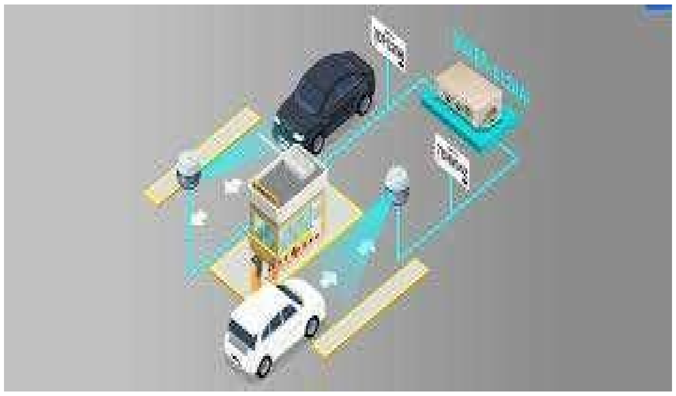
AI ENABLE CAR PARKING USING OPENCV

NAAN MUDHALVAN



COLLEGE NAME:THIRUMALAI COLLEGE OF ENGINEERING AND TECHNOLOGY

INTRODUCTION:

To finding parking availability for a specific time period is a very tedious job in urban areas. The Indian government now focusing on the smart city project, already they published city name for an upcoming smart city project. In smart city application , intelligent transportation system (ITS) plays an important role- in that finding parking place, specifically for the car owner to avoid time computation, as well as congestion in traffic is going to be very important. In this article, we propose an intelligent car parking system for the smart city using Circle Hough Transform (CHT). Keywords- Intelligent transportation system (ITS), Circle Hough Transform (CHT), Circle detection, Video-Image processing, smart city, parking system , OpenCV. For today's traffic monitoring and its management is a recent trend in research development. In this paper, we are focusing on parking component of the traffic parameter. Traffic very congested from last decade due to the increasing rise of automobile companies offers to a customer, privatization of that- mainly more and more used in present day compared to last decade and it's also increasing in the future may be with same or more speed. So now government thinking how to solve this problem in real time? Within specified time duration.

1.1 PROJECT OVERVIEW

S.NO	TOPICS
1	INTRODUCTION
2	IDEATION & PROPOSED SOLUTION
3	REQUIREMENTS ANAYSIS
4	PROJECT DESIGN
5	CODING AND SOLUTIONING
6	RESULTS
7	ADVANTAGES & DISADVANTAGES
8	CONCLUSION
9	FUTURE SCOPE
10	APPENDIX

1.2 PURPOSE:

It allows car park operators and companies to track their facilities, vehicle entry, and real-time reporting of the availability of parking spots. This helps companies manage their parks in a central digital hub offered with parking software.

1. Superior Technology

Parking management systems are known for their integration with technology. Most of these systems are based on improved models and technological innovations, due to which they are suited to be used in various car parks.

2. Better parking experience

Better car park management means happier customers. A parking management system enhances the customer journey by providing them with a unified procedure.

3. Increased Protection

Parking management systems have technologically advanced security features that enable you to prevent parking misuse and suspicious activity in your parking facility

4. Reduced traffic and pollution

Vehicles that keep circling an area in search of an empty parking space cause most of the city traffic. Moreover, significantly driving around or waiting for a parking space to be vacant burns through a lot of fuel and releases emissions daily.

5. Easy implementation and management

Another of the benefit of a parking management system is that it can efficiently be designed and implemented. These systems have a well-organised structure

6. Cost-effective

Another advantage that you obtain from installing a smart parking management system is the cost. It runs on a low workforce, so you can save money and time

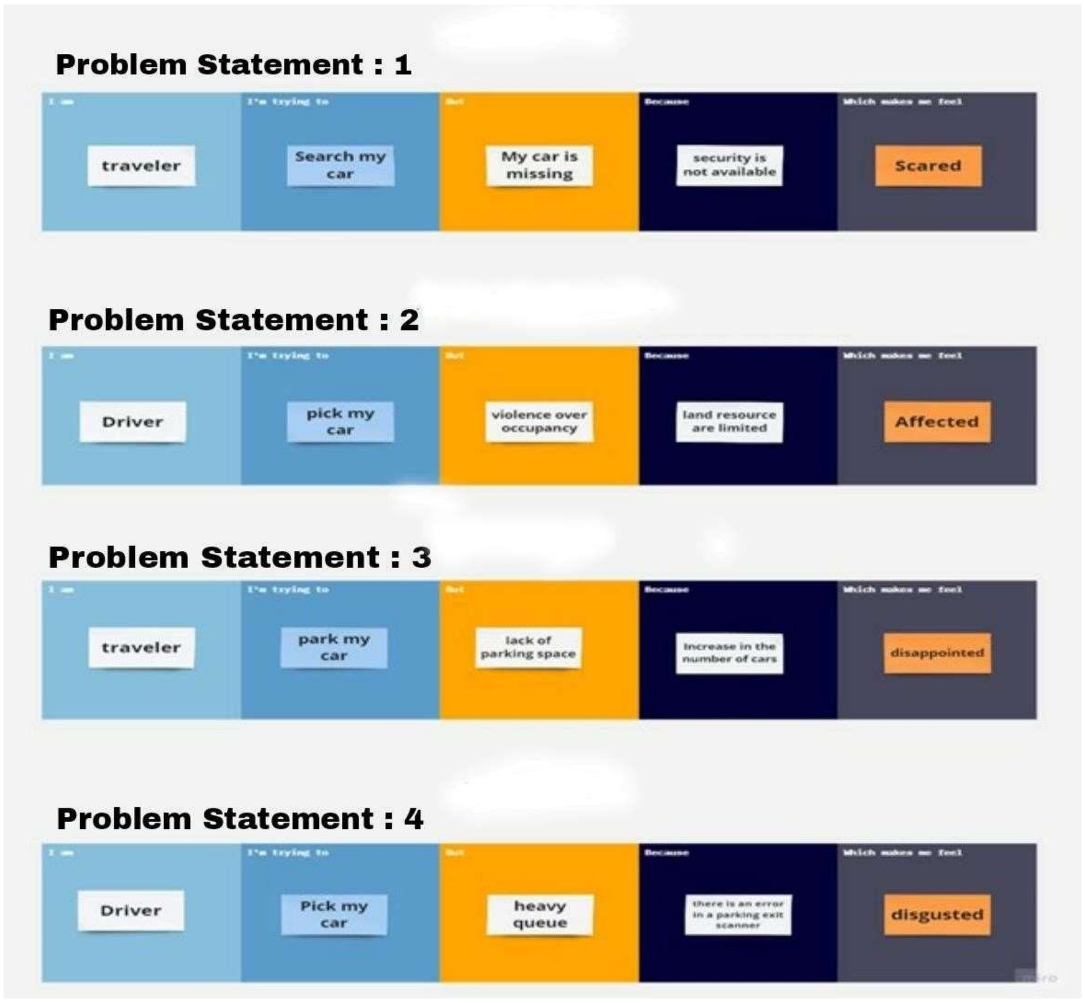
7. Uses integrated software and applications

Parking management solutions use software and applications that can be combined with another. Depending on your car park’s requirements, there are lots of customisations available.

