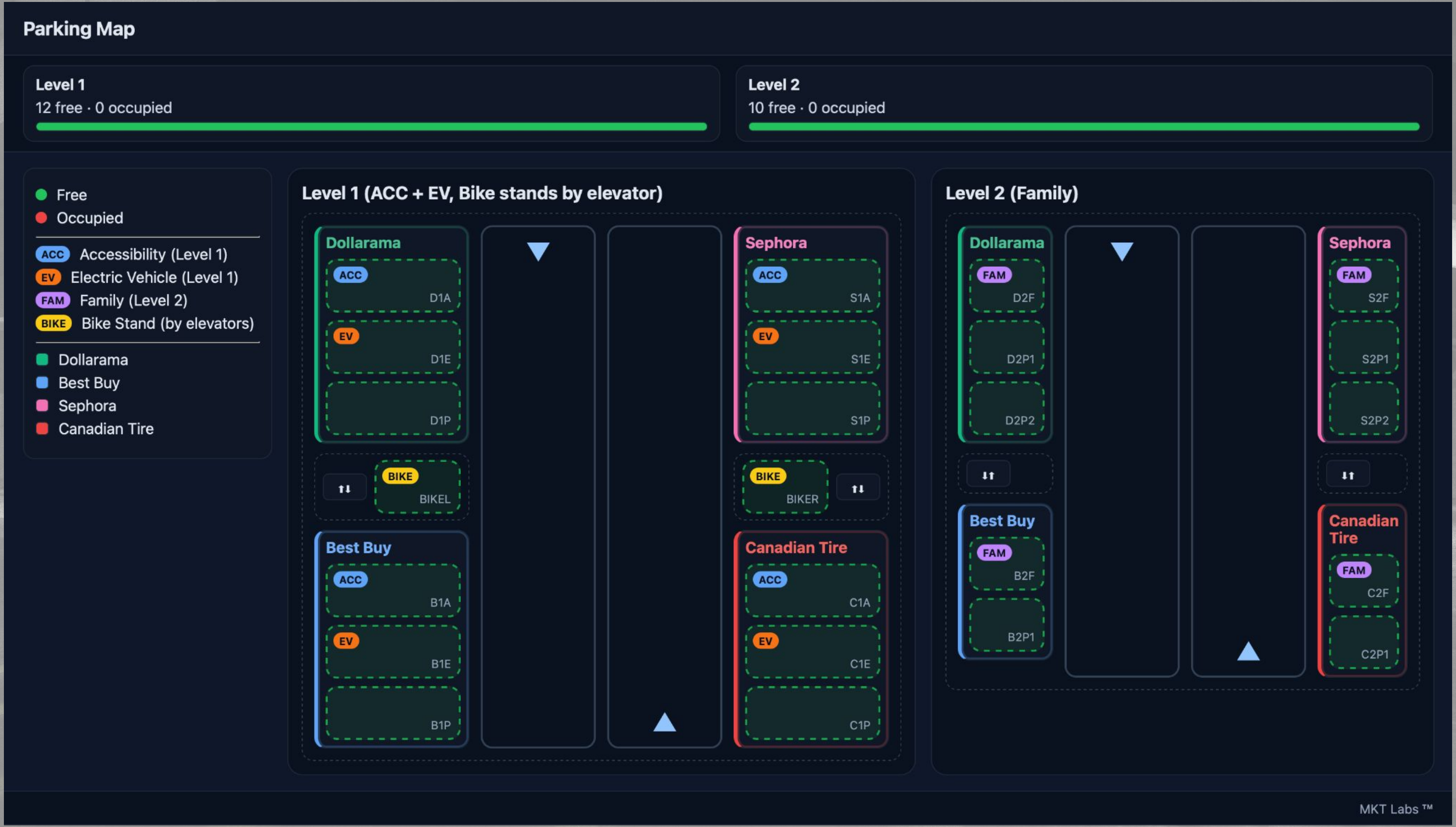
Python App



```python
def simulate_loop():
    """Demo mode: generate random updates if no Arduino is connected."""
    import random
    L1 = ["D1A","D1E","D1P","B1A","B1E","B1P","S1A","S1E","S1P","C1A","C1E","C1P","BIKEL","BIKER"]
    L2 = ["D2F","D2P","B2F","B2P1","B2P2","S2F","S2P","C2F","C2P1","C2P2"]
    allspots = [("L1", s) for s in L1] + [("L2", s) for s in L2]
    while True:
        level, spot = random.choice(allspots)
        occupied = random.choice([True, False, False])
        emit_update(level, spot, occupied)
        time.sleep(0.9)


if __name__ == "__main__":
    print(f"[config] SERIAL_PORT={SERIAL_PORT}, SERIAL_BAUD={SERIAL_BAUD}, SIMULATE={SIMULATE}")
    if SIMULATE:
        threading.Thread(target=simulate_loop, daemon=True).start()
    else:
        threading.Thread(target=serial_loop, daemon=True).start()
    socketio.run(app, host="0.0.0.0", port=int(os.environ.get("PORT", "5000")))
```

