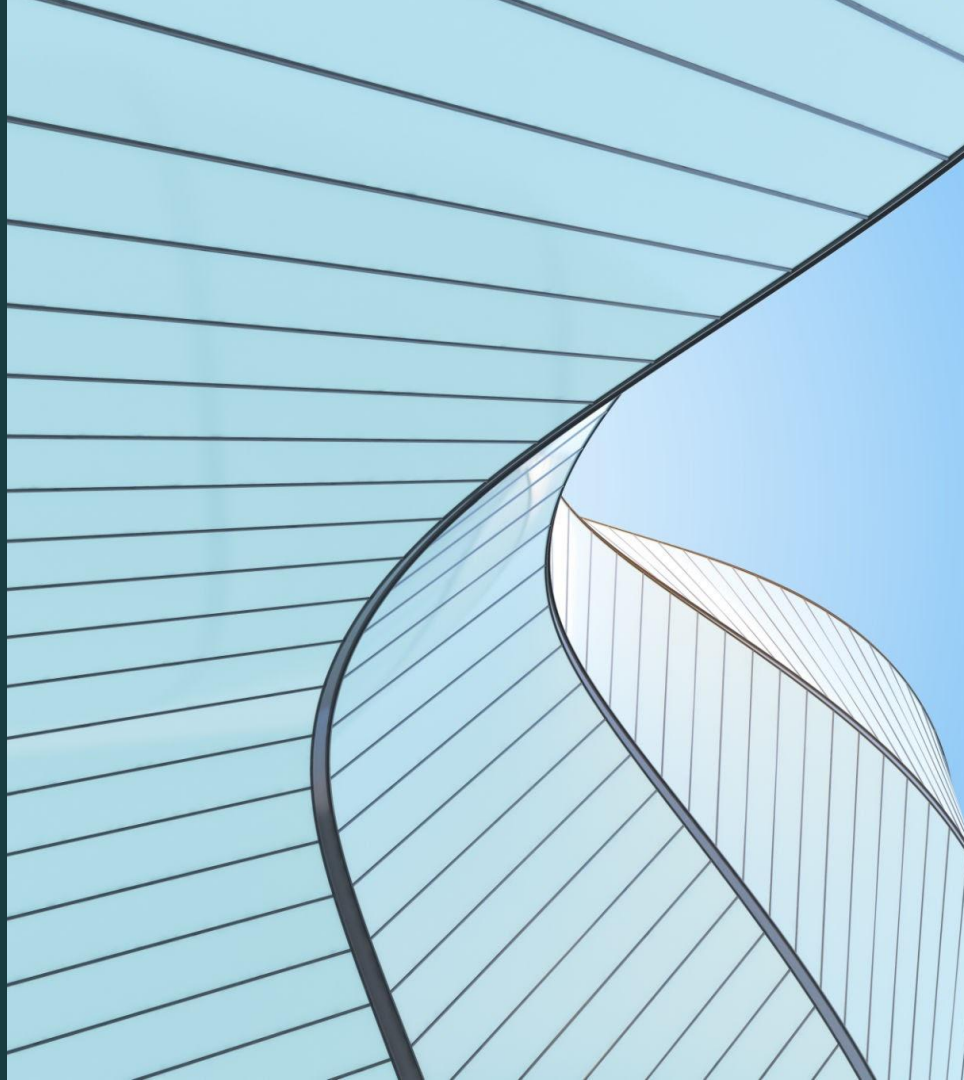
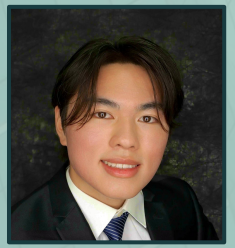
WINLAB SUMMER INTERNSHIP

CityOS

Aryan Upadhyay, Rohan Sada, Steven Huang
Under guidance of Dr. Jorge Ortiz



OUR TEAM



Steven Huang

RIsing Junior at Rutgers University-New Brunswick pursuing a B.S in Electrical Engineering and a minor in Computer Science.

I am also a 2D visual artist and graphic designer in my spare time.