2.IDEATION & PROPOSED SOLUTION:

2.1 PROBLEM STATEMENT DEFINITION:

Problem Statement – AI Enable Car Parking Using Open CV:
By using ultrasonic sensors be able to keep a record of the number of cars parked inside of a parking garage. Consequently, once a car enters a parking garage followed by a parking space, a ping ultrasonic sensor will then be able to determine if a car is parked in the space or not.



Problem Statement : 1

I am	I'm trying to	But	Because	Which makes me feel
traveler	Search my car	My car is missing	security is not available	Scared

Problem Statement : 2

I am	I'm trying to	But	Because	Which makes me feel
Driver	pick my car	violence over occupancy	land resource are limited	Affected

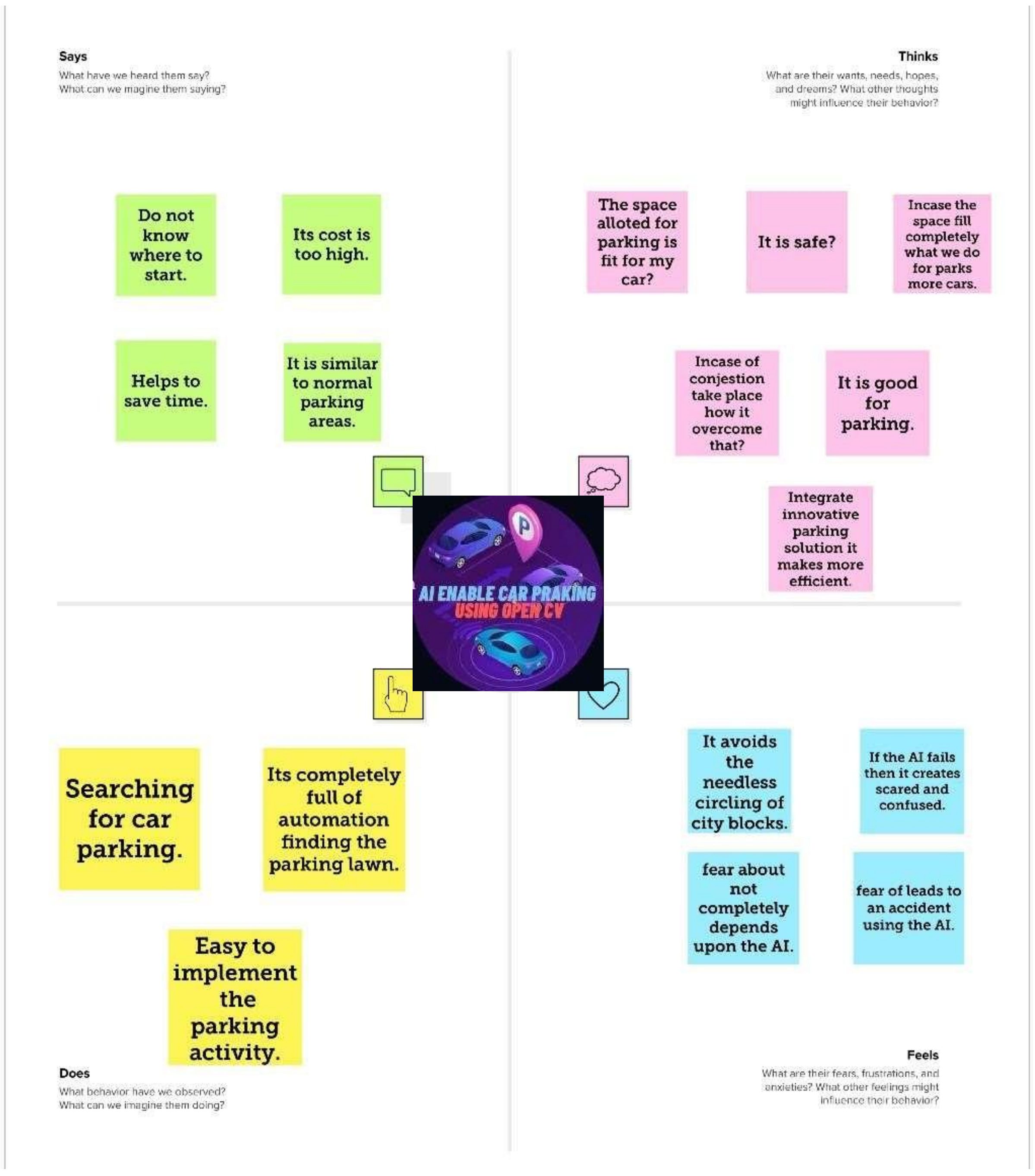
Problem Statement : 3

I am	I'm trying to	But	Because	Which makes me feel
traveler	park my car	lack of parking space	Increase in the number of cars	disappointed

Problem Statement : 4

I am	I'm trying to	But	Because	Which makes me feel
Driver	Pick my car	heavy queue	there is an error in a parking exit scanner	disgusted

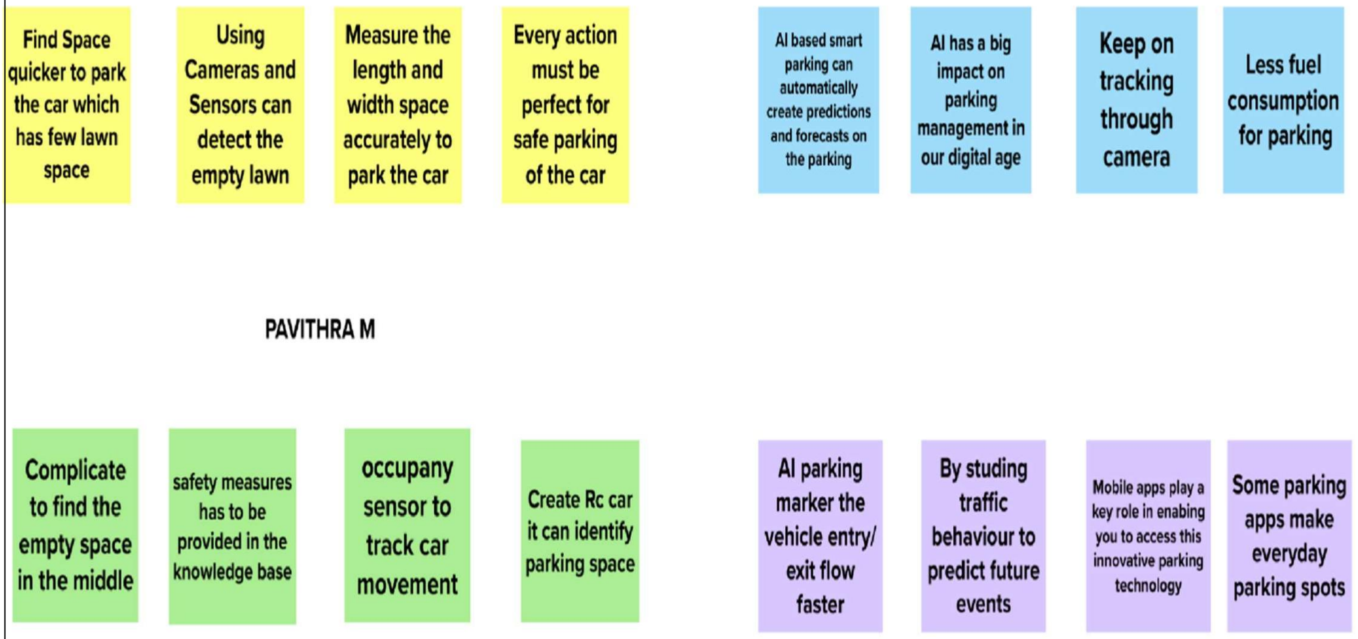
2.2 EMPATHY MAP CANVAS:


Says
What have we heard them say?
What can we imagine them saying?
Do not know where to start.
Its cost is too high.
Helps to save time.
It is similar to normal parking areas.
Thinks
What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?
The space alloted for parking is fit for my car?
It is safe?
Incase the space fill completely what we do for parks more cars.
Incase of conjestion take place how it overcome that?
It is good for parking.
Integrate innovative parking solution it makes more efficient.
AI ENABLE CAR PRAKING USING OPEN CV
Searching for car parking.
Its completely full of automation finding the parking lawn.
Easy to implement the parking activity.
Does
What behavior have we observed?
What can we imagine them doing?
It avoids the needless circling of city blocks.
If the AI fails then it creates scared and confused.
fear about not completely depends upon the AI.
fear of leads to an accident using the AI.
Feels
What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

2.3 IDEATION AND BRAINSTORMING:

Brainstorm & Idea Prioritization Template: AI Enable Car Parking Using Open CV

Step 1 : Brainstorm,Idea Listening and Grouping



Find Space quicker to park the car which has few lawn space

Using Cameras and Sensors can detect the empty lawn

Measure the length and width space accurately to park the car

Every action must be perfect for safe parking of the car

AI based smart parking can automatically create predictions and forecasts on the parking

AI has a big impact on parking management in our digital age

Keep on tracking through camera

Less fuel consumption for parking

PAVITHRA M

Complicate to find the empty space in the middle

safety measures has to be provided in the knowledge base

occupany sensor to track car movement

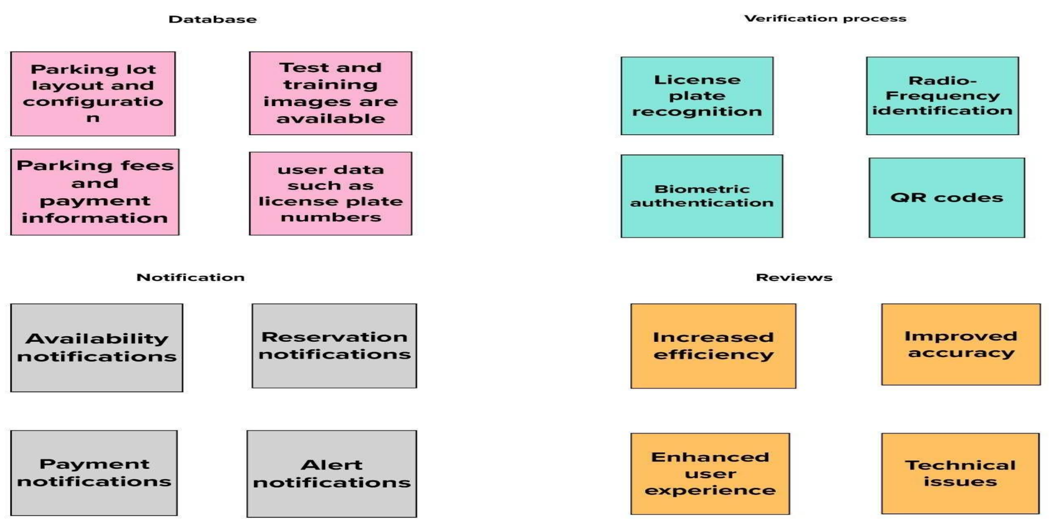
Create Rc car it can identify parking space

AI parking marker the vehicle entry/ exit flow faster

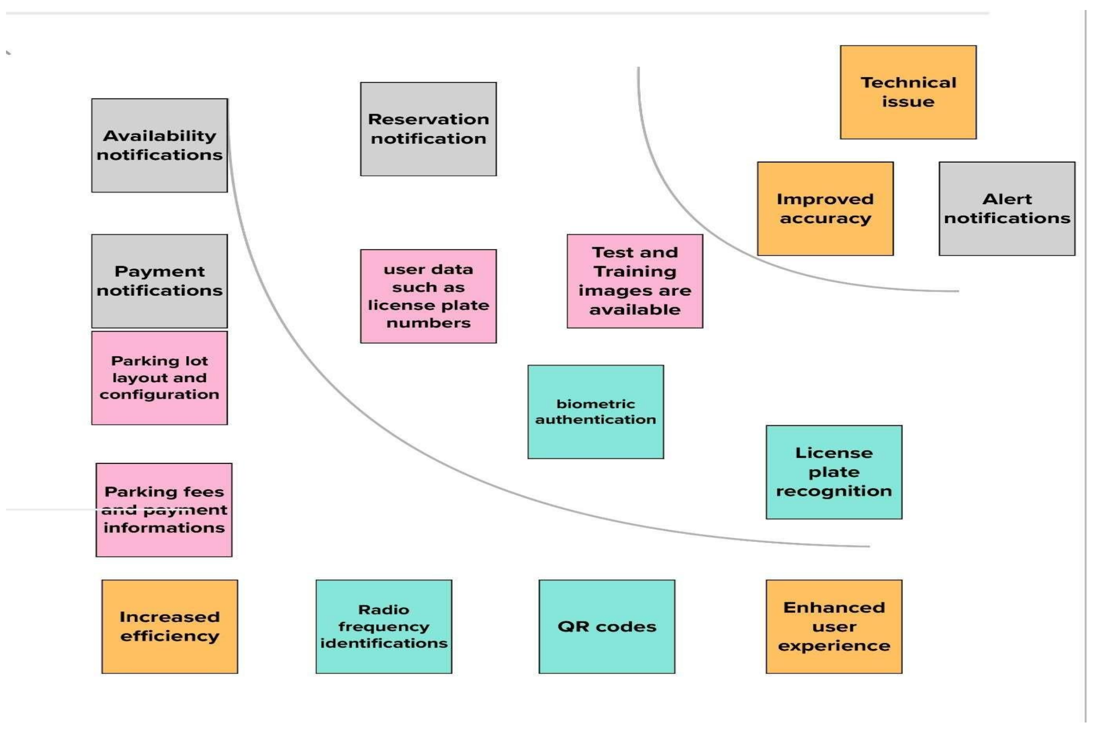
By studing traffic behaviour to predict future events

Mobile apps play a key role in enabing you to access this innovative parking technology

Some parking apps make everyday parking spots


Database
Parking lot layout and configuratio n
Test and training images are available
Parking fees and payment information
user data such as license plate numbers
Verification process
License plate recognition
Radio-Frequency identification
Biometric authentication
QR codes
Notification
Availability notifications
Reservation notifications
Payment notifications
Alert notifications
Reviews
Increased efficiency
Improved accuracy
Enhanced user experience
Technical issues

Step 2 : Idea Prioritization


Technical issue
Availability notifications
Reservation notification
Improved accuracy
Alert notifications
Payment notifications
user data such as license plate numbers
Test and Training images are available
Parking lot layout and configuration
biometric authentication
License plate recognition
Parking fees and payment informations
Increased efficiency
Radio frequency identifications
QR codes
Enhanced user experience

2.4 PROPOSED SYSTEM:

S. No	Parameter	Description
1.	Problem Statement (Problem to be solved)	To find the free parking slot in a minimum distance from a starting point.
2.	Idea / Solution description	The idea of this project to find the difference between empty slot and occupied slot and given numbers for each slot in ascending order to find minimum distance unoccupied slot.
3.	Novelty / Uniqueness	By above approach the parking slot is segregated as occupied and unoccupied slots
4.	Social Impact / Customer Satisfaction	This technique will reduce the time taken park their car and thereby improving the customer satisfaction.
5.	Business Model (Revenue Model)	The result of this project could be implemented in public places and they will be able to achieve the accuracy.
6.	Scalability of the Solution	The outcome of this project will be very helpful in parking management system.