UI Overview (No Vehicles)

Python App



Parking Map

Level 1
12 free · 0 occupied

Level 2
10 free · 0 occupied

- 🟢 Free
- 🔴 Occupied
- ACC Accessibility (Level 1)
- EV Electric Vehicle (Level 1)
- FAM Family (Level 2)
- BIKE Bike Stand (by elevators)
- 🟩 Dollarama
- 🟦 Best Buy
- 🟪 Sephora
- 🟥 Canadian Tire

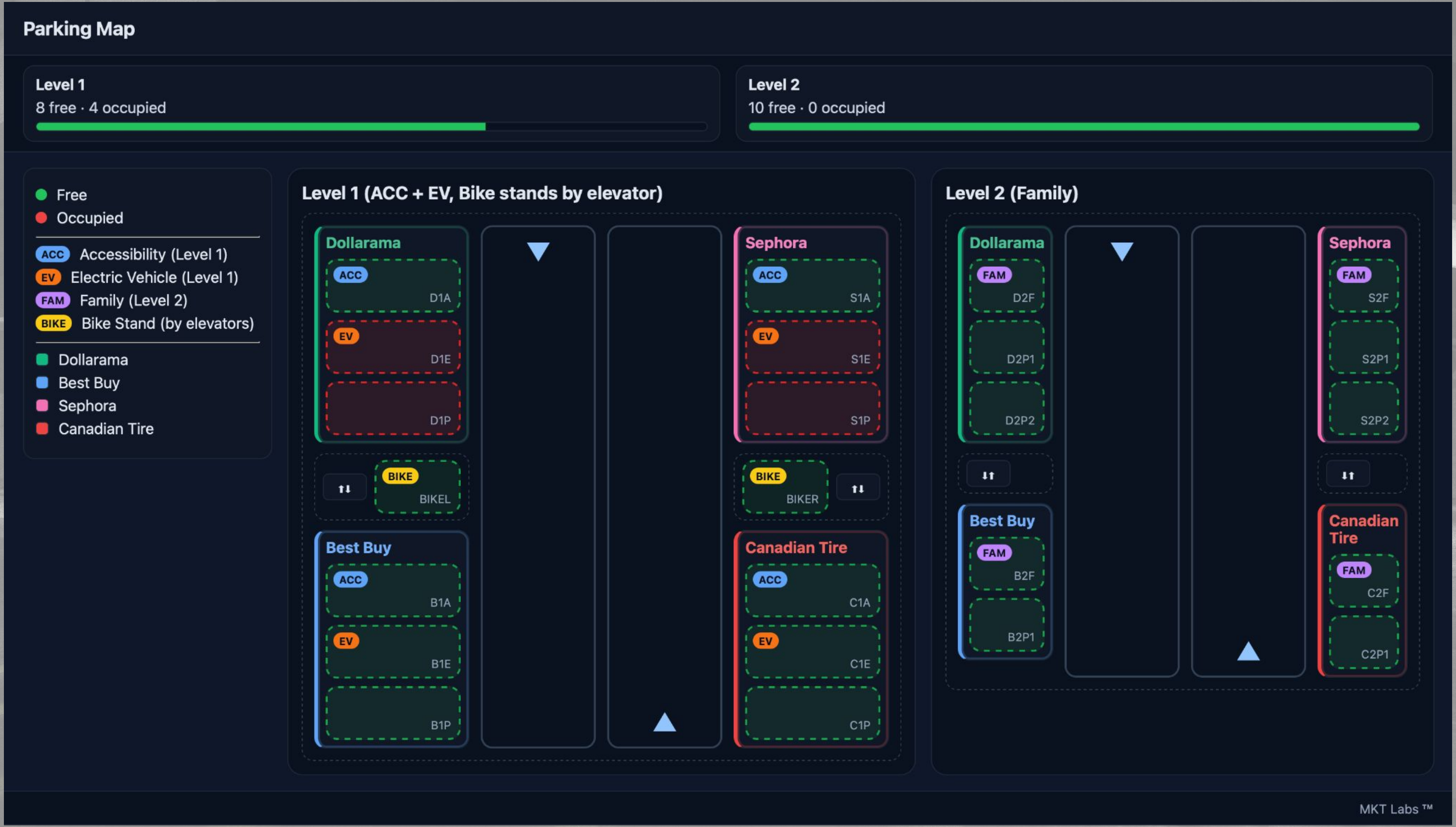
Level 1 (ACC + EV, Bike stands by elevator)

