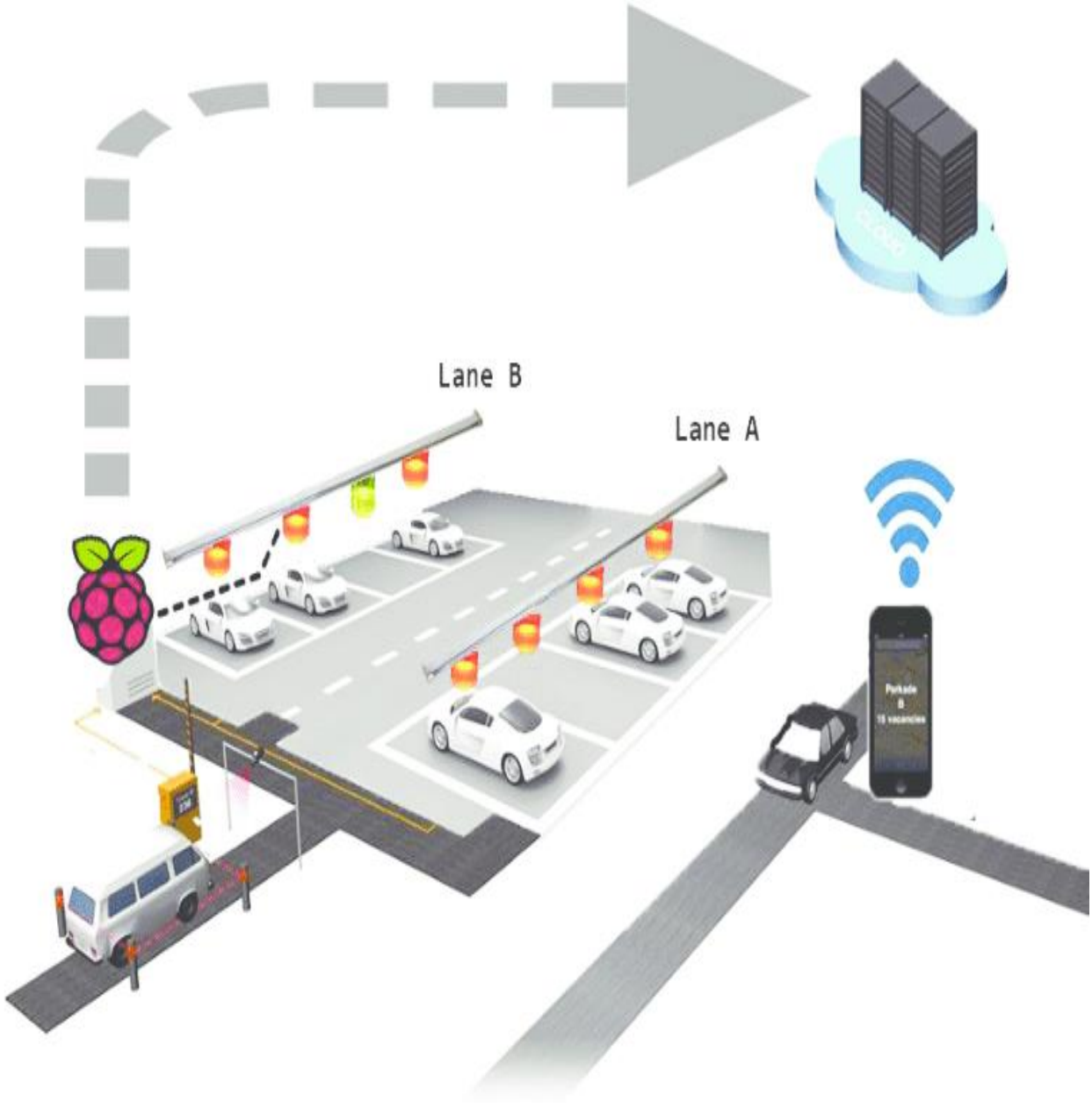
DEVELOPMENT PART 2

SMART PARKING



TEAM DETAILS

Mentor	**Mrs.M.Maheswari**
Leader	**Monika M**
Team Members	**Anupriya R** **Bhavani G** **Divyabharathi C L** **Kanimozhi S**
Problem Description	**Integrating a ultrasonic sensor in a raspberry pi and simulating in a wokwi platform and by creating a web application to display the vacant and occupied spaces**

