PROJECT REPORT

AI CAR PARKING USING OPEN CV

Submitted By

R.VISHALI(953520104002)
V.SUBHASHINI(953520104001)
M.JOTHILAKSHMI (953520205001)
M.KAVIYA (953520205002)

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING/TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING
/
INFORMATION TECHNOLOGY

VPMM ENGINEERING COLLEGE FOR WOMEN

KRISHNANKOVIL – 626190.

(Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai)

ANNA UNIVERSITY::CHENNAI 600 025
MAY 2023

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROJECT OVERVIEW

This automated system is used to find the vacancy in parking spaces available and navigate the driver to reach the desired space using visuals and in an effective manner, thus reducing search time. This system is required for malls, multistorey parking structures, IT hubs and parking facilities. This makes sure the requirement of labour is insubstantial.

1.2 PURPOSE

This project deals with an effective way of finding empty spaces and managing the number of vehicles moving in and out in complex multi storeyed parking structures by detecting a vehicle using IR sensors and thus providing a feedback. The fully automated smart car parking system is rudimental and does not require heavy lines of code nor expensive equipment. It is a simple circuit built for the exact need of purpose.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Car parking is a major problem in urban areas in both developed and developing countries. Following the rapid incense of car ownership, many cities are suffering from lacking of car parking areas with imbalance between parking supply and demand which can be considered the initial reason for metropolis parking problems. This imbalance is partially due to ineffective land use planning and miscalculations of space requirements during first stages of planning. Shortage of parking space, high parking tariffs, and traffic congestion due to visitors in search for a parking place are only a few examples of everyday parking problems.

2.2 REFERENCES

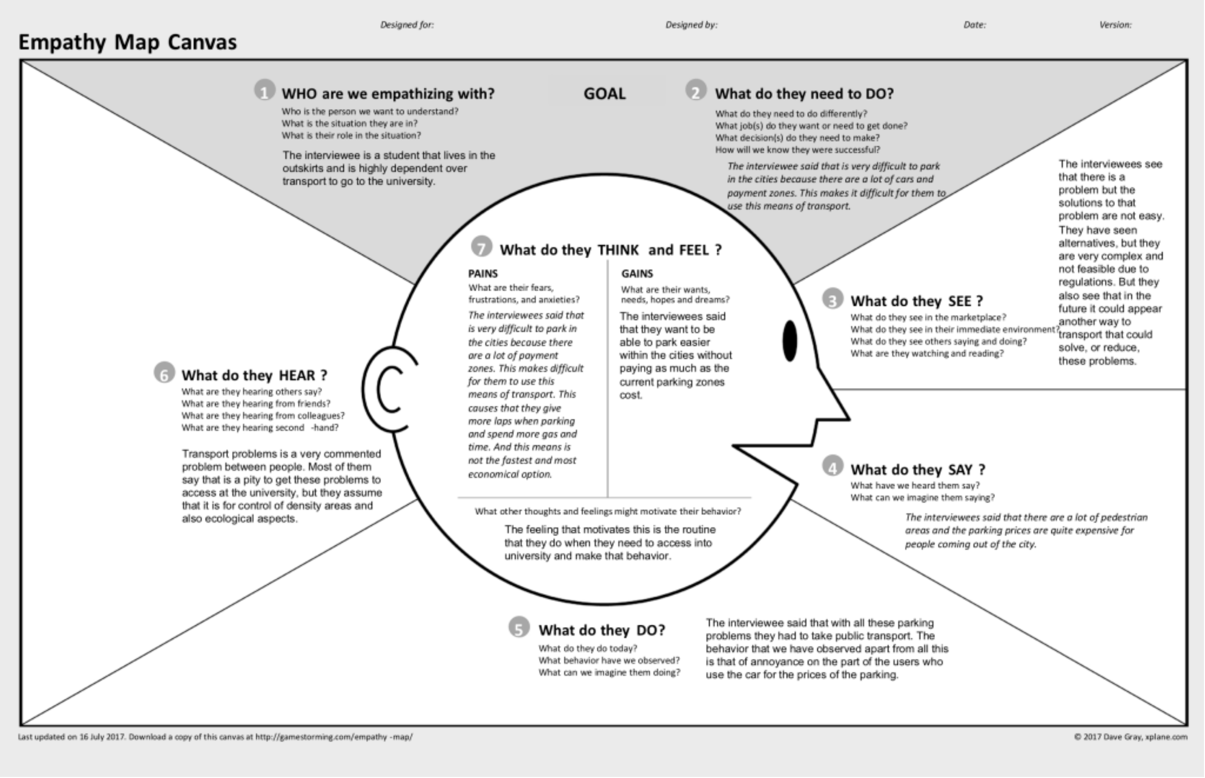
Bagula, Antoine, Lorenzo Castelli, and Marco Zennaro. On The Design of Smart Parking Networks in the Smart Cities: An Optimal Sensor Placement Model. Open Access Sensors 15 (2015): 15443-15467. Print.

2.3 PROBLEM STATEMENT DEFINITION

This imbalance is partially due to ineffective land use planning and miscalculations of space requirements during first stages of planning. Shortage of parking space, high parking tariffs, and traffic congestion due to visitors in search for a parking place are only a few examples of everyday parking problems.

3.IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



Designed for: Designed by: Date: Version:

Empathy Map Canvas

GOAL

1 **WHO are we empathizing with?**

Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

The interviewee is a student that lives in the outskirts and is highly dependent over transport to go to the university.

2 **What do they need to DO?**

What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

The interviewee said that is very difficult to park in the cities because there are a lot of cars and payment zones. This makes it difficult for them to use this means of transport.

3 **What do they SEE ?**

What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

The interviewees see that there is a problem but the solutions to that problem are not easy. They have seen alternatives, but they are very complex and not feasible due to regulations. But they also see that in the future it could appear another way to transport that could solve, or reduce, these problems.

4 **What do they SAY ?**

What have we heard them say?
What can we imagine them saying?

The interviewees said that there are a lot of pedestrian areas and the parking prices are quite expensive for people coming out of the city.

5 **What do they DO?**

What do they do today?
What behavior have we observed?
What can we imagine them doing?

The interviewee said that with all these parking problems they had to take public transport. The behavior that we have observed apart from all this is that of annoyance on the part of the users who use the car for the prices of the parking.

6 **What do they HEAR ?**

What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second -hand?

Transport problems is a very commented problem between people. Most of them say that is a pity to get these problems to access at the university, but they assume that it is for control of density areas and also ecological aspects.

7 **What do they THINK and FEEL ?**

PAINS

What are their fears, frustrations, and anxieties?

The interviewees said that is very difficult to park in the cities because there are a lot of payment zones. This makes difficult for them to use this means of transport. This causes that they give more laps when parking and spend more gas and time. And this means is not the fastest and most economical option.

GAINS

What are their wants, needs, hopes and dreams?

The interviewees said that they want to be able to park easier within the cities without paying as much as the current parking zones cost.

