report on project

**TOPIC: Fractionating tower or steam distillation in augmented reality with iot.**

**PROJECT DETAILS: Temperature sensors CAD modelling in solidworks unity.**

Team Members

Dheeraj Maurya

Pratima

Mentor

Mr. Abhay & Mr. Ashish

Index

1. Abstract
2. Components Used
3. Working
4. Procedure
5. Problem Encountered And Tackled
6. Result

**ABSTRACT**

**Augmented Reality Based Sensor Data Display**

Augmented Reality and Mixed Reality are the future of human interaction with the real world. Augmented Reality is a technology that allows projecting live images, videos or media in the real-world environment along with the real objects visible through a camera. The technology allows interacting with the real world environment while manipulating it through computer-generated graphical elements and virtual interfaces. It is a way of modifying the view of real world environment by a computer.

In this project, Augmented Reality is used to monitor the sensor data supplied through a microcontroller (NodeMCU) board. The IOT platform used in the project is Thinger.io. The NodeMCU (ESP8266 Wi-Fi SoC) is installed in a remote place with DHT 11 Temperature and Humidity Sensor interfaced to it. The board is connected to the cloud server i.e., Thinger.io and sends sensor data over the air to an Augmented Reality . The app projects the sensor data as the camera of the Android phone focus on a target image.

The Augmented Reality App used in the project is built using Vuforia Augmented Reality Software Development Kit along with the Unity Editor. The development platform allows compiling a readily available database uploads target images and creates UI elements on the Unity Editor. The user interface can be easily linked with the downloaded database by dragging and dropping UI elements. The application can be then linked with the data logging URL of the ESP8266 OR (NodeMCU) board to capture data in real time. The platform allows compiling the project to a cross platform mobile application for use on any smartphone.

**COMPONENTS USED**

Circuit Device:

1. Microcontroller (NodeMCU /ESP8266 WiFi SoC)
2. DHT11 Temperature and Humidity sensor
3. Resistor 10k ohm
4. Jumper Wires

Mobile Device:

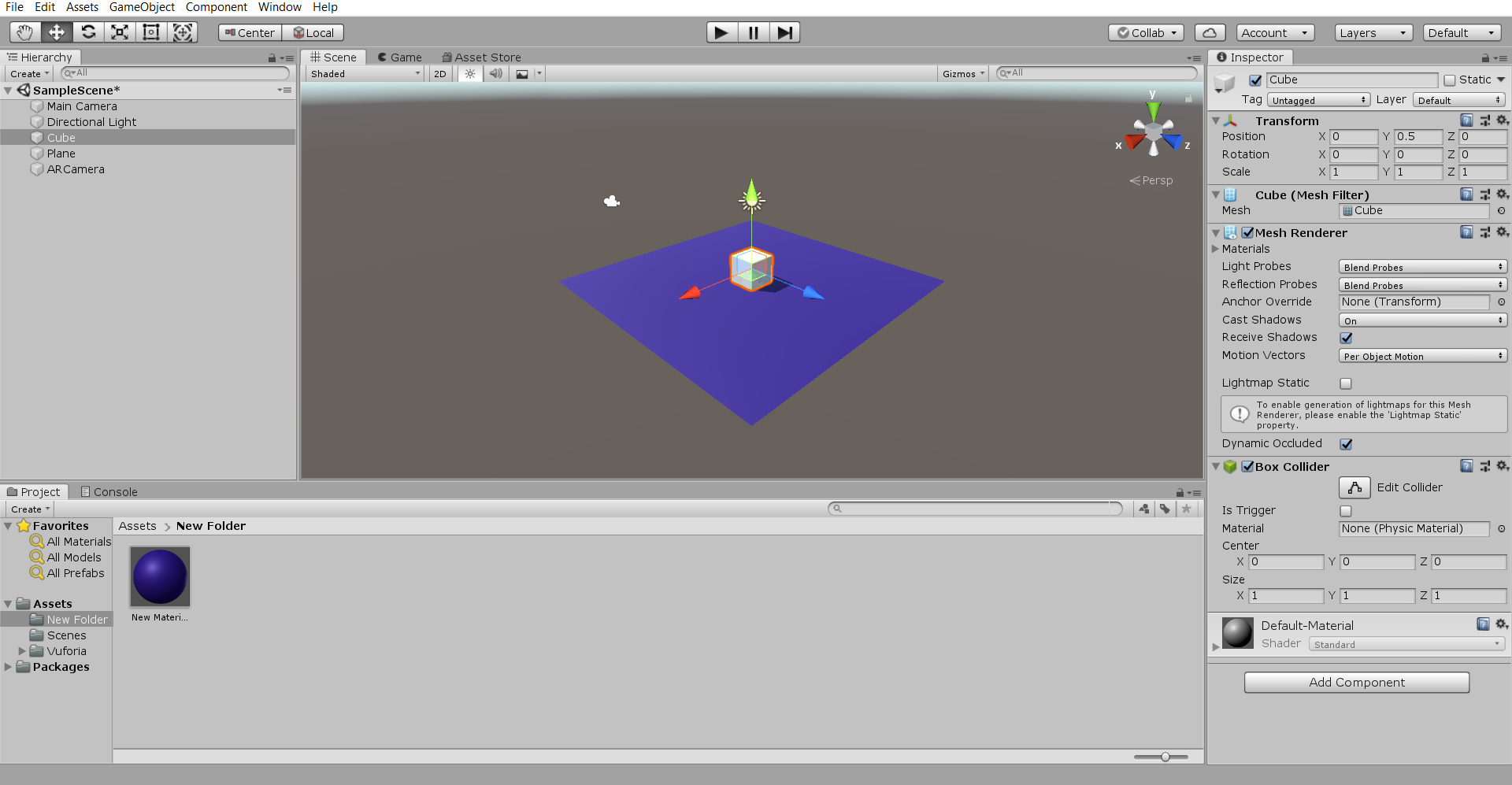
* Any Android Smartphone with internet connection

