American University of Sharjah

College of Engineering

Fall 2024

Embedded Systems

COE 410-01

Instructor Name: Prof. Abdulrahman Al Ali, Prof. Ahmad Al Nabulsi

Midterm 2

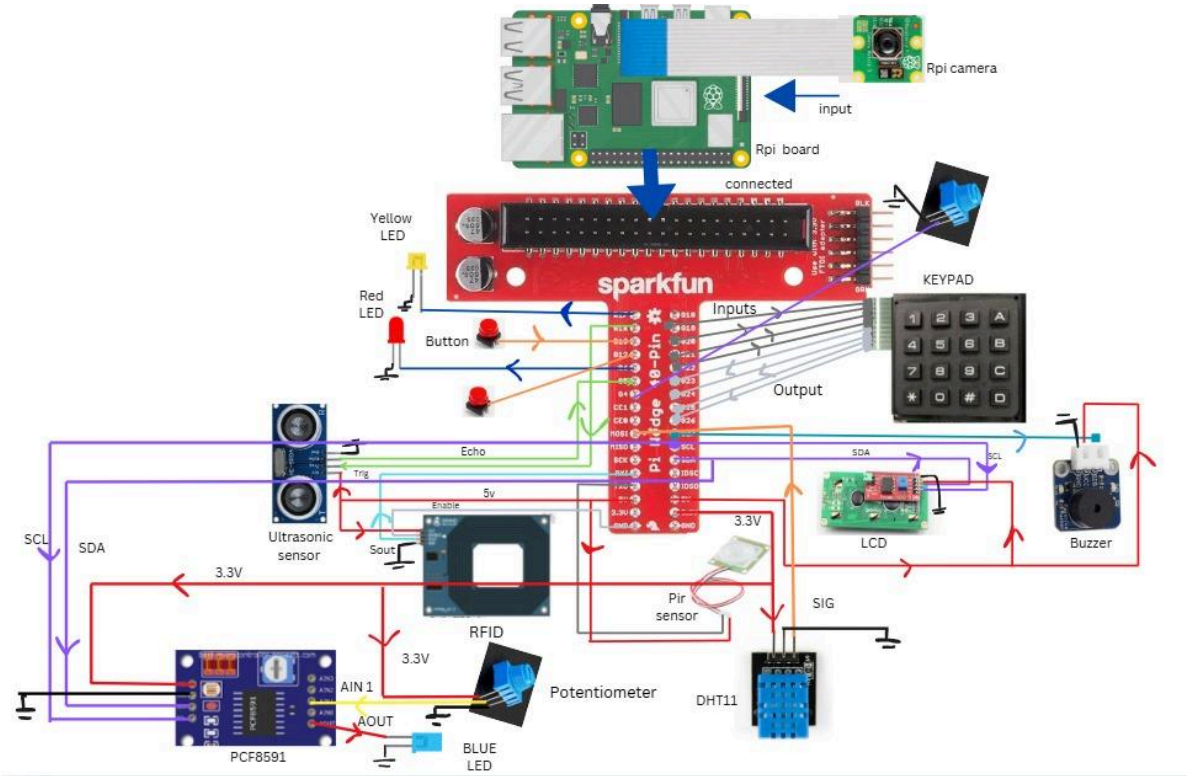
Submission Date: 3rd December, 2024

Sivapriya - g00093499

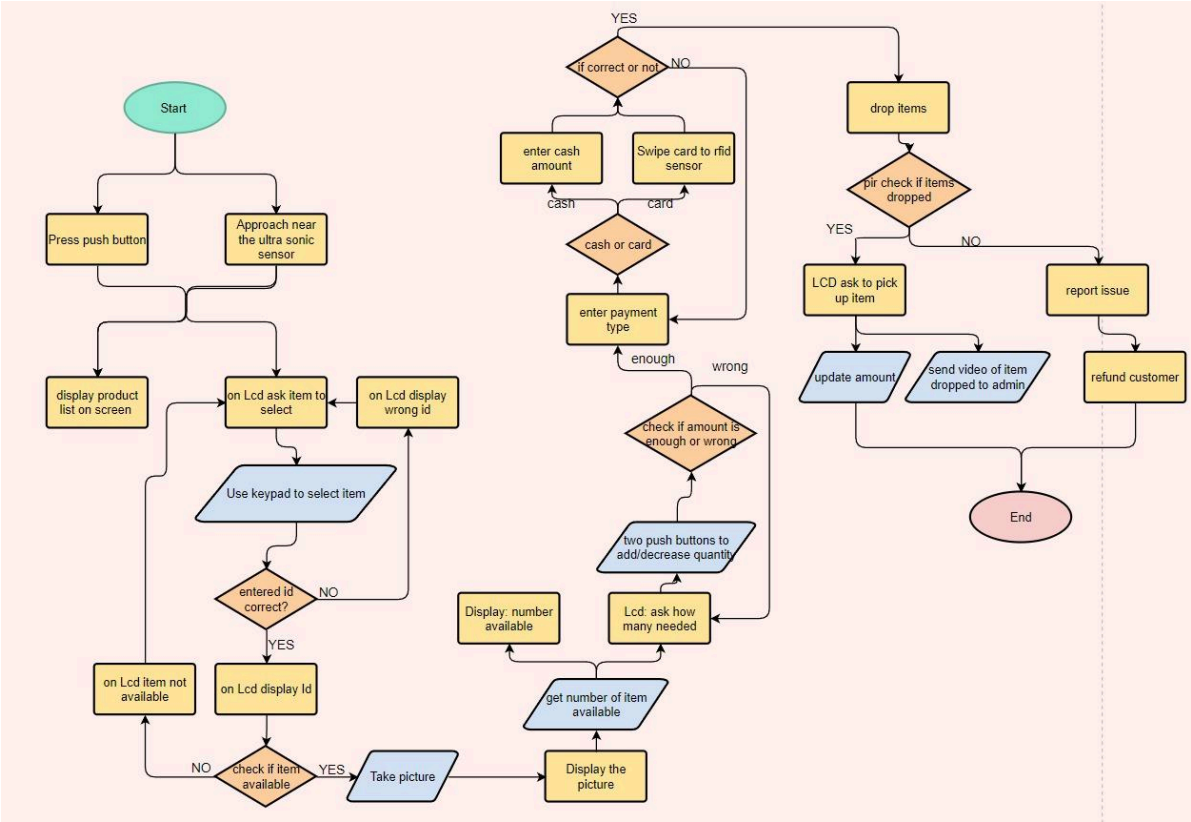
Mohammed Ali - b00088658

Kareem Ahmed - b00088375

Hardware Diagram:



Software Flowchart:



Commented Code:

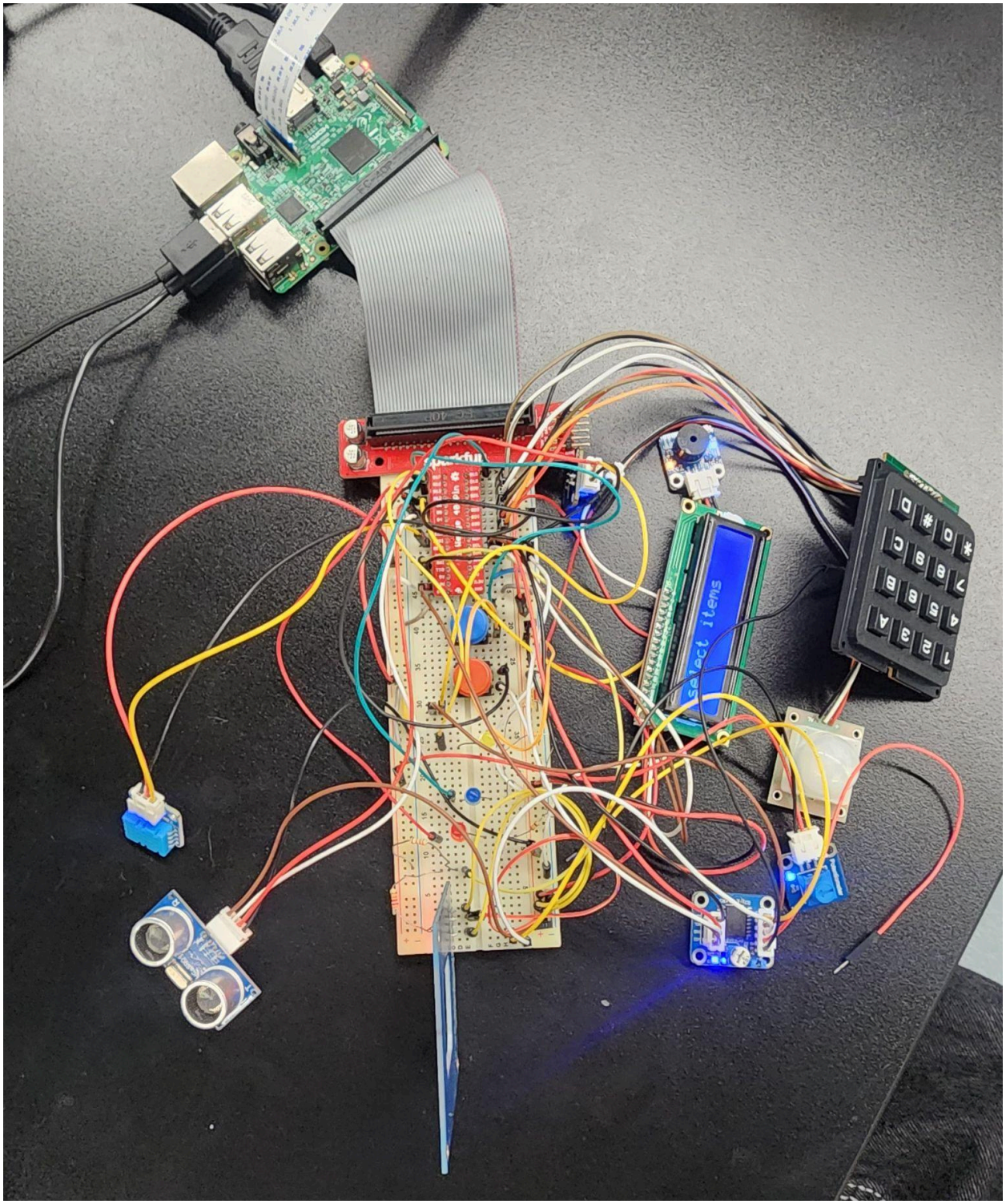
```
# Import necessary libraries
import RPi.GPIO as GPIO  # For GPIO control
from datetime import datetime  # To get current date and time
from flask import send_file, render_template, jsonify, request, redirect,
url_for  # For Flask server
from picamera import PiCamera  # For controlling the Raspberry Pi Camera
from flask import Flask  # For Flask application
import LCD1602 as LCD  # For controlling LCD
import time  # For sleep and timing
from keypadfunc import keypad  # Keypad input module
import PCF8591 as ADC  # ADC module for analog readings
ADC.setup(0x48)  # Setup ADC at address 0x48
import urllib.request  # For HTTP requests
import serial  # For serial communication
import DHT11  # DHT11 sensor module

# Define constants and initialize components
API_KEY = "GTY4GI5KQ431Q1SF"  # API key for external services
WRITE_KEY = "QHNWT7SWTJLSB43J"  # Write key for ThingSpeak
SERIAL_PORT = '/dev/ttyS0'  # Serial port for RFID reader
values = []  # List to store sensor readings
adps = 55  # Passcode for access
item = 0  # Current item quantity
counter = 0  # Counter for selected items
enable_count = 0  # Enable/disable counting for button callbacks
rBut = 16  # GPIO pin for red button
bBut = 17  # GPIO pin for blue button

# Setup GPIO pins and components
GPIO.setmode(GPIO.BCM)  # Use BCM pin numbering
GPIO.setup(6, GPIO.OUT)  # Buzzer pin
buzz = GPIO.PWM(6, 1)  # PWM for buzzer
buzz.start(1)  # Start buzzer with a low frequency

# Setup LCD and camera
LCD.init(0x27, 1)  # Initialize LCD at I2C address 0x27
Mycamera = PiCamera()  # Initialize camera

# Pins for ultrasonic sensor
```

```
TRIG = 5  # Trigger pin
GPIO.setup(TRIG, GPIO.OUT)
ECHO = 4  # Echo pin
GPIO.setup(ECHO, GPIO.IN)

# Motion sensor pin
MOTION = 14
GPIO.setup(MOTION, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# Button setup
GPIO.setup(rBut, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(bBut, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

# LEDs for status
dRed = 12
dYellow = 13
dGreen = 14
GPIO.setup(dRed, GPIO.OUT)
GPIO.setup(dYellow, GPIO.OUT)
GPIO.setup(dGreen, GPIO.OUT)

# Constants for ThingSpeak
elementnum = 5
Ch_ID = 2751326
Field_No1 = 1
Field_No2 = 2
No_readings = 10

# Notes for tune playback
G5 = 784
C4 = 261
D4 = 294
E4 = 329
F4 = 349
G4 = 392
G4_SHARP = 415
A4 = 440
B4 = 493
C5 = 523
D5 = 587
```