Aryan Upadhyay

I'm a rising sophomore at Rutgers University, double majoring in both Computer Science and Data Science.

Rohan Sada

I am pursuing my masters degree in Electrical and Computer Engineering at Rutgers University, specializing in Machine Learning.

Anjali Kapilavai

I am a rising senior at Edison Academy Magnet School studying Electrical Engineering and Computer Science. I wish to pursue a career in the fields of electrical engineering and machine learning.

What is *City*OS?

- Infrastructure for smart cities to incorporate built-in privacy for urban data collection.

- Uses privacy loss accounting, putting users' devices in control of their own privacy

When urban data is collected – whether from traffic sensors, public transit, or even personal devices – how do we ensure it's not misused or exposed?

The *Three* API's

- The first part has already been completed, real-time data usage without compromising user privacy.
- App that helps disabled individuals cross streets etc.
- The last 2 are still in progress of being developed.

API #1 On-Scene

Real-Time data which cannot be tracked at all.

API #2 Single-Locality Aggregation

Differential Privacy localized with continuous statistics

API #3 Cross-Locality Aggregation

Differential Privacy which is applied citywide and is mediated by user devices

API #2 Single-Locality Aggregation
Differential Privacy localized with continuous statistics

The Focus

What We Focused on

- We are laying the groundwork for API 2
- Studying a single locality (WINLAB Parking Lot)
- These are all examples of continuous statistics—they are measurements on a continuous numerical scale (time, duration)

The Parking Application - CityOS

Detection

Prediction

User Interface

Why is it important?

Popularity based Meters/Tickets

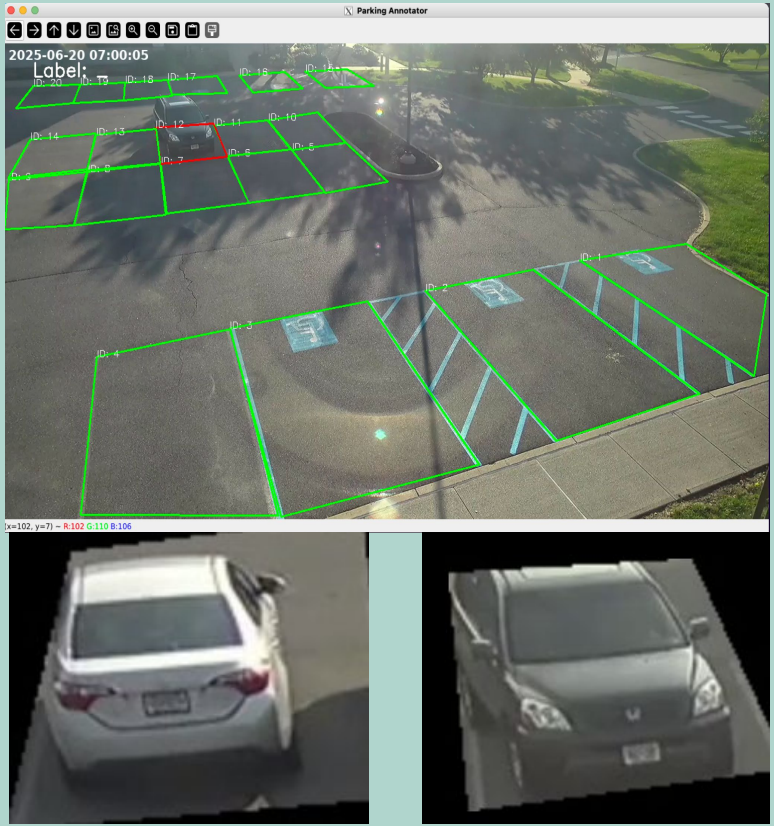
Predict what spots could be open when you arrive

Reduce congestion

Detecting Parking Spot Occupancy

Teaching the system about parking spots

- We manually cropped out each parking spot from the camera view.
- Then we trained a custom YOLO (You Only Look Once) classifier on these images
- Each image was labelled as free (0) or occupied (1)
- The model learns to classify any spot as free or occupied