3.REQUIREMENT ANALYSIS:

3.1 FUNCTIONAL REQUIREMENT:

Following are the functional requirements of the proposed solution.

FR NO.	Functional Requirement (Epic)	Sub requirement(story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through familiarize with the system Registration through mobile app
FR-2	User Confirmation	Confirmation via Email Confirmation via approval of parking pass
FR-3	Object Detection	The system should be able to detect the presence of a car in a parking spot.
FR-4	Parking monitoring	The system should be able to monitor the parked cars and detect any illegal activities, such as double parking or parking in a handicap spot.
FR-5	Real-time updates	The system should provide real-time updates on parking availability and other relevant information to drivers and parking lot staff.
FR-6	User-friendly interface	The system should have a user-friendly interface that is easy to use and understand, to ensure a smooth and hassle-free parking experience for drivers.

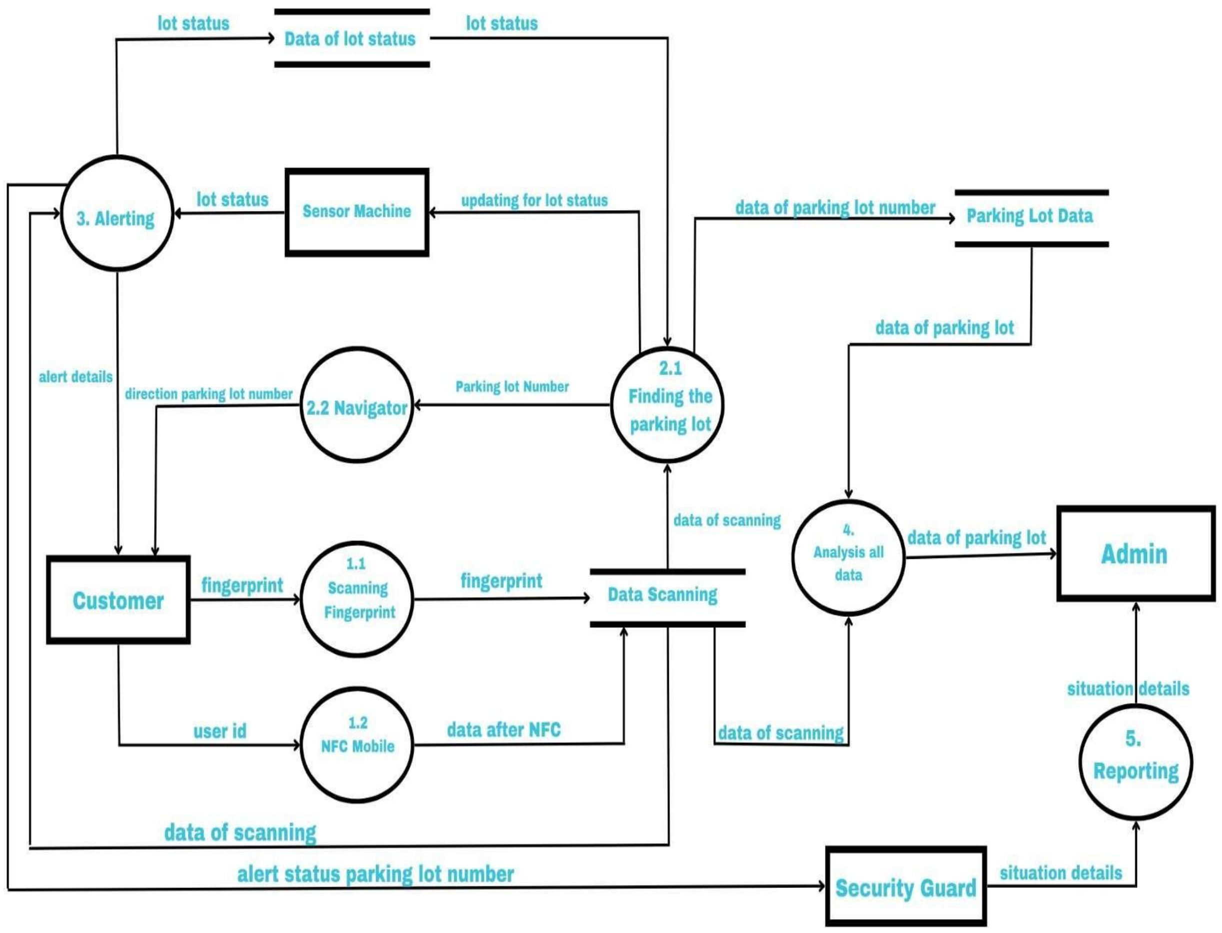
3.2 Non – Functional Requirements:

Following are the non-functional requirements of the proposed solution.

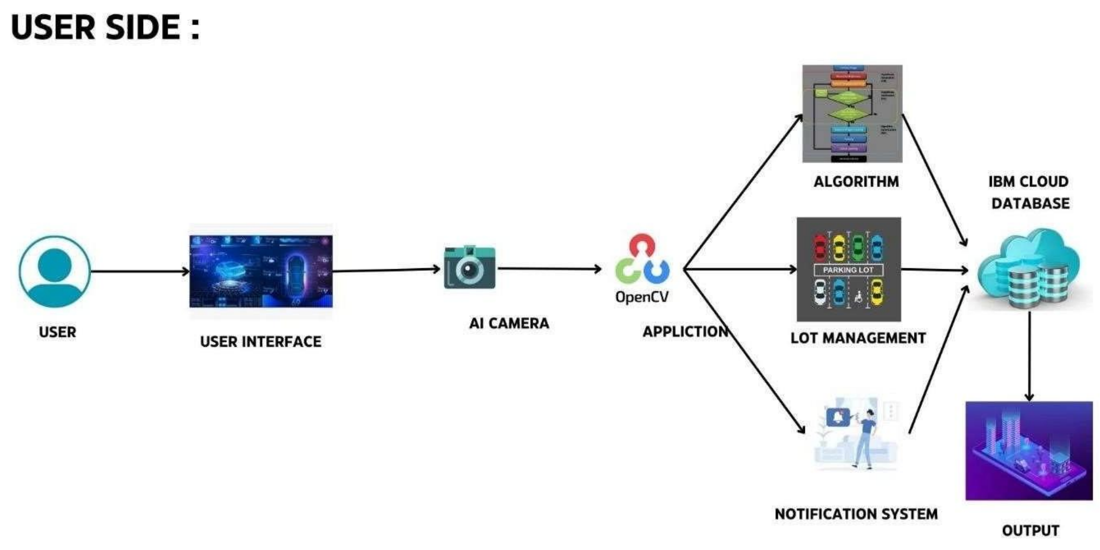
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system should be user-friendly and intuitive, with a simple and easy-to-use interface that is accessible to all users.
NFR-2	Security	The system should be designed with robust security features, to ensure the privacy and safety of drivers and their vehicles, and to prevent unauthorized access and data breaches.
NFR-3	Reliability	The system should be reliable and stable, with high availability and minimal downtime.
NFR-4	Performance	The system should be able to process data quickly and accurately, with minimal delay and high efficiency.
NFR-5	Availability	The system should be highly available, with minimal downtime and interruption to the parking service
NFR-6	Scalability	The system should be able to handle a large number of parking spots and users, and be easily scalable as the demand increases.