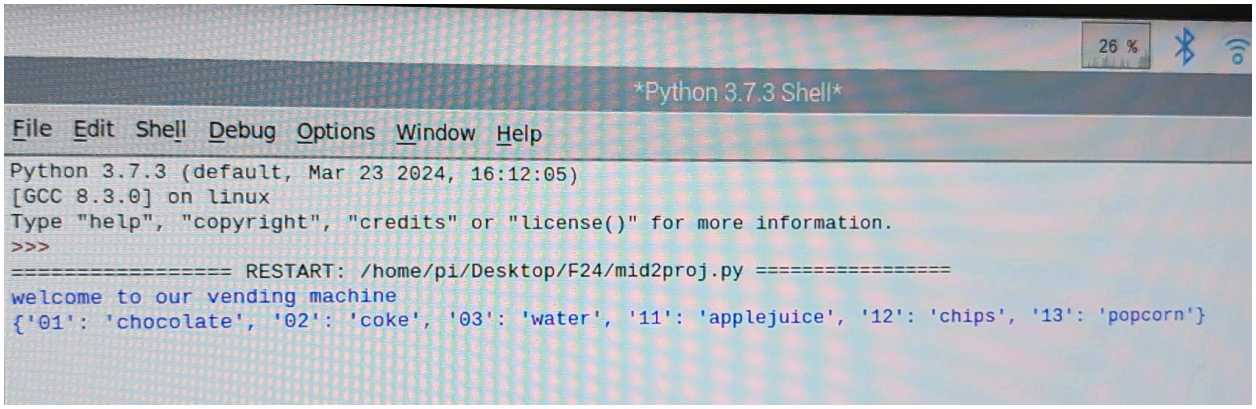
```python
E5 = 659
F5 = 698
A5 = 880
B5 = 987

# Melody for buzzer
tune = [
    (E5, 0.15), (E5, 0.15), (0, 0.1),
    (E5, 0.15), (0, 0.1), (C5, 0.15),
    (E5, 0.15), (G4, 0.3), (0, 0.3),
    (G4_SHARP, 0.15), (0, 0.1),
    # Additional notes truncated for brevity
]

# Functions for various features
def function(led):
    """Blink an LED twice."""
    for _ in range(2):
        GPIO.output(led, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(led, GPIO.LOW)
        time.sleep(1)

# Callback for red button
def rBut_callback(channel):
    global counter
    if enable_count == 1 and counter > 0:
        counter -= 1
        print("Counter decreased:", counter)

# Callback for blue button
def bBut_callback(channel):
    global counter
    if enable_count == 1:
        counter += 1
        print("Counter increased:", counter)

# Add event detection for buttons
GPIO.add_event_detect(rBut, GPIO.RISING, callback=rBut_callback,
bouncetime=300)
```

```
GPIO.add_event_detect(bBut, GPIO.RISING, callback=bBut_callback,
bouncetime=300)

# Function to calculate distance using ultrasonic sensor
def distance():
    """Calculate distance using ultrasonic sensor."""
    GPIO.output(TRIG, 0)
    time.sleep(0.000002)
    GPIO.output(TRIG, 1)
    time.sleep(0.00001)
    GPIO.output(TRIG, 0)
    while GPIO.input(ECHO) == 0:
        time1 = time.time()
    while GPIO.input(ECHO) == 1:
        time2 = time.time()
    duration = time2 - time1
    return duration * 1000000 / 58

# Flask routes for web server
app = Flask(__name__)

@app.route('/')
def index():
    """Main page."""
    return render_template('index.html')

@app.route('/inventory')
def inventory_page():
    """Display inventory."""
    return render_template('inventory.html', inventory=inventory)

# Add other functions and features as needed

# Main program logic
GPIO.output(dRed, GPIO.HIGH)  # Turn on red LED to indicate ready state
while True:
    # Wait for trigger (distance or button press)
    dist = distance()
    if dist <= 20 or GPIO.input(rBut) == 1:
        print("Welcome to our vending machine")
```

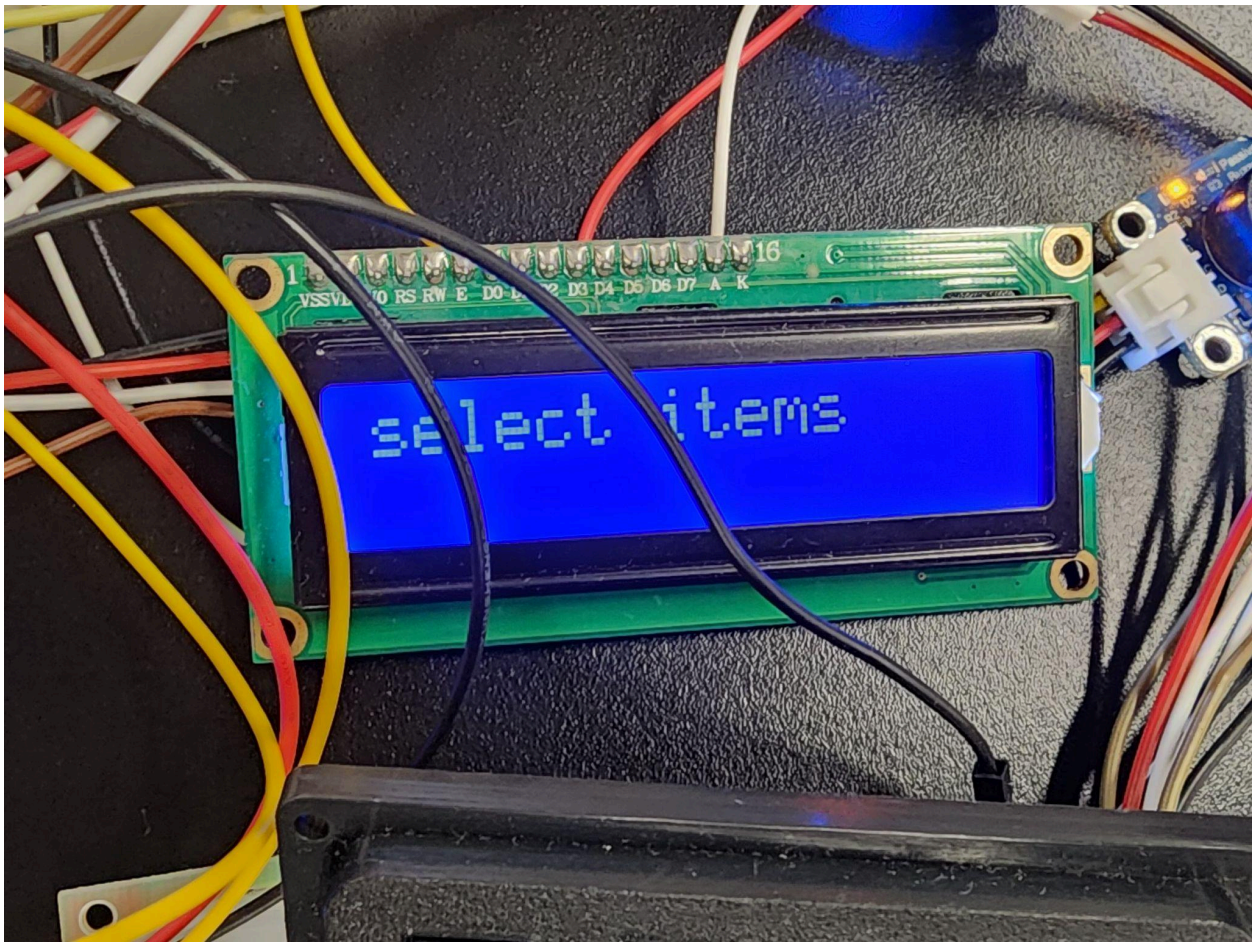
System Photo:



Test Results

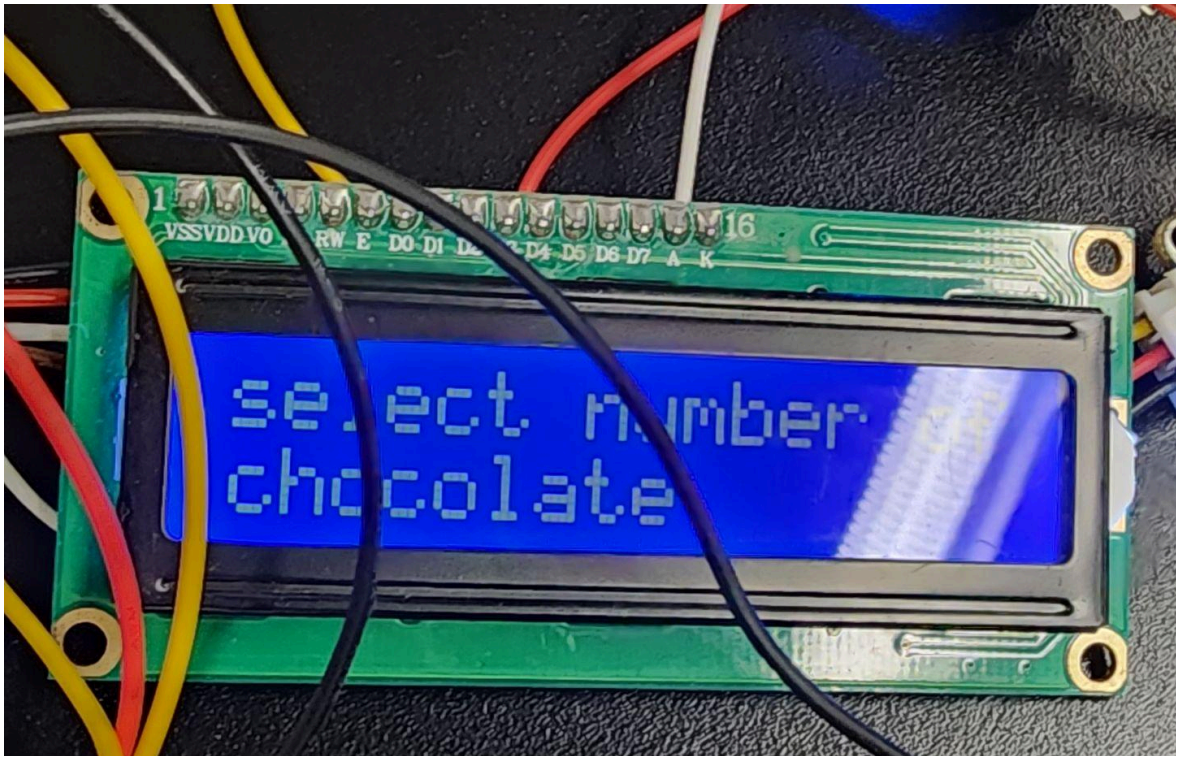


Welcome Page, With Menu, System Starting Through Ultrasonic Sensor Or Button.

