KINGSTON ENGINEERING COLLEGE

AI Enabled Car Parking Using Open CV

IBM NAAN MUDHALVAN

PROJECT REPORT

Submitted By

T. Shammikumar (511320205025)
G. Janarathanan (511320205011)
J. Kishore Kumar (511320205017)
R. Ram Prasad (511320205020)

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

ANNA UNIVERSITY::CHENNAI 600 025
MAY 2023

TABLE OF CONTENTS

Ex.No:	Name of the Exercise	Page. No:
1.	**Introduction**	
	Project Overview	4
	Purpose	4
2.	**Ideation And Proposed Solution**	
	Problem Statement Definition	5
	Empathy Map Canvas	6
	Ideation And Brainstorming	7
	Proposed Solution	8
3.	**Requirement Analysis**	
	Functional Requirement	9
	Non-Functional Requirement	10
4.	**Project Design**	
	Data Flow Diagram	11
	Solution & Technical Architecture	12
	User Stories	13
5.	**Coding And Solution**	
	Feature 1	14
	Feature 2	14

Ex.No:	Name of the Exercise	Page. No:
6.	**Advantages And Disadvantages**	15
7.	**Conclusion**	16
8.	**Future Scope**	17
9.	**Appendix**	19
	Source code	19
	Git-Hub & Project Video Demo Link	23
10.	**References**	24

EX.NO: 1	INTRODUCTION

PROJECT OVERVIEW

In recent years, advancements in Artificial Intelligence (AI) and computer vision technologies have revolutionized various industries, including the automotive sector. One notable application of AI in automobiles is the development of AI-enabled car parking systems. These systems utilize computer vision techniques, particularly OpenCV (Open Source Computer Vision Library), to enhance the efficiency and accuracy of parking vehicles. The traditional process of parking a car can be challenging and time-consuming for many drivers. However, with the integration of AI and OpenCV, car parking becomes a seamless and automated experience. By leveraging real-time video processing and object detection algorithms, AI-enabled car parking systems can assist drivers in finding parking spaces, navigating tight spots, and avoiding collisions. OpenCV, a popular open-source library, provides a wide range of functions and tools for computer vision tasks. It offers robust capabilities for image processing, object detection, and motion tracking, making it an ideal choice for implementing AI-enabled car parking systems. With the aid of cameras and sensors strategically placed in parking lots or vehicles, OpenCV algorithms can analyze the surrounding environment, detect obstacles, and generate accurate guidance for parking maneuvers.

The core principle behind AI-enabled car parking is the ability to recognize and interpret visual data in real-time. OpenCV algorithms leverage machine learning techniques, such as deep neural networks, to train models that can identify various objects, including cars, pedestrians, and obstacles. This enables the system to accurately perceive the parking environment and make intelligent decisions based on the detected objects' positions, sizes, and trajectories.

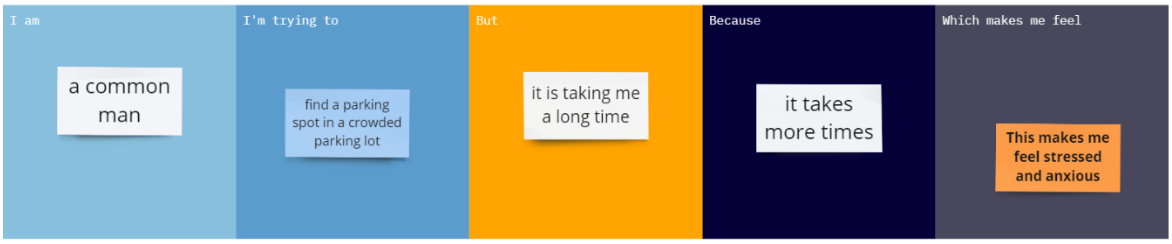
PURPOSE

The purpose of AI-enabled car parking using OpenCV is to leverage advanced technologies to transform the parking experience, making it more efficient, safe, and user-friendly. By automating parking processes, optimizing space utilization, and enhancing driver guidance, these systems offer significant benefits for both individual drivers and urban infrastructure planning.
The integration of AI and OpenCV in parking systems serves several key purposes:
Efficient Space Utilization, Enhanced Safety, Time Savings, User-Friendly Experience, Smart City Integration

EX.NO: 2	**IDEATION & PROPOSED SOLUTION**

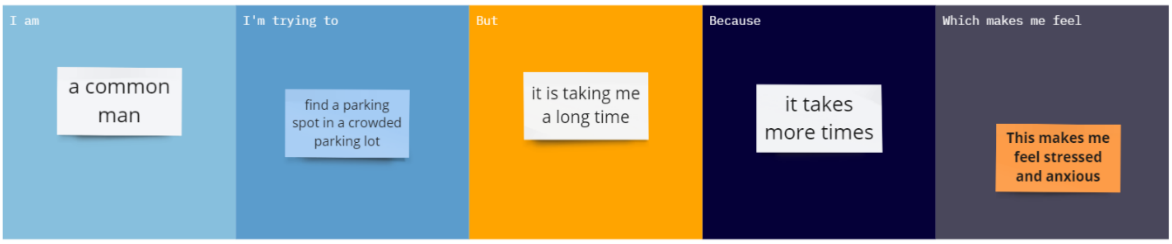
PROBLEM STATEMENT DEFINITION

Problem statement 1:



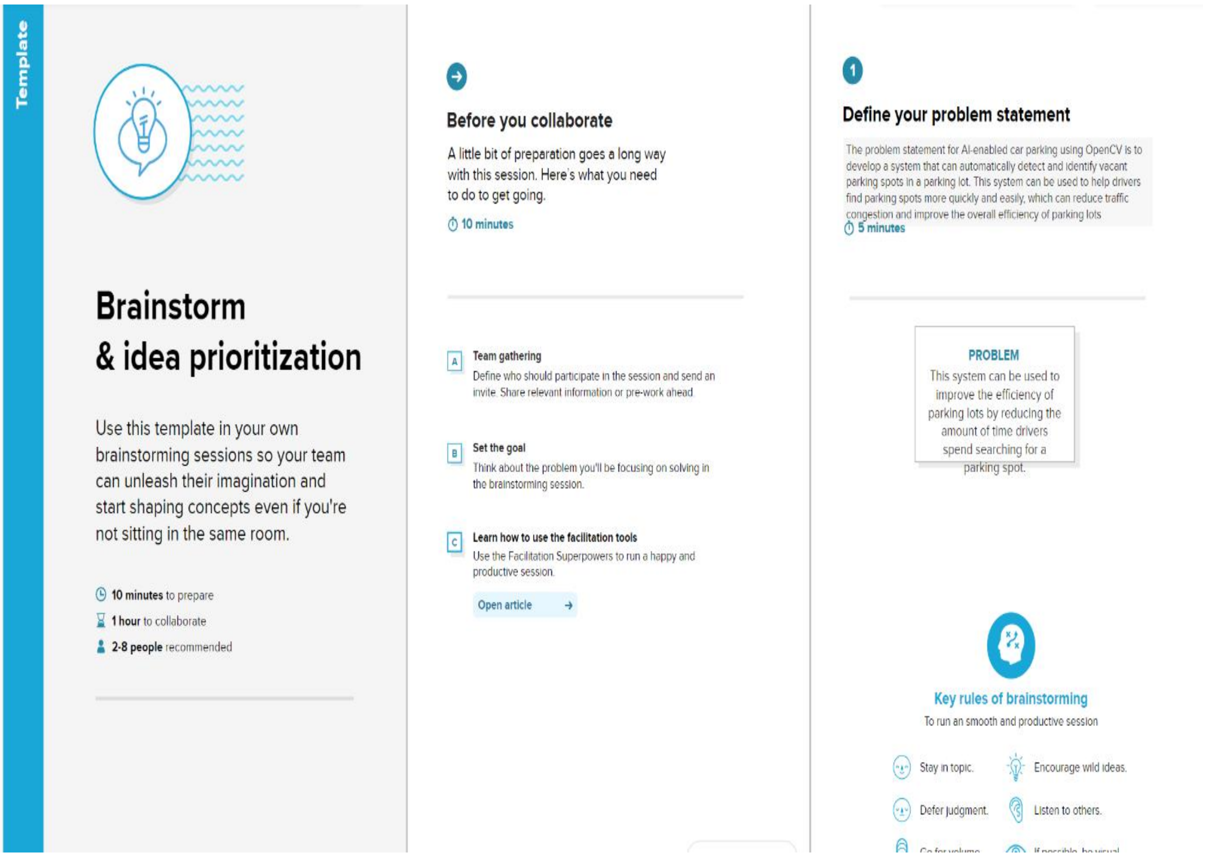
I am	I'm trying to	But	Because	Which makes me feel
a common man	find a parking spot in a crowded parking lot	it is taking me a long time	it takes more times	This makes me feel stressed and anxious

Problem statement 2:



I am	I'm trying to	But	Because	Which makes me feel
a common man	find a parking spot in a crowded parking lot	it is taking me a long time	it takes more times	This makes me feel stressed and anxious

EMPATHY MAP CANVAS



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

The problem statement for AI-enabled car parking using OpenCV is to develop a system that can automatically detect and identify vacant parking spots in a parking lot. This system can be used to help drivers find parking spots more quickly and easily, which can reduce traffic congestion and improve the overall efficiency of parking lots