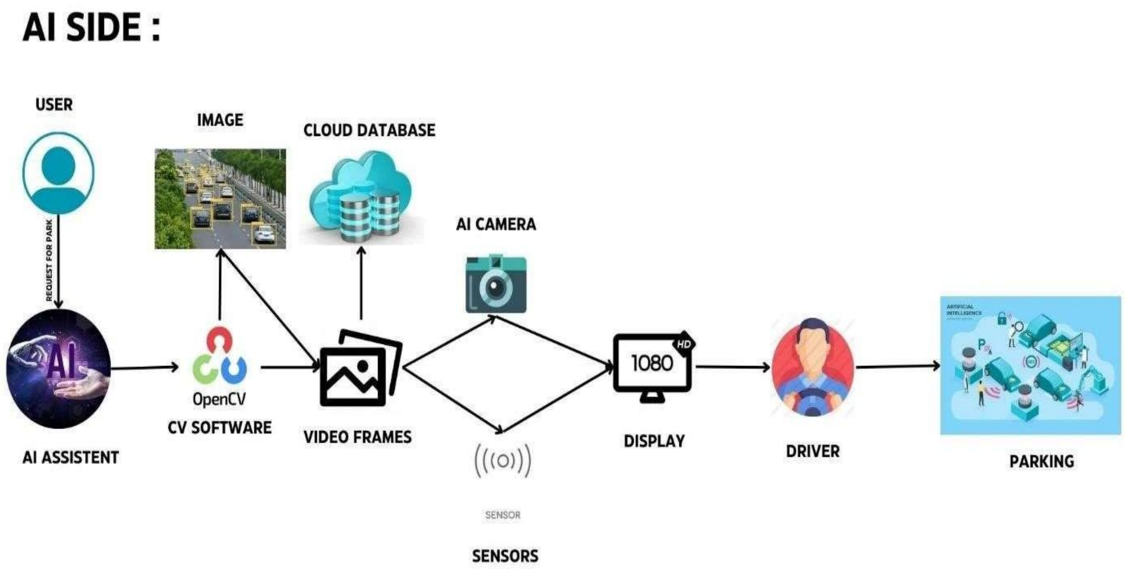
4.PROJECT DESIGN:

4.1 DATA FLOW DIAGRAM:



4.2 SOLUTION & TECHNICAL ARCHITECTURE:





TECHNOLOGICAL ARCHITECTURE:

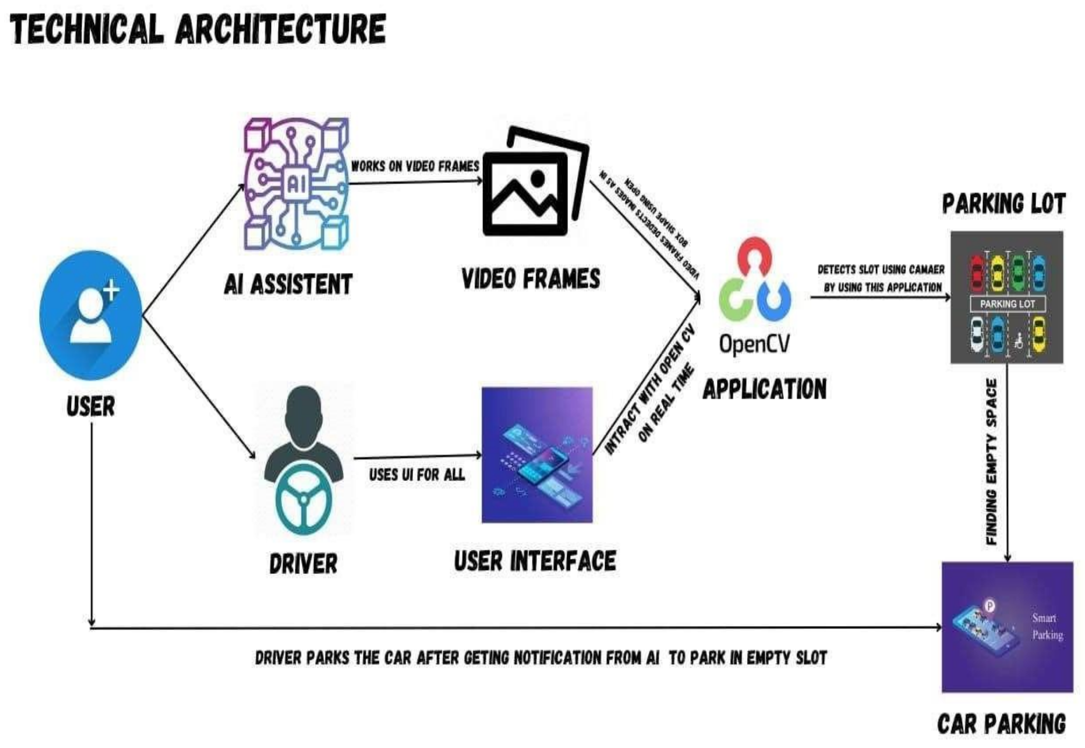

TECHNICAL ARCHITECTURE
USER
AI ASSISTENT
WORKS ON VIDEO FRAMES
VIDEO FRAMES
VIDEO FRAMES DEDECTS IMAGES AS IN BOX SHAPE USING OPEN
DRIVER
USES UI FOR ALL
USER INTERFACE
INTRACT WITH OPEN CV ON REAL TIME
OpenCV
APPLICATION
DETECTS SLOT USING CAMAER BY USING THIS APPLICATION
PARKING LOT
PARKING LOT
FINDING EMPTY SPACE
Smart Parking
CAR PARKING
DRIVER PARKS THE CAR AFTER GETING NOTIFICATION FROM AI TO PARK IN EMPTY SLOT

Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	User Interface is used by user in mobile application or In Build in car display itself	HTML, CSS, JavaScript / Angular JS / React JS etc.
2.	User Logic-1	Framework used for design the software	Python , python-flask

3.	User Logic-2	Access the software in the car by the driver to detect spot	Python, Open CV
4.	Application Logic-1	Open CV is an open-source platform for providing real time computer vision technology	Open CV
5.	Data Base	Contains images and video frames stores in data base	MySQL, NoSQL, etc.
6.	Cloud Data Base	Data Base Service on cloud	IBM DB2, IBM Cloud etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or local File system
8.	External API-1	They make it easy for developers to store manage and deploy container images	Container registry
10.	Machine Learning Model	Uses test and trained data images and video to learn the environment	Object recognition models, etc.
11.	Infrastructure (server / cloud)	Application Development on Local system / cloud	Local, cloud Foundry, python-flask, etc.