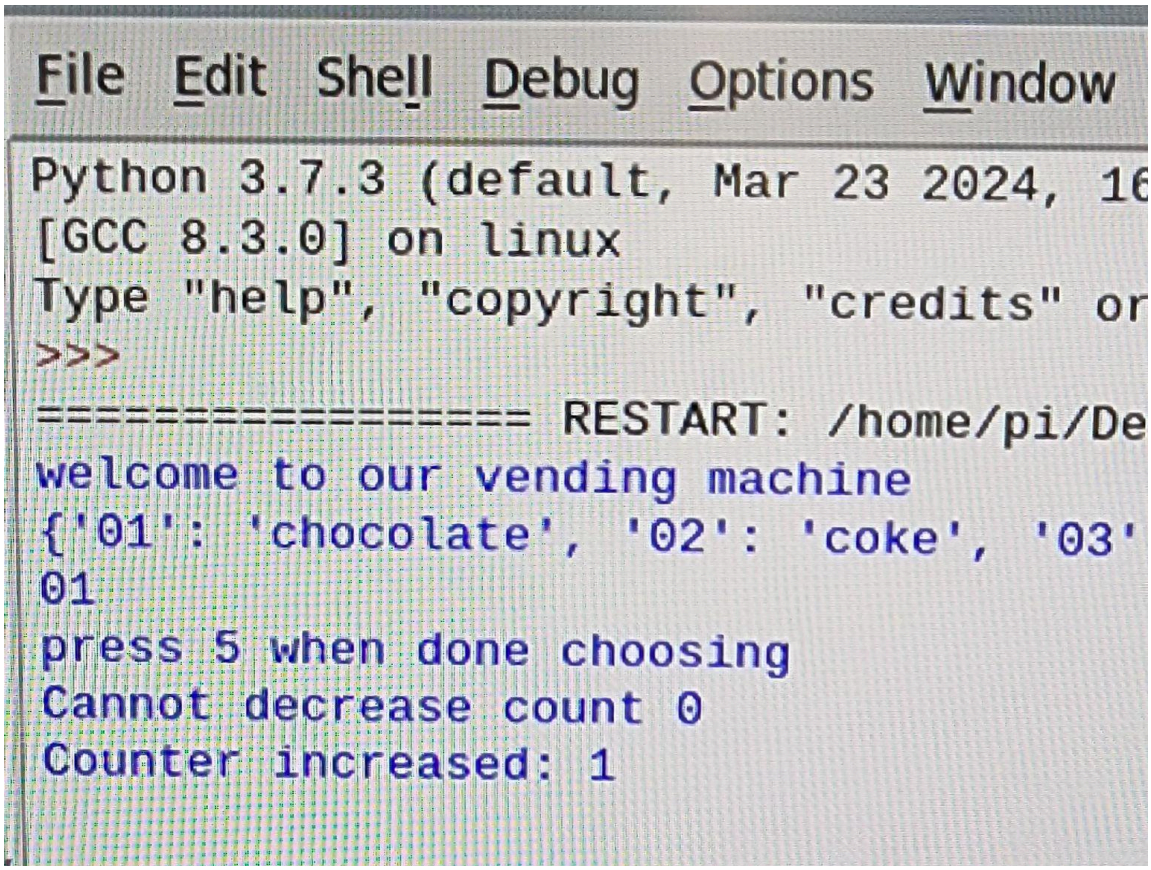


Lcd Output To Select Items

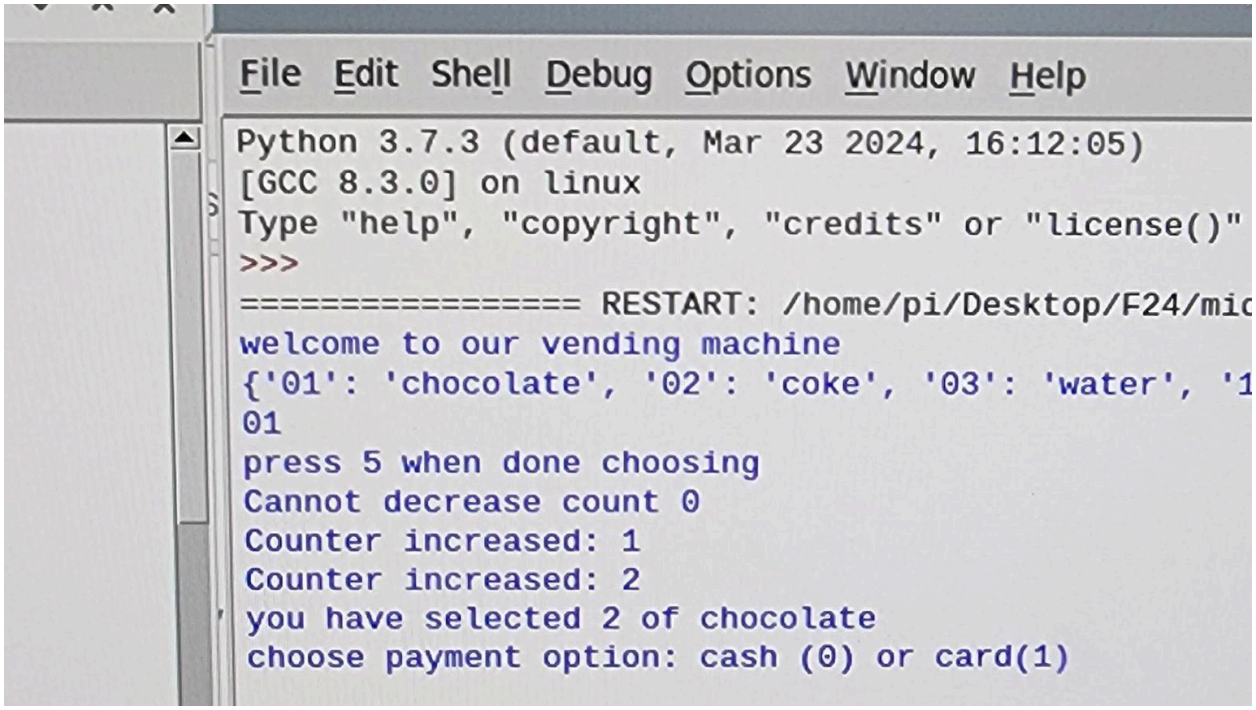
Showing User The Item Dropping



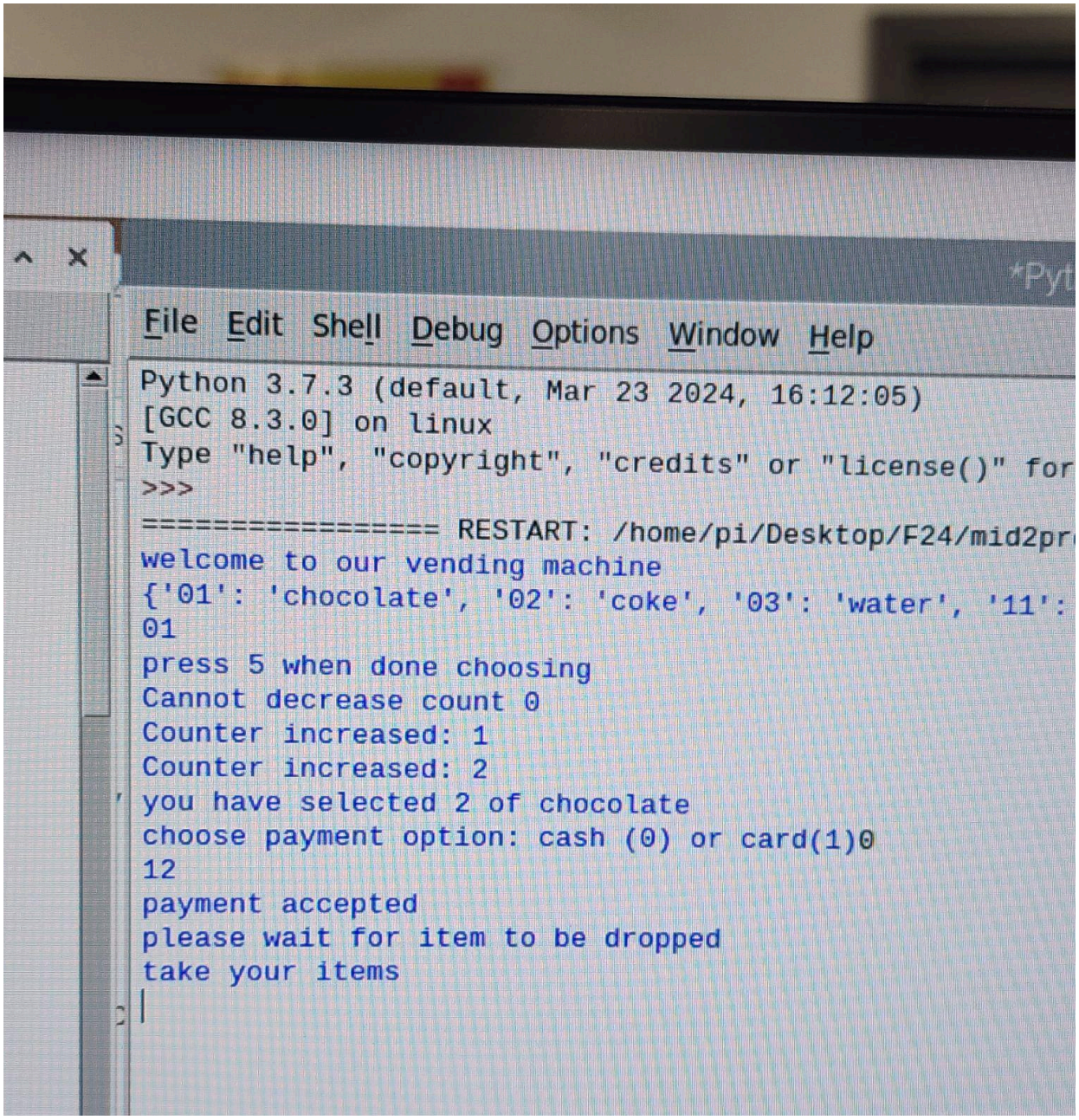
Lcd Output Prompting User To Either Increase Or Decrease Number Of Item



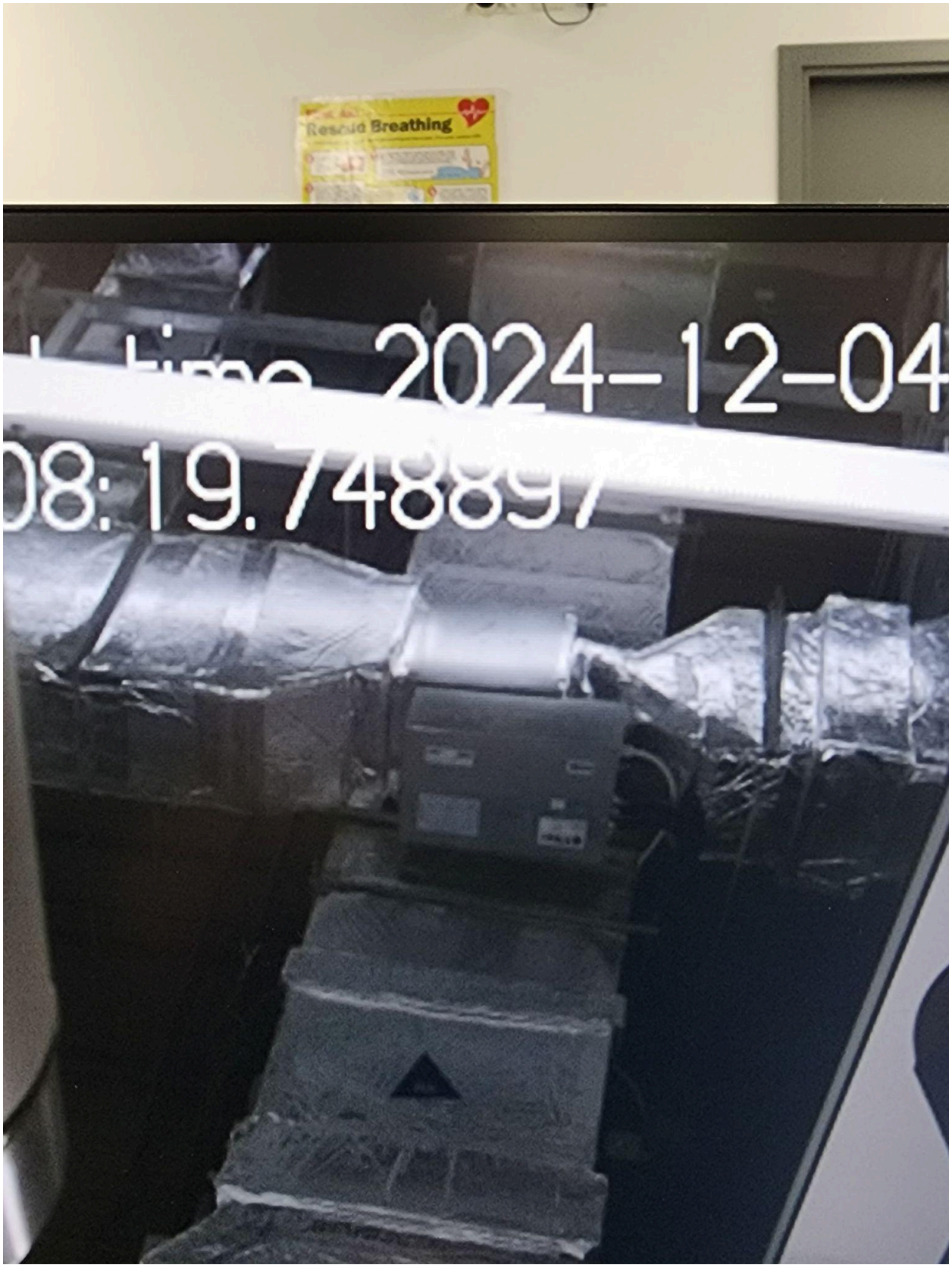
Cannot Decrease Further Than Zero Number Of Chocolates



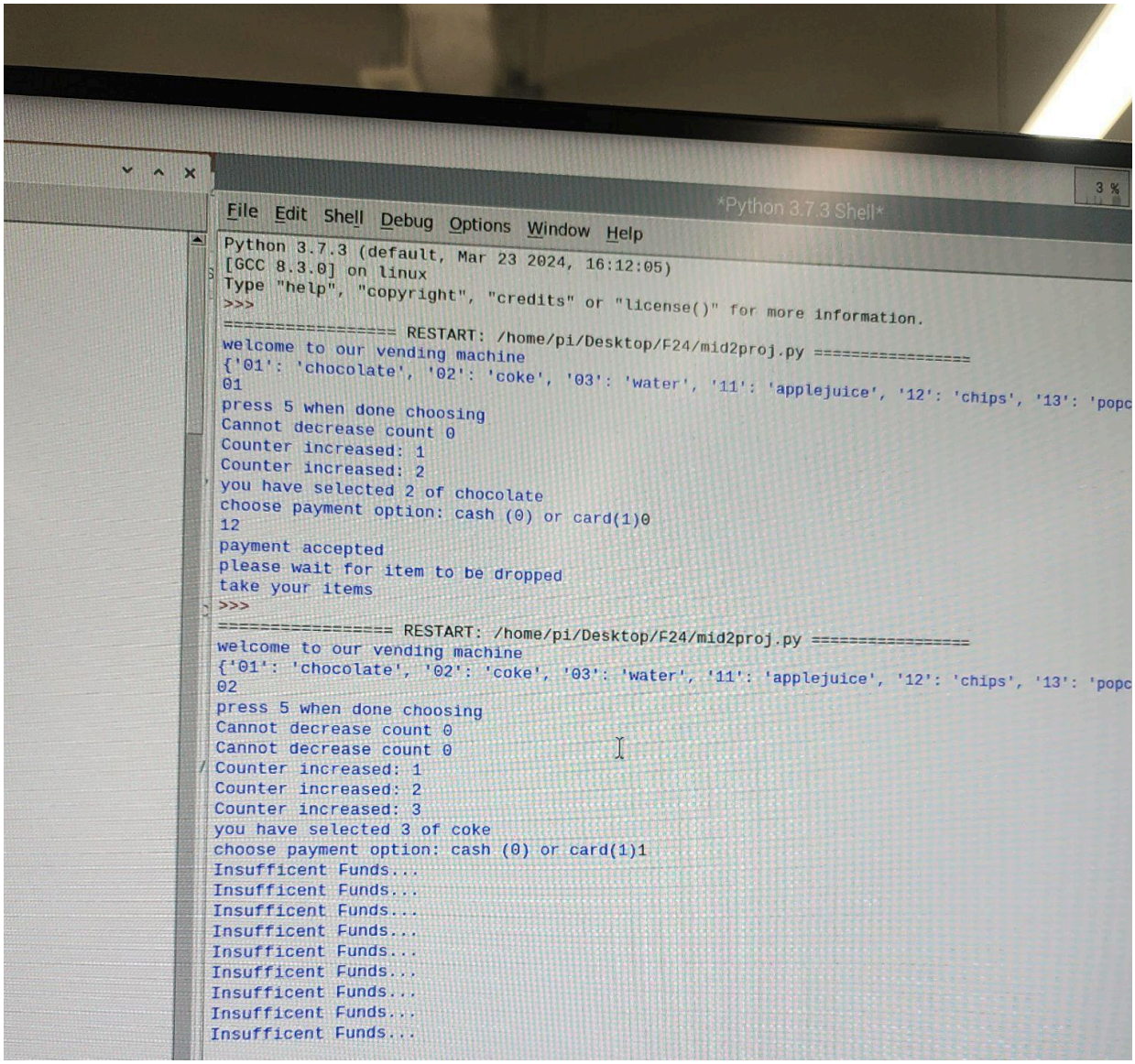
Asking For Payment Method (Cash Selected)