5 minutes

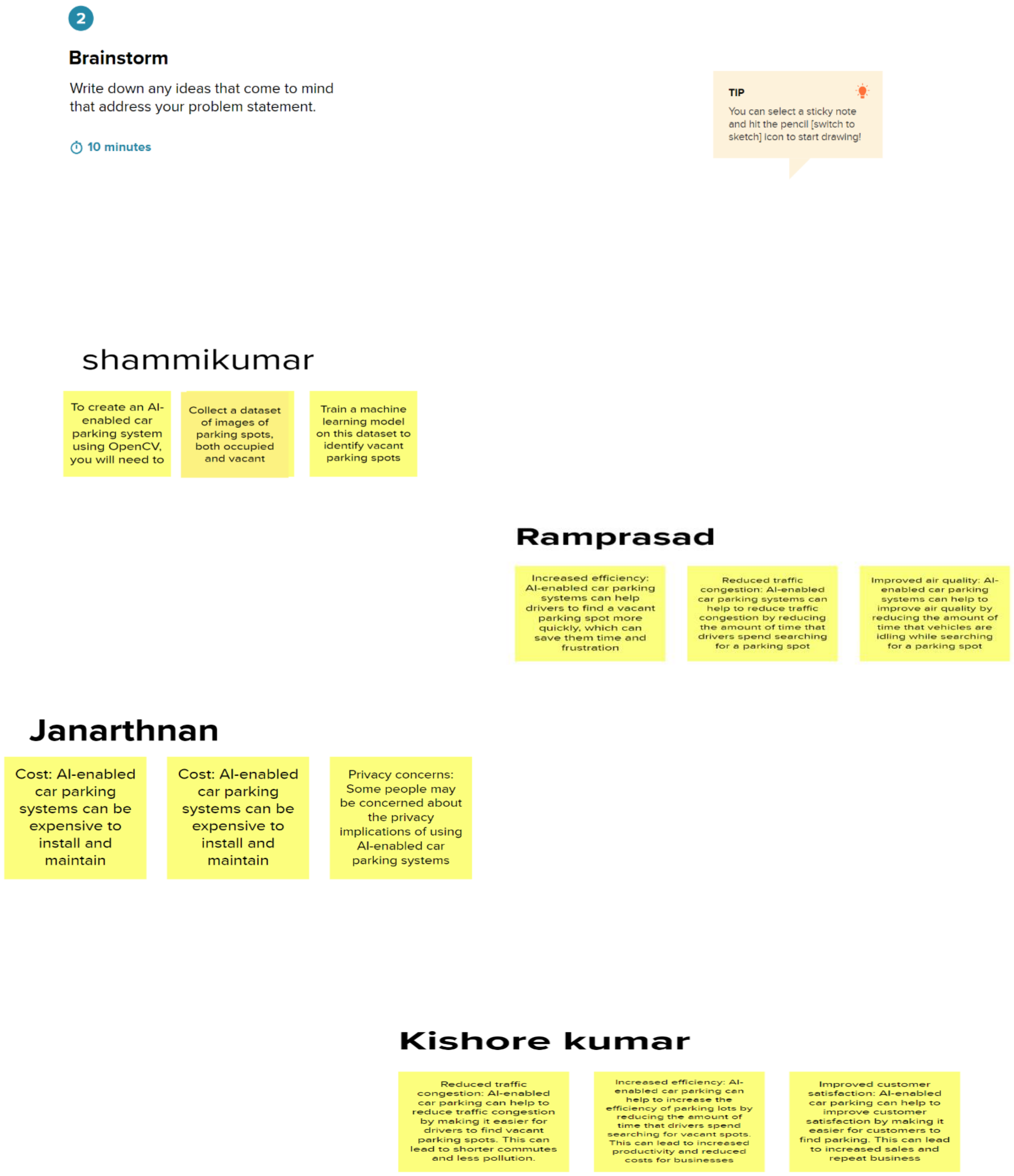
PROBLEM
This system can be used to improve the efficiency of parking lots by reducing the amount of time drivers spend searching for a parking spot.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

IDEATION & BRAINSTORMING:



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

shammikumar

To create an AI-enabled car parking system using OpenCV, you will need to

Collect a dataset of images of parking spots, both occupied and vacant

Train a machine learning model on this dataset to identify vacant parking spots

Ramprasad

Increased efficiency: AI-enabled car parking systems can help drivers to find a vacant parking spot more quickly, which can save them time and frustration

Reduced traffic congestion: AI-enabled car parking systems can help to reduce traffic congestion by reducing the amount of time that drivers spend searching for a parking spot

Improved air quality: AI-enabled car parking systems can help to improve air quality by reducing the amount of time that vehicles are idling while searching for a parking spot

Janarthnan

Cost: AI-enabled car parking systems can be expensive to install and maintain

Cost: AI-enabled car parking systems can be expensive to install and maintain

Privacy concerns: Some people may be concerned about the privacy implications of using AI-enabled car parking systems

Kishore kumar

Reduced traffic congestion: AI-enabled car parking can help to reduce traffic congestion by making it easier for drivers to find vacant parking spots. This can lead to shorter commutes and less pollution.

