

EzParking

Real Time Parking

Recommendation System

Explore Next Generation No Hassle Parking Experience

>>>>

.....



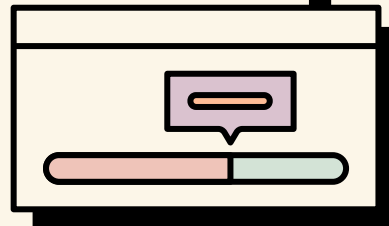


>>>>

About EzParking

- A software dedicate to provide user the optimized parking lot selection
- Provide service to all age group and background

Meet us





Meet the team Apollo



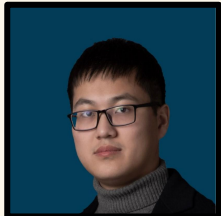
Ziwen Tan

Backend Developer



Zhuo (Kyle) Chen

Frontend Developer



Yilin (Kevin) Ren

Frontend Developer &
Scrum Manager



Kecheng Yu

AI Developer &
Presentation Hoster



Why?



**Vehicle is part of
our life**



**Hard to find a parking
spot when occur in a
heavy traffic time**



**Even find a parking
spot but far away
from the destination**



**No indoor
navigation**



What sets us apart?



Simplicity

In just three simple steps, this project fulfills the users' requirements



Flexibility

The waypoint map can be customized by administrators as per their requirements



Reliability

The system offers up to five recommended alternative parking lots that are optimal

Project commercial



01

.....

>>>>

~~~~~



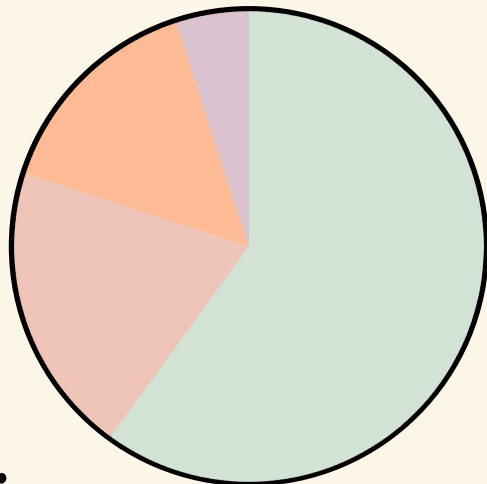
# Technology Use





# Tech Stack

>>>>>



~~~~~  
.....

60%

Python

AI Detector

20%

Java

Backend services

15%

JavaScript

Frontend, React and Redux

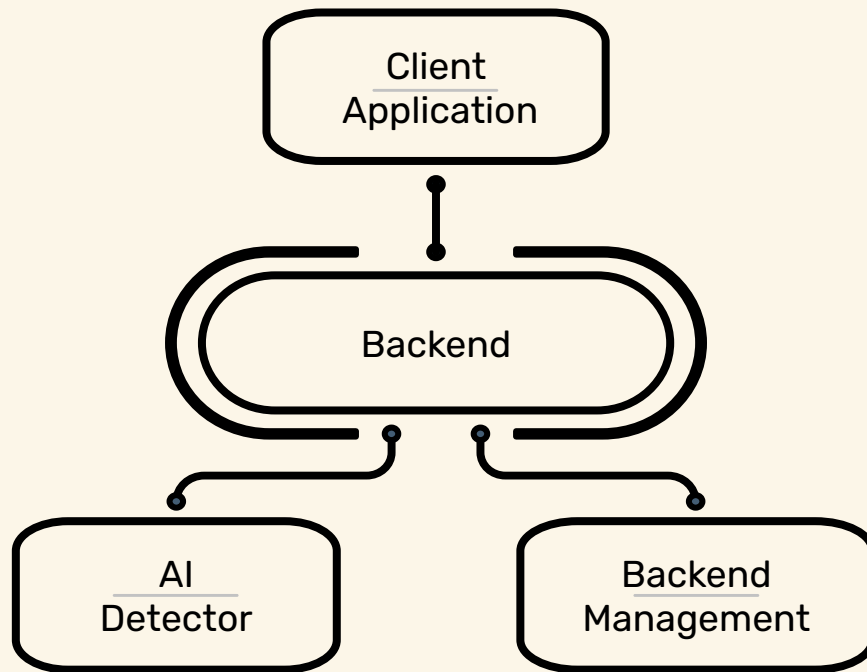
5%

Other

HTML, TypeScript, Shell, etc.