**LEARNING BASICS OF UNITY**

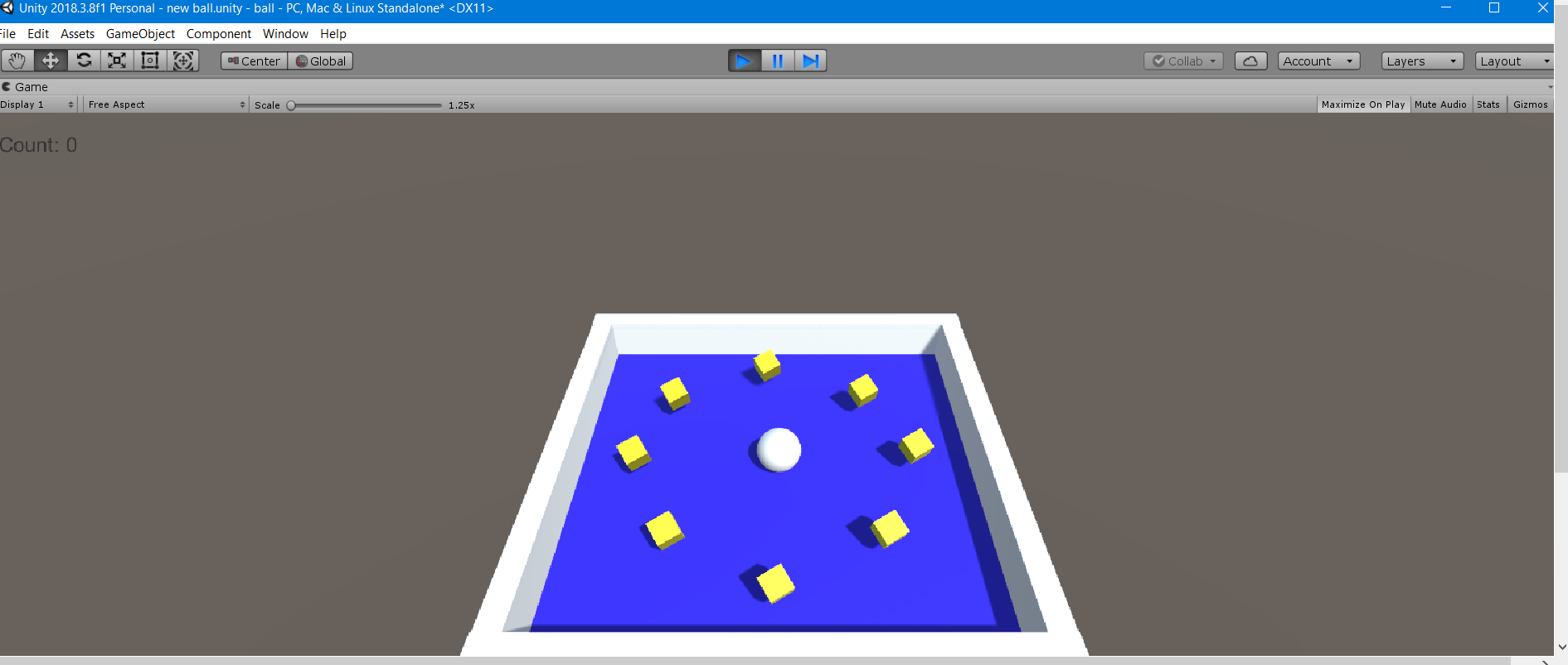
1. Inorder to understand the working procedure of unity ,we have learned how to make a game in unity.

