



## **Introduction to Information Technology Project**

*Student 1:* Tran Manh Duc - 10422019  
*Student 2:* Pham Quang Dung - 10422093  
*Student 3:* Luu Danh Thanh Khoi - 10422039  
*Student 4:* Nguyen Huu Oanh - 10422112  
*Student 5:* Truong Thanh Hai - 10422025  
*Student 6:* Luu Huynh Phuc Hau - 10422026  
*Student 7:* Nguyen Do Quynh Nguyen - 10422108

## Content

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Adafruit Feeds</b>	<b>2</b>
<b>3</b>	<b>Adafruit Dashboard</b>	<b>3</b>
<b>4</b>	<b>Python Source Code</b>	<b>3</b>
<b>5</b>	<b>Extra Features</b>	<b>5</b>
5.1	Trading function . . . . .	5
5.2	Total calculating . . . . .	5
<b>6</b>	<b>Conclusion</b>	<b>5</b>

## 1 Introduction

The function of the project is to provide a concise and user-friendly dashboard of financial information for investors and traders. Trading in the world is much concerned by many of us, from big investing funds to individual investors. Viewing values of different materials is highly important. Most applications on the web provide users with one entry at a time. Viewing multiple values simultaneously in real time is efficient and necessary in planning future investing plans and decisions. This project allows such to enable more accuracy and better views of the stock market.

## 2 Adafruit Feeds

The repository consists of the following feeds:

**pricegold:** This feed is used to display the price of gold in a specific time.

**priceoil:** This feed is used to display the price of oil in a specific time.

**pricesteel:** This feed is used to display the price of steel in a specific time.

**pricecoffee:** This feed is used to display the price of coffee in a specific time.

**buttontrade:** This feed is used to illustrate if trading is enabled or not.

**buttonwithdraw:** This feed is used to illustrate if withdrawal is enabled or not.

**equation:** This feed is used to illustrate the sum of values in a specific time.

**sum:** This feed outputs the sum of the aforementioned equation onscreen.

<input type="checkbox"/> <a href="#">buttontrade</a>	button1	0
<input type="checkbox"/> <a href="#">buttonwithdraw</a>	button2	0
<input type="checkbox"/> <a href="#">equation</a>	equation	$x1 + x2 + x3 + x4$
<input type="checkbox"/> <a href="#">pricecoffee</a>	pricecoffee	10
<input type="checkbox"/> <a href="#">pricegold</a>	sensor1	4196
<input type="checkbox"/> <a href="#">priceoil</a>	sensor2	169
<input type="checkbox"/> <a href="#">pricesteel</a>	sensor3	1404
<input type="checkbox"/> <a href="#">sum</a>	sum	8013