Increased efficiency: AI-enabled car parking can help to increase the efficiency of parking lots by reducing the amount of time that drivers spend searching for vacant spots. This can lead to increased productivity and reduced costs for businesses

Improved customer satisfaction: AI-enabled car parking can help to improve customer satisfaction by making it easier for customers to find parking. This can lead to increased sales and repeat business

PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problemto be solved)	I am the long traveller , I’m trying to park my car in lawn , but I am not able to park because there is insufficient space in the lawn which makes me feel bad .
2.	Idea / Solution description	The solution given by car owners is park manually with help of our car parking AI in outside the park area (or) Wait for few minutes to some one give u space to park your car.
3.	Novelty / Uniqueness	The uniqueness is We have to park our car based upon the lawn place & the environment situation .
4.	Social Impact / CustomerSatisfaction	The owner feels very happy by got space to park his car safely and the owner gets satisfy .
5.	Business Model (RevenueModel)	On the revenue based its profit for the users No need of any external maintenance and System needed .
6.	Scalability of the Solution	The searching for car parking space time become reduced and to park our car safely without any crash .

EX.NO: 3	REQUIREMENT ANALYSIS

Functional Requirements:

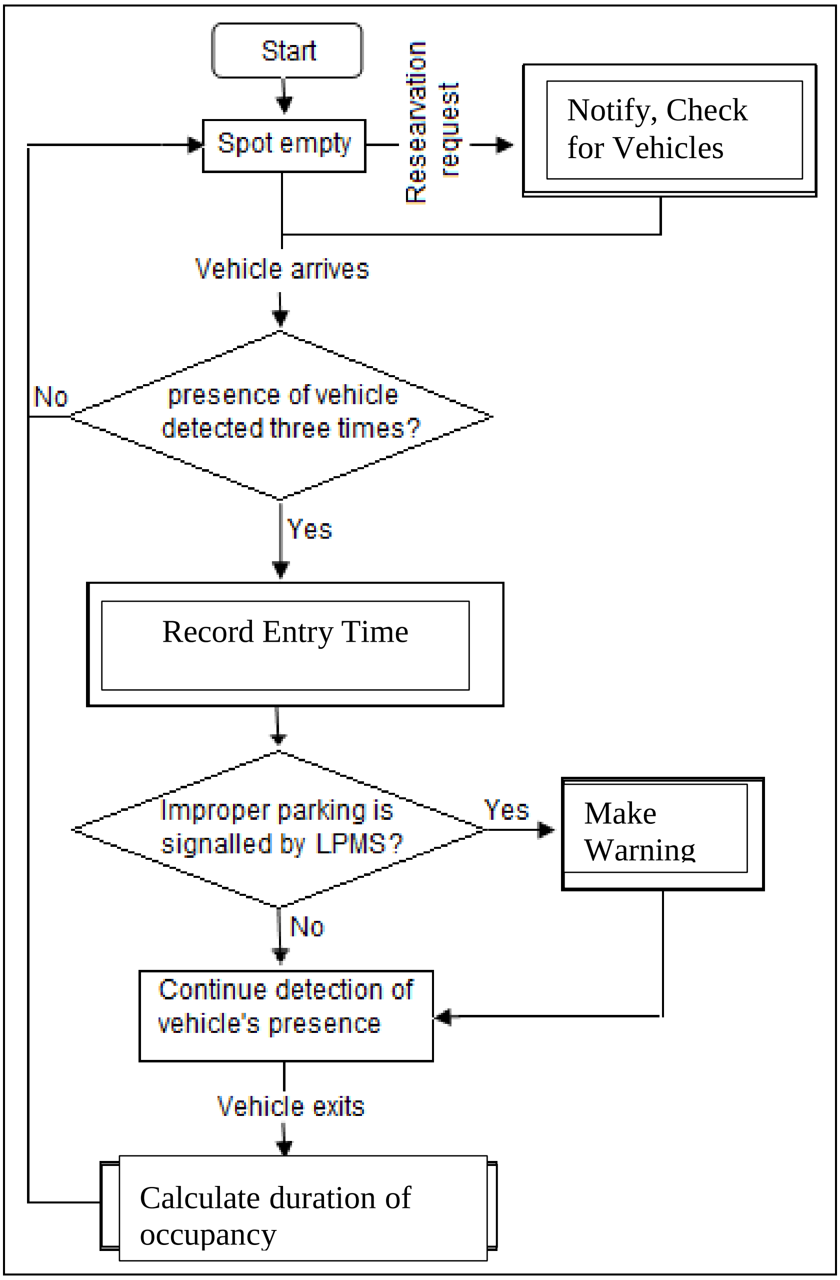
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Object Detection and Recognition	The AI-enabled car parking system needs to detect and recognize cars present in the parking lot using OpenCV and object detection algorithms.
FR-2	Determine Parking Space Availability	The AI-enabled car parking system should determine the availability of parking spaces based on the detected cars in the parking lot. Acceptance Criteria: 1.The system should analyze the detected cars' positions and sizes to determine parking space availability. 2.It should identify and differentiate between occupied and unoccupied parking spaces. 3.The system should provide real-time updates on the number of available parking spaces. 4.It should handle complex parking lot layouts and account for different parking space sizes and orientations
FR-3	Parking Guidance and Assistance	The AI-enabled car parking system should generate accurate guidance for drivers, directing them to available parking spaces. Acceptance Criteria: The system should utilize the detected car positions and available parking space information to generate optimal parking guidance. It should consider factors such as distance, maneuverability, and parking lot layout while generating guidance. The guidance should be displayed to the driver, either through visual indicators or audio instructions, to assist in parking the vehicle.
FR-4	Parking Confirmation and Feedback	The AI-enabled car parking system should provide confirmation and feedback to the driver upon successful parking.