2. The game which we have learned is rolling a ball game .

# **( Initial stage of learning)**

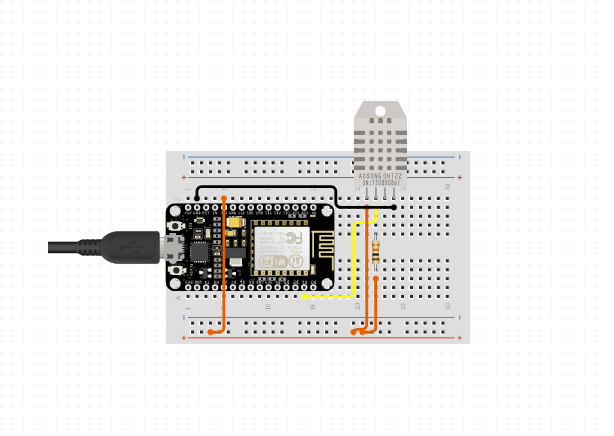


**# (Finalised result)**



**WORKING**

Circuit connection



* Place NodeMCU
* Place DHT11
* Place resistor10k ohm

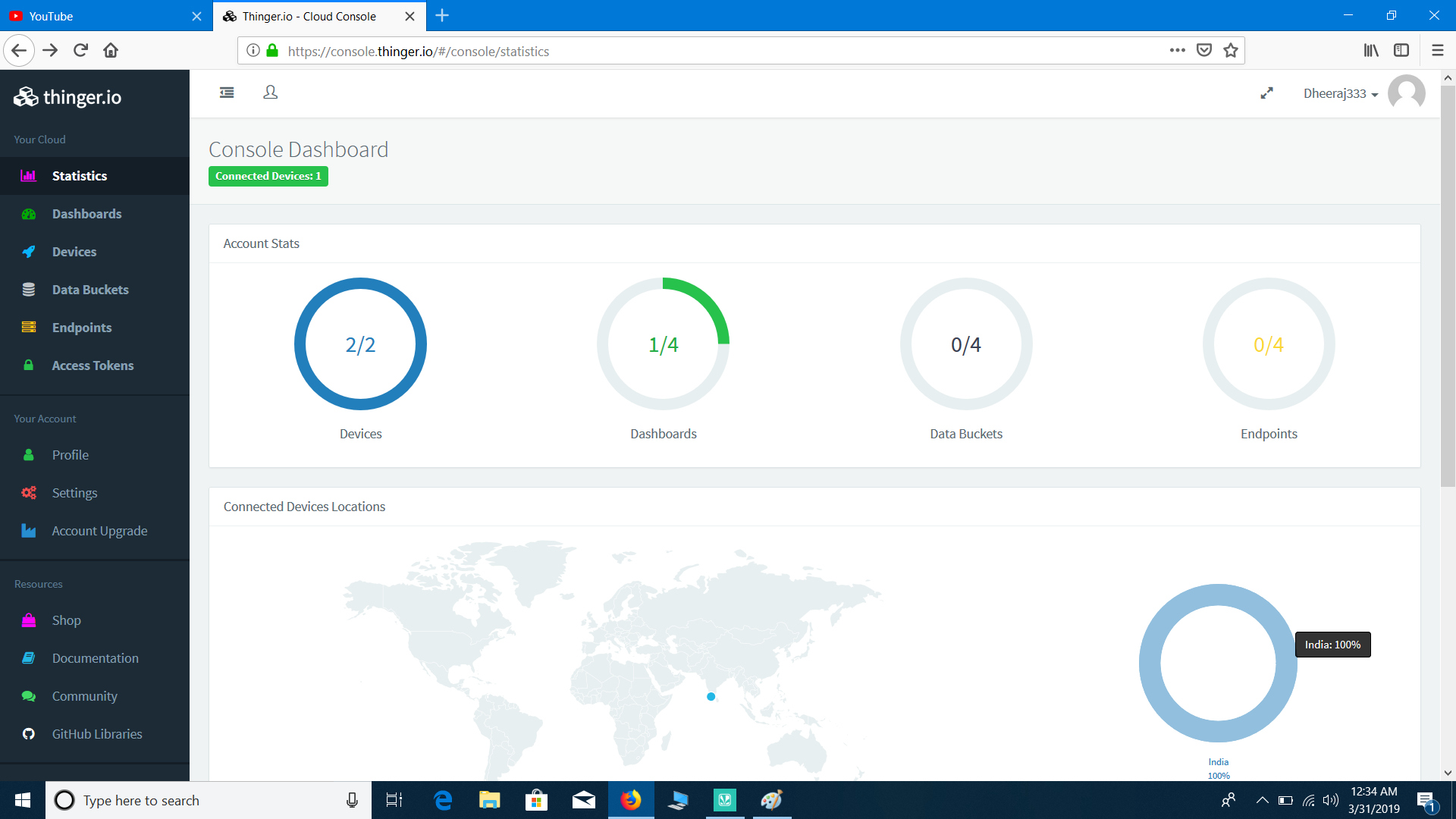
1. Connect NodeMCU
   * + NodeMCU 3V3 to Bus POS
2. Connect DHT11
   * DHT11 data to NodeMCU D1
   * DHT11 VCC to Bus POS
   * DHT11 GND to NodeMCU GND
3. Connect resistor 10k ohm
   * + Resistor 10k ohm to Bus POS
4. Connect to computer and power supply