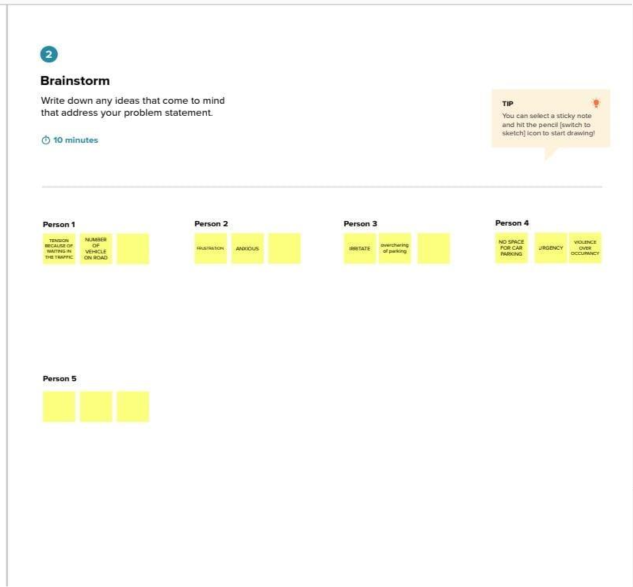
What other thoughts and feelings might motivate their behavior?

The feeling that motivates this is the routine that they do when they need to access into university and make that behavior.

Last updated on 16 July 2017. Download a copy of this canvas at http://gamestorming.com/empathy -map/