4.3 USER STORIES:

the below template to list all the user stories for the product

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint - 1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint - 1
		USN-3	As a user, I can register for the application through Face book	I can register & access the dashboard with Face book Login	Low	Sprint - 2
		USN-4	As a user, I can register for the application through Gmail	I can register the app with email account	Medium	Sprint - 1
	Login	USN-5	As a user, I can log into the	I can register &access user	High	Sprint - 1

			application by entering email & password	profile/account with Gmail account		
	Requesting/ conferrer	USN-6	As a conferrer I can request vacant parking space to park my car	I can get information about parking rates	High	Sprint - 2
Customer (Web user)	profile	USN-7	As a user I can see registration page ,login page and Chabot for I can check availability of parking spots in real time	I can login through email and social media account for registration	Medium	Sprint - 2
Customer Care Executive	Help desk /user support	USN-8	As a customer care executive I can solve the queries of the users	I can reply to their queries and solve their related problems	High	Sprint - 3
Administrator	Registration	USN-9	As an administrator ,I can view the database of the registered users	I can check and verify the persons who are the registered their mail id's and information's	Medium	Sprint - 4
	Dash board	USN-10	As an administrator ,I can view how many	I can check the number of requirements and monitor	Low	Sprint - 4

			members requested for what trouble occurs in parking a vehicle	the availability		
chatbot	User interface	USN-11	In addition to the customer care executive I can solve all the queries of the customer as well as the conferrer	I can reply to all the questions which are asked by the users that are related to the service we provided	Medium	Sprint - 4

5.CODING & SOLUTION:

5.1 FEATURE 1:

1.Add Vechicle Records

```
#Impor
t Time
import
time

Vehicle_Number=['XXXX-XX-
XXXX']Vehicle_Type=['Bike']
vehicle_Name=['Intruder']
Owner_Name=['Unknown']
Date=['22-22-3636']
Time=['22:22:22']
```

In this code block, we are importing the time module to implement its methods and function in the project. We have initialized the variables vehicle number, vehicle type, vehicle name, owner name date, and time to some default value. As well as bikes, cars, and bicycles with some initial value.

2. Create a while loop block to display the options in Vehicle Parking Management Project

```
def main():
   global
   bikes,cars,bicycles
   try:
      while True:
         print("..................................................................          ")print("\t\tParki
         print("..................................................................          ")print("1.Vehi
         print("2.Remove Entry" )
         print("3.View Parked Vehicle
         ") print("4.View Left Parking
         Space ")print("5.Amount
         Details ") print("6.Bill")
         print("7.Close Programme ")
```

```
        print("+.......................................................    +")ch=int(input("\tSel
```

In this code block, we have initialized the bikes, cars, and bicycles as global variables. They are accessible through the entire main block. Here we are providing the options to choose the service options from the list, for the vehicle parking management system.

3.Code for vehicle number entry

```
if
   c
   h
   =
   =
   1:
   n
   o
   =
   T
   ru
   e
```

Ch is for choice, Once we select the ch option as 1 which is for vehicle entry number, then we provide the while loop. while the number(no is True). We will store the vehicle number in Vno. If the vno is empty i.e vno==“”.The user asks to enter the vehicle number, else If the vno entered is already present in the vehicle number then it prints the vehicle number already exists. Else if len(vno)==12, It will ask to append the info to the vehicle number variable.

5.Code to enter the vehicle type

```
typee=True
while typee==True:
   Vtype=str(input("\tEnter vehicle
   type(Bicycle=A/Bike=B/Car=C):")).lower()if Vtype=="":
      print("###### Enter Vehicle Type
   ######")elif Vtype=="a":
      Vehicle_Type.append("Bi
      cycle")bicycles-=1
      typee=not
   Trueelif
```

```
        bikes-=1
        typee=not
        True
    elif Vtype=="c":
        Vehicle_Type.append(
        "Car")cars-=1
        typee=not
```

Here we have to initialize the typee variable to true. While the condition is True, the system asks to enter the vehicle type i.e a,b, or c which will accept the input in the lower case. Here A is for bicycle, B is for Bike and C is for Car. Any vehicle type you enter is stored in the variable Vtype. If the Vtype==””(empty). It will ask to enter the vehicle type. According to the type of variable you enter the vehicle type will be stored in the variable and typee variable is set to not True.

6.Code to enter the vehicle name

```
name=True
while   name==True:
    vname=input("\tEnter vehicle
    name - ")if vname=="":
        print("########Please Enter Vehicle Name
    ########")else:
        vehicle_Name.append(vname)
```

Here we have set the name== True. While the name == True i.e until we enter the name.vname store the value i.e. vehicle name. if the vname is empty system asks to enter the vehicle name, else it will store the name using the append function to the vehicle name variable The name variable is initialized to not True.

5.2 FEATURES 2

7.Code to enter the owners name

```
o=True
while o==True:
    OName=input("\tEnter owner
    name - ")if OName=="":
        print("###### Please Enter Owner Name
    ######")else:
        Owner_Name.append(ON
        ame)o=not True
```

O is initialized to True. While the condition satisfies the Owner's name is stored in the OName variable. If the OName is empty it system asks to enter the owner name else it will store the Owner name in the owner name variable. O is now initialized to Not True.

8.Code enter the date and time

```
d=True
while d==True:
    date=input("\tEnter Date (DD-MM-
    YYYY) - ")if date=="":
        print("###### Enter Date
    ######")elif len(date)!=10:
        print("###### Enter Valid Date
    ######")else:
        Date.append(dat
        e)d=not True
t=True
while t==True:
    time=input("\tEnter Time
    (HH:MM:SS) - ")if t=="":
        print("###### Enter Time
    ######")elif len(time)!=8:
        print("###### Please Enter Valid Time
    ######")else:
        Time.append(ti
```

Similarly, we have to create a while loop to enter the date and time initializing d and t to 0. Date variable stores the date and the time-variable stores time. The date and time variable checks the condition and accordingly execute further.

9.Code to remove the entry from the register

```
elif
    c
    h
```

```
        Vno=input("\tEnter vehicle number to Delete(XXXX-XX-XXXX) -
").upper()
        if Vno=="":
            print("###### Enter Vehicle No.
        ######")elif len(Vno)==12:
            if Vno in Vehicle_Number:
                i=Vehicle_Number.inde
                x(Vno)
                Vehicle_Number.pop(i)
                Vehicle_Type.pop(i)
                vehicle_Name.pop(i)
                Owner_Name.pop(i)
                Date.pop(i)
                Time.pop(i
                )no=not
                True
                print("\n............................................................ Removed
Sucessfully......................................................................")
            elif Vno not in Vehicle_Number:
                print("###### No Such Entry
```

In this code block, we are writing the code to remove the particular entry of a vehicle from the database. Users have entered a valid vehicle number to Create a Vehicle Parking Management in Python. If the vehicle number is present in our database and if the length of the vehicle number is 12 then it uses a pop function to remove the particular entry. Else if the vehicle number is not present in the database it will print “No such Entry” and asks to enter a valid vehicle number.

e
d
i
f

10.Code to display the vehicles present in the parking area

```
 print("Vehicle                    Vehicle              Owner
   print("-------------------------------------------------------------------------------------------------
----------------------------------------")
   for i in
      range(len(Vehicle_Number)
      ):count+=1
      print(Vehicle_Number[i],"\t   ",Vehicle_Type[i],"\t
",vehicle_Name[i],"\t       ",Owner_Name[i],"      " ,Date[i],"
                                                ",Time[i])
   print("-------------------------------------------------------------------------------------------------
----------------------------------------")
```

Here ch==3 is for displaying the parked vehicles in the parking area. For this, we have to use the for loop function. It counts the length of the vehicle number. This will display the whole information of the vehicles.