Level 2 (Family)

MKT Labs ™

UI Overview (With Vehicles)

Python App



Parking Map

Level 1
8 free · 4 occupied

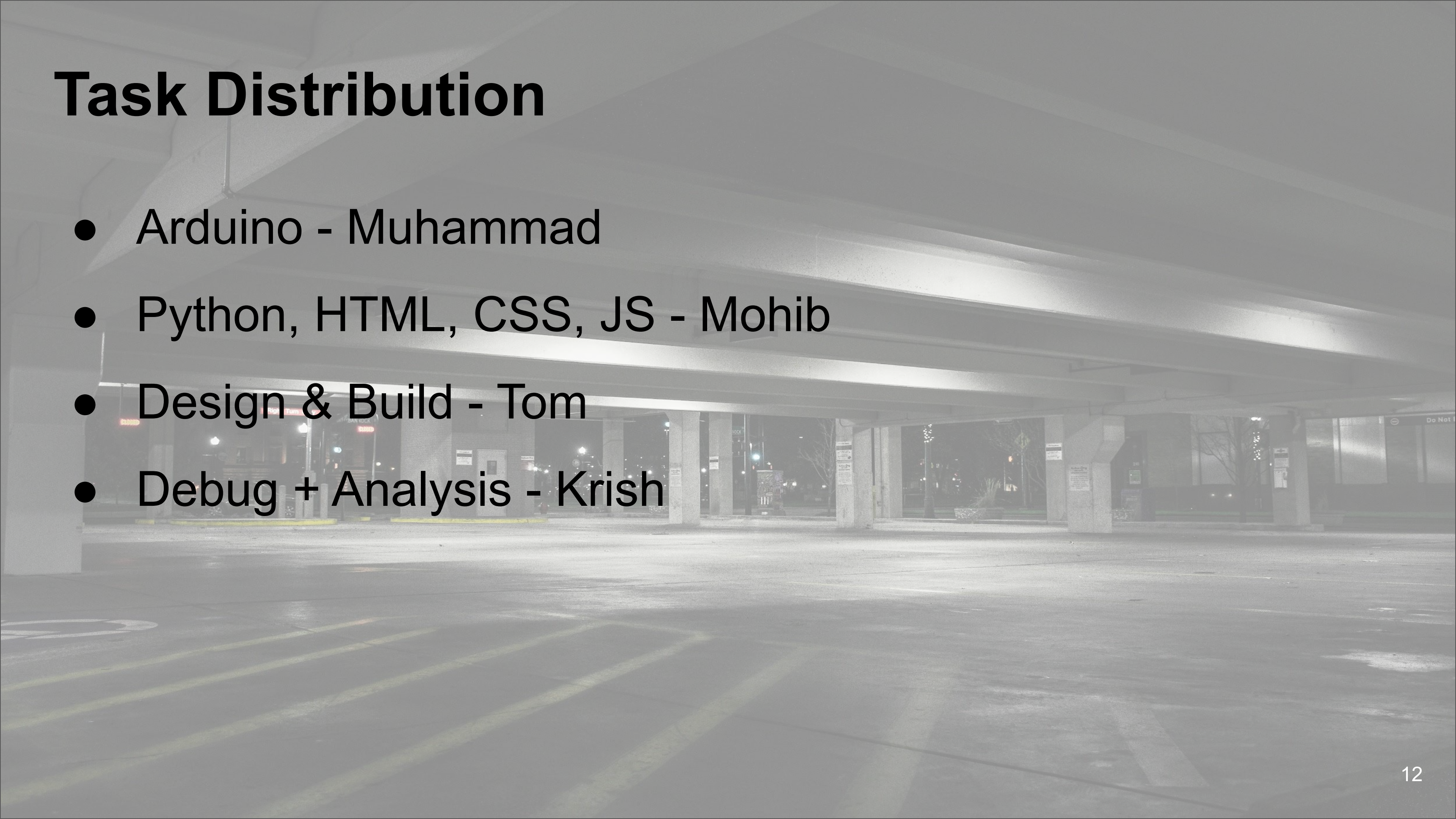
Level 2
10 free · 0 occupied

- ● Free
- ● Occupied
- ACC Accessibility (Level 1)
- EV Electric Vehicle (Level 1)
- FAM Family (Level 2)
- BIKE Bike Stand (by elevators)
- 🟩 Dollarama
- 🟦 Best Buy
- 🟪 Sephora
- 🟥 Canadian Tire

Level 1 (ACC + EV, Bike stands by elevator)