Non-functional Requirements:

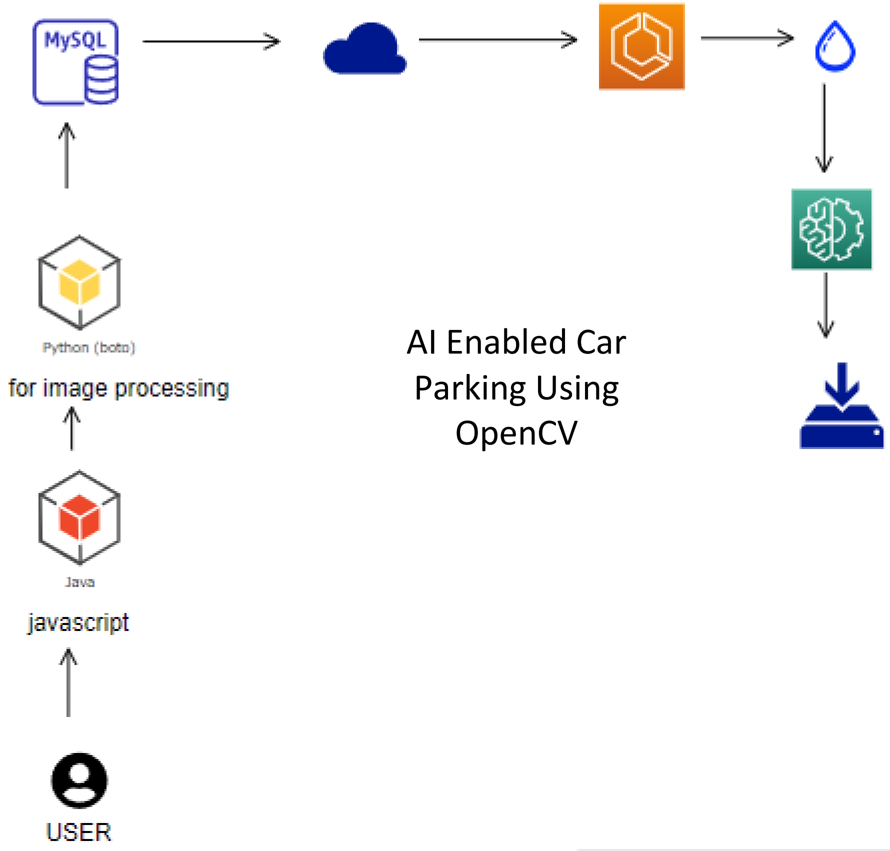
FR No.	Non-Functional Requirement	Description
NFR-1	Performance and Real-Time Processing	The AI-enabled car parking system using OpenCV should meet specific performance criteria and ensure real-time processing of video feeds and object detection**.**
NFR-2	Reliability and Availability	The AI-enabled car parking system using OpenCV should ensure reliability and availability to provide a seamless parking experience.
NFR-3	User Interface and Usability	The AI-enabled car parking system should provide a user-friendly interface and ensure ease of use for both drivers and administrators.
NFR-4	Security and Privacy	The AI-enabled car parking system using OpenCV should prioritize security and privacy to protect the data and ensure the confidentiality of users
NFR-5	Compatibility and Integration	The AI-enabled car parking system using OpenCV should be compatible with existing infrastructure and capable of seamless integration with other systems or components.
NFR-6	Performance Optimization and Resource Efficiency.	The AI-enabled car parking system using OpenCV should optimize performance and utilize system resources efficiently

EX.NO: 4	PROJECT DESIGN

DATA FLOW DIAGRAMS:


Start
Spot empty
Researvation request
Notify, Check for Vehicles
Vehicle arrives
No
presence of vehicle detected three times?
Yes
Record Entry Time
Improper parking is signalled by LPMS?
Yes
Make Warning
No
Continue detection of vehicle's presence
Vehicle exits
Calculate duration of occupancy

SOLUTION & TECHNICAL ARCHITECTURE:



User Stories

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	Jana
		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	shammi
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	ram
		USN-4	As a user, I can register for the application through Gmail		Medium	kishore
	Login	USN-5	As a user, I can log into the application by entering email & password		High	shammi
	Dashboard		Mobile support. Users must be able to interact in the same roles and on the same tasks on computers and mobile devices where practical, given mobile capabilities.			ram
Customer (Web user)			First version of your assistant operates on one channel and within one initial domain or department			jana
Customer Care Executive			Customers hate hitting dead ends when they're trying to get something done with your business. This means that 100% of the questions need some method of resolution. It's likely that, in this first launch, you'll be able to handle 40–70% of questions with the assistant itself between your top topics and			kishore

EX.NO: 5	CODING & SOLUTIONING

FEATURE 1

An AI-enabled car parking system using OpenCV (Open Source Computer Vision) can incorporate several features to enhance the parking experience. OpenCV is a popular open-source computer vision library that provides tools and algorithms for image and video processing. Here are some features that can be implemented: Vehicle Detection: OpenCV can be used to detect and track vehicles in real-time using video or image input. It can analyze the video feed from parking lot cameras and identify the presence of vehicles. Space Detection: By analyzing the video feed or images, OpenCV can determine the availability of parking spaces. It can detect whether a parking space is occupied or vacant by analyzing the size, shape, and color of the vehicles in the scene. Object Tracking: OpenCV can track the movement of vehicles within the parking lot. It can assign unique identifiers to vehicles and track their positions over time. This information can be used to optimize parking operations and monitor the usage of parking spaces. Parking Guidance: The AI system can guide drivers to available parking spaces using visual indicators or signage. OpenCV can analyze the parking lot layout and provide real-time information on the availability of spaces, guiding drivers to the nearest vacant spot. Automatic Number Plate Recognition (ANPR): OpenCV can be used for automatic number plate recognition to read and identify vehicle license plates. This feature can help in managing parking access control, detecting unauthorized vehicles, or facilitating automated payment systems

FEATURE 2

Parking Slot Size Detection: OpenCV can measure the dimensions of parking spaces and classify them based on vehicle size compatibility. This information can be used to guide drivers to appropriate parking spots that match their vehicle size.Parking Lot Occupancy Prediction: By analyzing historical parking data and considering factors such as time of day, day of the week, and special events, OpenCV can predict parking lot occupancy levels. This feature can help drivers plan their parking in advance and avoid crowded areas.Parking Violation Detection: OpenCV can be used to detect parking violations such as parking in no-parking zones, handicap spots, or fire lanes. The system can analyze the video feed or images and issue alerts or notifications to parking attendants or authorities.

.

EX.NO: 6	ADVANTAGES AND DISADVANTAGES

Advantages of AI Enabled Car Parking Using Open CV

Advantages:

- Improved Parking Efficiency: AI-enabled car parking systems optimize space utilization, maximizing the number of vehicles accommodated in parking lots.

- Time Savings: The system provides real-time guidance, reducing the time spent searching for parking spaces and manoeuvring into them.

- Enhanced Safety: AI algorithms and object detection techniques help prevent accidents by identifying obstacles, pedestrians, and other vehicles.

- User-Friendly Experience: The system provides intuitive visual guidance, making the parking process easier and less stressful for drivers.

Disadvantages of AI Enabled Car Parking Using Open CV

Disadvantages:

- Initial Cost: Implementing an AI-enabled car parking system requires upfront investment in hardware, software, and installation, which can be costly.

- System Complexity: Developing and maintaining an AI system with OpenCV requires expertise in computer vision, AI algorithms, and system integration, which may pose challenges for some organizations.

- Maintenance and Updates: AI systems require regular maintenance and updates to ensure optimal performance, which can involve additional costs and effort.

EX.NO: 7	CONCLUSION

AI-enabled car parking systems using OpenCV offer several benefits that enhance the parking experience for drivers and improve overall parking efficiency. These systems optimize parking space utilization, reduce the time spent searching for parking spots, and provide real-time guidance to drivers, resulting in time savings and a user-friendly experience. The integration of AI algorithms and object detection techniques enhances safety by identifying obstacles and preventing accidents during parking manoeuvres. Additionally, AI-enabled car parking systems generate valuable data insights that can be used for parking management, traffic planning, and urban development.

However, there are some considerations to keep in mind. The initial cost of implementing an AI-enabled car parking system can be significant, and the technical complexity involved in developing and maintaining the system requires expertise in computer vision and AI. Privacy concerns may arise due to the collection and processing of video feeds for object detection. Regular maintenance and updates are necessary to ensure optimal system performance. Additionally, the reliability of these systems relies on stable infrastructure, including power supply, network connectivity, and functioning cameras.