11. Code for spaces left in the parking area

```
elif ch==4:
   print("-------------------------------------------------------------------------------------------------
----------------------------------------")
   print("\t\t\t\tSpaces Left For
   Parking")
   print("-------------------------------------------------------------------------------------------------
----------------------------------------")
   print("\tSpaces Available for Bicycle -
   ",bicycles)print("\tSpaces Available for
```

This block of code displays the spaces left for parking in the parking area.

12. Code for displaying the parking rate

```
elif ch==5:
   print("-------------------------------------------------------------------------------------------------
--------------------------------       ")print("\t\t\t\tParking Rate")
   print("-------------------------------------------------------------------------------------------------
----------------------------------------")
   print("*1.Bicycle      Rs20 /
   Hour")print("*2.Bike
                         Rs40/
   Hour") print("*3.Car Rs60/
```

It displays the parking rate of different types of vehicles.

13. Code to generate bills for different types of vehicles parked

```
elif ch==6:
   print("................................................................ Generating Bill
............................................................................")
   no=True
   while no==True:
      Vno=input("\tEnter vehicle number to Delete(XXXX-XX-XXXX) -
").upper()
      if Vno=="":
         print("###### Enter Vehicle No. ######")
      elif len(Vno)==12:
         if Vno in Vehicle_Number:
            i=Vehicle_Number.index(Vno)
            no=not True
         elif Vno not in Vehicle_Number:
            print("###### No Such Entry ######")
         else:
            print("Error")
      else:
         print("###### Enter Valid Vehicle Number ######")
   print("\tVehicle Check in time - ",Time[i])
   print("\tVehicle Check in Date  -  ",Date[i])
   print("\tVehicle Type - ",Vehicle_Type[i])
   inp=True
   amt=0
   while inp==True:
      hr=input("\tEnter No. of Hours Vehicle Parked - ").lower()
      if hr=="":
         print("###### Please Enter Hours ######")
      elif int(hr)==0 and Vehicle_Type[i]=="Bicycle":
         amt=20
         inp=not True
      elif int(hr)==0 and Vehicle_Type[i]=="Bike":
         amt=40
         inp=not True
      elif int(hr)==0 and Vehicle_Type[i]=="Car":
         amt=60
         inp=not True
      elif int(hr)>=1:
         if Vehicle_Type[i]=="Bicycle":
            amt=int(hr)*int(2
            0)inp=not True
         elif
            Vehicle_Type[i]=="B
            ike":
            amt=int(hr)*int(40)
            inp=not True
         elif
            Vehicle_Type[i]=="
            Car":
            amt=int(hr)*int(60)
            inp=not True
   print("\t Parking Charge -
   ",amt)ac=18/100*int(amt)
   print("\tAdd. charge 18 %
   - ",ac)
   print("\tTotal           Charge           -
   ",int(amt)+int(ac)) print("................................ Thank you for
   using our
```

The billing section generates the bill for the vehicle parked. We have to enter the correct vehicle number and check the length of the vehicle number and the vehicle number present in the database. After this check the time and the type of vehicle. Depending upon the vehicle type the system calculates the charges. The last choice ch==7 is to come out of the service options and quit the program.

6.RESULTS:

6.1 PERFORMANCE METRICS:

Performance of the PMS, which employs resource allocation, real-time and dynamic path planning, and elevator scheduling algorithms are assessed and evaluated using the metrics

Performance metric	Definition
Average travel distance for storage (ATDS)	This performance parameter measures the average travel distance in terms of cell movements for all storage requests during the morning rush hours
Average travel distance for retrieval (ATDR)	This performance parameter measures the average travel distance in terms of cell movements for all retrieval requests during the evening rush hours
Range of utilization rate for elevators	This performance parameter measures the range of values for the ratio of time during which an elevator is busy transporting a vehicle to the 2-h simulation time, and separately for each of the two rush hours, namely, the morning rush hour and the afternoon rush hour
Customer waiting time for storage (WTS)	This performance metric measures the time that elapses from the instant the customer arrives at the end of the first-in-first-out (FIFO) queue extending from the front of delivery bays of the parking ground floor outward until (s)he reaches the delivery/retrieval bay on the ground floor
Customer waiting time for retrieval (WTR)	This performance parameter measures the time that elapses from the instant a customer arrives at the parking structure to retrieve his/her

Time to full capacity utilization :This performance parameter records the occurrence time for the full space evaluation metrics and for parking structure utilization for the parking structure during the morning rush hours their definitions

7.ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

Recent increments in car ownership and urbanization, coupled with poor city planning, have contributed greatly to the rising parking problems across the US[6]. Fortunately, the adoption of AI in smart parking has completely revolutionized how we look for parking spaces. Here are a few advantages of smart parking for drivers and business owners.

Less fuel consumption
Driving around looking for a parking space consumes a lot of fuel. And considering the current fuel prices, that's a lot of money going down the drain. Smart parking solutions provide easy access to parking spots, thereby saving precious resources such as time, fuel, and space.

A reduction of search traffic on the streets
Nearly a third of traffic in urban areas is created by drivers looking for a parking spot. By leveraging AI-based smart parking solutions, municipalities can manage and reduce search traffic on busy streets. Smart parking solutions not only minimize search traffic but also smoothen traffic flow. The result? People spend less time looking for a parking spot since they already know where to get one.

Reduced parking stress
Looking for a good parking spot in a congested part of the city is simply overwhelming. You might find yourself driving across the same street several times, only to end up parking far away from your destination.

AI-based smart parking solutions incorporate smart parking technologies with IoT devices so drivers can find parking spots easily using their smartphones and computers. This way, they can see all parking spots available in the area they plan to travel to long before they get there. Some parking lots even allow you to reserve a space. This means you don't have to drive around looking for a space to park.

Benefits of AI-based smart parking for businesses:
Surface parking lots take up 5% of all urban land in the US [7]. This means more competition among parking lot businesses. You'd think parking lots around busy streets get a lot of business, but sadly, that's not the case. In most cases, drivers can't find these parking lots, and when they do, there are tons of irregularly parked cars, making the parking lot inefficient.

The result is that some drivers opt to drive longer distances for safer and more accessible parking lots, which means less business for the better-situated but poorly managed parking lots. Fortunately, by incorporating smart parking solutions in their management process, parking lot managers can improve their lots' efficiency, boost customer satisfaction and ultimately boost profits. Here are a few other benefits parking lot businesses stand to gain from leveraging smart parking solutions.