Level 2 (Family)

MKT Labs ™

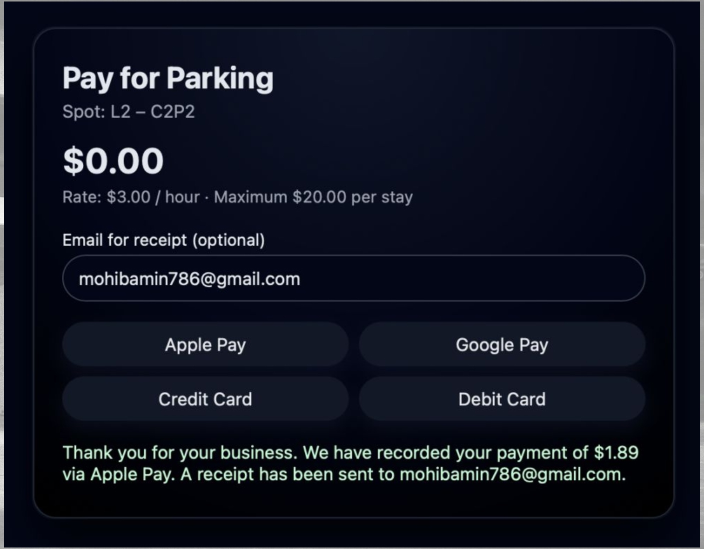


Task Distribution

- Arduino - Muhammad
- Python, HTML, CSS, JS - Mohib
- Design & Build - Tom
- Debug + Analysis - Krish

Thoughts & Future Improvements

- Payment Processing Plan
- 2 way transmission
  - Currently using 1 way
- Solar Panel Integration on Slanted Roof
  - Battery Backup incase of Power Failure