Figure 1: List of feeds on Adafruit IO

### 3 Adafruit Dashboard

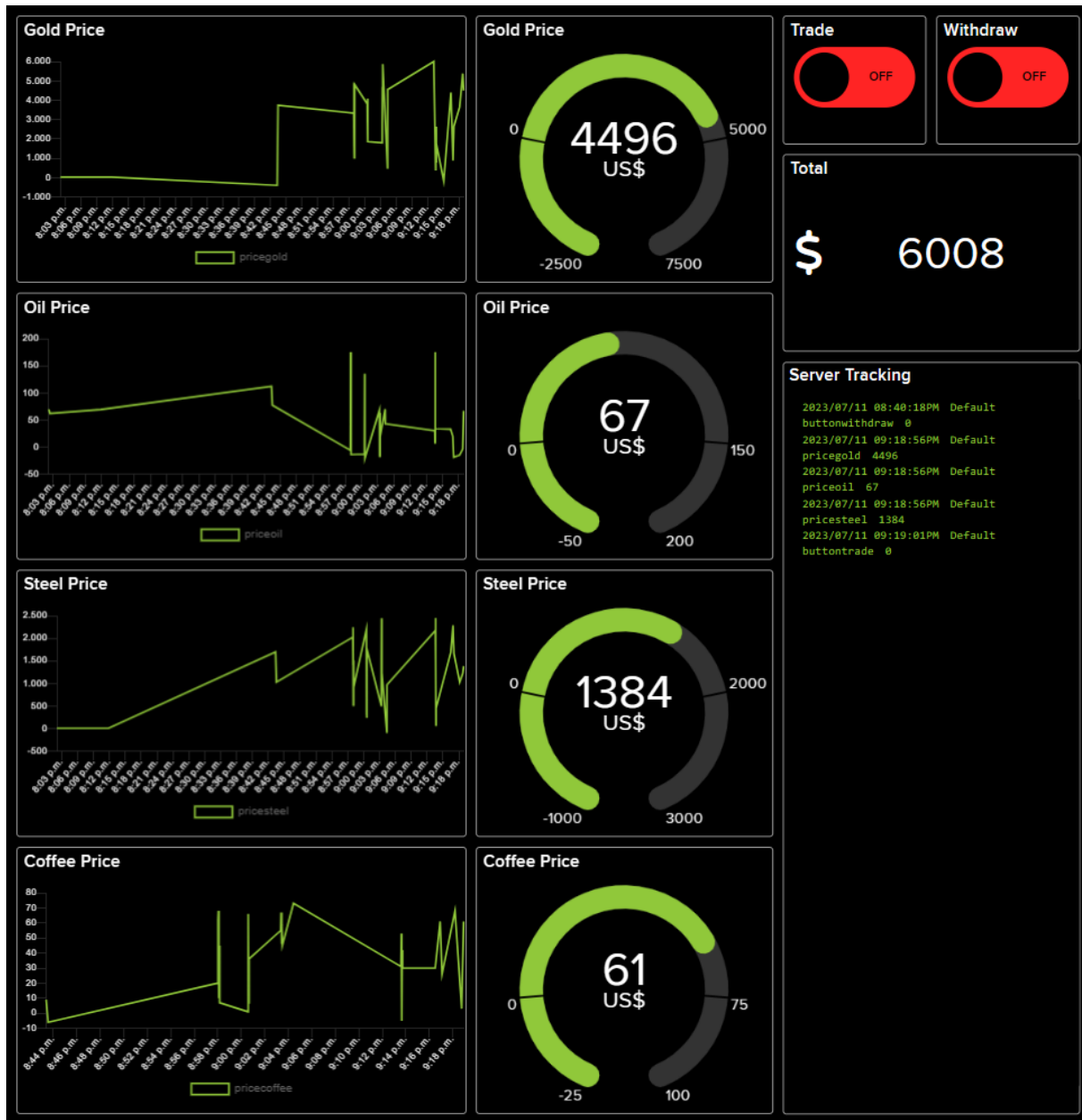


Figure 2: Dashboard on Adafruit IO

### 4 Python Source Code

The complete source code for our project is as follows:

Listing 1: Main Python source code

```
1 import sys
2 import time
3 import random
```

```
4 from Adafruit_IO import MQTTClient
5 import requests
6 import warnings
7 AIO_USERNAME = "" # username of feeds
8 AIO_KEY = "" # key changes over time
9 equation= 'x1+x2+x3+x4'
10 sum_count = False
11 warnings.filterwarnings("ignore")
12 def init_global_equation():
13     global global_equation
14     headers = {}
15     aio_url = "" # URL for the 'sum' feed
16     x = requests.get(url=aio_url, headers=headers, verify=False)
17     data = x.json()
18     global_equation = "x1+x2+x3+x4"
19     print("Get latest value:", global_equation)
20 def modify_value(x1,x2,x3,x4):
21     result= eval(equation)
22     print(result)
23     return result
24 def connected(client):
25     print("Server connected ...")
26     client.subscribe("button1")
27     client.subscribe("button2")
28     client.subscribe("equation")
29     client.subscribe("pricecoffee")
30     client.subscribe("sensor1")
31     client.subscribe("sensor2")
32     client.subscribe("sensor3")
33     client.subscribe("sum")
34 def subscribe(client , userdata , mid , granted_qos):
35     print("Subscribed!")
36 def disconnected(client):
37     print("Disconnected from the server!")
38     sys.exit (1)
39 def message(client , feed_id , payload):
40     global sum_count
41     print("Received: " + payload)
42     if feed_id == "buttontrade":
43         if payload == "1":
44             sum_count = True
45         elif payload == "0":
46             sum_count = False
47 client = MQTTClient(AIO_USERNAME , AIO_KEY)
48 client.on_connect = connected # function pointer/callback
```

```
49 client.on_disconnect = disconnected
50 client.on_message = message
51 client.on_subscribe = subscribe # meaning: on_cue = call function
52 client.connect()
53 client.loop_background()
54 global_equation = 'x1+x2+x3+x4'
55 init_global_equation()
56
57 while True:
58     time.sleep(10) # updating every interval
59     x1 = random.randint(-500, 6000)
60     x2 = random.randint(-25, 175)
61     x3 = random.randint(-500, 2500)
62     x4 = random.randint(-10, 80)
63     #x = random.randint(0, 1)
64     result = modify_value(x1, x2, x3, x4)
65     client.publish("sensor1", x1)
66     client.publish("sensor2", x2)
67     client.publish("sensor3", x3)
68     client.publish("priccoffee", x4)
69     if sum_count:
70         client.publish("sum", result)
```

---

## 5 Extra Features

There are two extra features in this project: Trading function and Total calculating function.

### 5.1 Trading function

This feature is an on-off button allowing users to trade and buy when opportunities appear. It will activate the second extra feature.

### 5.2 Total calculating

This second feature activates as soon as the first one is turned on. Then, the total price of all indexes on the dashboard will be calculated.

## 6 Conclusion

This project, with the ability to showcase numerous values of worldwide goods at the same time, is aimed to aid traders, investors and the like to make better conclusions of the stock market situation, hopefully increasing their profits.