© 2017 Dave Gray, xplane.com

3.2 IDEATION AND 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!

Person 1

TENSION BECAUSE OF WAITING IN THE TRAFFIC | NUMBER OF VEHICLE ON ROAD

Person 2

FRUSTRATION | ANXIOUS

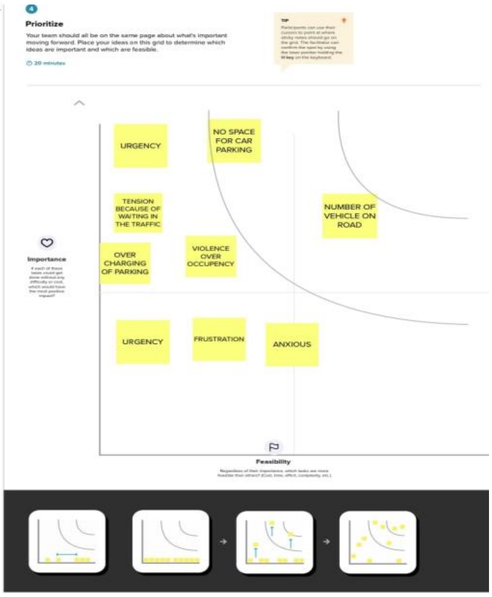
Person 3

IRRITATE | overcharing of parking

Person 4

NO SPACE FOR CAR PARKING | URGENCY | VIOLENCE OVER OCCUPANCY

Person 5



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

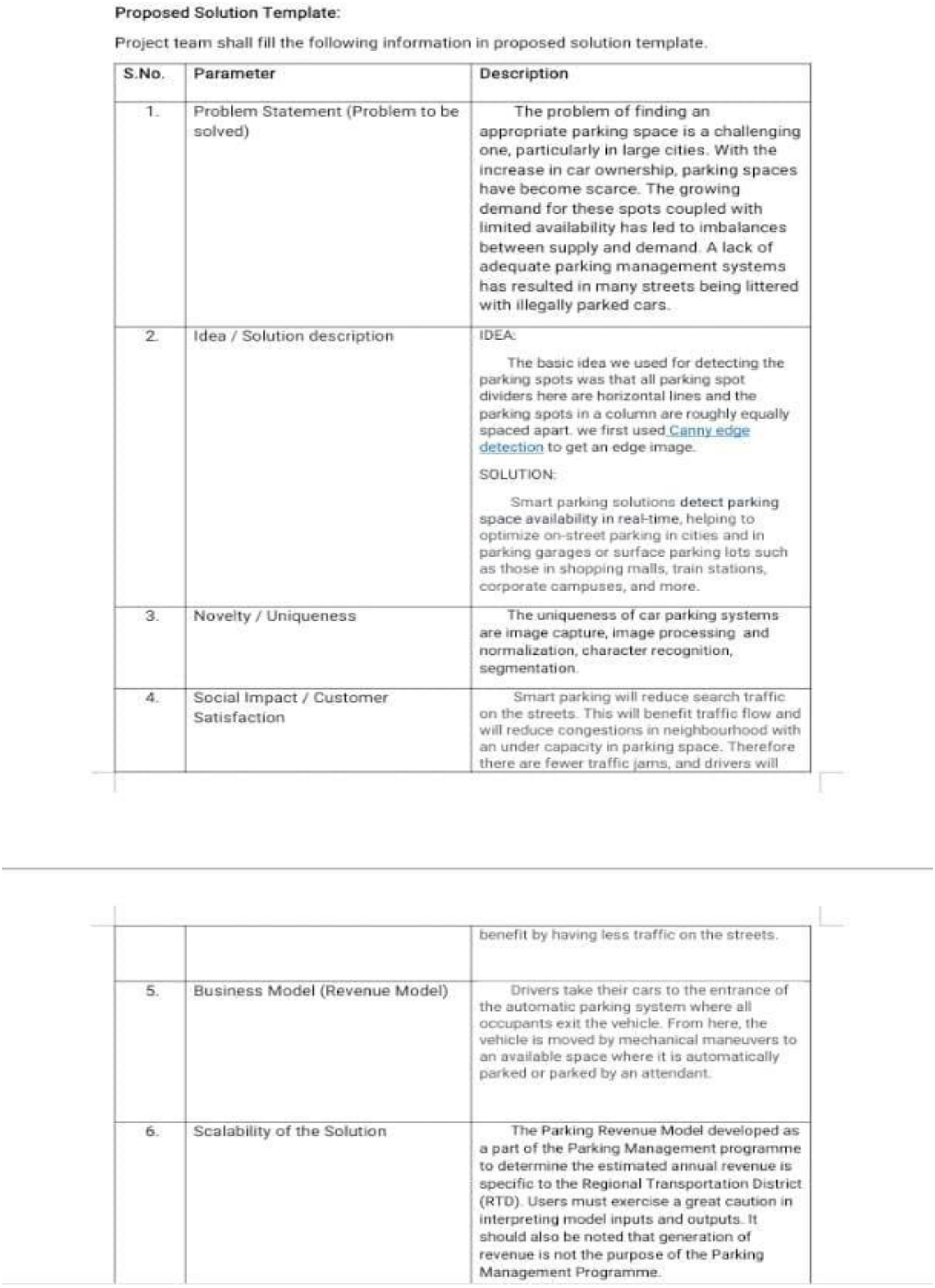
20 minutes

TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 PROPOSED SOLUTION

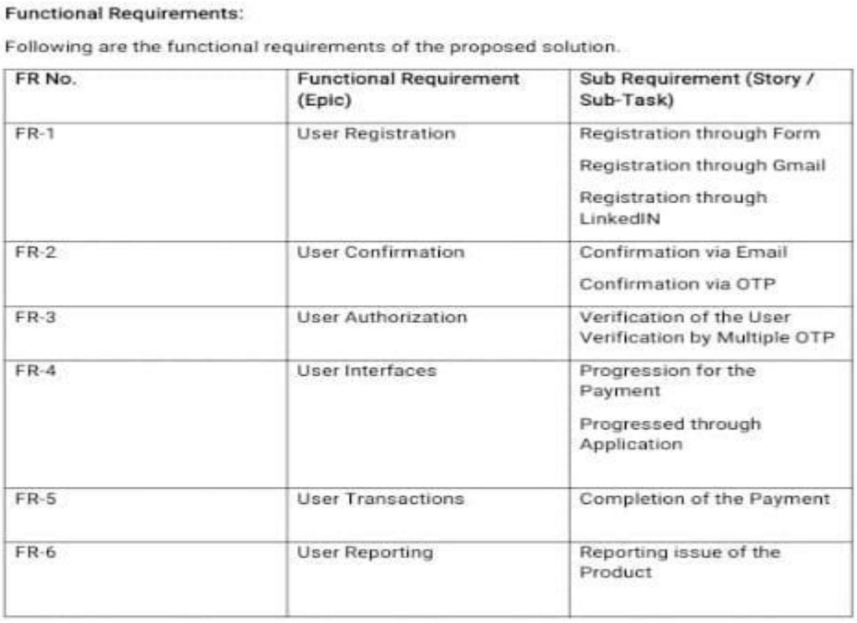


Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The problem of finding an appropriate parking space is a challenging one, particularly in large cities. With the increase in car ownership, parking spaces have become scarce. The growing demand for these spots coupled with limited availability has led to imbalances between supply and demand. A lack of adequate parking management systems has resulted in many streets being littered with illegally parked cars.
2.	Idea / Solution description	IDEA: The basic idea we used for detecting the parking spots was that all parking spot dividers here are horizontal lines and the parking spots in a column are roughly equally spaced apart. we first used Canny edge detection to get an edge image. SOLUTION: Smart parking solutions detect parking space availability in real-time, helping to optimize on-street parking in cities and in parking garages or surface parking lots such as those in shopping malls, train stations, corporate campuses, and more.
3.	Novelty / Uniqueness	The uniqueness of car parking systems are image capture, image processing and normalization, character recognition, segmentation.
4.	Social Impact / Customer Satisfaction	Smart parking will reduce search traffic on the streets. This will benefit traffic flow and will reduce congestions in neighbourhood with an under capacity in parking space. Therefore there are fewer traffic jams, and drivers will benefit by having less traffic on the streets.
5.	Business Model (Revenue Model)	Drivers take their cars to the entrance of the automatic parking system where all occupants exit the vehicle. From here, the vehicle is moved by mechanical maneuvers to an available space where it is automatically parked or parked by an attendant.
6.	Scalability of the Solution	The Parking Revenue Model developed as a part of the Parking Management programme to determine the estimated annual revenue is specific to the Regional Transportation District (RTD). Users must exercise a great caution in interpreting model inputs and outputs. It should also be noted that generation of revenue is not the purpose of the Parking Management Programme.

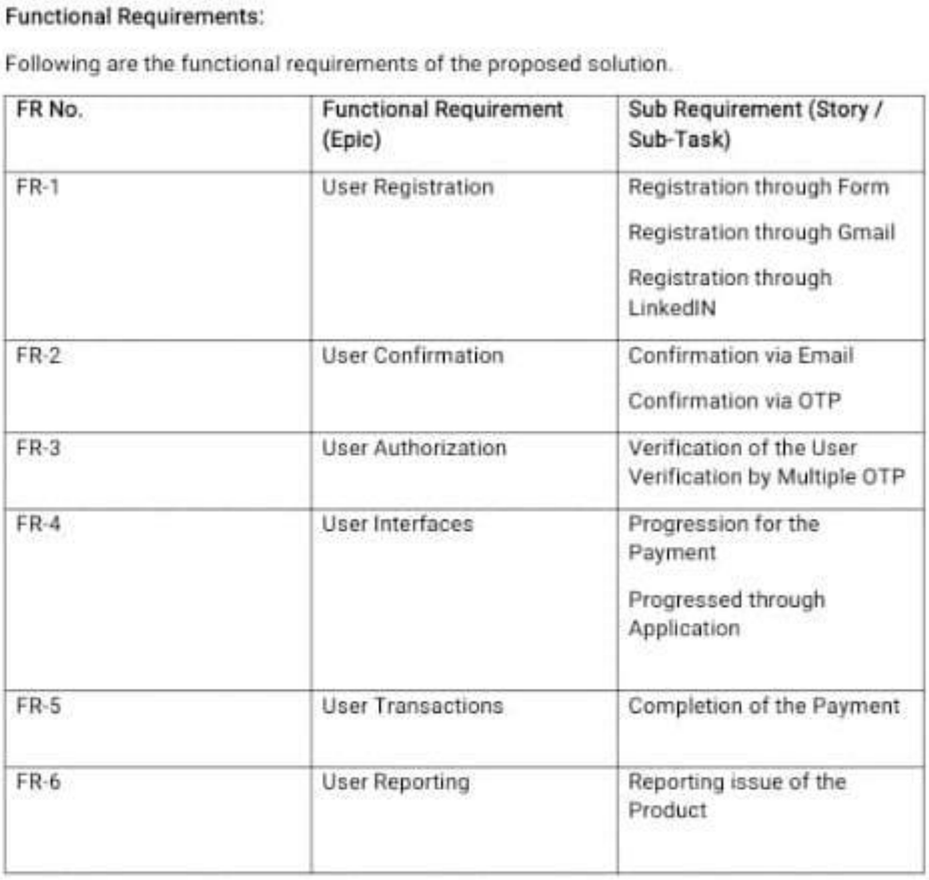
4.REQUIREMENT ANALYSIS



Functional Requirements:

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 Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Authorization	Verification of the User Verification by Multiple OTP
FR-4	User Interfaces	Progression for the Payment Progressed through Application
FR-5	User Transactions	Completion of the Payment
FR-6	User Reporting	Reporting issue of the Product