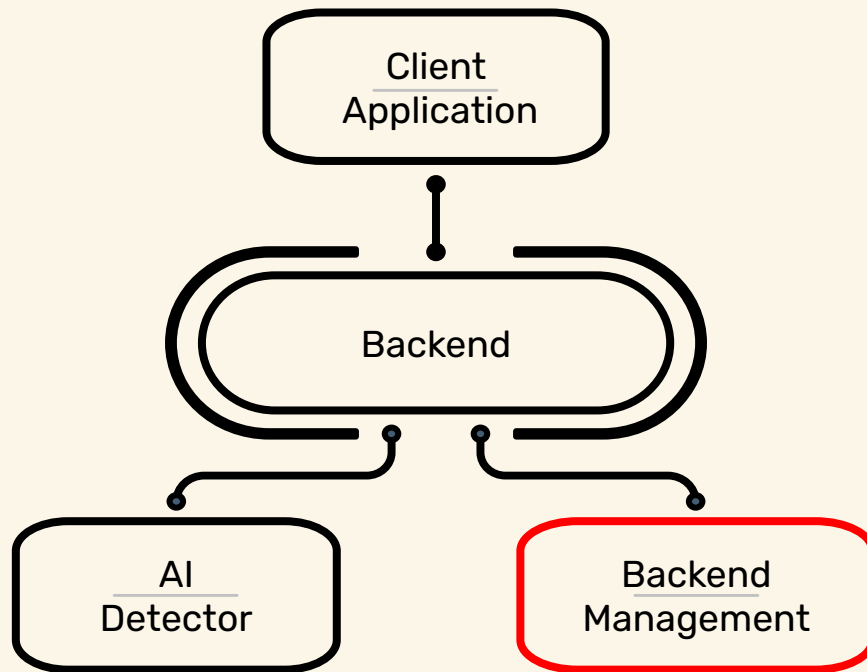


System Workflow





System Workflow



Waypoint Graph



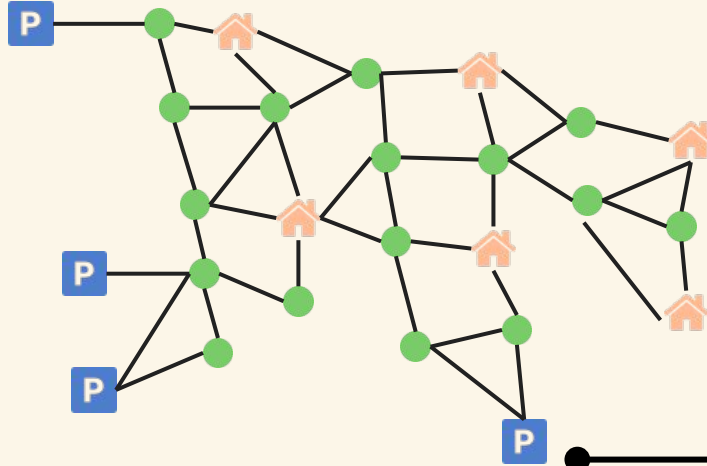
Destination



Parking Lot



Mid Way point



Destination

Name: Gym

Latitude: 50.4155

Longitude: 104.5878

Parking lot

Name: Lot 10

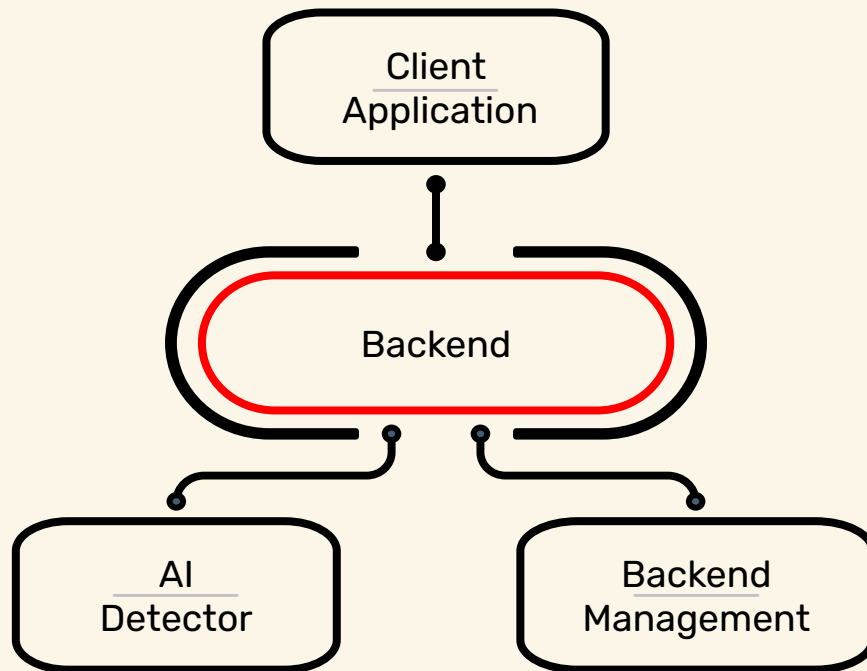
Available Spot: 55