Pay for Parking

Spot: L2 – C2P2

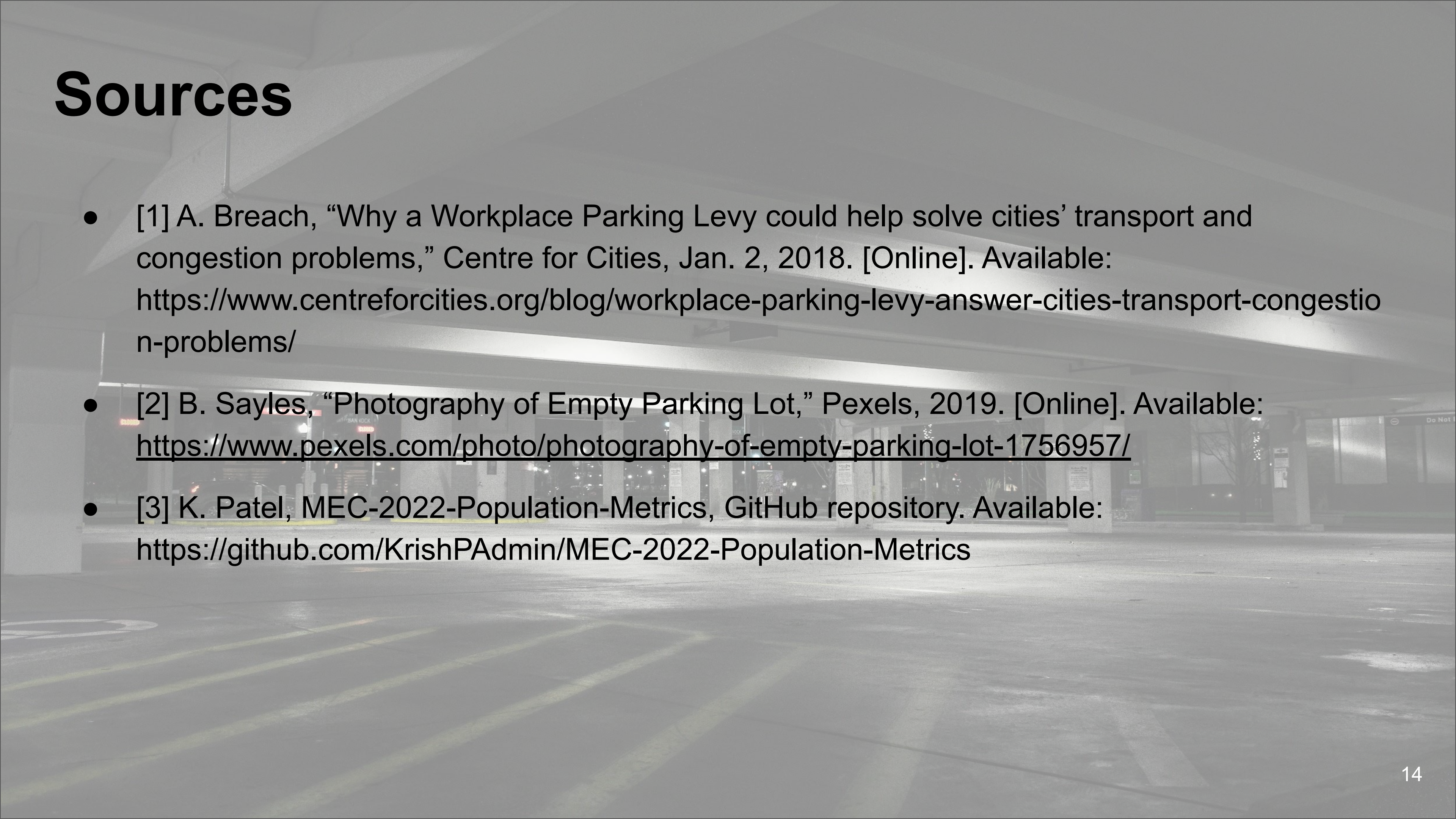
$0.00

Rate: $3.00 / hour · Maximum $20.00 per stay

Email for receipt (optional)

mohibamin786@gmail.com

Thank you for your business. We have recorded your payment of $1.89 via Apple Pay. A receipt has been sent to mohibamin786@gmail.com.



Sources

- [1] A. Breach, “Why a Workplace Parking Levy could help solve cities’ transport and congestion problems,” Centre for Cities, Jan. 2, 2018. [Online]. Available: https://www.centreforcities.org/blog/workplace-parking-levy-answer-cities-transport-congestion-problems/
- [2] B. Sayles, “Photography of Empty Parking Lot,” Pexels, 2019. [Online]. Available: https://www.pexels.com/photo/photography-of-empty-parking-lot-1756957/
- [3] K. Patel, MEC-2022-Population-Metrics, GitHub repository. Available: https://github.com/KrishPAdmin/MEC-2022-Population-Metrics