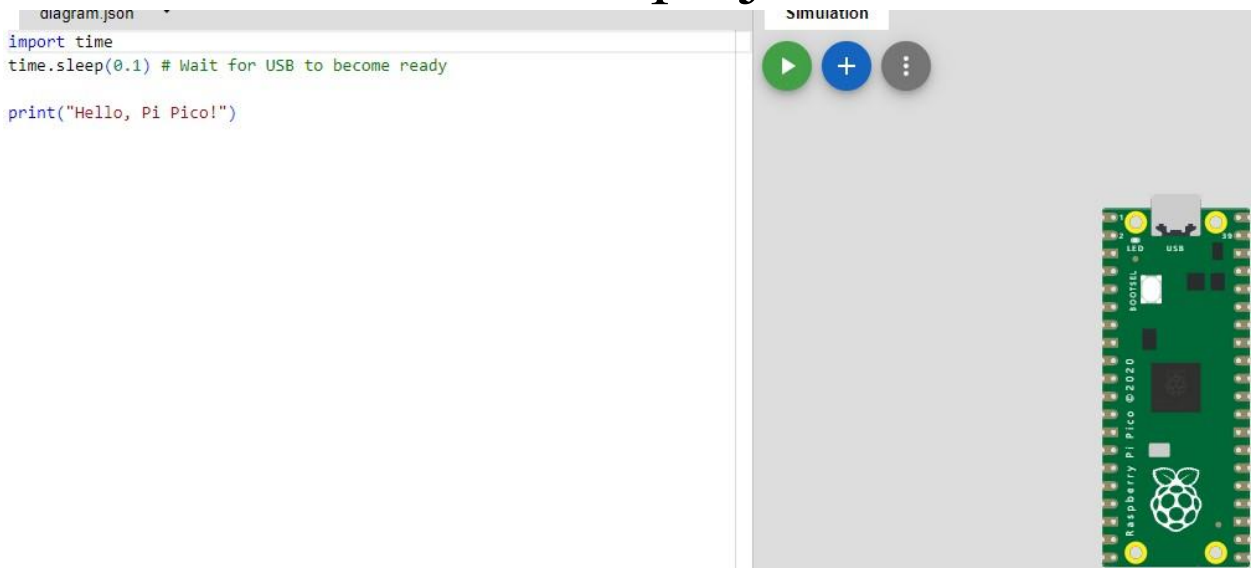
Steps For Simulating Raspberry Pi

Step 1: Access wokwi

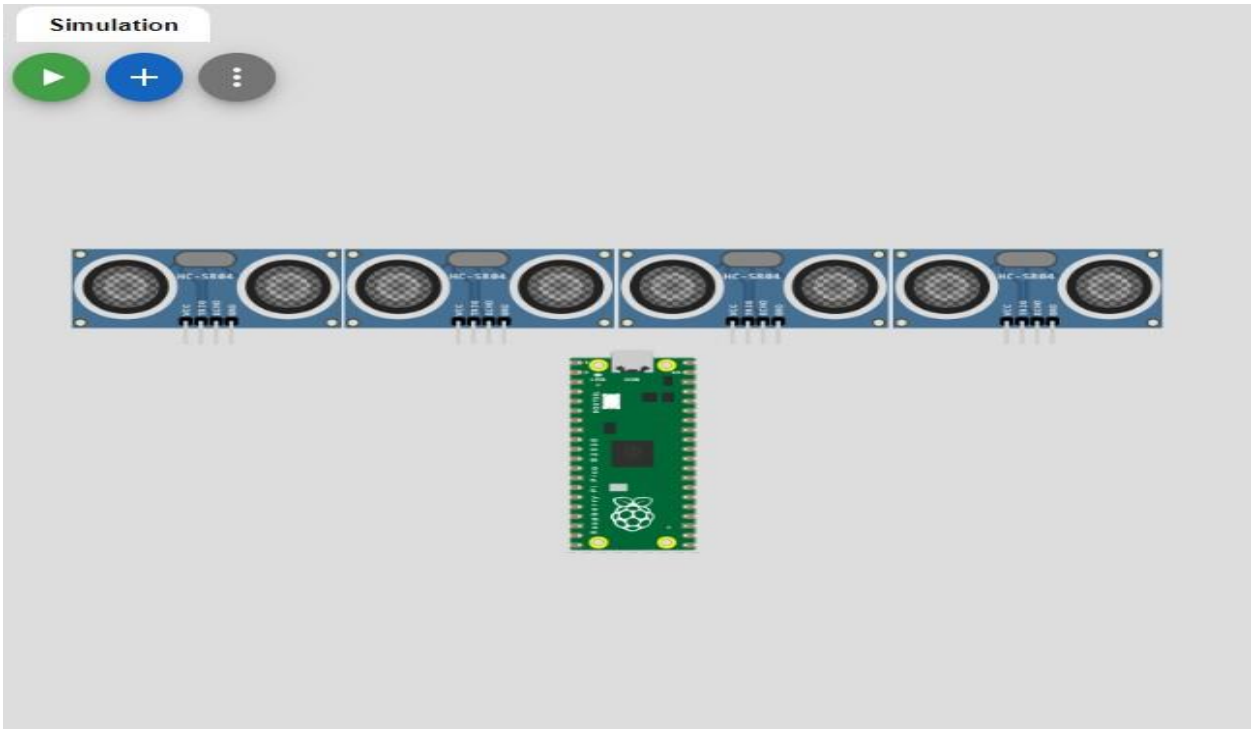
- Go to the wokwi website and choose raspberry pi project(https://wokwi.com/projects/new/pi-pico)

Step 2: Create a New Project

- Click on “create a new project”