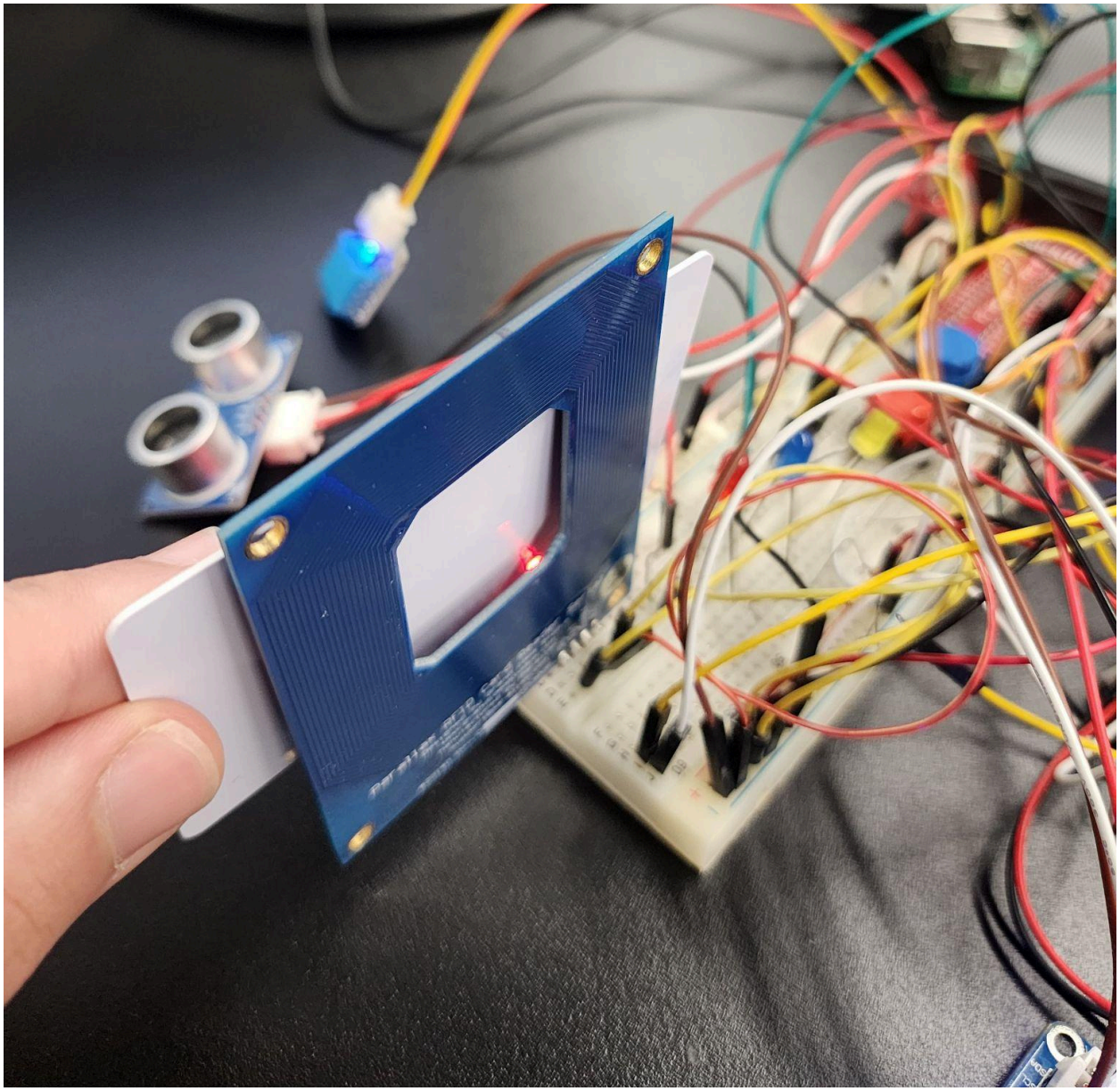
Asking User To Wait For Item To Be Dropped



Camera Image Of Item Being Dropped



Payment Method Chosen As Card



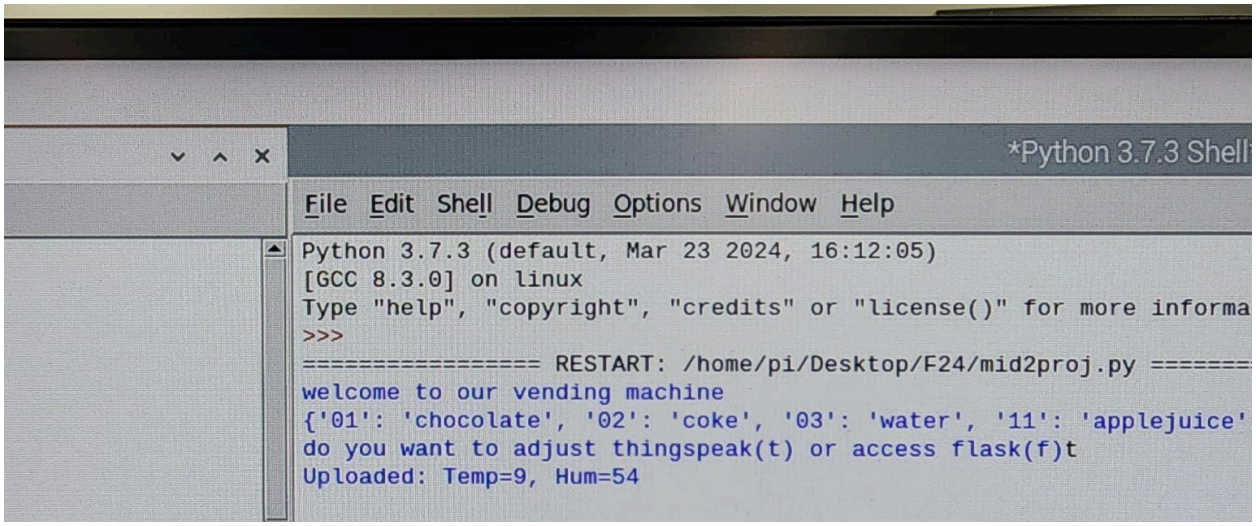
Rfid Is Used As Credit Card For Payment

```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================= RESTART: /home/pi/Desktop/F24/mid2proj.py ==============
welcome to our vending machine
{'01': 'chocolate', '02': 'coke', '03': 'water', '11': 'applejuice', '12':
01
press 5 when done choosing
Cannot decrease count 0
Counter increased: 1
Counter increased: 2
you have selected 2 of chocolate
choose payment option: cash (0) or card(1)0
12
payment accepted
please wait for item to be dropped
take your items
>>>
================= RESTART: /home/pi/Desktop/F24/mid2proj.py ==============
welcome to our vending machine
{'01': 'chocolate', '02': 'coke', '03': 'water', '11': 'applejuice', '12':
02
press 5 when done choosing
Cannot decrease count 0
Cannot decrease count 0
Counter increased: 1
Counter increased: 2
Counter increased: 3
you have selected 3 of coke
choose payment option: cash (0) or card(1)1
Insufficent Funds...
Insufficent Funds...
Insufficent Funds...
Insufficent Funds...
Insufficent Funds...
Insufficent Funds...
Insufficent Funds...
Insufficent Funds...
Insufficent Funds...
Insufficent Funds...
payment accepted 5200129DAB
please wait for item to be dropped
take your items
```

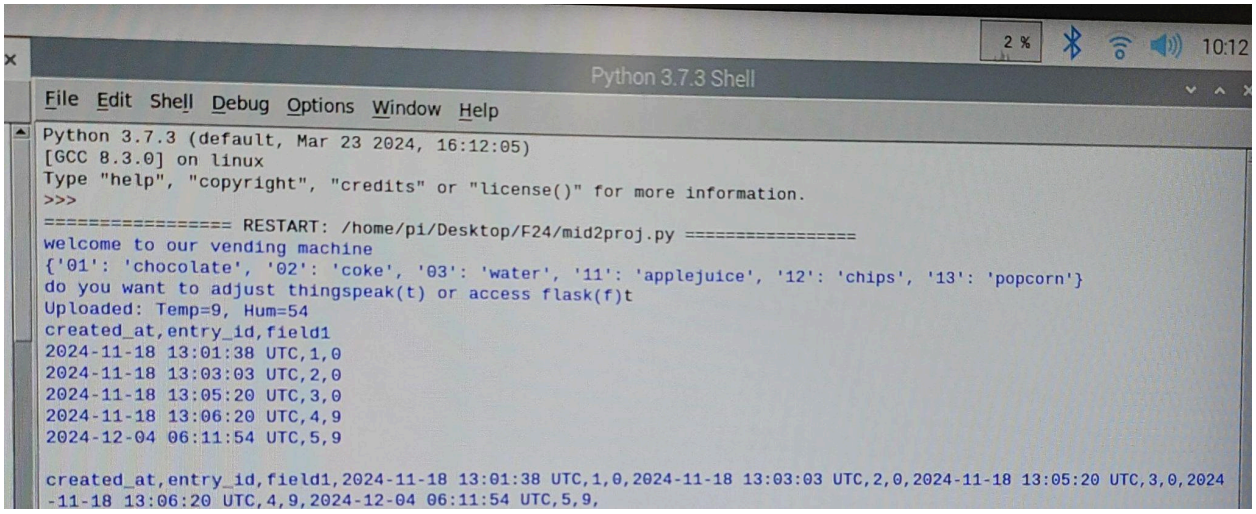
Release Item If Credit Card Is Acceptable

```
*Python 3.7.3 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.3 (default, Mar 23 2024, 16:12:05)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================= RESTART: /home/pi/Desktop/F24/mid2proj.py ================
welcome to our vending machine
{'01': 'chocolate', '02': 'coke', '03': 'water', '11': 'applejuice', '12': 'chips', '13': 'popcorn'}
do you want to adjust thingspeak(t) or access flask(f)
```