Latitude: 50.2113

Longitude: 104.1266



System Workflow



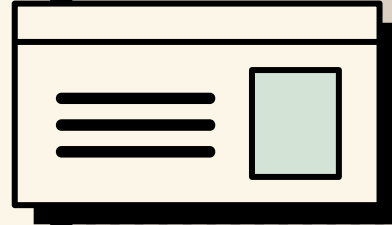
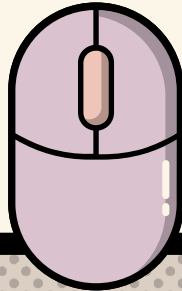
Dijkstra's algorithm

- Why Ezparking use dijkstra's algorithm?
- What problem does the dijkstra's algorithm help

EzParking solve?

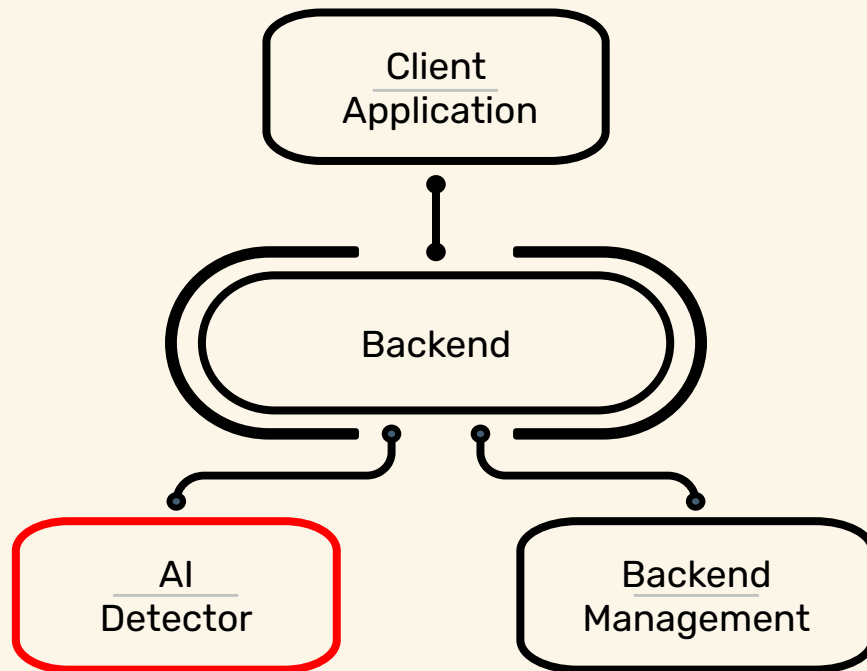
.....

>>>>>





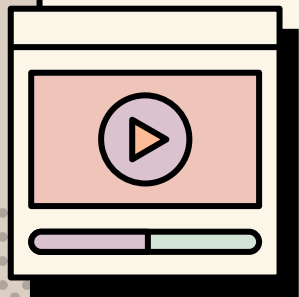
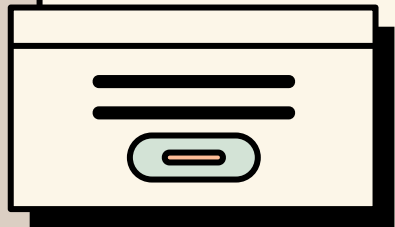
System Workflow





AI Vehicle Detection

>>>>



– Camera Detect

Easy to place and more versatile than sensors



– YOLO v5

The best choice of both performance and requirements

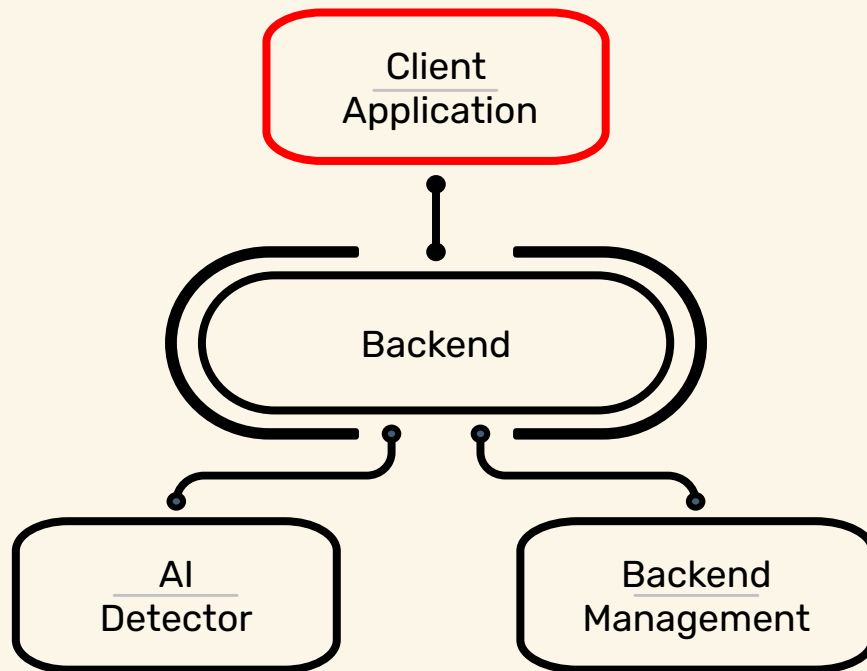


– Multi-process APIs Call

No pausing issue anymore, won't miss any frame in the video



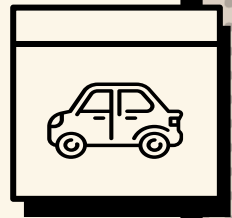
System Workflow





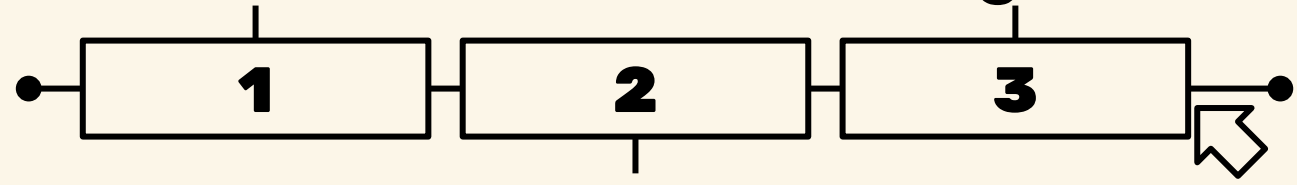
.....