Despite these considerations, AI-enabled car parking systems using OpenCV offer a promising solution to address parking challenges in urban areas. By leveraging AI technology, these systems improve space management, safety, and efficiency, contributing to smarter, more connected cities. With careful planning, implementation, and ongoing maintenance, organizations can harness the advantages of AI-enabled car parking systems to provide a seamless and convenient parking experience for drivers while optimizing parking resources.

EX.NO: 8

FUTURE SCOPE

The future scope for AI-enabled car parking using OpenCV is promising, with several potential advancements and opportunities for development. Here are some future possibilities:

Enhanced Object Detection: Continued advancements in AI algorithms and computer vision techniques will likely lead to improved object detection capabilities. This can include better recognition of various vehicle types, more accurate identification of parking spaces, and the ability to detect and classify other objects or obstacles in the parking environment.

Integration with Connected Vehicles: As connected and autonomous vehicles become more prevalent, AI-enabled car parking systems can integrate with these vehicles. This integration can enable seamless communication between the parking system and vehicles, allowing for automated parking processes, reservation systems, and enhanced navigation assistance.

Smart Parking Management: AI-enabled car parking systems can integrate with larger smart city initiatives to create a comprehensive parking management ecosystem. This can involve real-time data sharing, dynamic pricing models based on parking demand, integration with public transportation systems, and smart parking guidance through mobile applications or digital signage.

Sustainability and Energy Efficiency: Future developments may focus on incorporating sustainability and energy-efficient practices into AI-enabled car parking systems. This can include the integration of renewable energy sources for powering parking facilities, intelligent lighting systems that optimize energy consumption, and the implementation of green infrastructure in parking structures.

Advanced Analytics and Predictive Insights: AI-enabled car parking systems can leverage data analytics and machine learning techniques to provide predictive insights. By analyzing historical parking data, traffic patterns, and external factors, the system can anticipate parking demand, optimize space allocation, and provide recommendations for parking availability and efficient traffic flow.

Augmented Reality (AR) Integration: AR technology can enhance the user experience in AI-enabled car parking systems. Drivers can use AR interfaces to visualize parking space availability, receive real-time navigation guidance through visual overlays, and obtain additional information such as parking rates or nearby amenities.

Integration with Smart Home Systems: AI-enabled car parking systems can integrate with smart home systems, allowing homeowners to remotely monitor and manage their parking spaces. This can include features like remote access control, automated gate opening, and the ability to reserve parking spaces for guests.

Integration with Parking Payment Systems: Seamless integration with digital payment systems and mobile wallets can enhance the convenience of AI-enabled car parking. Drivers can easily make payments, receive digital receipts, and access loyalty programs or discounts through integrated payment platforms.

These are just a few potential future developments for AI-enabled car parking using OpenCV. As technology continues to advance, the scope for innovation in this field is vast. The integration of AI, computer vision, IoT, and other emerging technologies will likely lead to even more sophisticated and efficient parking solutions that revolutionize the way we park our vehicles.