Collecting Occupancy Data

Turning Camera Footage into Usable Data

- For each parking spot we record its status over time
- Whenever we analyze a video frame, we log the results as:
  - SpotID
  - Date and Time
  - Status: Free or Occupied (0 or 1)
- All this is saved in a CSV file (like an Excel Sheet)

SpotID	Day	Month	Date	Year	Hour	Minute	Second	Status
2	Tuesday	7	15	2025	07	23	49	0
4	Monday	7	14	2025	06	22	24	1

Predicting Parking Spot Occupancy

- Random forest has proven to be the best model through experimentation and past literature.
- Imagine this camera has been watching the parking lot for months, taking notes
- Random forest builds a bunch of these trees and tells us its confidence in a result.
    - Like having 200 parking experts at once instead of 1

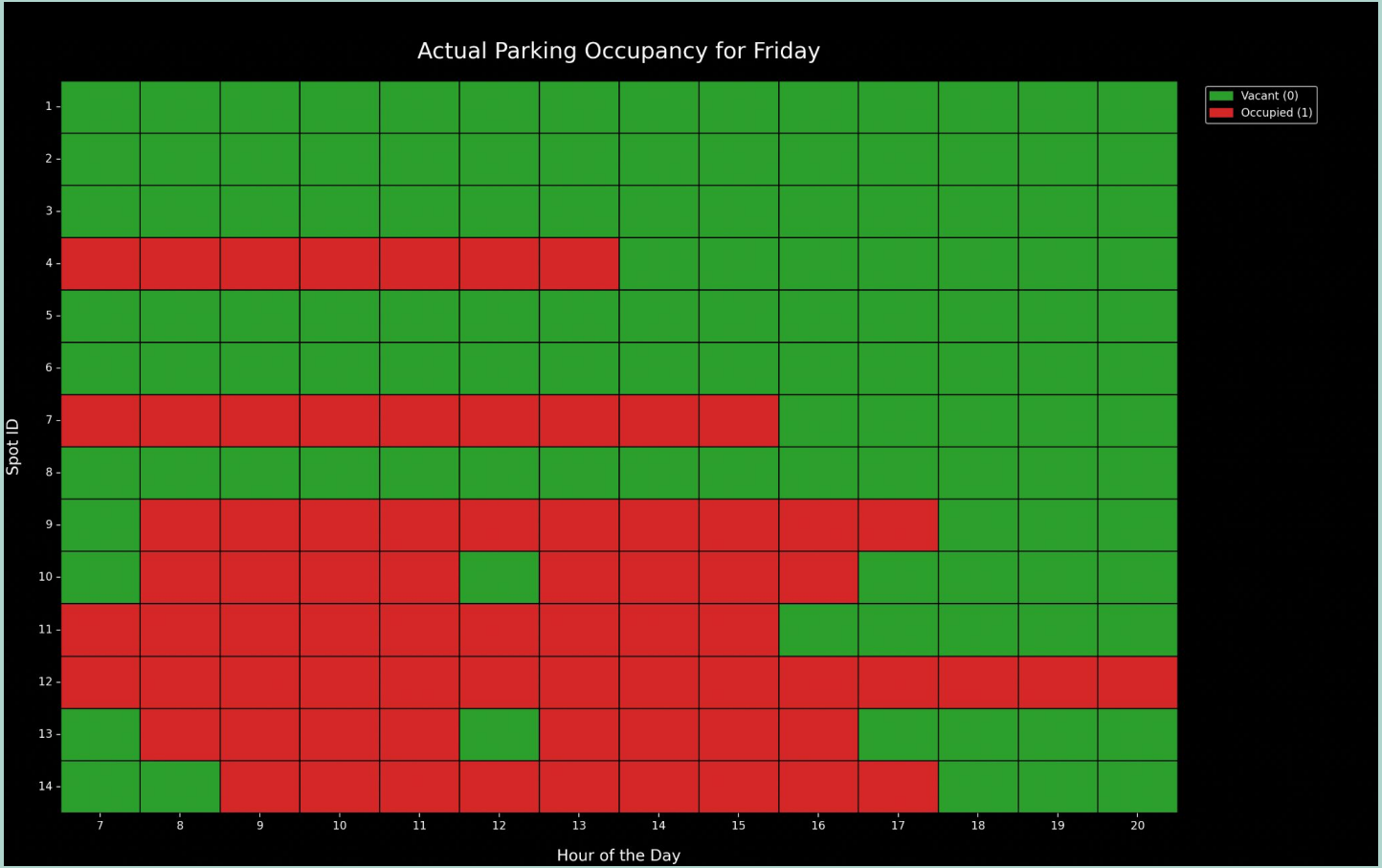
Spot 2 on mondays at 5PM?