Client Application



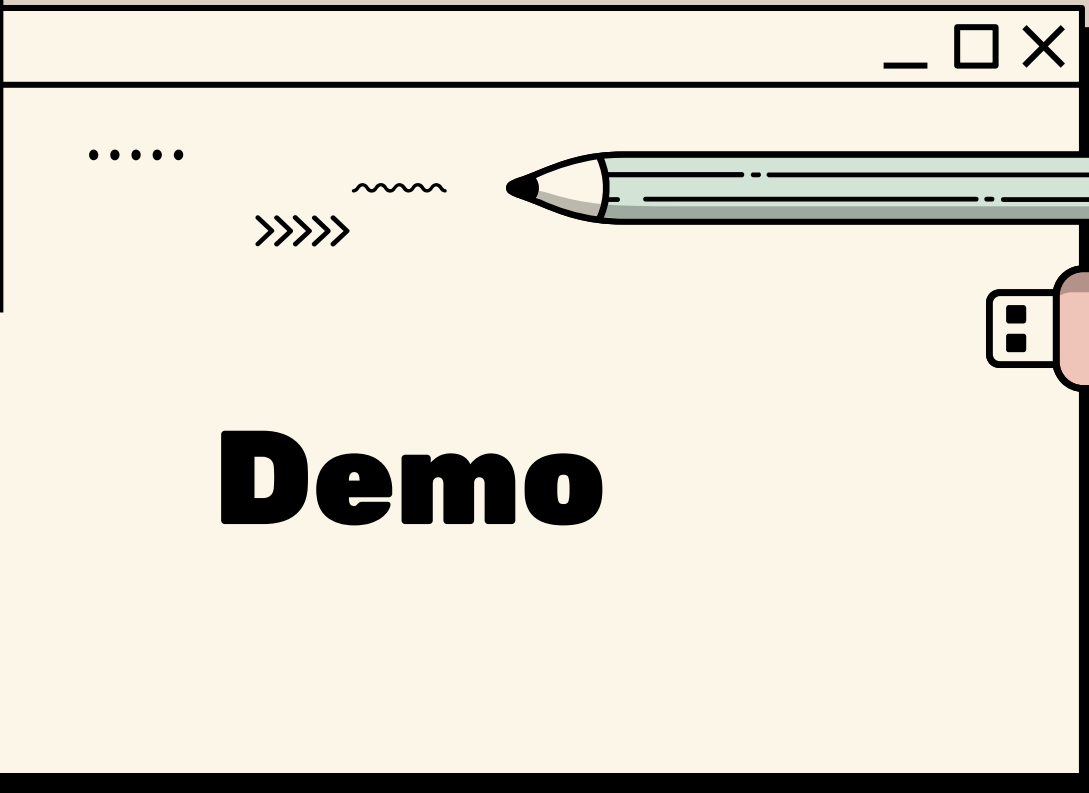
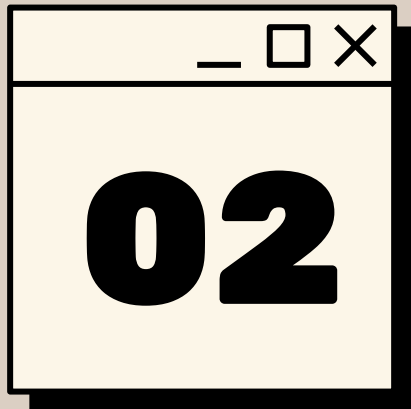
**Select
destination**

**Start
navigation**



**Select
parking lot**

~~~~~  
>>>>>



# AI Demo



The screenshot displays an AI demo interface. On the left, a 3D simulation shows a parking lot with several cars. The cars are labeled with IDs: 10-100, 10-124, 10-113, 10-126, 10-135, and 10-130. A yellow line indicates a path or boundary. On the right, a table titled "EzParking" displays parking data. The table has two columns, "Left" and "Right", and five rows of data. The bottom of the interface shows a Windows taskbar with the time 2:00.

| EzParking |     |
|-----------|-----|
| 17        | 5   |
| 2         | 40  |
| 15        | 38  |
| 3         | 18  |
| 14        | 138 |

03

.....  
>>>>>  


# Never Stop Exploring





# Future

## Implemente advanced analytics

- More personalized recommendations



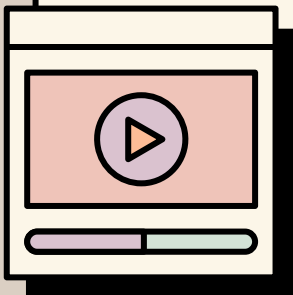
## Develop additional features

- Additional features



## Enhance algorithm and server performance

- Improving the performance



# Acknowledgement

We would like to thank the following people for their involvement with our project:

- Project Advisor: Dr. Timothy Maciag
- Project Mentor: Dr. Christine Chan

.....

>>>>>





**Thanks!**



# Question

**CREDITS:** This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**