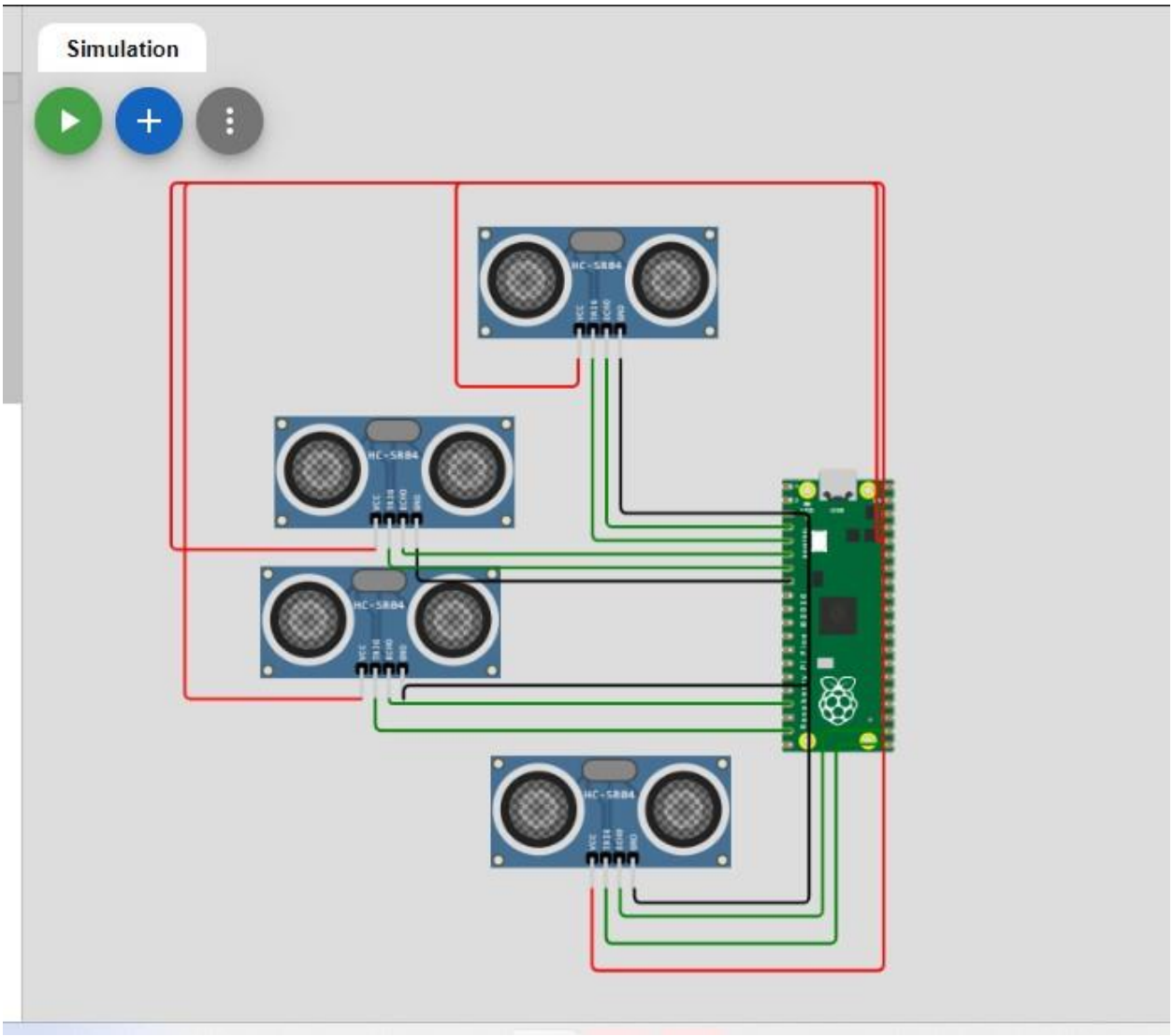
Step 3: Add Ultrasonic sensor



Step 4: Make connections to ultrasonic sensor with a Raspberry pi

- Connect ultrasonic1 VCC pin to Pico 3V3
- Connect ultrasonic1 TRIG pin to Pico GP3
- Connect ultrasonic1 ECHO pin to Pico GP2
- Connect ultrasonic1 GND pin to Pico GND1
- Connect ultrasonic2 VCC pin to Pico 3V3
- Connect ultrasonic2 TRIG pin to Pico GP14
- Connect ultrasonic2 ECHO pin to Pico GP13
- Connect ultrasonic2 GND pin to Pico GND1
- Connect ultrasonic3 VCC pin to Pico 3V3
- Connect ultrasonic3 TRIG pin to Pico GP5
- Connect ultrasonic3 ECHO pin to Pico GP4
- Connect ultrasonic3 GND pin to Pico GND2
- Connect ultrasonic4 VCC pin to Pico 3V3
- Connect ultrasonic4 TRIG pin to Pico GP16
- Connect ultrasonic4 ECHO pin to Pico GP17
- Connect ultrasonic4 GND pin to Pico GND1

CIRCUIT DIAGRAM:



Step 5: code

- Click on the “code” tab in wokwi to access the code editor

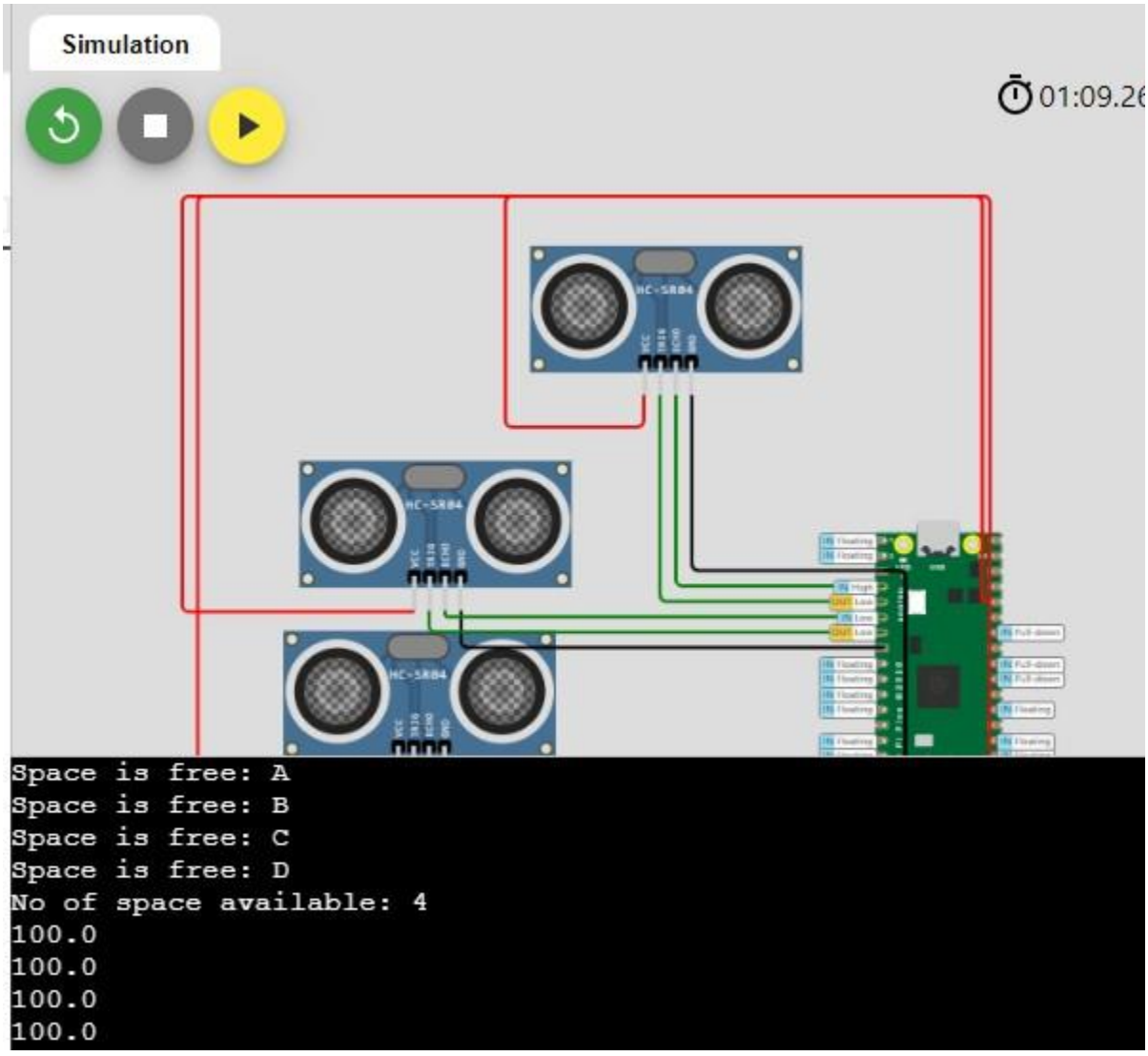
```
from ultra import DistanceSensor
from time import sleep
dsa = DistanceSensor(echo=2, trigger=3)
dsb = DistanceSensor(echo=4, trigger=5)
dsc = DistanceSensor(echo=13, trigger=14)
dsd = DistanceSensor(echo=17, trigger=16)
while True:
distance_a = dsa.distance * 100
distance_b = dsb.distance * 100
distance_c = dsc.distance * 100
distance_d = dsd.distance * 100
a = float(distance_a)
b = float(distance_b)
c = float(distance_c)
d = float(distance_d)  # Convert to a floating-point number
print(a)
print(b)
print(c)
print(d)
A="A"
B="B"
C="C"
D="D"
no=0
def parking(distance, n,slot):
if distance < 30:
# Code to execute if the distance is less than 30
print("Space is not free:"+slot)
if(n==0):
n=0
else:
n=n-1
else:
# Code to execute if the distance is not less than 30
```

```
print("Space is free: "+slot)
if(n==4):
n=4
else:
n=n+1
return n
no=parking(a,no,A)
no=parking(b,no,B)
no=parking(c,no,C)
no=parking(d,no,D)
no=no
print("No of space available:",no)
sleep(0.1)
```

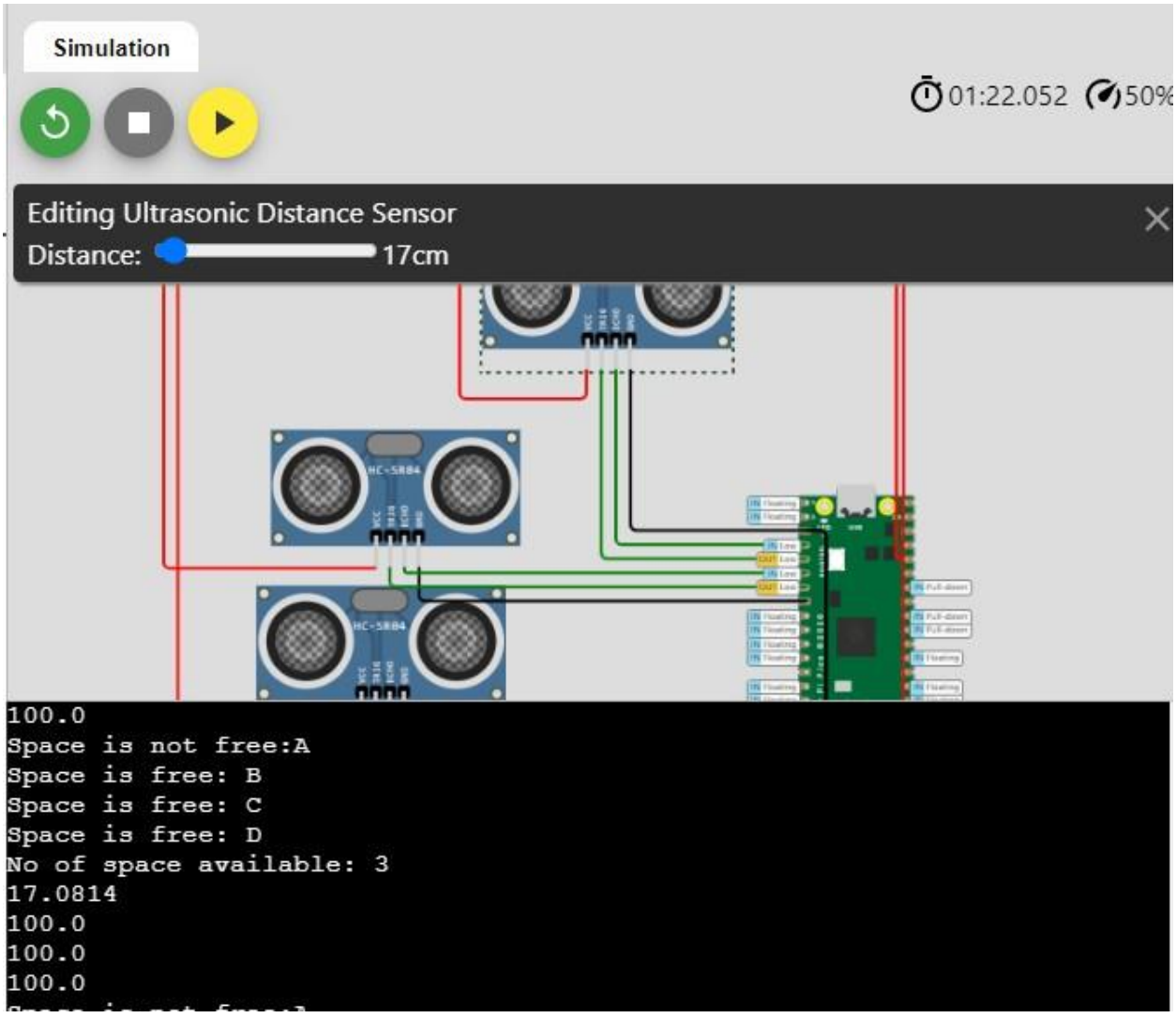
Step 6: Simulation

- Click on the “simulate” button to start the simulation
- Based on the distance echoed by the ultrasonic sensor ,the raspberry pico will show the available parking slot is either free or vacant

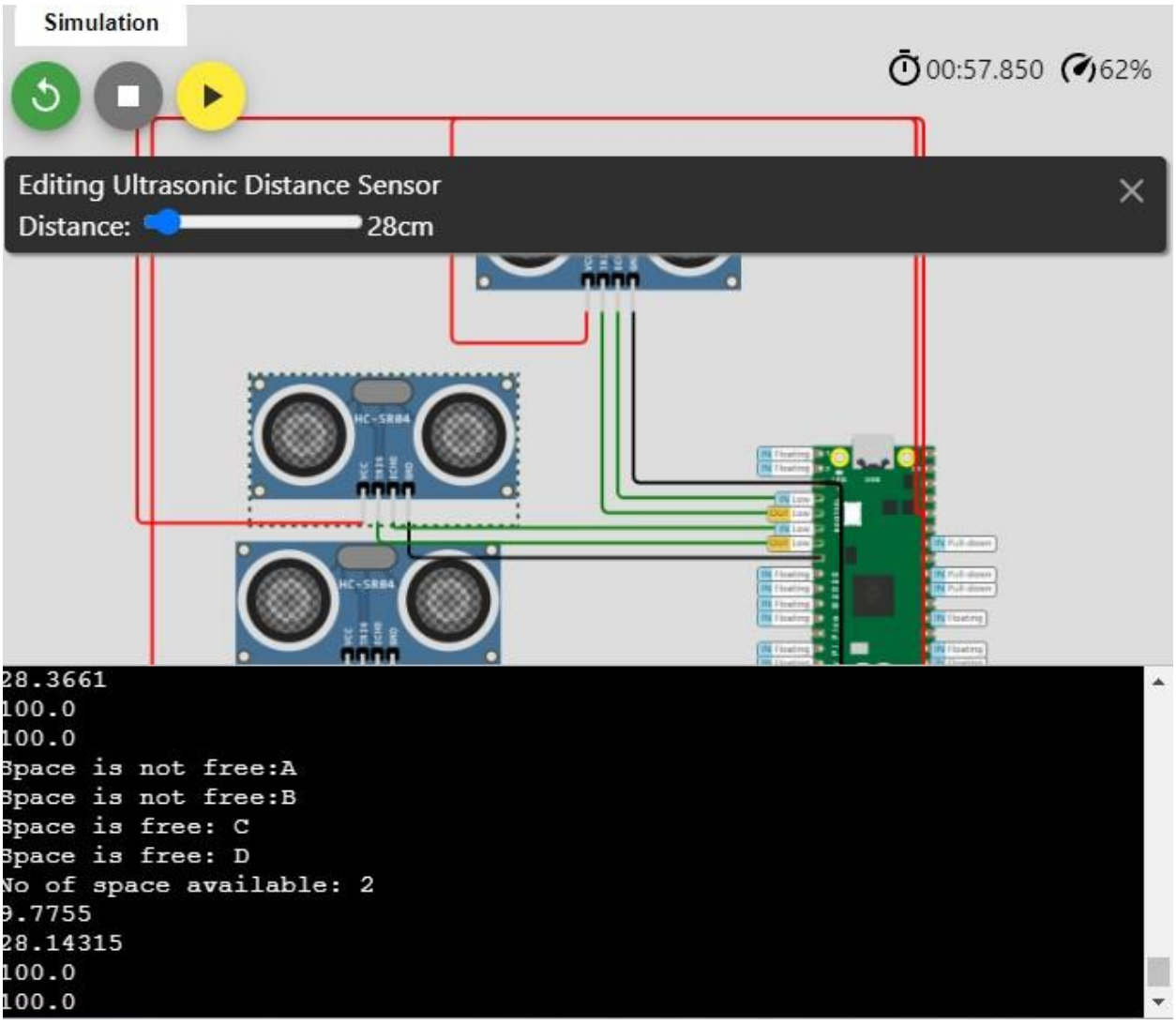
Step 1:when the distance of all the ultrasonic sensor is above 30 cm then it will display all the free spaces along with its echoed distance



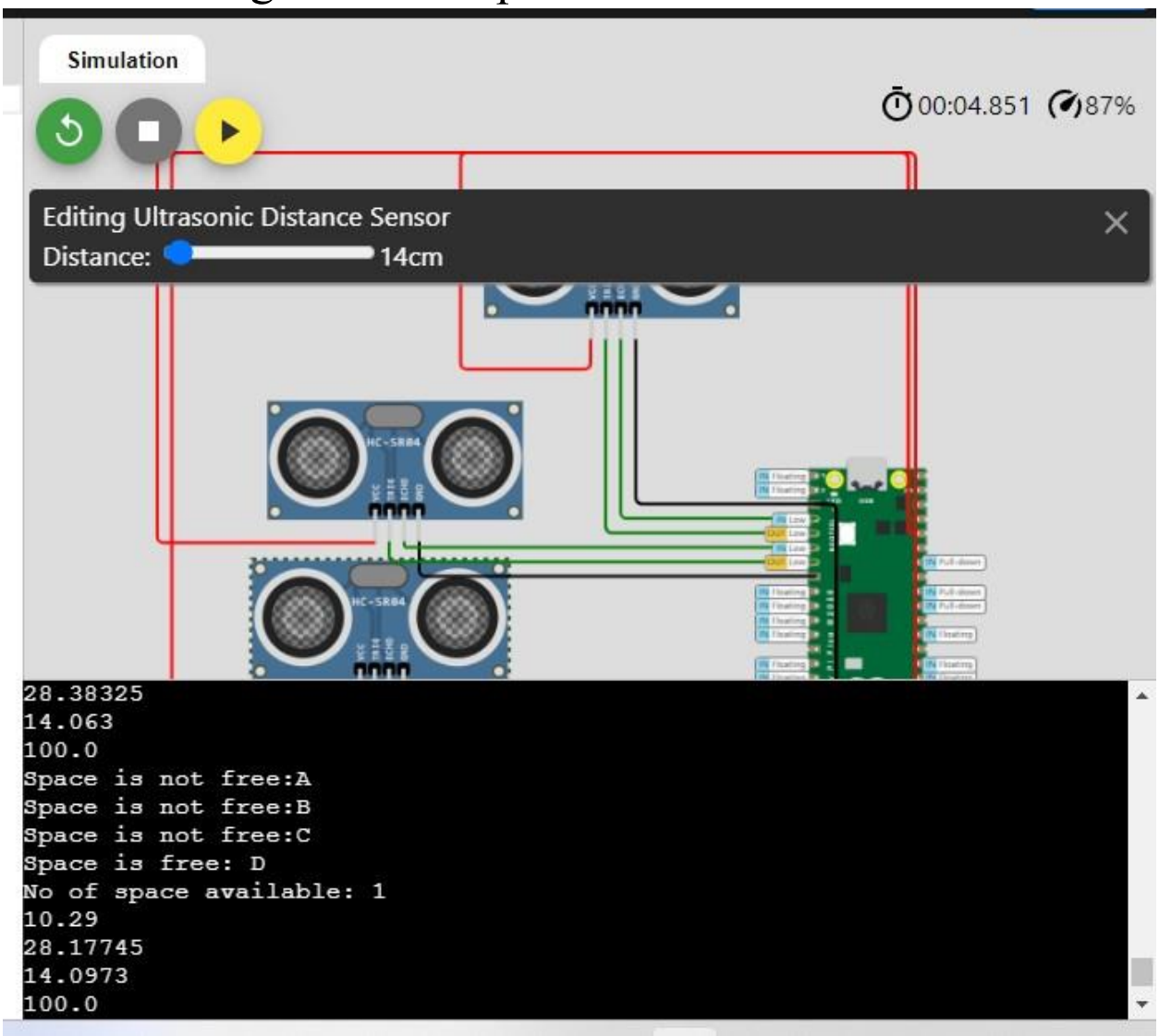
Step 2: when the distance of the ultrasonic1 sensor is below 30 cm (i.e.,17cm) then it will display that the slot A is occupied along with the echoed distance of slot A and it will display the remaining available spaces as free



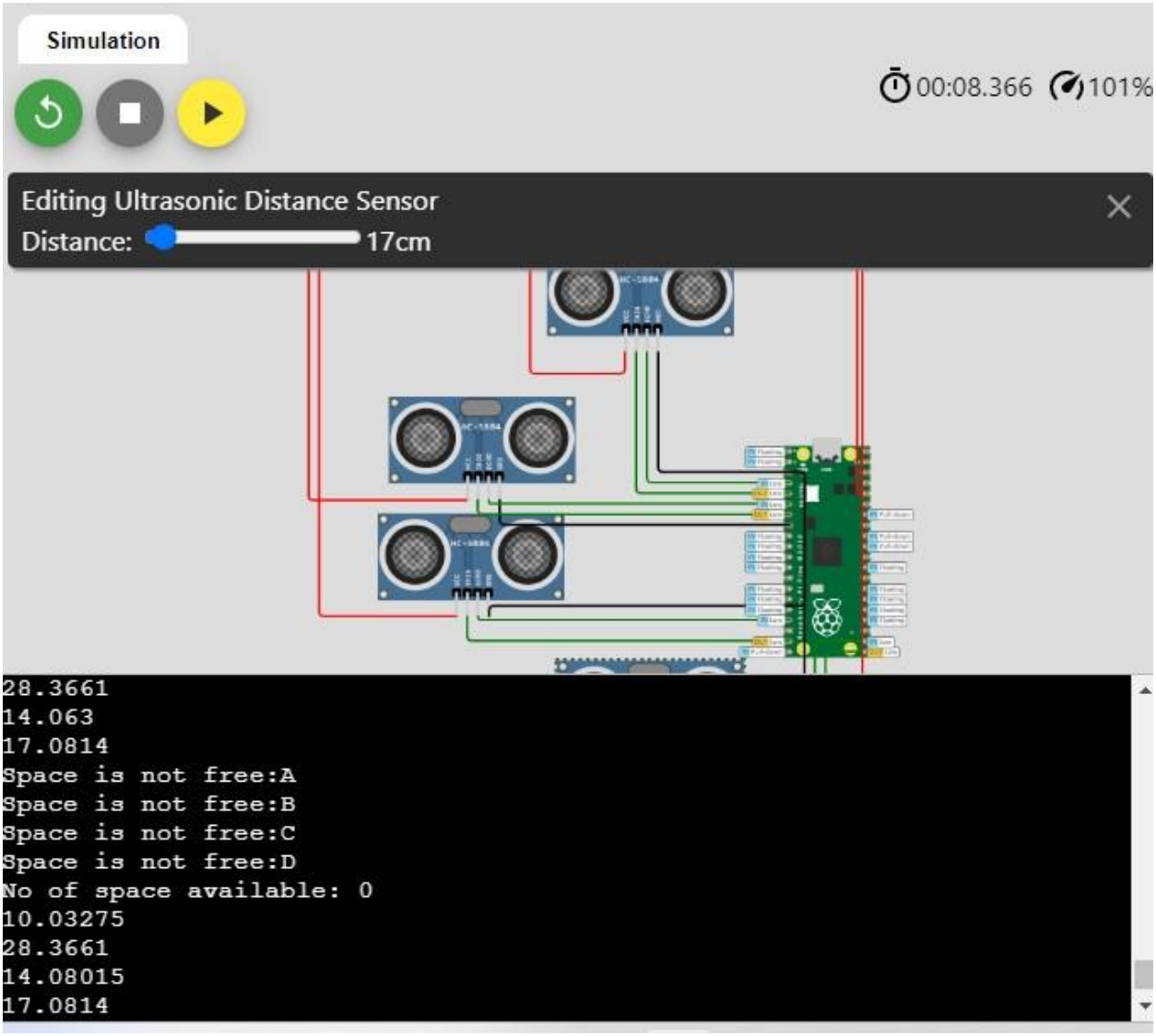
Step 3: when the distance of the ultrasonic1 and ultrasonic2 sensor is below 30 cm (i.e.,17cm and 28cm) then it will display that the slot A and B is occupied along with the echoed distance of slot A and B and it will display the remaining available spaces as free



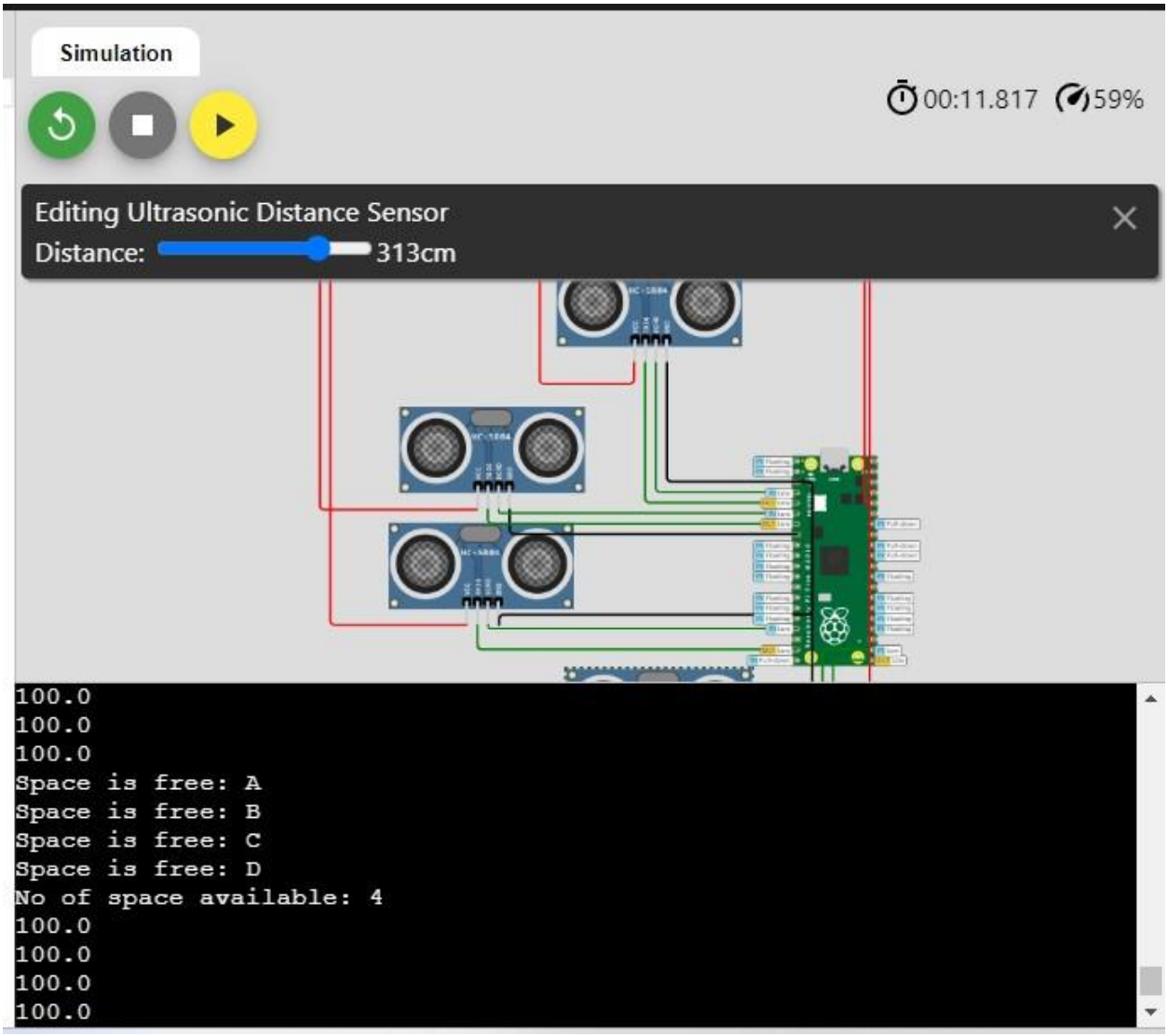
Step 4: when the distance of the ultrasonic1 ,ultrasonic2 and ultrasonic3 sensor is below 30 cm (i.e.,17cm,28cm and 14cm) then it will display that the slot A , B and C is occupied along with the echoed distance of slot A , B and C and it will display the remaining available spaces as free



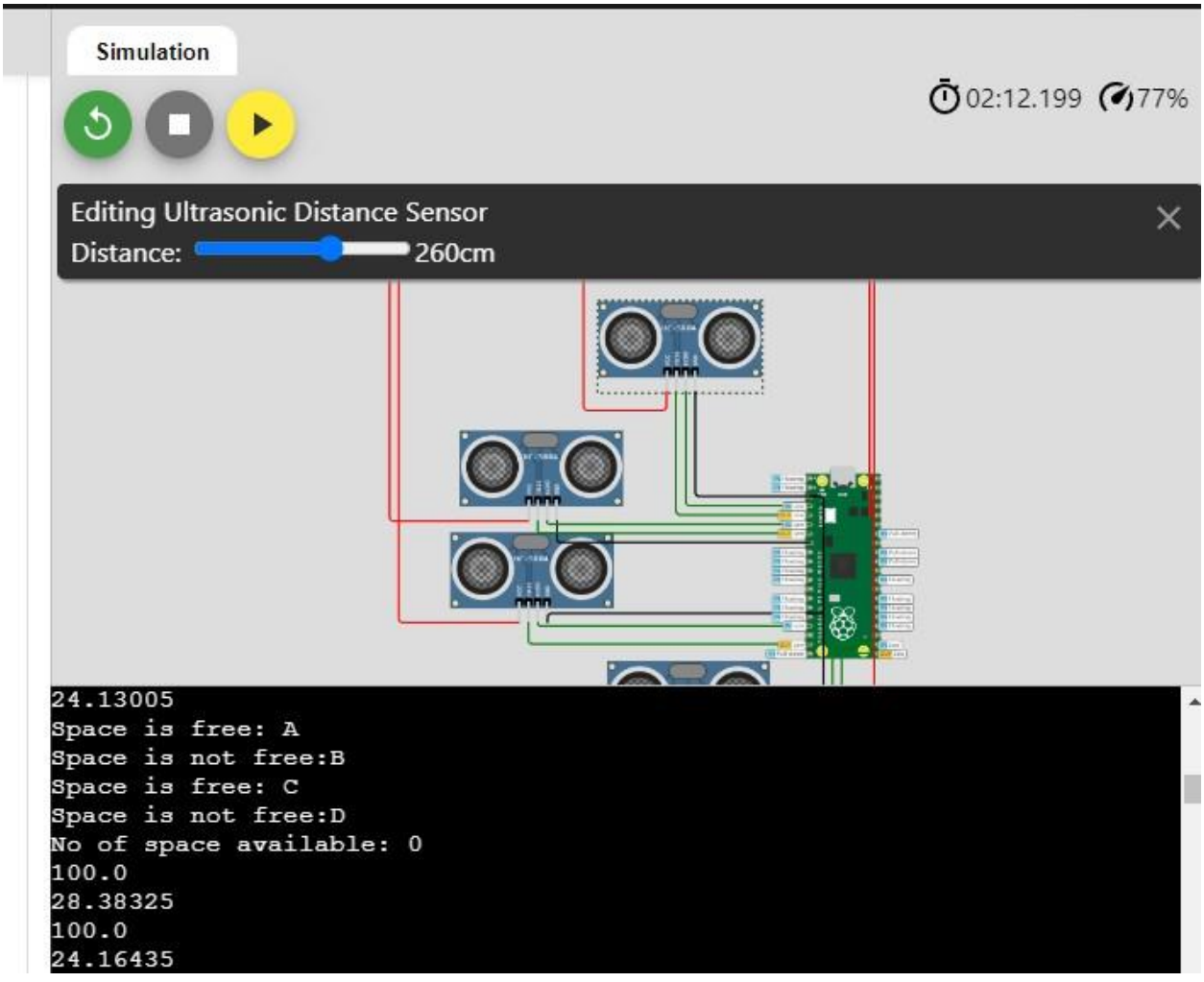
Step 5: when the distance of the ultrasonic1 ,ultrasonic2 ,ultrasonic3 and ultrasonic4 sensor is below 30 cm (i.e.,17cm,28cm ,14cm and 2cm) then it will display that the slot A , B , C and D is occupied along with the echoed distance of slot A , B , C and D and it will display there is no available spaces. All slots are occupied.



Step 6: when the distance of the ultrasonic1 ,ultrasonic2 ,ultrasonic3 and ultrasonic4 sensor is above 30 cm (i.e.,157cm,321cm,245cm and 298cm) then it will display that the slot A , B , C and D is free along with the echoed distance of slot A , B , C and D and it will display there is available spaces. All slots are free.



Step 7: when the distance of the ultrasonic1 and ultrasonic3 sensor is above 30 cm (i.e.,157cm,321cm) then it will display that the slot A and C is free along with the echoed distance of slot A and C. when the distance of ultrasonic2 and ultrasonic4 sensor is below 30cm (i.e.,14cm and 12cm)then it will display that the slot B and D is Occupied along with the echoed distance of Slot B and D.



WEB APPLICATION FOR SMART PARKING

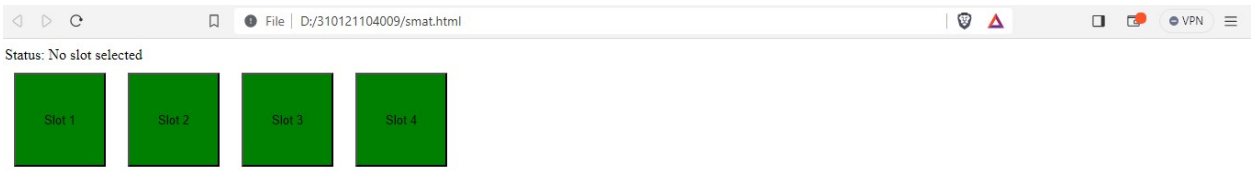
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
.parking-slot {
width: 100px;
height: 100px;
margin: 10px;
cursor: pointer;
}
.empty {
font-size:72;
background-color: green;
}
```

```
.full {
font-size:72;
background-color: red;
}
body
{
font-size:72;
font-family:Times New Roman;
}
</style>
<title>Smart Parking System</title>
</head>
<body>
<div id="status-message">Status: No slot selected</div>
<button class="parking-slot empty"
onclick="toggleSlot(this)">Slot 1</button>
<button class="parking-slot empty"
onclick="toggleSlot(this)">Slot 2</button>
<button class="parking-slot empty"
onclick="toggleSlot(this)">Slot 3</button>
```

```
<button class="parking-slot empty"
onclick="toggleSlot(this)">Slot 4</button>

<script>

function toggleSlot(slot) {

if (slot.classList.contains('empty')) {

slot.classList.remove('empty');

slot.classList.add('full');

document.getElementById('status-message').innerText =
`Status:  ${slot.innerText} is full`;

} else {

slot.classList.remove('full');

slot.classList.add('empty');

document.getElementById('status-message').innerText =
`Status:  ${slot.innerText} is empty`;

}

}

</script>

</body>

</html>
```

Link:///D:/310121104009/smat.html

Step 1: Initially no slots are selected all the slots are empty. So it shows the status No slot Selected



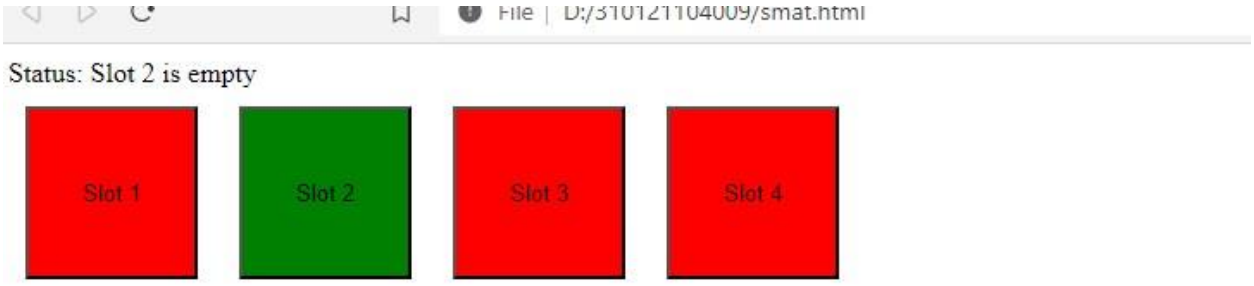
Step 2: when slot 1 is clicked it will show the status slot 1 is full which means the slot is occupied and indicates in red colour

Step 3: when slot 2 is clicked it will show the status slot 2 is full which means the slot is occupied and indicates in red colour and other slots are free

Step 4: when slot 3 is clicked it will show the status slot 3 is full which means the slot is occupied and indicates in red colour and other slots are free

Step 5: when slot 4 is clicked it will show the status slot 4 is full which means the slot is occupied and indicates in red colour and there is no more slots left . All are occupied

Step 6: when slot 2 is clicked it will show the status slot 2 is empty which means the slot is free and indicates in green colour whereas the other slots are occupied.



Step 7: when slot 4 is clicked it will show the status slot 4 is empty which means the slot is free and indicates in green colour whereas the other slots are occupied.