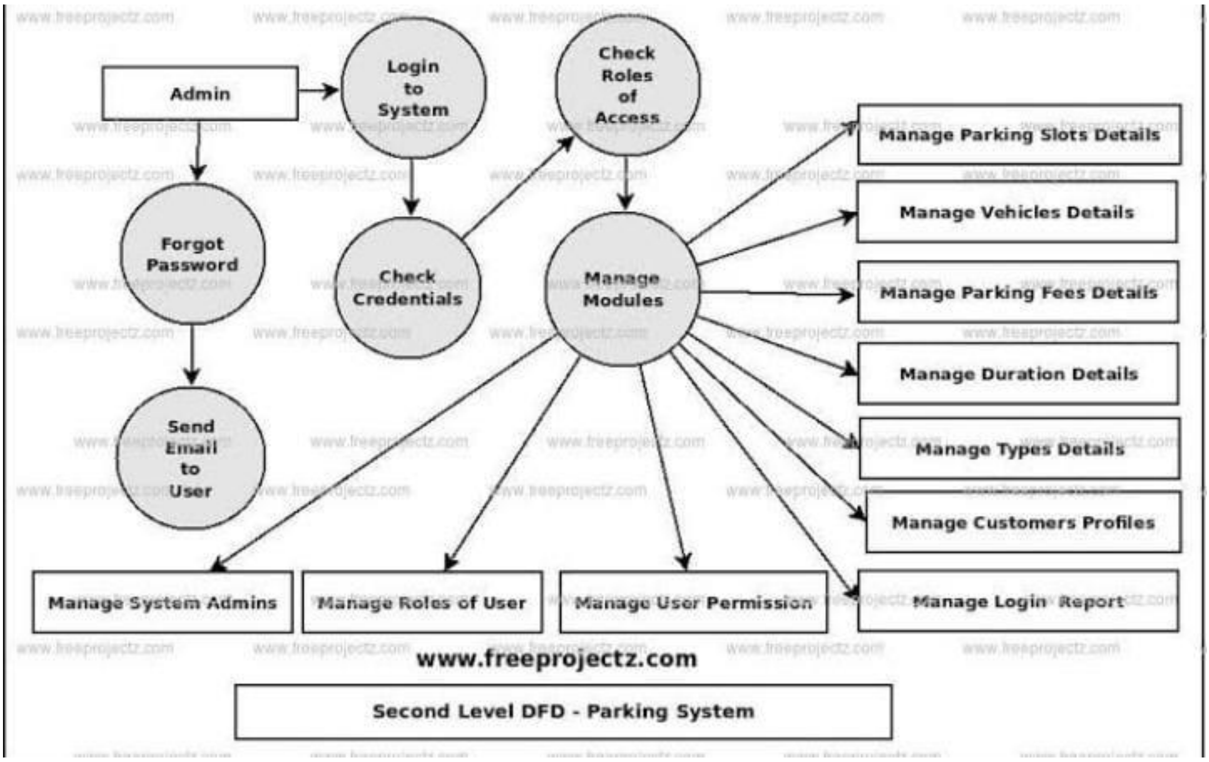
Functional Requirements:

Following are the functional requirements of the proposed solution.

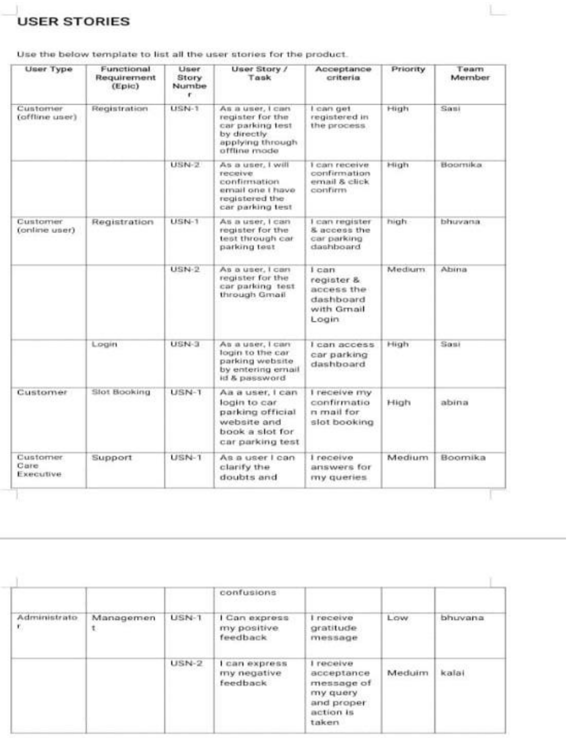
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Authorization	Verification of the User Verification by Multiple OTP
FR-4	User Interfaces	Progression for the Payment Progressed through Application
FR-5	User Transactions	Completion of the Payment
FR-6	User Reporting	Reporting issue of the Product

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 USER STORIES

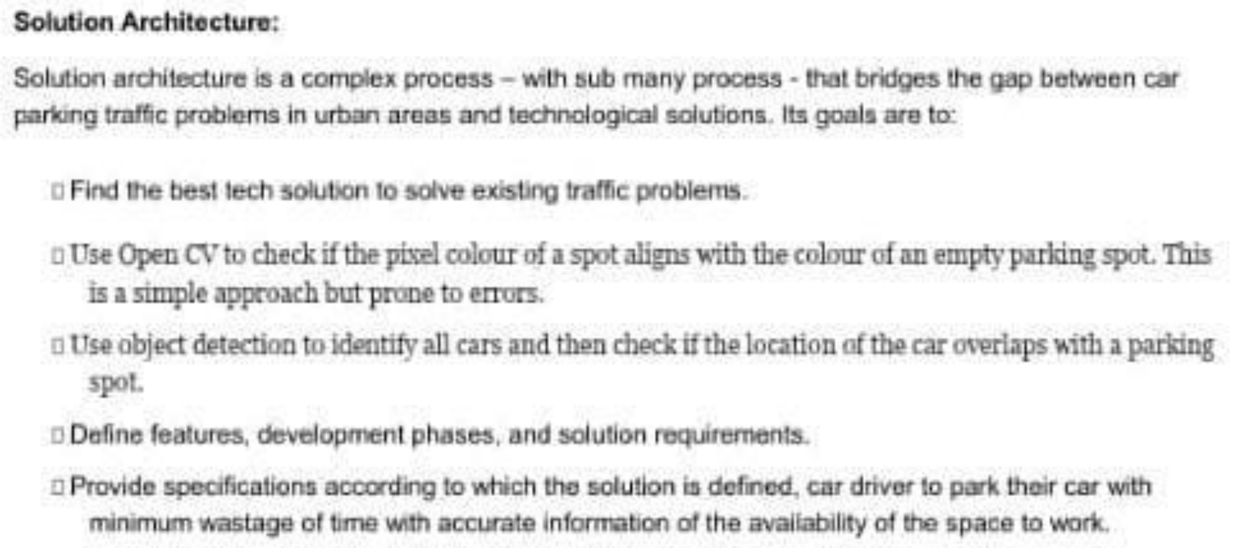


USER STORIES

Use 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 (offline user)	Registration	USN-1	As a user, I can register for the car parking test by directly applying through offline mode	I can get registered in the process	High	Sasi
		USN-2	As a user, I will receive confirmation email one I have registered the car parking test	I can receive confirmation email & click confirm	High	Boomika
Customer (online user)	Registration	USN-1	As a user, I can register for the test through car parking test	I can register & access the car parking dashboard	high	bhuvana
		USN-2	As a user, I can register for the car parking test through Gmail	I can register & access the dashboard with Gmail Login	Medium	Abina
	Login	USN-3	As a user, I can login to the car parking website by entering email id & password	I can access car parking dashboard	High	Sasi
Customer	Slot Booking	USN-1	Aa a user, I can login to car parking official website and book a slot for car parking test	I receive my confirmation mail for slot booking	High	abina
Customer Care Executive	Support	USN-1	As a user I can clarify the doubts and confusions	I receive answers for my queries	Medium	Boomika
Administrator	Management	USN-1	I Can express my positive feedback	I receive gratitude message	Low	bhuvana
		USN-2	I can express my negative feedback	I receive acceptance message of my query and proper action is taken	Meduim	kalai