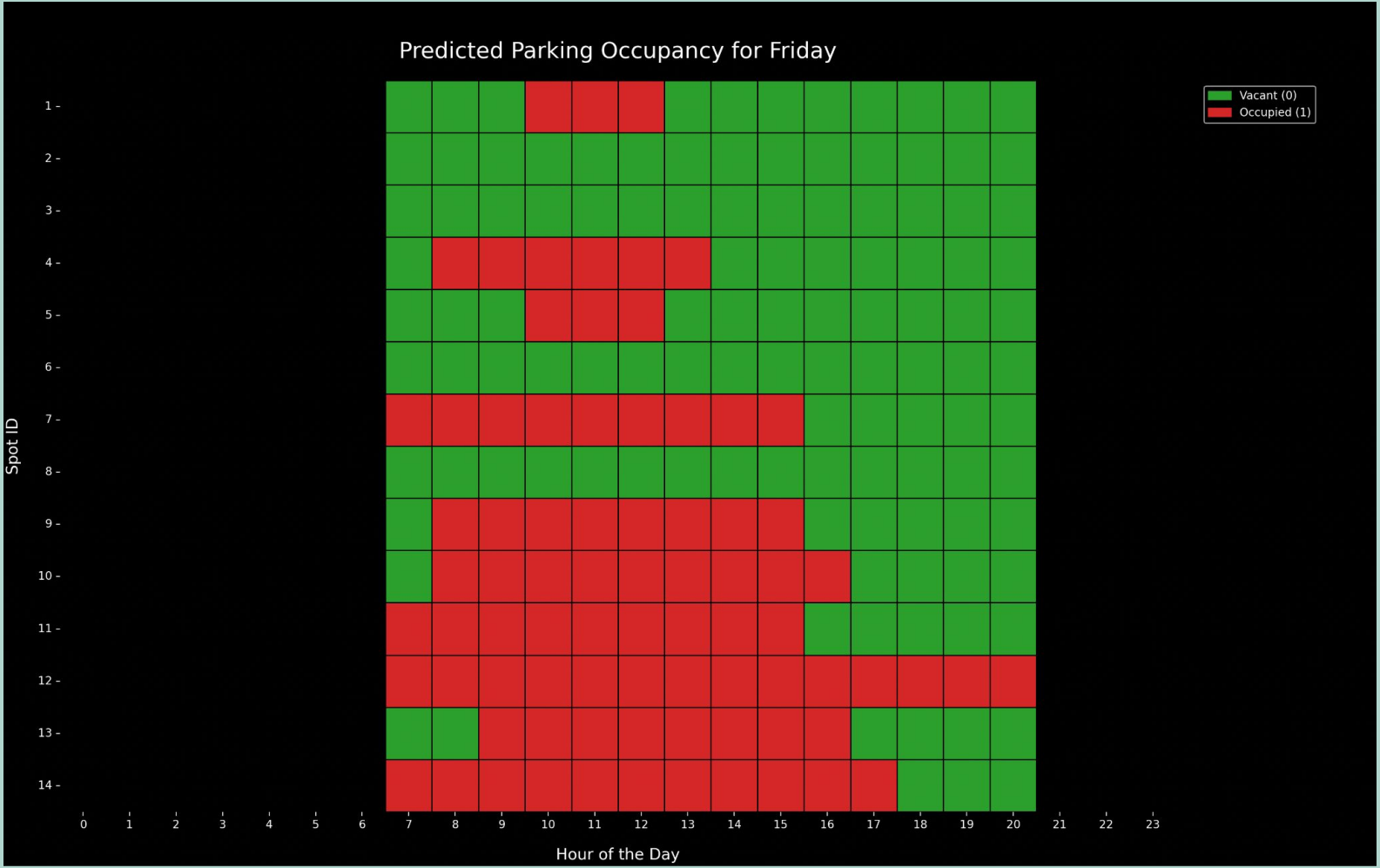
Tree 1:
"Monday + 2 PM + Spot 5 = Usually free because it's after lunch rush"