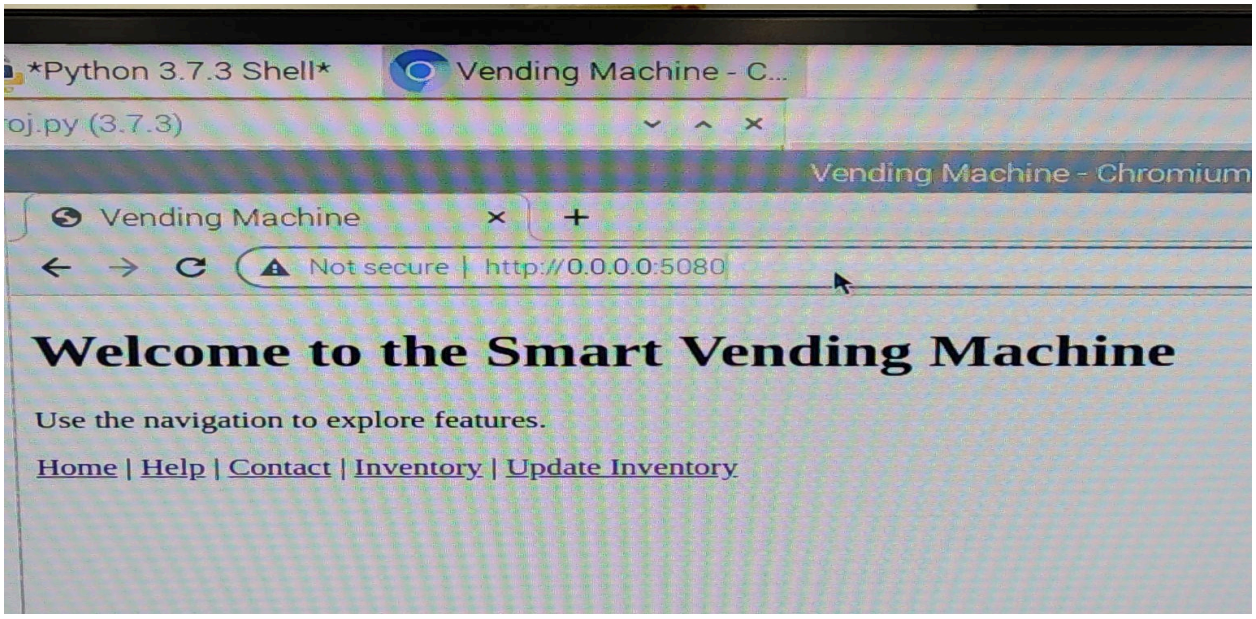
Welcome Page



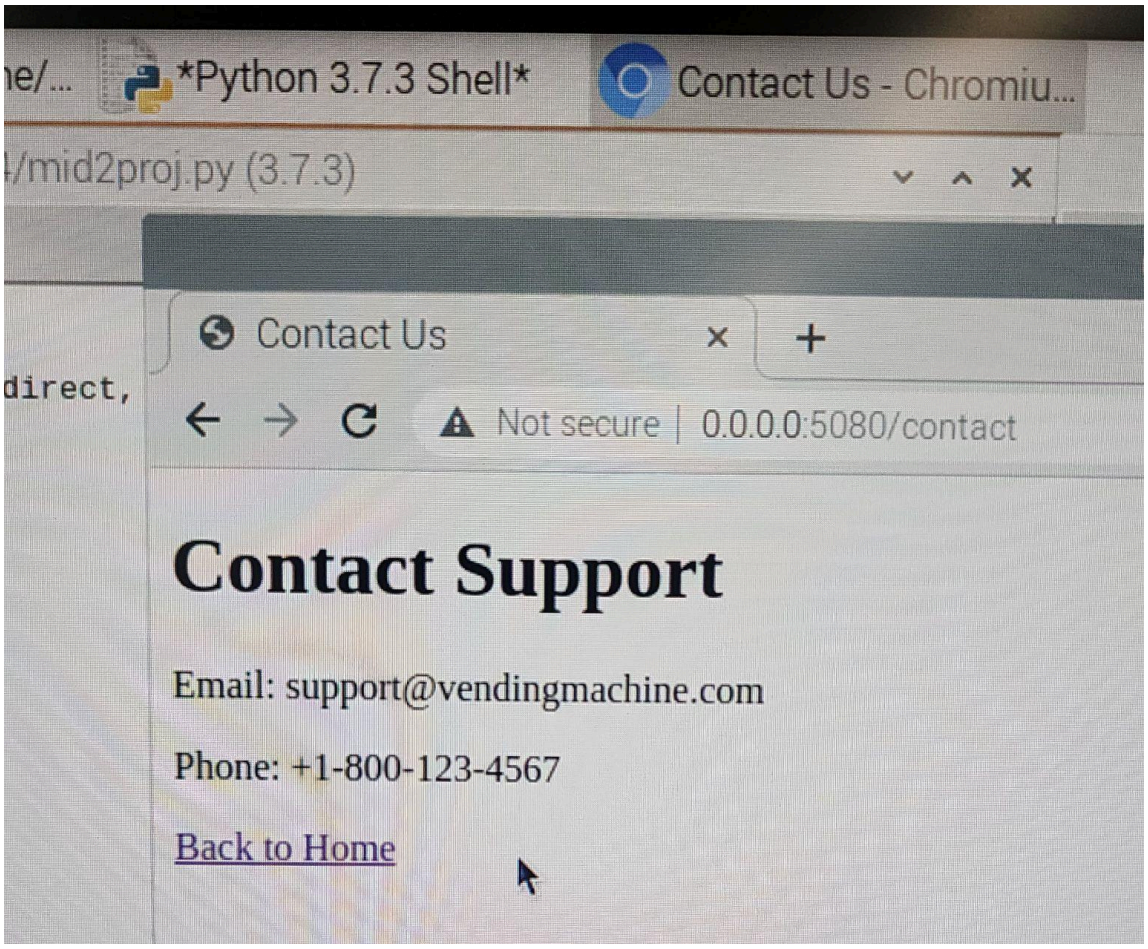
Accessing Admin Option Of Flask And Thinkspeak Through Keypad Password 55



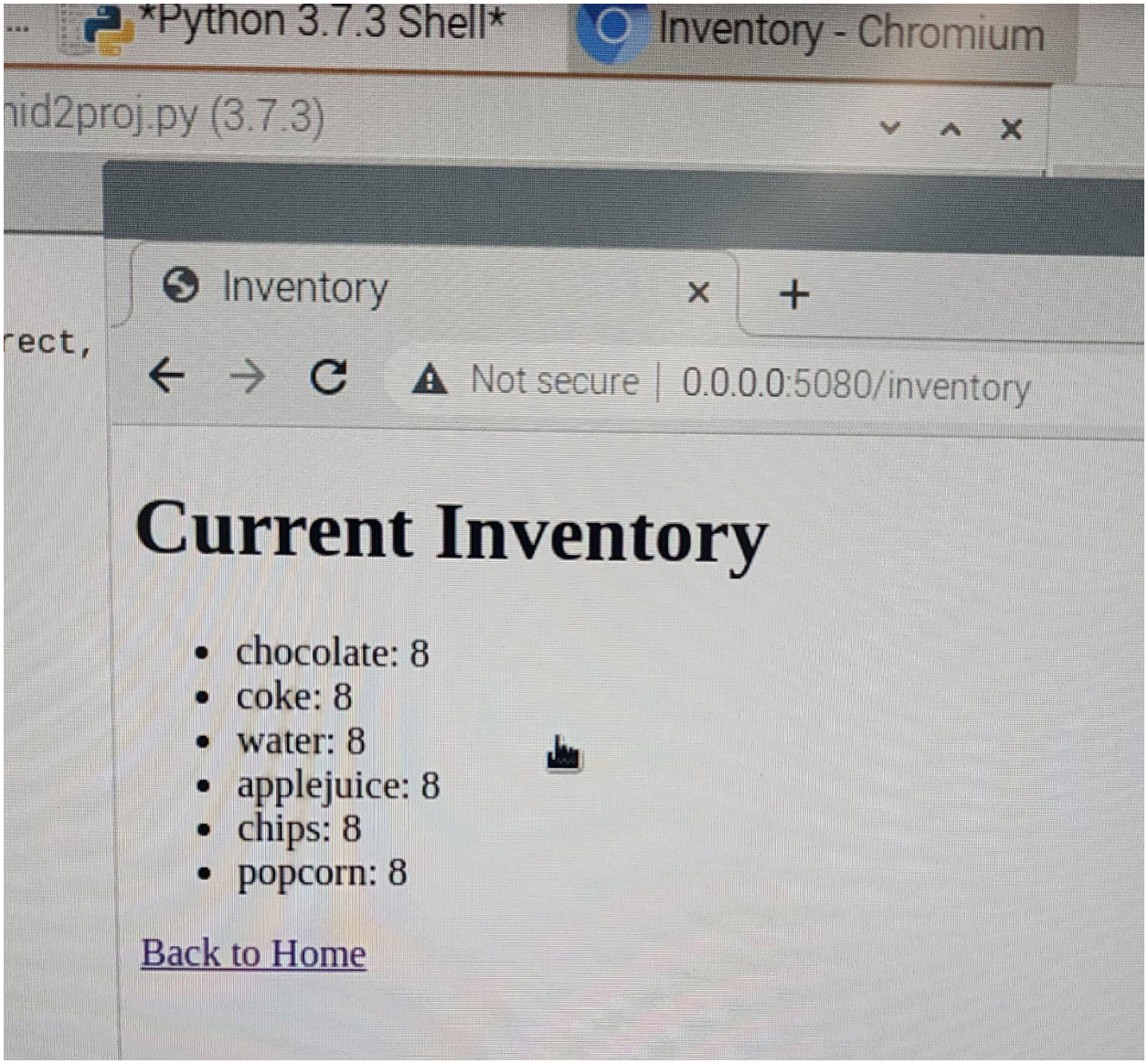
Uploading Current Values To Thinkspeak Server