**How the circuit works -**

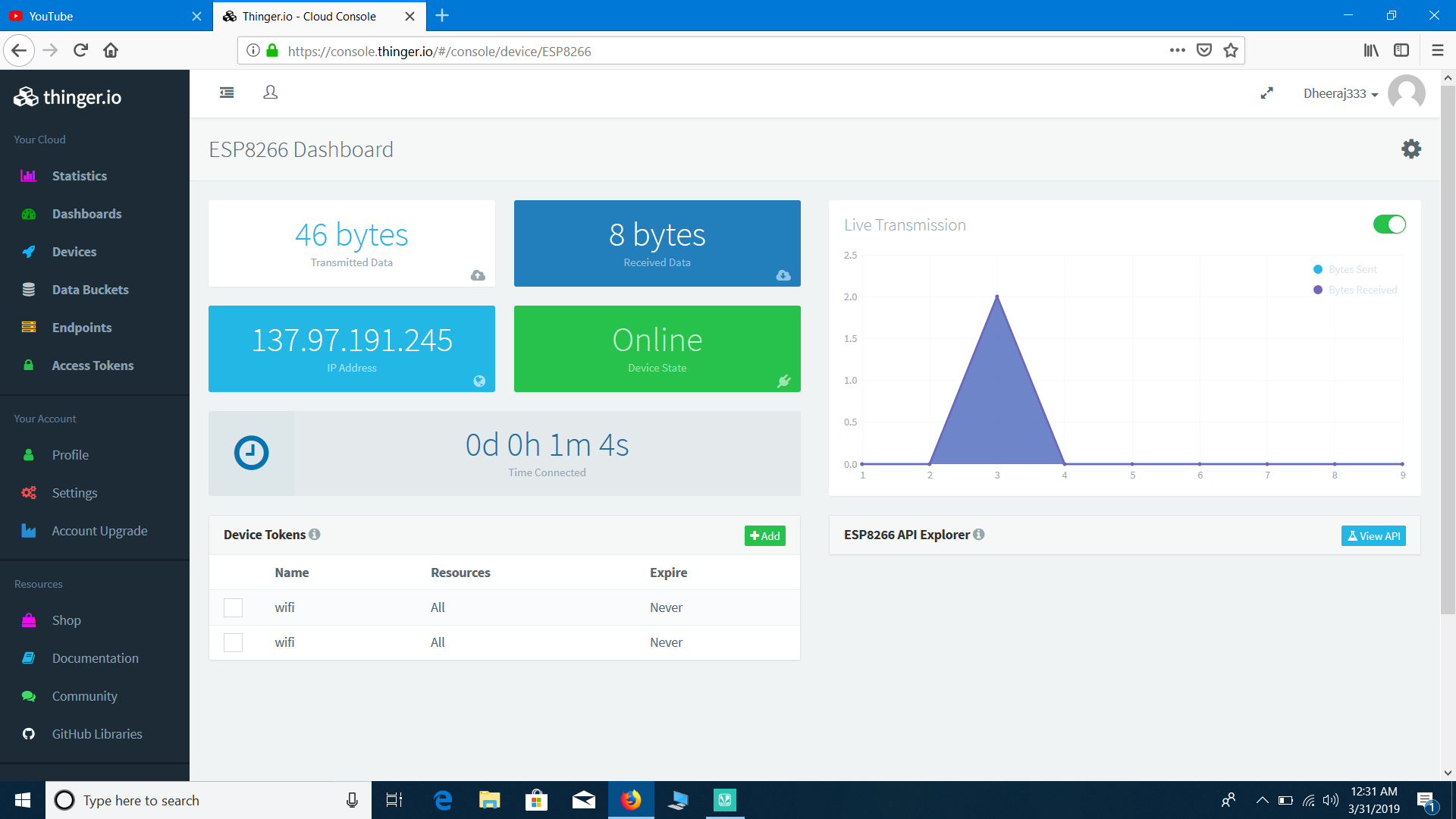
Once the program code is transferred to NodeMCU board it starts operating according to it. The board should be connected to any internet hotspot via Wi-Fi. It reads the data from sensors.