EX.NO: 9	APPENDIX

SOURCE CODE:

```
from flask import Flask,render_template,request,session
import cv2
import pickle
import cvzone
import numpy as np
import ibm_db
import re

app=Flask(_name_)
app.secret_key='a'
conn =
ibm_db.connect("DATABASE=bludb;PORT=32459;SECURITY=SSL;Se
rverSecurityCertificate=;HOSTNAME=http://9938aec0-8105-433e-8bf9-
0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;UI=
fml40618;PWD=7hgxTYgbTNdOjrB6","","")
print("connected")
@app.route('/')
def project():
    return render_template('index.html')

@app.route('/home')
def home():
    return render_template('index.html')

@app.route('/model')
def model():
    return render_template('model.html')
```

```
@app.route('/login')
def login():
    return render_template('login.html')

@app.route("/reg",methods=['POST','GET'])
def signup():
    msg="
    if request.method=="POST":
     name=request.form["name"]
     email=request.form["email"]
     password=request.form["password"]
     sql="SELECT * FROM REGISTER WHERE name=?"
     stmt=ibm_db.prepare(conn,sql)
     ibm_db.bind_param(stmt,1,name)
     ibm_db.execute(stmt)
     account=ibm_db.fetch_assoc(stmt)
     print(account)
    if account:
        return render_template('login.html',error=True)
    elif not re.math(r'[^@]+@[^@]+\.[^@]+',email):
        msg="Invalid Email Address!"
    else:
        insert_sql="INSERT INTO REGISTER VALUES(?,?,?)"
        prep_stmt=ibm_db.prepare(conn,insert_sql)
        #this username & password should be as same as db-2 details &
order also
        ibm_db.bind_param(prep_stmt,1,name)
        ibm_db.bind_param(prep_stmt,2,email)
        ibm_db.bind_param(prep_stmt,3,password)
        ibm_db.execute(prep_stmt)
        msg="You Have successfully registered ! "
        return render_template('login.html',msg=msg)

@app.route("/log",methods=['POST','GET'])
```

```
    def login1():
        if request.method=="POST":
            email=request.form["email"]
            password=request.form["password"]
            sql="SELECT * FROM REGISTER WHERE EMAIL=? AND
PASSWORD=?"
            stmt=ibm_db.prepare(conn,sql)
            #this username & password whould be same as db-2 details &
order also
            ibm_db.bind_param(stmt,1,email)
            ibm_db.bind_param(stmt,2,password)
            ibm_db.execute(stmt)
            account=ibm_db.fetch_assoc(stmt)
            print(account)
            if account :
                session['Loggedin']=True
                session['id']=account['EMAIL']
                session['email']=account['EMAIL']
                return  render_template('model.html')
            else:
                msg="Incorrect Email/password"
                return render_template('login.html',msg=msg)
        else:
            return render_template('login.html')

    @app.route('/predict_live')
    def liv_pred():
        # Video feed
        cap = cv2.VideoCapture('carParkingInput.mp4')
        with open('parkingSlotPosition', 'rb') as f:
            posList = pickle.load(f)
        width, height = 107, 48
        def checkParkingSpace(imgPro):
            spaceCounter = 0
            for pos in posList:
```

```
            x, y = pos
            imgCrop = imgPro[y:y + height, x:x + width]
            # cv2.imshow(str(x * y), imgCrop)
            count = cv2.countNonZero(imgCrop)
            if count < 900:
                color = (0, 255, 0)
                thickness = 5
                spaceCounter += 1
            else:
                color = (0, 0, 255)
                thickness = 2
            cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), color,
thickness)
            """cvzone.putTextRect(img, str(count), (x, y + height - 3),
scale=1,
                                  thickness=2, offset=0, colorR=color)"""
        cvzone.putTextRect(img, f'Free:
{spaceCounter}/{len(posList)}',(100, 50), scale=3,
                              thickness=5, offset=20, colorR=(200, 0, 0))
    while True:
        if cap.get(cv2.CAP_PROP_POS_FRAMES) ==
cap.get(cv2.CAP_PROP_FRAME_COUNT):
            cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
        success, img = cap.read()
        imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        imgBlur = cv2.GaussianBlur(imgGray, (3, 3), 1)
        imgThreshold = cv2.adaptiveThreshold(imgBlur, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                                cv2.THRESH_BINARY_INV, 25, 16)
        imgMedian = cv2.medianBlur(imgThreshold, 5)
        kernel = np.ones((3, 3), np.uint8)
        imgDilate = cv2.dilate(imgMedian, kernel, iterations=1)
        checkParkingSpace(imgDilate)
        cv2.imshow("Image", img)
        # cv2.imshow("ImageBlur", imgBlur)
        # cv2.imshow("ImageThres", imgMedian)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break



if _name=="__main_":
    app.run(debug=True)
```

GITHUB & PROJECT VIDEO DEMO LINK

GITHUB LINK:

https://github.com/naanmudhalvan-SI/PBL-NT-GP--20776-1682831745

PROJECT VIDEO LINK:

https://www.dropbox.com/s/jbcbq9oefeca86j/VID-20230522-WA0024%5B2%5D.mp4?dl=0

https://www.dropbox.com/s/zqz63n23d27lli7/VID-20230522-WA0023%5B2%5D.mp4?dl=0

REFERENCES

1) "Intelligent Car Parking System Using OpenCV and Arduino" by S. K. Bharti and P. K. Gupta (International Journal of Engineering Research & Technology, 2019): This research paper discusses the implementation of an intelligent car parking system using OpenCV for vehicle detection and Arduino for controlling the parking process.

2) "Smart Car Parking System using OpenCV" by M. Kannan, M. Dhanasekaran, and K. Sudharsan (International Journal of Science and Research, 2018): The paper presents a smart car parking system that utilizes OpenCV for vehicle detection and counting to optimize parking space utilization.

3) "Automatic Car Parking System Using Image Processing Techniques" by A. Gupta and N. Mittal (International Journal of Computer Applications, 2015): This research paper focuses on the automatic car parking system using OpenCV for image processing and vehicle detection to assist drivers in finding available parking spaces.

4) "Smart Parking System Using OpenCV and Arduino" by N. S. Mungara, S. S. Kannekanti, and V. P. Marupudi (International Journal of Innovative Research in Computer and Communication Engineering, 2016): The paper presents a smart parking system that employs OpenCV and Arduino for vehicle detection, tracking, and occupancy detection to optimize parking management.

5) "An Intelligent Car Parking System using OpenCV" by R. Jena and M. R. Jena (International Journal of Scientific & Engineering Research, 2015): This research paper describes an intelligent car parking system that uses OpenCV for vehicle detection and recognition to provide real-time guidance to drivers for finding available parking spaces

6) "Design and Implementation of Intelligent Car Parking Management System" by S. K. Yadav, P. R. Ahire, and M. R. Fating (2017 International Conference on Inventive Communication and Computational Technologies