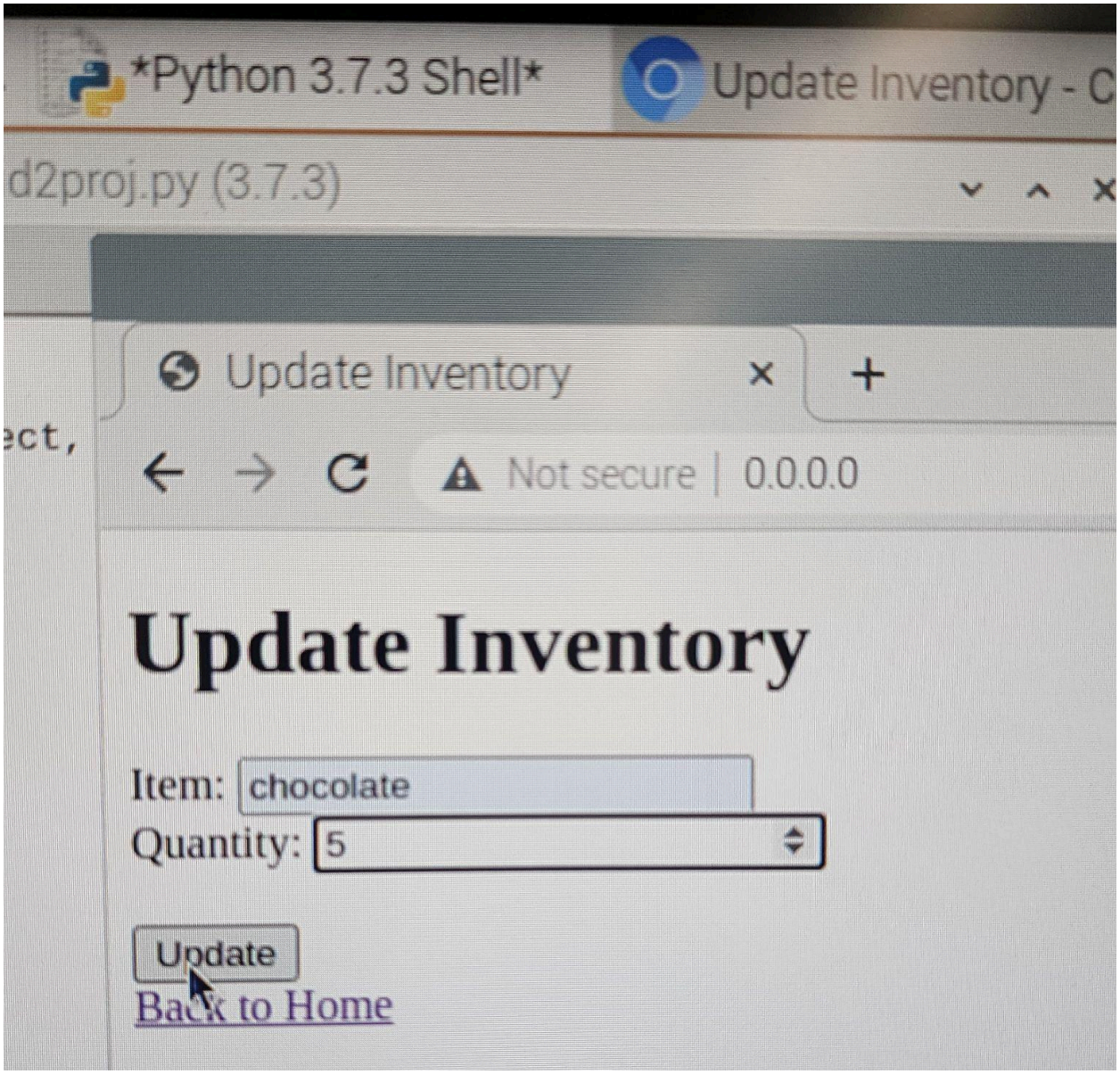
Flask Index



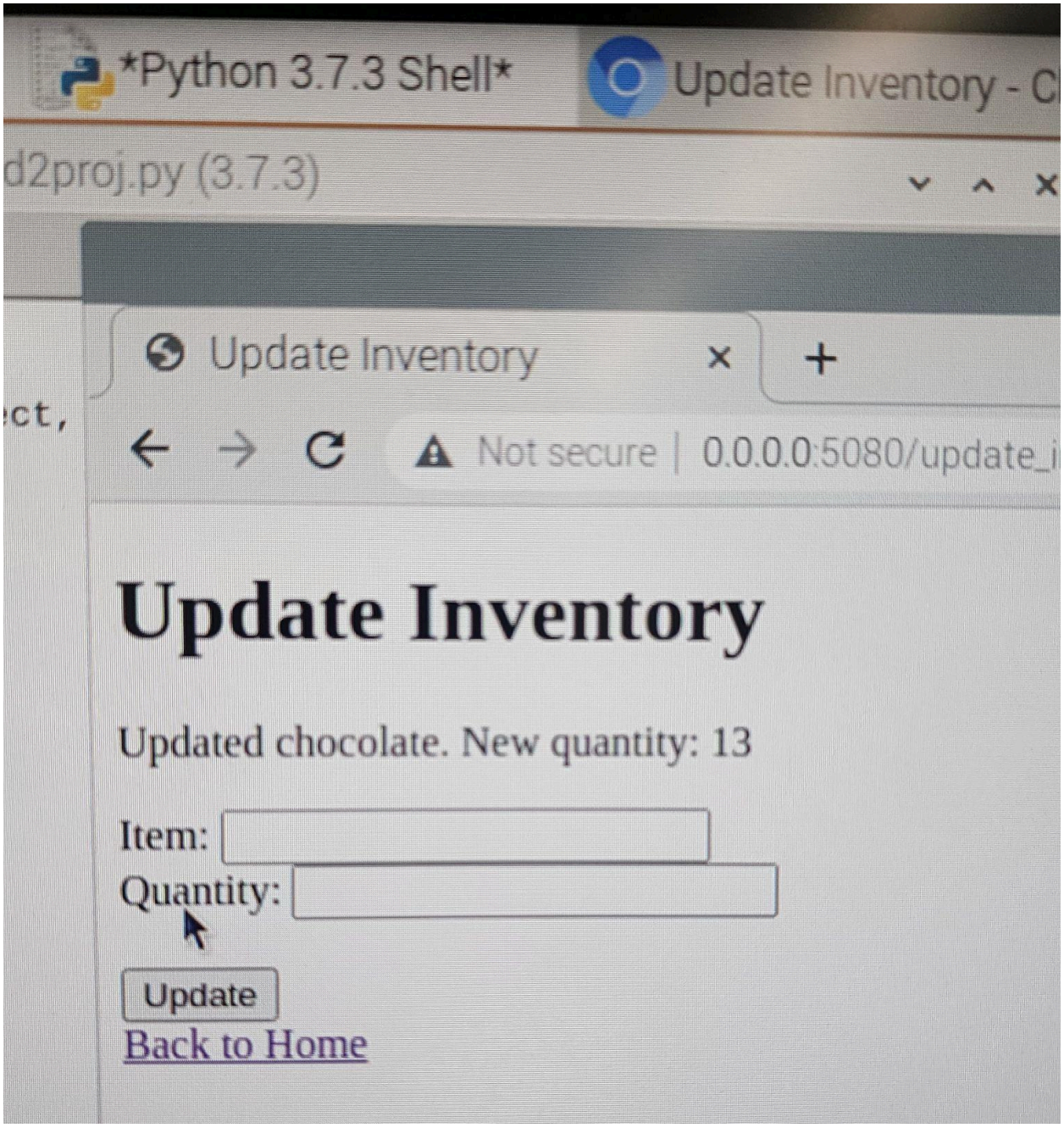
Flask Static Contact Page



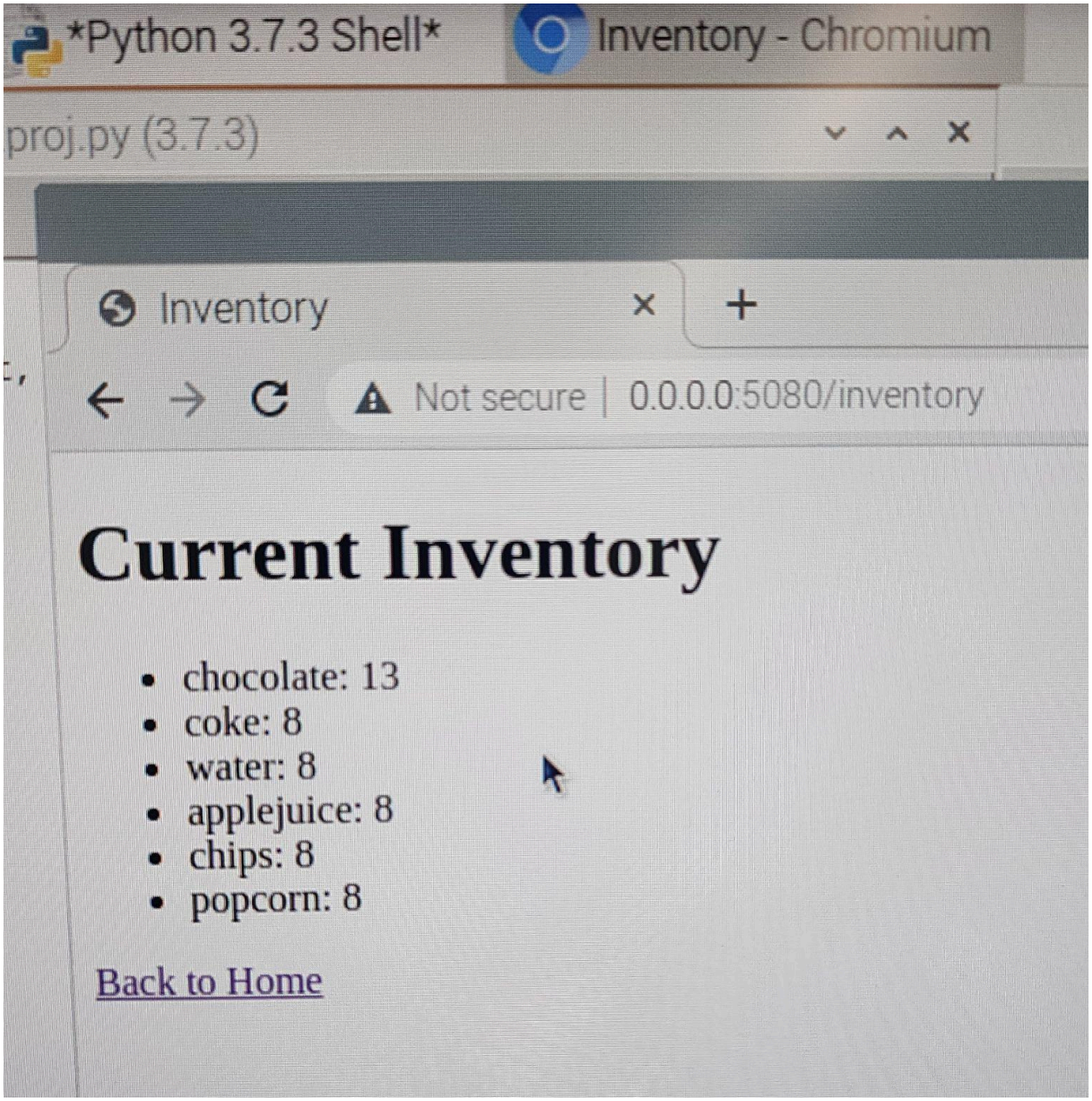
Flask Dynamic Inventory Page



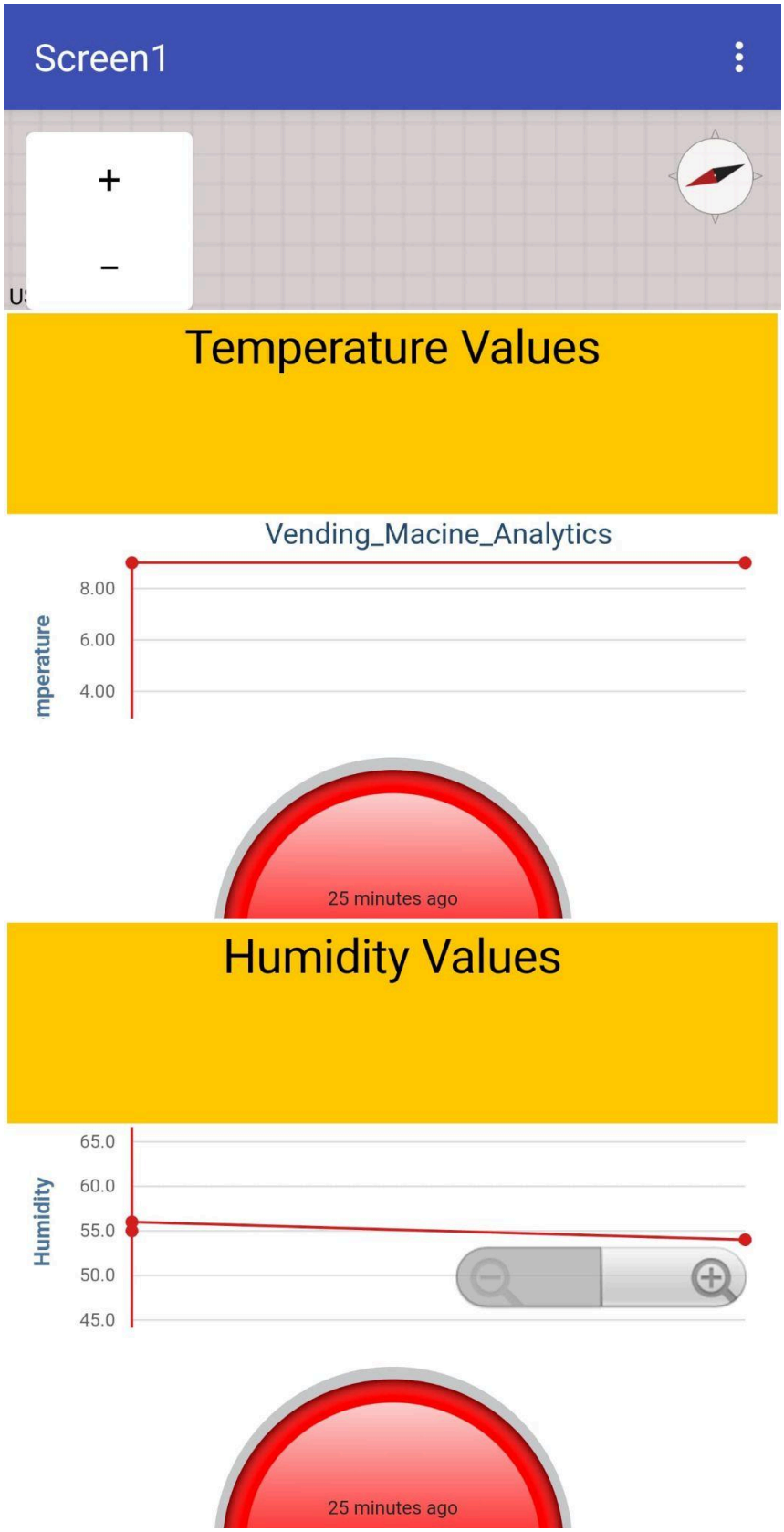
Flask Dynamic Inventory Updater



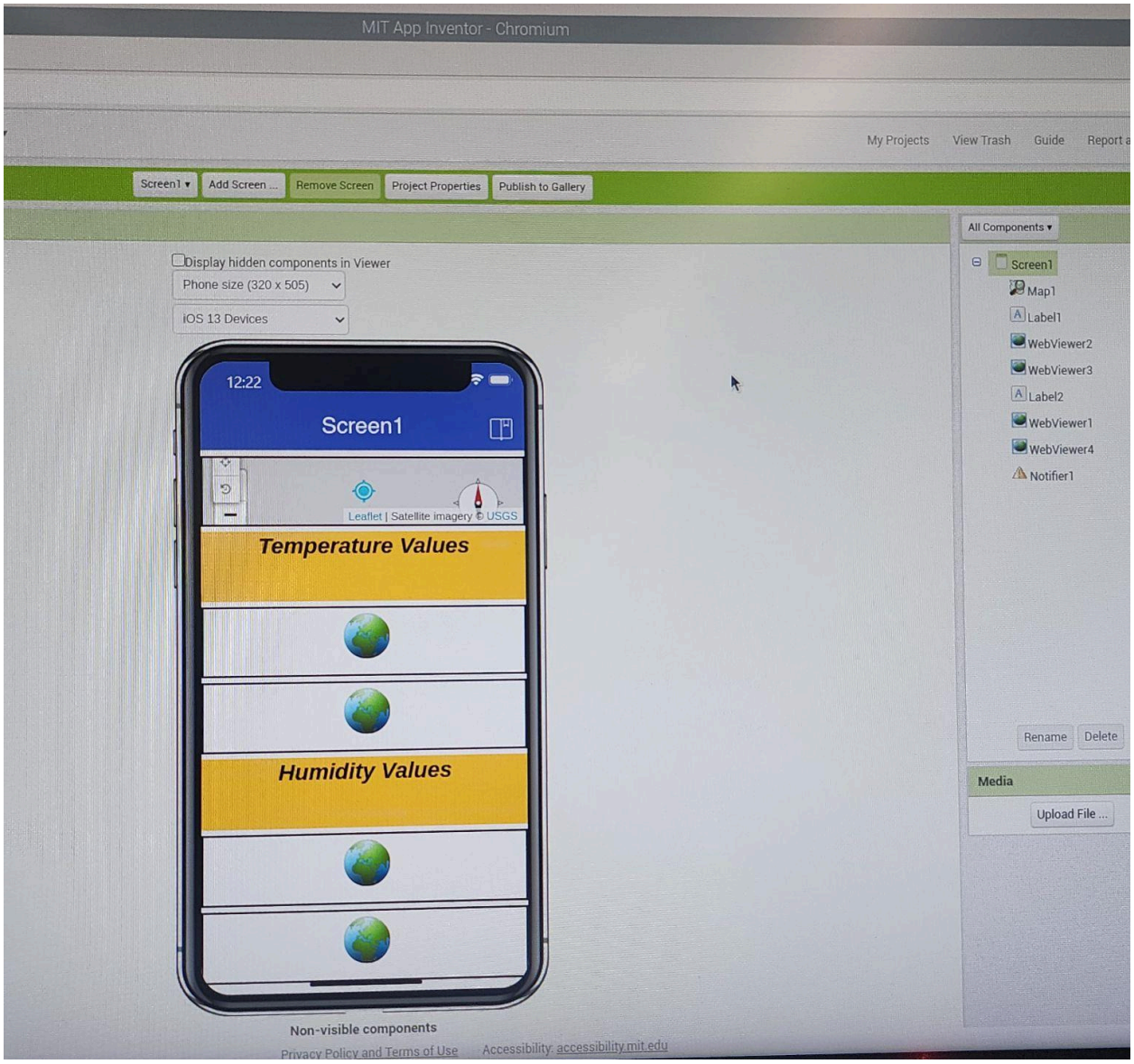
Updating Chocolate Amount In Real-Time



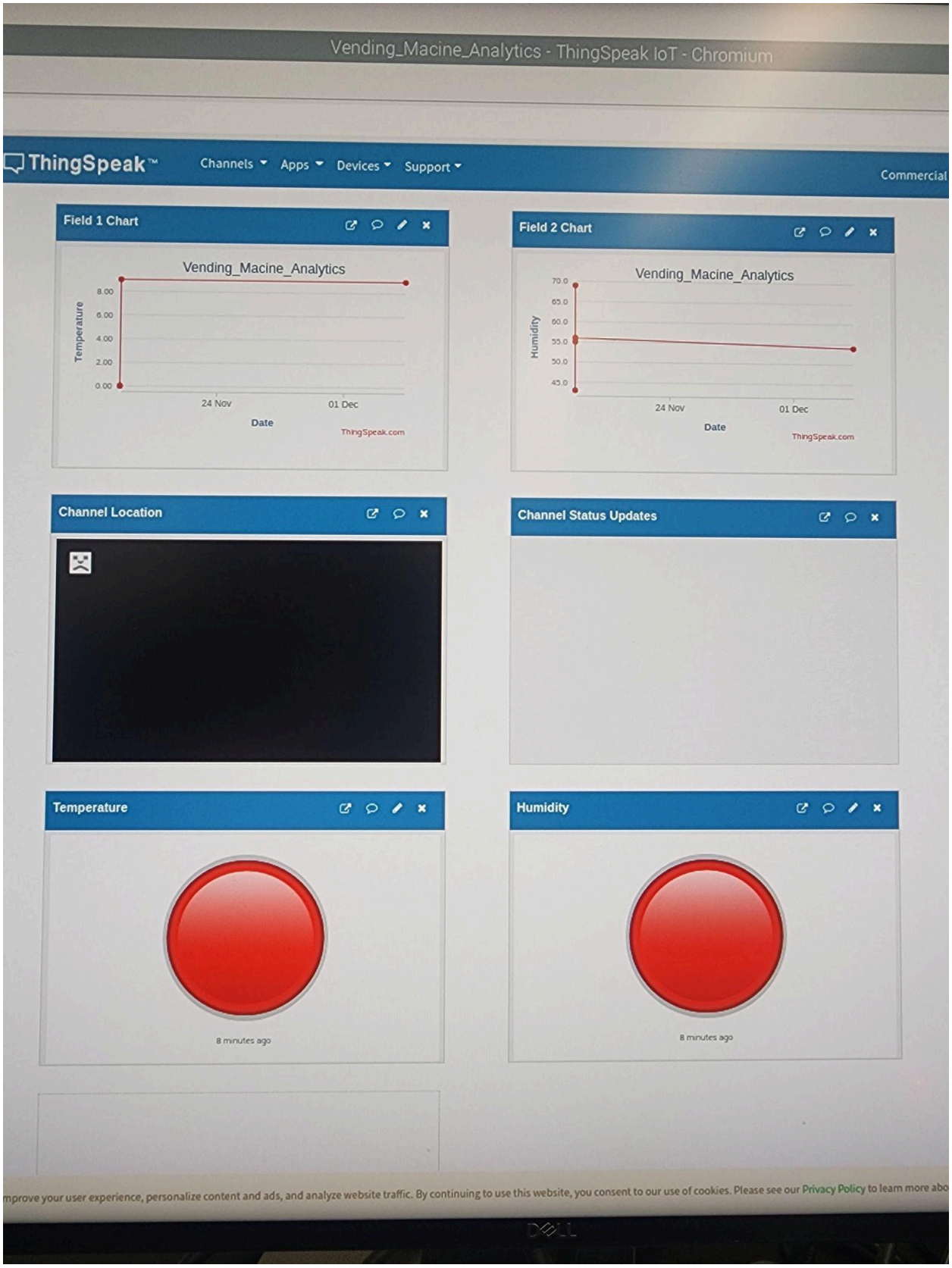
Updated Inventory



Mit App Companion



Mit App Inventor View



Thinkspeak View