Tree 2:
"Spot 5 + Monday + afternoon = Sometimes occupied, sometimes free"

Tree 3:
"2 PM + Monday + Spot 5 = Free because most people are at work"

Final result: "Spot 5 will likely be FREE with 80% confidence"


Actual Parking Occupancy for Friday
Vacant (0)
Occupied (1)
Spot ID
1
2
3
4
5
6
7
8
9
10
11
12
13
14
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Hour of the Day


Predicted Parking Occupancy for Friday
Vacant (0)
Occupied (1)
Spot ID
1
2
3
4
5
6
7
8
9
10
11
12
13
14
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Hour of the Day

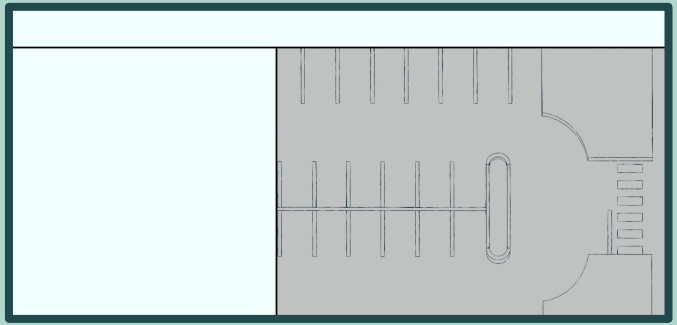
The UI Development Process

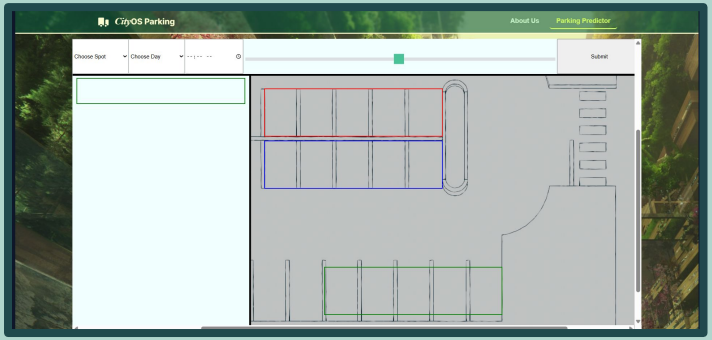
Divided into 3 main pages

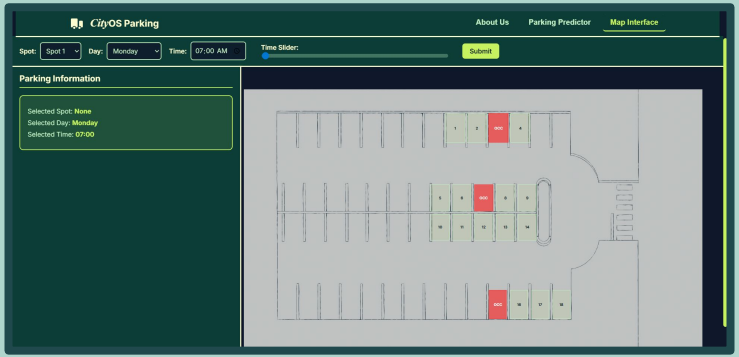
- The About Us Page
  - Has the project documentation and weekly progress

- Single Parking Predictor
  - First UI developed to connect with prediction model
  - Checks one spot at a time

- Map Interface
  - Checks all the spots and returns a map of all occupancies
    - Pictures of iterations → → →







The Full UI Demo