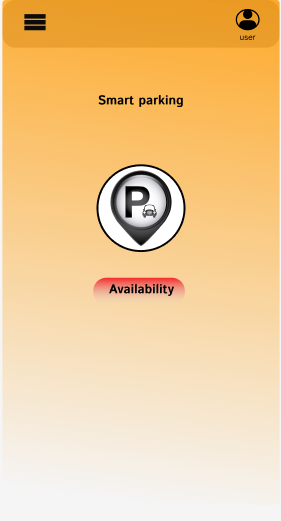
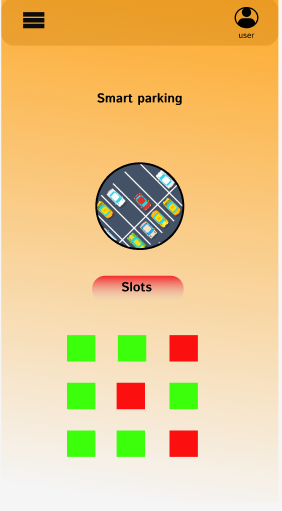
**SMART PARKING SYSTEM**

**Mobile Development App Screenshot:**

****

-> This project is about Parking Management. With growing, car parking increments with the number of vehicle users.

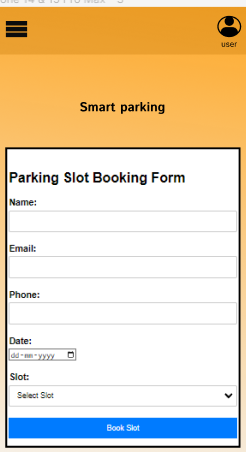
->With the increase of smart phones, internet, users prefer web based applications or smart phone applications for these solutions.

****

**-> The number of vehicles in cities has incremented dramatically dur to rapid economic development.**

**-> However, the infrastructure for accommodating these vehicles has grown relatively slow.**

**->Incrementation of the pressure on the urban transport system and solving the ‘parking difficulty’ problem have thus become hot topics recently. In this project, an intelligent parking system based on filed.**

****

**OBJECTIVES:**

-> Smart parking technologies ensure to reduce the number of cars circling around the streets for finding a parking spot.

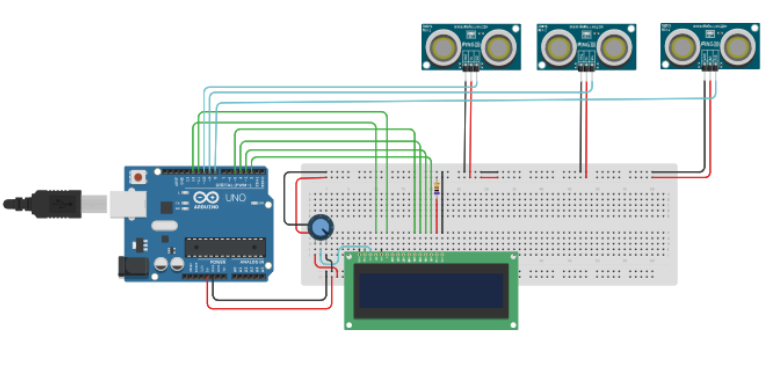
-> This ultimately smoothens the traffic flow and minimize the search traffic on streets as much as possible.

***\*Accuracy of detecting a vehicle presence/absence***

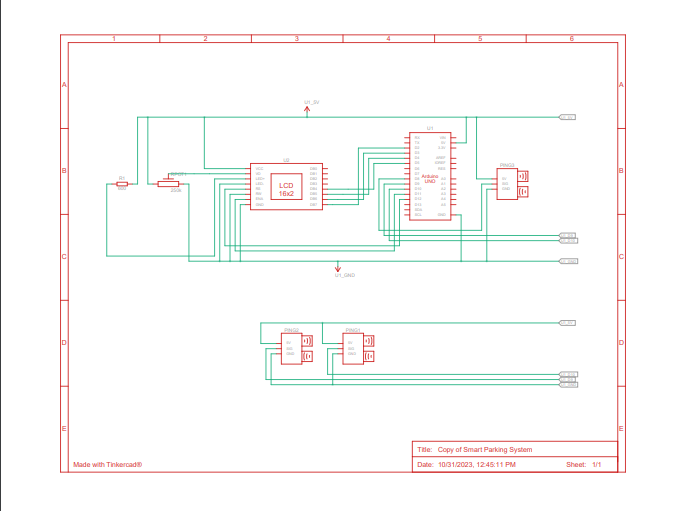
***\*Total cost of solution***

***\*Privacy concerns***

**Diagram:**

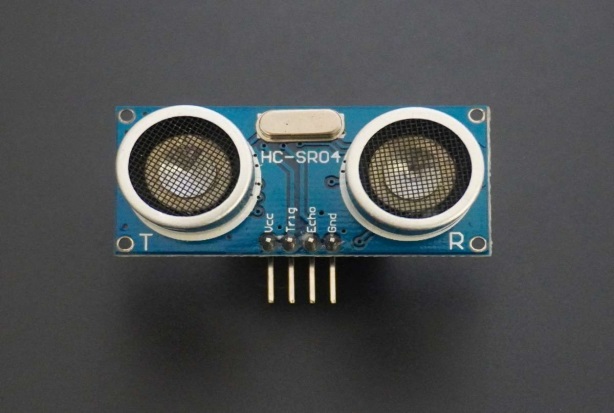
****

***Schematic Diagram:***

******

***SENSOR:***

***-> The vehicle is available for slots filled or non filled checking the sensors.***

******

**DISPLAY**:

**-> The vehicle parking slots show in output of display.**

**BENEFITS:**

* ***Use Smart Parking Protection Easily.***
* ***Monitor Your Parking Lots.***
* ***Manage Parking Lot Using Real-time Data.***
* ***Minimize Carbon Footprint.***
* ***Maximize Revenue.***
* ***Save Time, Gas, And Money.***
* ***Integrate Them Into Any Smart City System.***

***RASSPERRY PI CODE:***

***import RPi.GPIO as GPIO***

***import time***

***# Set up GPIO mode and pins for the ultrasonic sensor***

***GPIO.setmode(GPIO.BCM)***

***TRIG = 23 # The GPIO pin connected to the TRIG pin on the sensor***

***ECHO = 24 # The GPIO pin connected to the ECHO pin on the sensor***

***GPIO.setup(TRIG, GPIO.OUT)***

***GPIO.setup(ECHO, GPIO.IN)***

***def distance\_measurement():***

***# Trigger the ultrasonic sensor to send a pulse***

***GPIO.output(TRIG, True)***

***time.sleep(0.00001)***

***GPIO.output(TRIG, False)***

***# Measure the time it takes for the pulse to return***

***while GPIO.input(ECHO) == 0:***

***pulse\_start = time.time()***

***while GPIO.input(ECHO) == 1:***

***pulse\_end = time.time()***

***# Calculate the distance based on the pulse duration***

***pulse\_duration = pulse\_end - pulse\_start***

***distance = pulse\_duration \* 17150 # Speed of sound = 34300 cm/s***

***distance = round(distance, 2)***

***return distance***

***try:***

***while True:***

***distance = distance\_measurement()***

***print(f"Distance: {distance} cm")***

***# Define a threshold distance to detect if a car is parked***

***threshold\_distance = 20 # Adjust this value as needed***

***if distance < threshold\_distance:***

***print("Parking spot occupied")***

***else:***

***print("Parking spot vacant")***

***time.sleep(2) # You can adjust the polling interval***

***except KeyboardInterrupt:***

***GPIO.cleanup()***