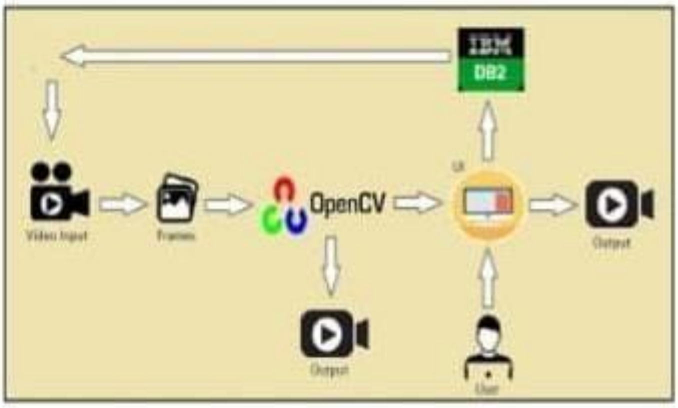
5.3 SOLUTION ARCHITECTURE



Solution Architecture:

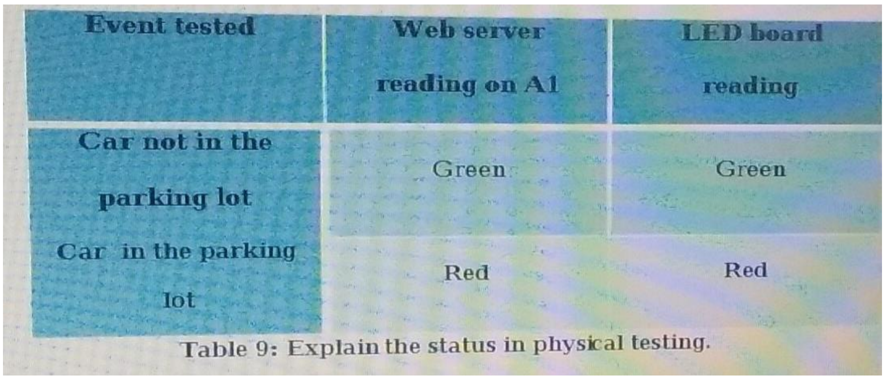
Solution architecture is a complex process – with sub many process - that bridges the gap between car parking traffic problems in urban areas and technological solutions. Its goals are to:

- □ Find the best tech solution to solve existing traffic problems.
- □ Use Open CV to check if the pixel colour of a spot aligns with the colour of an empty parking spot. This is a simple approach but prone to errors.
- □ Use object detection to identify all cars and then check if the location of the car overlaps with a parking spot.
- □ Define features, development phases, and solution requirements.
- □ Provide specifications according to which the solution is defined, car driver to park their car with minimum wastage of time with accurate information of the availability of the space to work.



6.RESULT OF THE EXPERIMENT

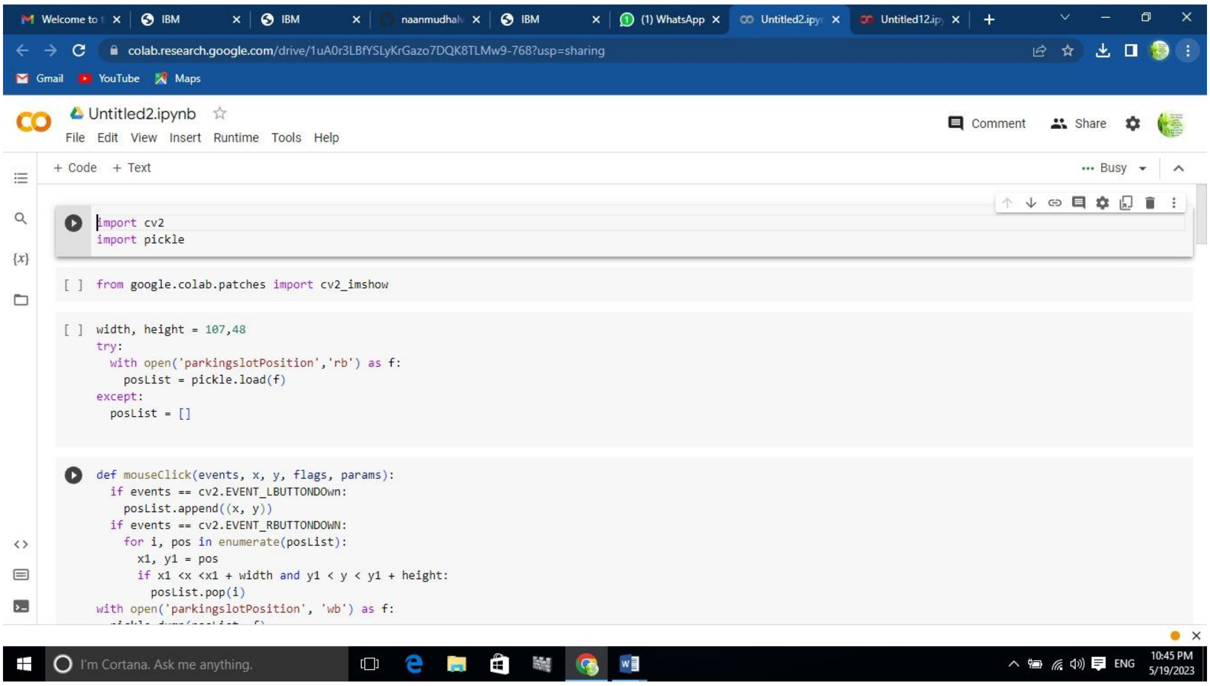
The smart car parking system is built for real-life applications.Thus, quality and consistency are mandatory. Thus, we carriedout some testing experiments using the prototype system to evaluate its dependability. The test area consisted of twoparking areas of ten lots each divided into section A and B.However; equipment were only installed in parking lot A1 where the physical testing was carried. It was not possible totest the result physically for both lots: A1 & B1, hence for thisproject we took the help of the simulation tool (XCTU) todemonstrate the occupancy of the car park A1 and B1.



Event tested	Web server reading on A1	LED board reading
Car not in the parking lot	Green	Green
Car in the parking lot	Red	Red

Table 9: Explain the status in physical testing.

7.CODING AND SOLUTIONING

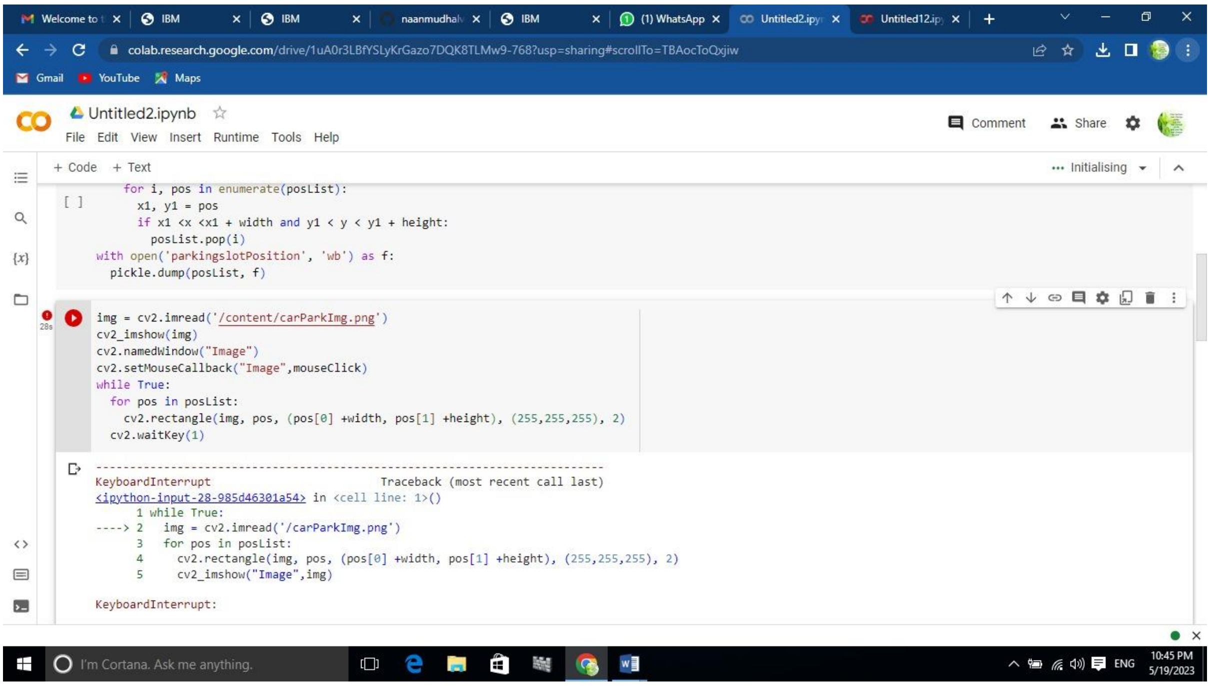


```
import cv2
import pickle
```

```
from google.colab.patches import cv2_imshow
```

```
width, height = 107,48
try:
  with open('parkingslotPosition','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('parkingslotPosition', 'wb') as f:
```

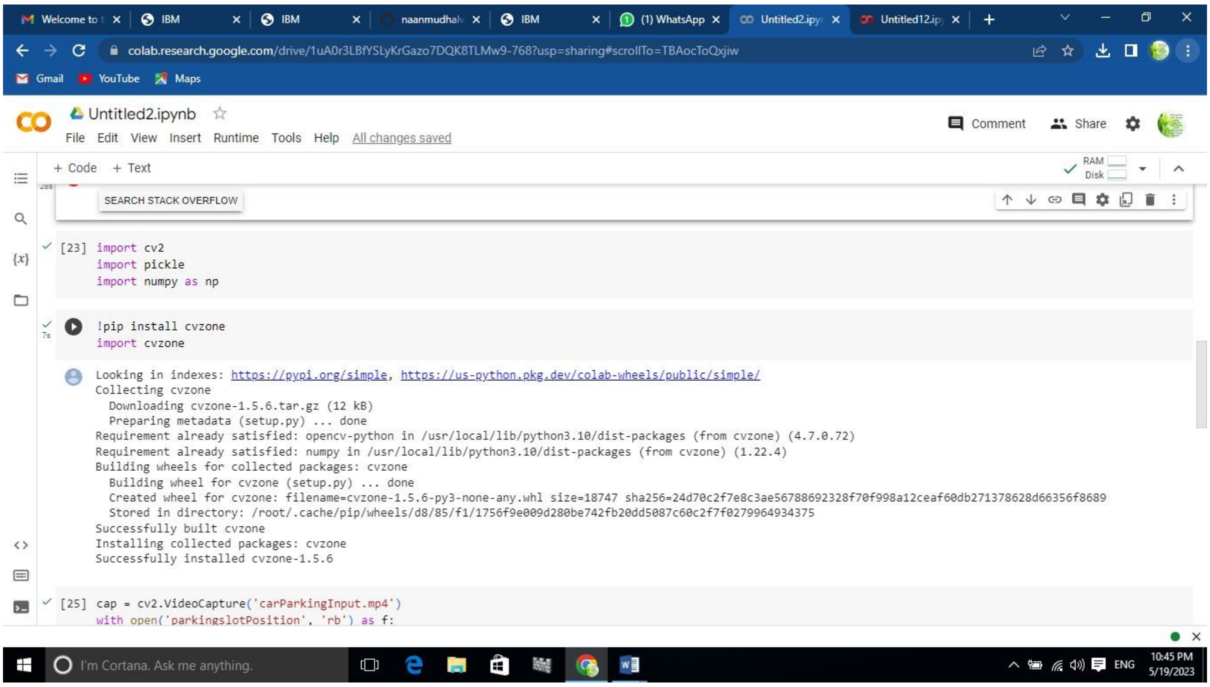


```
    for i, pos in enumerate(posList):
      x1, y1 = pos
      if x1 <x <x1 + width and y1 < y < y1 + height:
        posList.pop(i)
with open('parkingslotPosition', 'wb') as f:
  pickle.dump(posList, f)
```

```
img = cv2.imread('/content/carParkImg.png')
cv2_imshow(img)
cv2.namedWindow("Image")
cv2.setMouseCallback("Image",mouseClick)
while True:
  for pos in posList:
    cv2.rectangle(img, pos, (pos[0] +width, pos[1] +height), (255,255,255), 2)
  cv2.waitKey(1)
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-28-985d46301a54> in <cell line: 1>()
      1 while True:
----> 2   img = cv2.imread('/carParkImg.png')
      3   for pos in posList:
      4     cv2.rectangle(img, pos, (pos[0] +width, pos[1] +height), (255,255,255), 2)
      5     cv2_imshow("Image",img)

KeyboardInterrupt:
```

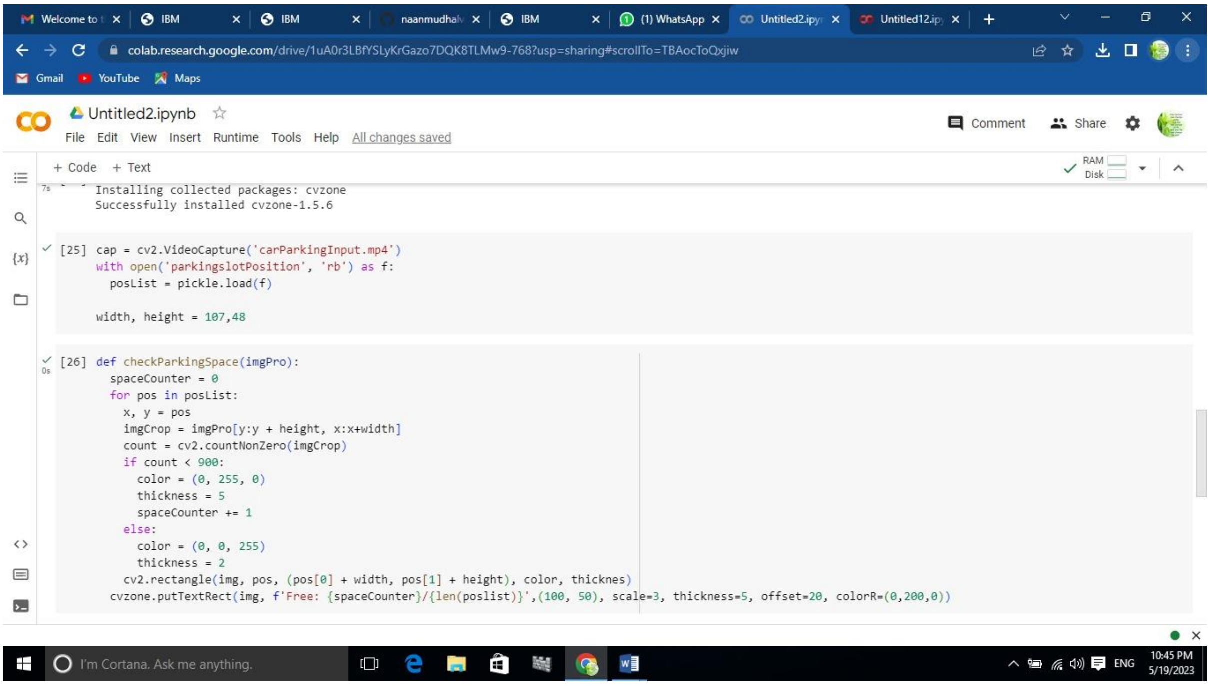


```
[23] import cv2
     import pickle
     import numpy as np
```

```
!pip install cvzone
import cvzone
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting cvzone
  Downloading cvzone-1.5.6.tar.gz (12 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-packages (from cvzone) (4.7.0.72)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from cvzone) (1.22.4)
Building wheels for collected packages: cvzone
  Building wheel for cvzone (setup.py) ... done
  Created wheel for cvzone: filename=cvzone-1.5.6-py3-none-any.whl size=18747 sha256=24d70c2f7e8c3ae56788692328f70f998a12ceaf60db271378628d66356f8689
  Stored in directory: /root/.cache/pip/wheels/d8/85/f1/1756f9e009d280be742fb20dd5087c60c2f7f0279964934375
Successfully built cvzone
Installing collected packages: cvzone
Successfully installed cvzone-1.5.6
```

```
[25] cap = cv2.VideoCapture('carParkingInput.mp4')
     with open('parkingslotPosition', 'rb') as f:
```

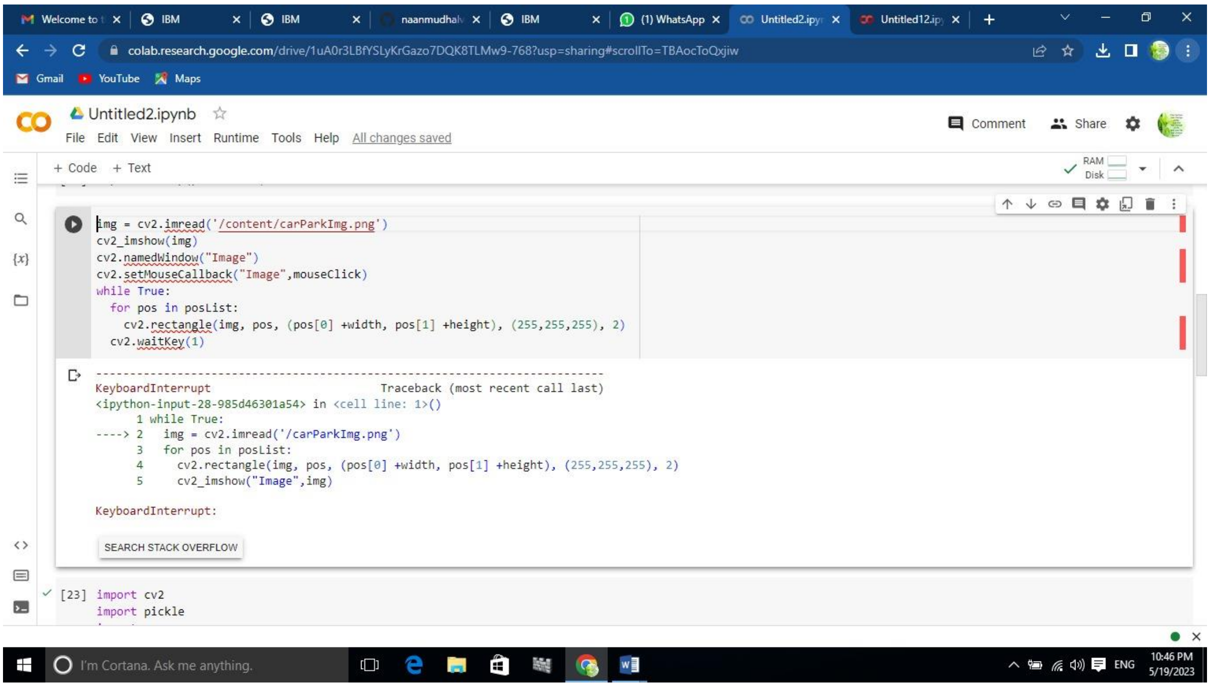


```
Installing collected packages: cvzone
Successfully installed cvzone-1.5.6
```

```
[25] cap = cv2.VideoCapture('carParkingInput.mp4')
     with open('parkingslotPosition', 'rb') as f:
       posList = pickle.load(f)

     width, height = 107,48
```

```
[26] def checkParkingSpace(imgPro):
       spaceCounter = 0
       for pos in posList:
         x, y = pos
         imgCrop = imgPro[y:y + height, x:x+width]
         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, thicknes)
       cvzone.putTextRect(img, f'Free: {spaceCounter}/{len(poslist)}',(100, 50), scale=3, thickness=5, offset=20, colorR=(0,200,0))
```

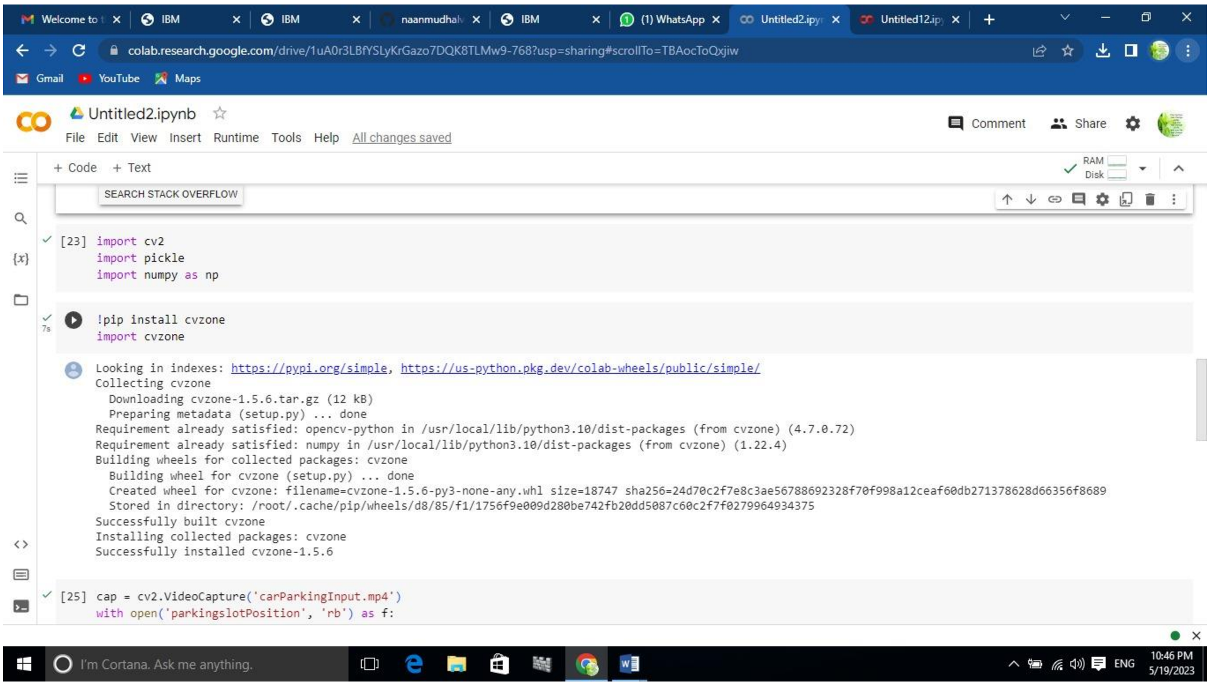


```
img = cv2.imread('/content/carParkImg.png')
cv2_imshow(img)
cv2.namedWindow("Image")
cv2.setMouseCallback("Image",mouseClick)
while True:
  for pos in posList:
    cv2.rectangle(img, pos, (pos[0] +width, pos[1] +height), (255,255,255), 2)
  cv2.waitKey(1)
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-28-985d46301a54> in <cell line: 1>()
      1 while True:
----> 2   img = cv2.imread('/carParkImg.png')
      3   for pos in posList:
      4     cv2.rectangle(img, pos, (pos[0] +width, pos[1] +height), (255,255,255), 2)
      5     cv2_imshow("Image",img)

KeyboardInterrupt:
```

```
[23] import cv2
     import pickle
```

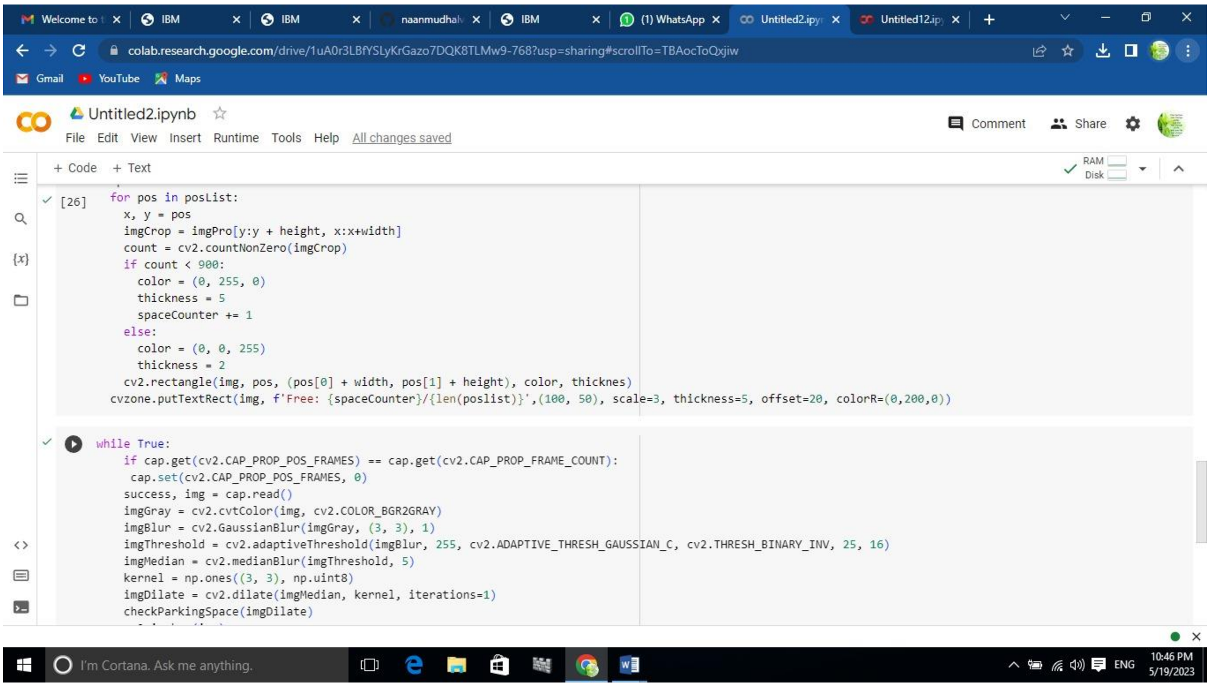


```
[23] import cv2
     import pickle
     import numpy as np
```

```
!pip install cvzone
import cvzone
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting cvzone
  Downloading cvzone-1.5.6.tar.gz (12 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-packages (from cvzone) (4.7.0.72)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from cvzone) (1.22.4)
Building wheels for collected packages: cvzone
  Building wheel for cvzone (setup.py) ... done
  Created wheel for cvzone: filename=cvzone-1.5.6-py3-none-any.whl size=18747 sha256=24d70c2f7e8c3ae56788692328f70f998a12ceaf60db271378628d66356f8689
  Stored in directory: /root/.cache/pip/wheels/d8/85/f1/1756f9e009d280be742fb20dd5087c60c2f7f0279964934375
Successfully built cvzone
Installing collected packages: cvzone
Successfully installed cvzone-1.5.6
```

```
[25] cap = cv2.VideoCapture('carParkingInput.mp4')
     with open('parkingslotPosition', 'rb') as f:
```



```
for pos in posList:
  x, y = pos
  imgCrop = imgPro[y:y + height, x:x+width]
  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, thicknes)
cvzone.putTextRect(img, f'Free: {spaceCounter}/{len(poslist)}',(100, 50), scale=3, thickness=5, offset=20, colorR=(0,200,0))
```

```
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)
```

8.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Improved parking efficiency
- Enhanced user experience
- Optimal space utilization
- Cost savings
- Increased safety and security
- Real-time monitoring and reporting

DISADVANTAGES

- Dependency on camera installation
- Limited accessibility for non-camera equpied vechicles

9.CONCLUSION

The AI enabled car parking system holds significant potential in addressing the parking challenges face in urban areas. With further development and integration into existing parking management systems. It can contribute to creating smarter and more efficient cities.

10.FUTURE SCOPE

- Scalability and integration
- Advanced parking analytic
- Mobile application and real-time updates

GITHUB

https://github.com/Vishalirv31/Vishalirv31.git

PROJECT DEMO

https://drive.google.com/file/d/1eJbOhwHBb9jVGljnS6b94DWToTM2w2uv/view?usp=sharing