Improved parking experience:

AI-based smart parking solutions leverage collected data to provide specialized services that ultimately lead to stress-free parking experiences. Besides letting nearby drivers know if there are available spots in the parking lot, managers can also install digital signs that receive real-time data from parking lot management software to direct drivers to their parking spots.
This eliminates frustration among drivers trying to find a spot within the parking lot, thus improving customer satisfaction.

Also, the mere fact that drivers don't have to drive around the parking looking for a free space means fewer emissions, which could improve the air quality in indoor parking lots.

Pinpoint inefficiencies in parking lot management:
Managing a parking lot takes a lot of work. For instance, parking lot managers often have to check the parking duration of certain vehicles or deal with irregularly and illegally parked vehicles. By leveraging real-time information from connected on-site devices, they can accurately determine the parking duration of all vehicles.

This comes in handy, especially in parking lots offering short-time parking services like supermarkets. With this data, the managers can monitor excessive parking durations, which in most cases imply unauthorized use of their parking areas. They can also monitor spaces that stay empty for extended periods, which could imply an issue with the spot in question.

Optimized usage of the facility:
Through data collected from various sensors installed around the streets and parking lot, businesses can monitor which areas have the highest and lowest parking traffic. With this data, they can better determine where to expand or cut back operations. Sensor data also enables businesses to monitor misuse of emergency access roads and dedicated parking spots.

DISADVANTAGES:

Smart parking solutions present a lot of advantages for both drivers and businesses. But they also present a few drawbacks that might cause some people to put back or even avoid incorporating them altogether. Here are some of the drawbacks of incorporating smart parking solutions.

High cost of installation
Numerous systems and technologies go into building an effective smart parking lot management system. Things like sensors, cameras, automated ticketing machines, and software cost a lot of money to install. Unfortunately, some businesses can’t afford these systems, making it nearly impossible to incorporate the system.

Regular maintenance requirements
Despite being automated, smart parking management systems require regular maintenance to ensure smooth operation. The frequency of maintenance all comes down to the system in question, but most systems require monthly maintenance. This further racks up the running costs of the parking lot, which might have a significant impact on profits.

8.CONCLUSION:

As conclusion, the objectives of this project have been achieved. The hassle in searching for available parking slots has been completely eliminated. The designed system could be applied everywhere due to its ease of usage and effectiveness. It facilitates the problems of urban livability , transportation mobility and environment sustainability. The Internet of Things integrates the hardware, software and network connectivity that enable objects to be sensed and remotely controlled across existing network. Such integration allows users to monitor available and unavailable parking spots that lead to improved efficiency, accuracy and economic benefit.

9.FUTURE SCOPE:

The smart parking management system can be broadly applied for many future applications. Apart from its basic role of parking management of cars it can also be applied for plane and ship and fleet management. With the ever growing field of Internet of Things many concepts can be interfaced along with our system. For residential and domestic parking system the device can be interfaced with Home Automation system which can control the various home appliances by sensing whether the user is arriving or departing from the parking space. For instance if the user has arrived then the module will sense the presence and will send information about arrival to the Home automation system which can accordingly switch on the selected appliances like HVAC (Heating Ventilation and Air Conditioning) units, Coffee maker, toaster, Wi-Fi routers etc. For commercial parking system the device can be interfaced with a module which can sense the arrival of employee and can switch on his computer and HVAC systems and accordingly switch off the appliances when the employee departs. The system can also be used to track the reporting and departing time of the employee for all days with precision thus acting as an attendance system. Thus many such modules can be interfaced with our system to provide better facility, security, and optimization of electricity and resources with the principle idea of flawless fleet management system.

10. APPENDIX:

SOURCE CODE:

Main source code:

```
import cv2

import pickle

import cvzone

import numpy as np


cap = cv2.VideoCapture('carPark.mp4')

width, height = 103, 43

with open('CarParkPos', 'rb') as f:

    posList = pickle.load(f)



def empty(a):

    pass



cv2.namedWindow("Vals")

cv2.resizeWindow("Vals", 640, 240)

cv2.createTrackbar("Val1", "Vals", 25, 50, empty)

cv2.createTrackbar("Val2", "Vals", 16, 50, empty)

cv2.createTrackbar("Val3", "Vals", 5, 50, empty)
```

```
def checkSpaces():
    spaces = 0
    for pos in posList:
        x, y = pos
        w, h = width, height

        imgCrop = imgThres[y:y + h, x:x + w]
        count = cv2.countNonZero(imgCrop)

        if count < 900:
            color = (0, 200, 0)
            thic = 5
            spaces += 1

        else:
            color = (0, 0, 200)
            thic = 2

        cv2.rectangle(img, (x, y), (x + w, y + h), color, thic)

        cv2.putText(img, str(cv2.countNonZero(imgCrop)), (x, y + h - 6),
cv2.FONT_HERSHEY_PLAIN, 1,
                color, 2)

    cvzone.putTextRect(img, f'Free: {spaces}/{len(posList)}', (50, 60),
thickness=3, offset=20,
                   colorR=(0, 200, 0))



while True:

    # Get image frame
    success, img = cap.read()
    if cap.get(cv2.CAP_PROP_POS_FRAMES) ==
cap.get(cv2.CAP_PROP_FRAME_COUNT):
        cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
    # img = cv2.imread('img.png')
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    imgBlur = cv2.GaussianBlur(imgGray, (3, 3), 1)
    # ret, imgThres = cv2.threshold(imgBlur, 150, 255,
cv2.THRESH_BINARY)

    val1 = cv2.getTrackbarPos("Val1", "Vals")
    val2 = cv2.getTrackbarPos("Val2", "Vals")
    val3 = cv2.getTrackbarPos("Val3", "Vals")
    if val1 % 2 == 0: val1 += 1
    if val3 % 2 == 0: val3 += 1
    imgThres = cv2.adaptiveThreshold(imgBlur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                     cv2.THRESH_BINARY_INV, val1, val2)
    imgThres = cv2.medianBlur(imgThres, val3)
    kernel = np.ones((3, 3), np.uint8)
    imgThres = cv2.dilate(imgThres, kernel, iterations=1)

    checkSpaces()
    # Display Output

    cv2.imshow("Image", img)
    # cv2.imshow("ImageGray", imgThres)
    # cv2.imshow("ImageBlur", imgBlur)
    key = cv2.waitKey(1)
    if key == ord('r'):
        pass

import cv2
import pickle

width, height = 107, 48

try:
    with open('CarParkPos', 'rb') as f:
        posList = pickle.load(f)
except:
    posList = []



def mouseClick(events, x, y, flags, params):
    if events == cv2.EVENT_LBUTTONDOWN:
        posList.append((x, y))
    if events == cv2.EVENT_RBUTTONDOWN:
        for i, pos in enumerate(posList):
            x1, y1 = pos
            if x1 < x < x1 + width and y1 < y < y1 + height:
                posList.pop(i)

    with open('CarParkPos', 'wb') as f:
        pickle.dump(posList, f)



while True:
    img = cv2.imread('carParkImg.png')
    for pos in posList:
        cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), (255, 0,
255), 2)
```

```
    cv2.imshow("Image", img)
    cv2.setMouseCallback("Image", mouseClick)
    cv2.waitKey(1)
```

GITHUB AND PROJECT VEDIO DEMO: