IoT PHASE 3

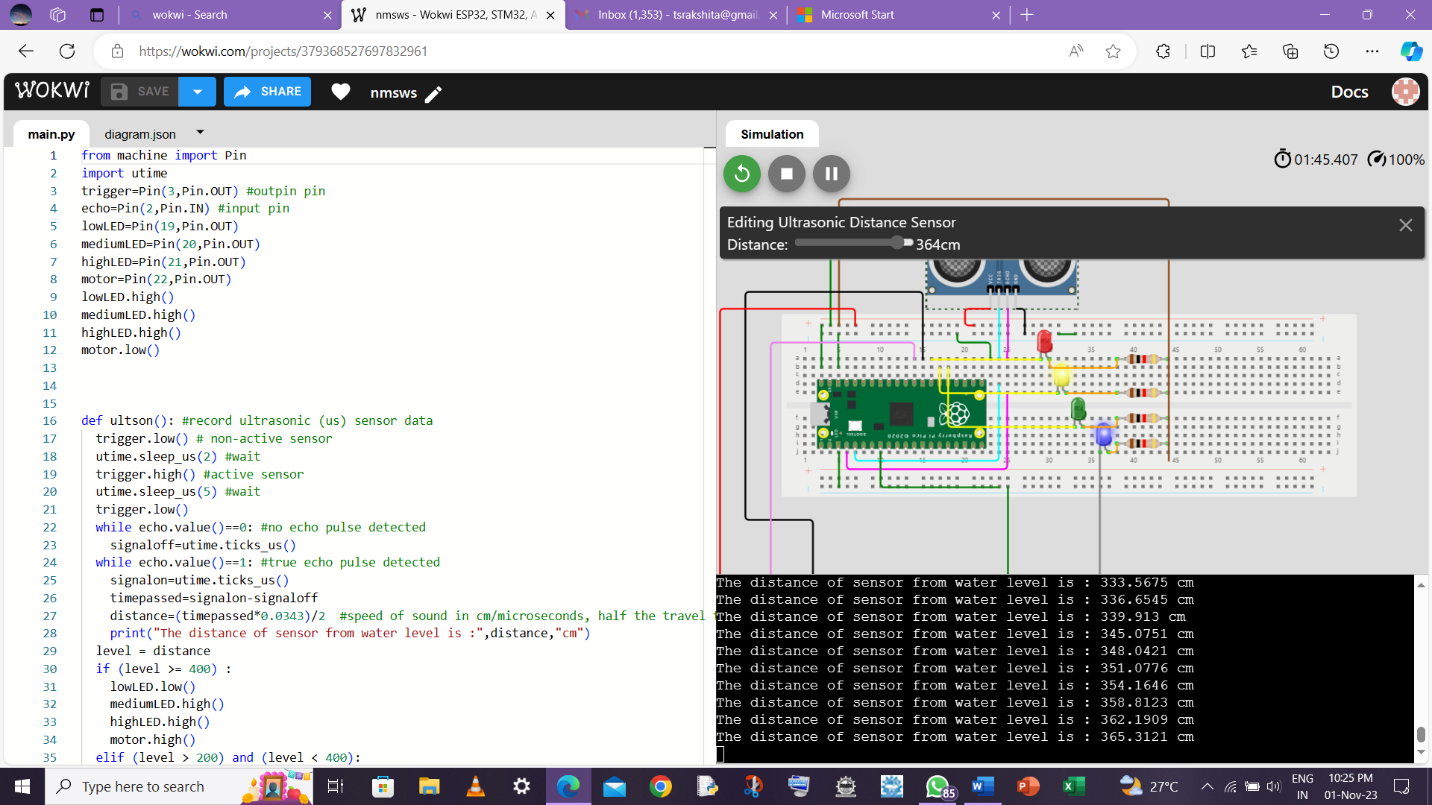
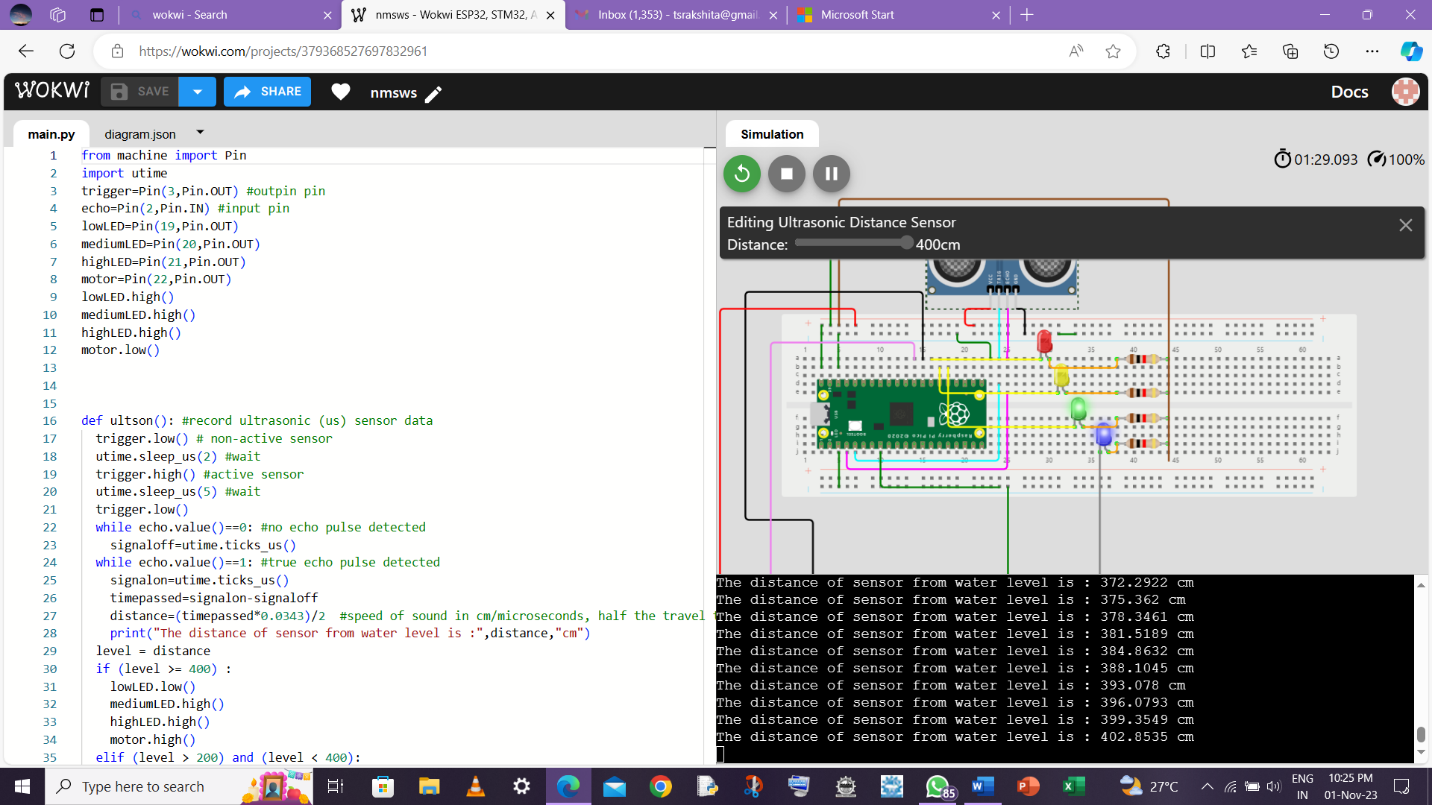
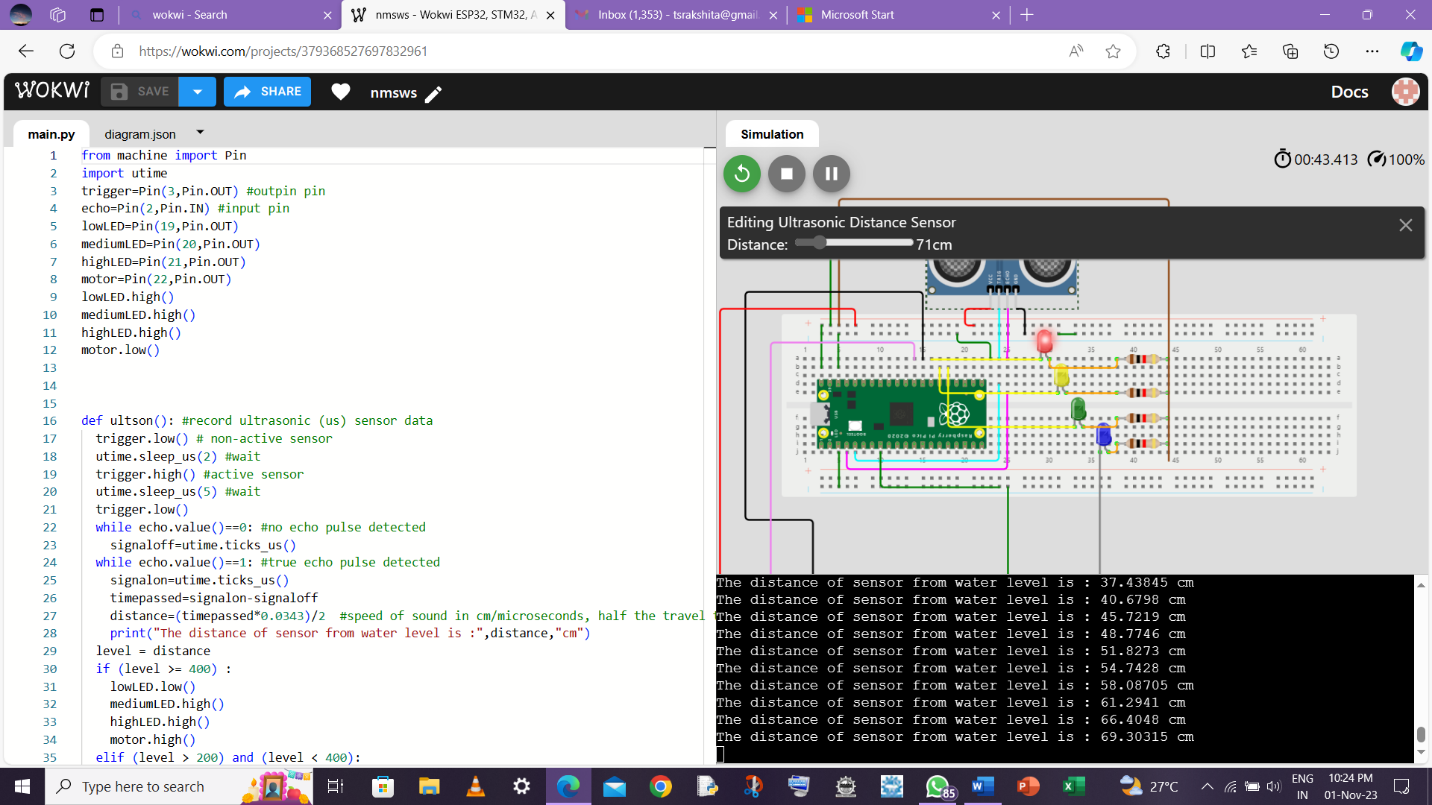
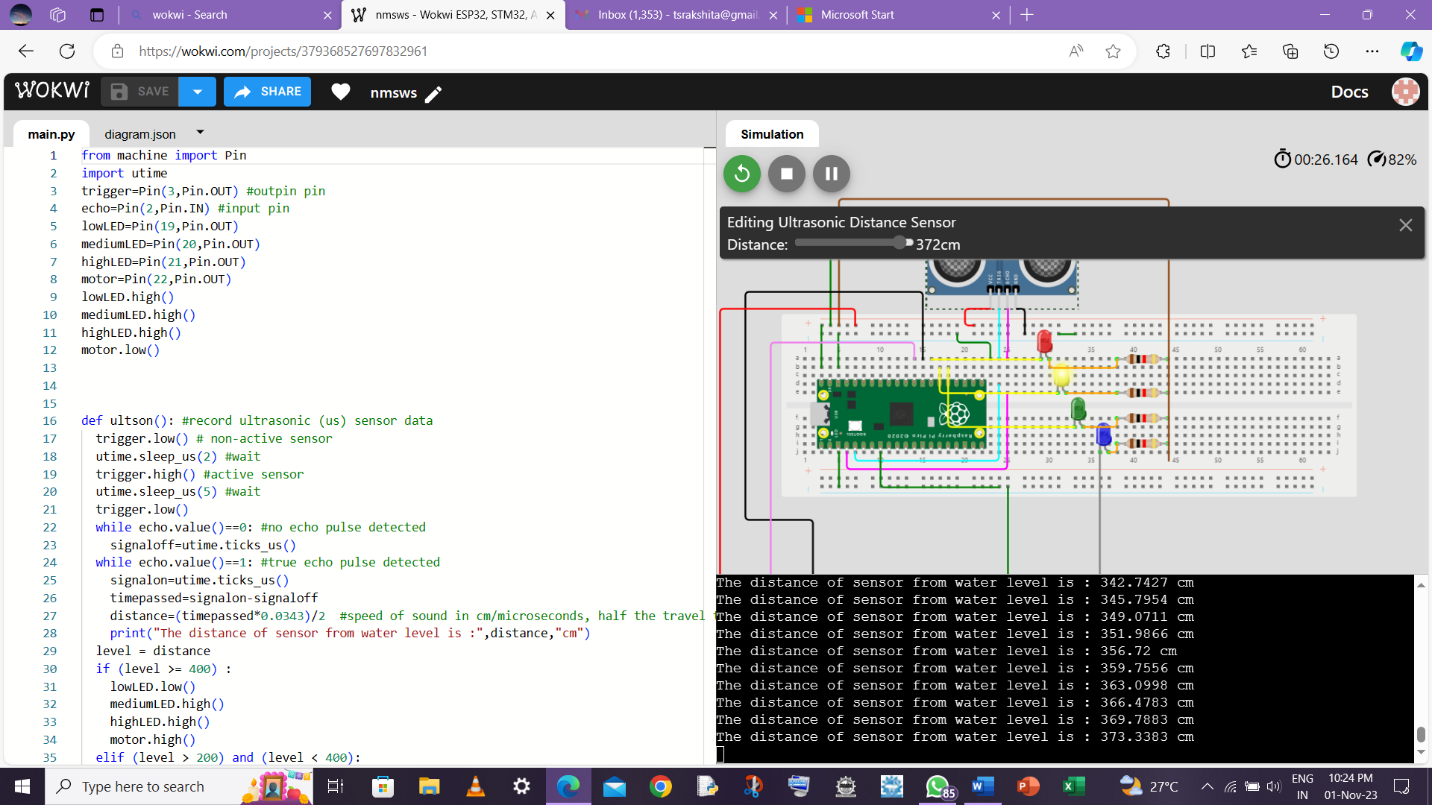
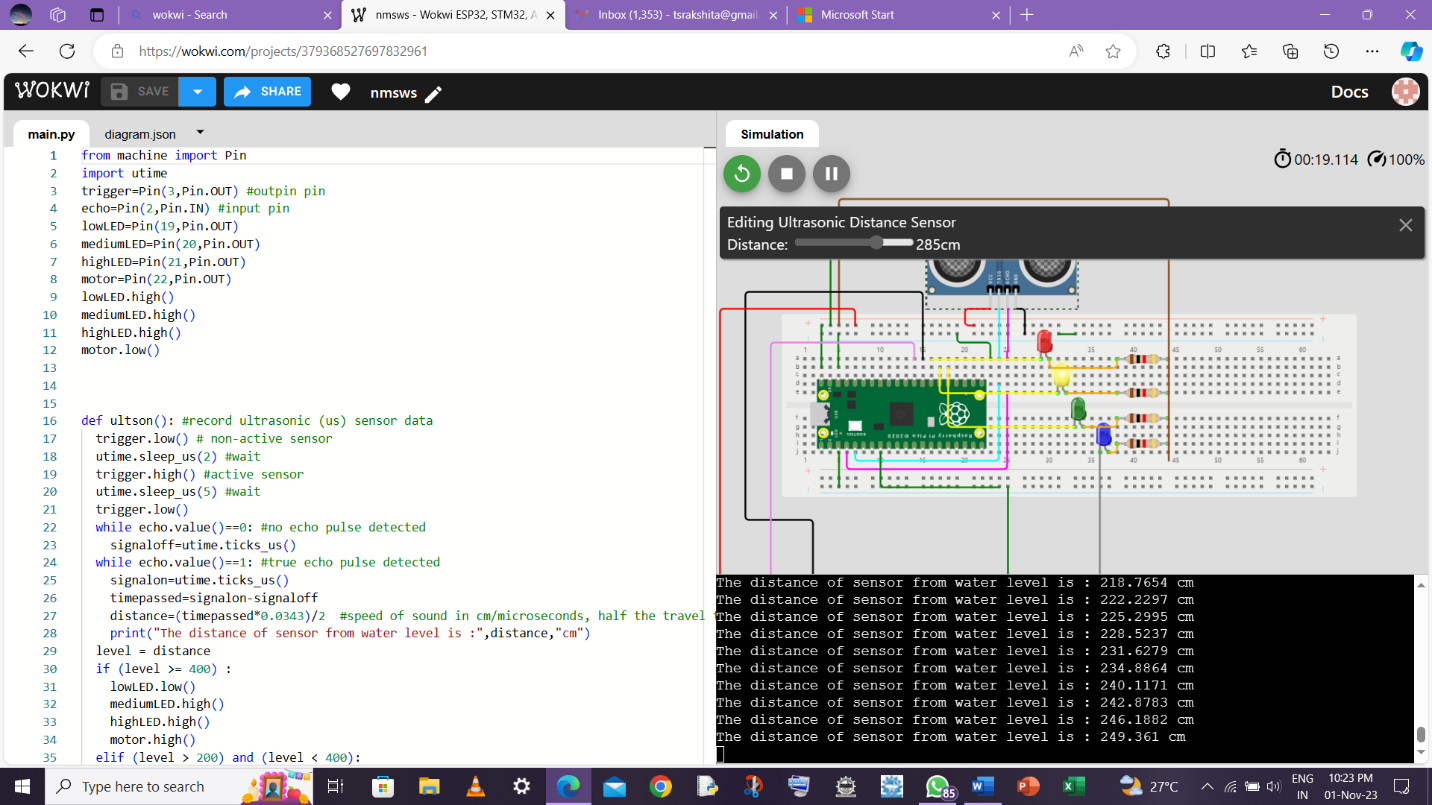
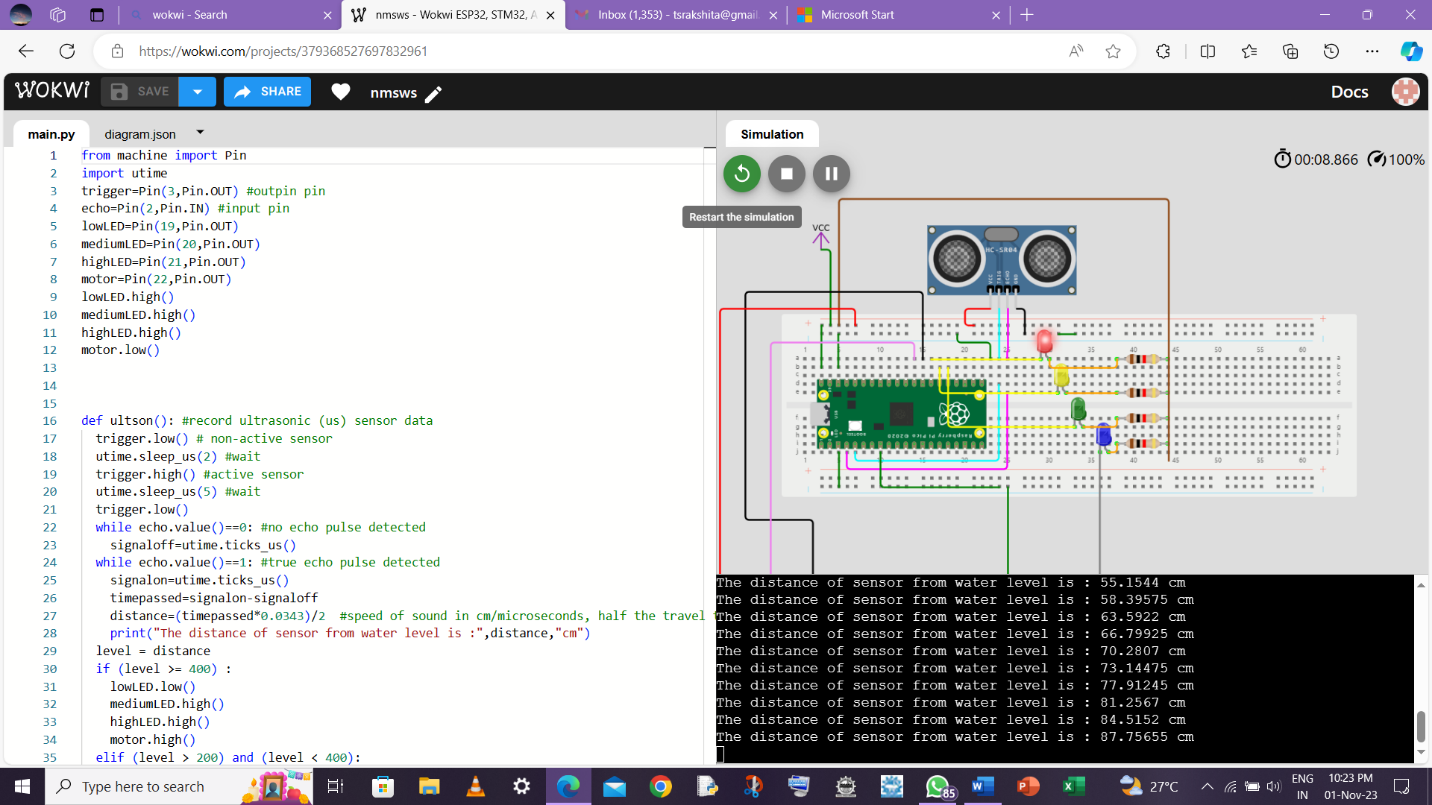
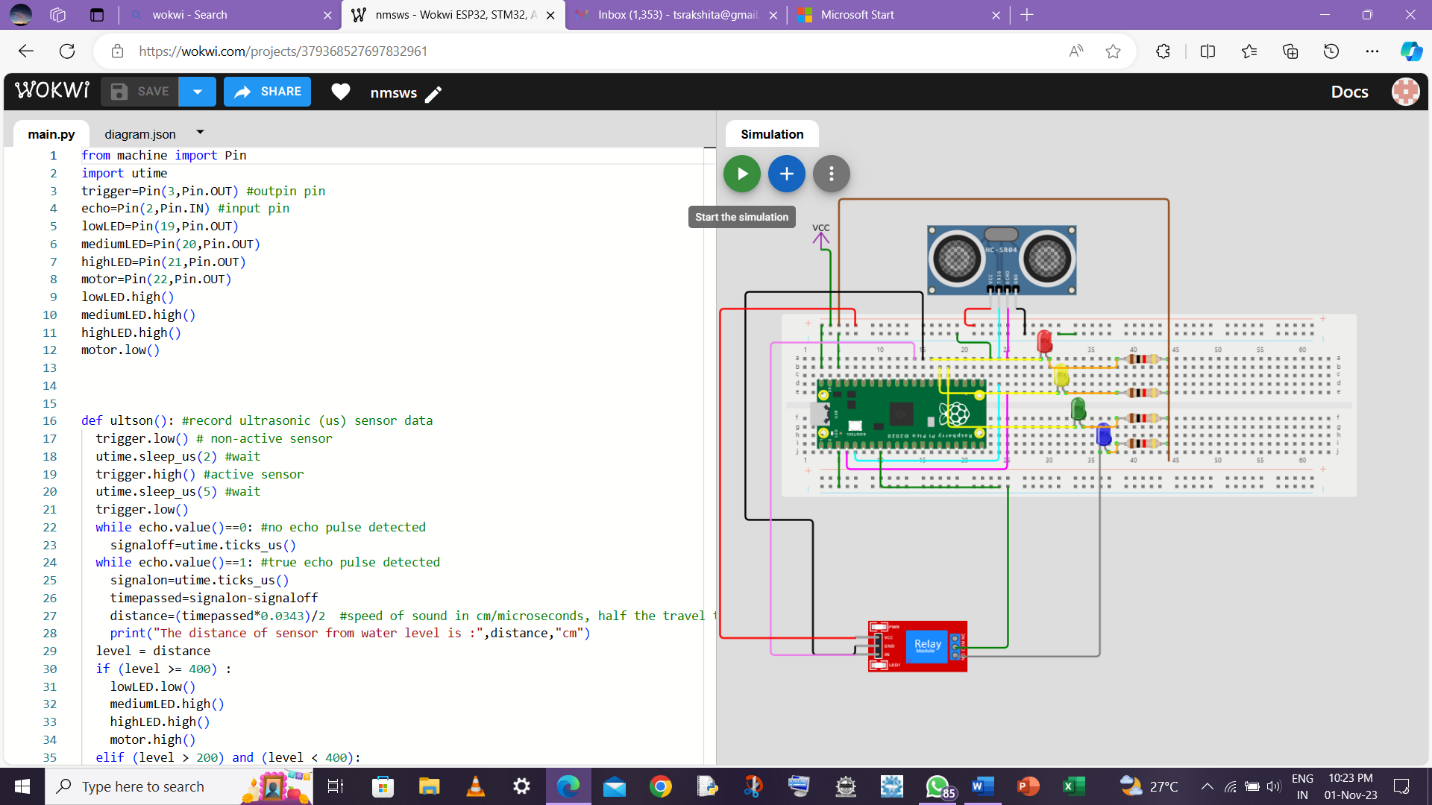
SMART WATER SYSTEM

PROJECT DESCRIPTION:

For the monitoring of water tanks used to supply water for drinking, irrigation, restrooms and cleaning, ultrasonics are placed at the top of the tanks. This sensor will send ultrasonic waves which would hit the surface of the water and get reflected back. Depending on the time taken to reflect back from the water level the ultrasound calculates the water level.

Here, the ultrasound in interfaced with Raspberri Pi Pico microcontroller. Four LEDs are connected as well as a relay for motor is connected.When the sensor detects the distance travelled by the reflected wave is greater than 400 cm, green LED glows and motor relay is switched on, indicating that the water level in the tank is low. When the distance is between 200 cm and 400 cm,yellow LED glows indicating water level s the middle range. When the distance is less than 100 cm, red LED glows, indicating the high water level in tank.

SCREENSHOTS:



MICROPYTHON CODE:

from machine import Pin

import utime

trigger=Pin(3,Pin.OUT) #outpin pin

echo=Pin(2,Pin.IN) #input pin

lowLED=Pin(19,Pin.OUT)

mediumLED=Pin(20,Pin.OUT)

highLED=Pin(21,Pin.OUT)

motor=Pin(22,Pin.OUT)

lowLED.high()

mediumLED.high()

highLED.high()

motor.low()

def ultson(): #record ultrasonic (us) sensor data

  trigger.low() # non-active sensor

  utime.sleep\_us(2) #wait

  trigger.high() #active sensor

  utime.sleep\_us(5) #wait

  trigger.low()

  while echo.value()==0: #no echo pulse detected

    signaloff=utime.ticks\_us()

  while echo.value()==1: #true echo pulse detected

    signalon=utime.ticks\_us()

    timepassed=signalon-signaloff

    distance=(timepassed\*0.0343)/2  #speed of sound in cm/microseconds, half the travel time

    print("The distance of sensor from water level is :",distance,"cm")

  level = distance

  if (level >= 400) :

    lowLED.low()

    mediumLED.high()

    highLED.high()

    motor.high()

  elif (level > 200) and (level < 400):

    lowLED.high()

    mediumLED.low()

    highLED.high()

  elif (level < 100):

    lowLED.high()

    mediumLED.high()

    highLED.low()

    motor.low()

while True:

  ultson()

  utime.sleep(5)