Thinger.io is an Arduino compatible IOT platform. For writing the program code, the developer needs to create an account on thinger.io website and register with his user account. The program code then can be written on Web IDE and transferred to a registered IOT board over the internet i.e., we have to upload the code and after uploading the code we have to check the website thinger.io whether it is online or not, if it shows online implies data is getting transferred.

# **Homepage of Thinger.io platform**



# **BOARD CONNECTED TO ANY INTERNET HOTSPOT VIA WIFI**



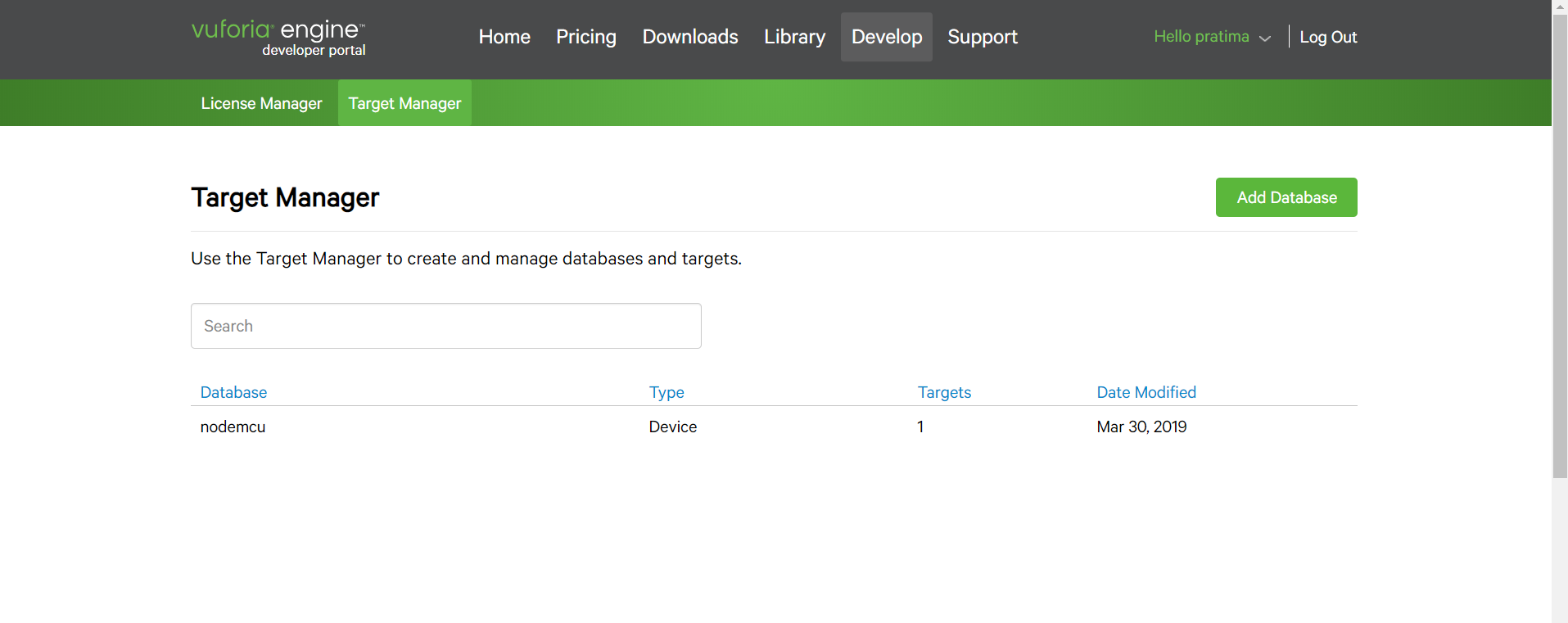
**IMPLEMENTATION OF AUGMENTED REALITY.**

The AR app has been created using Vuforia and involves using Unity Editor. The Augmented Reality app development using Vuforia involves the following steps:

- Open Vuforia Developer Portal and register an account.

- Verify email ID and log in to the registered user account.

- Tap "Develop" from the navigation bar and click on "Target Manager”.

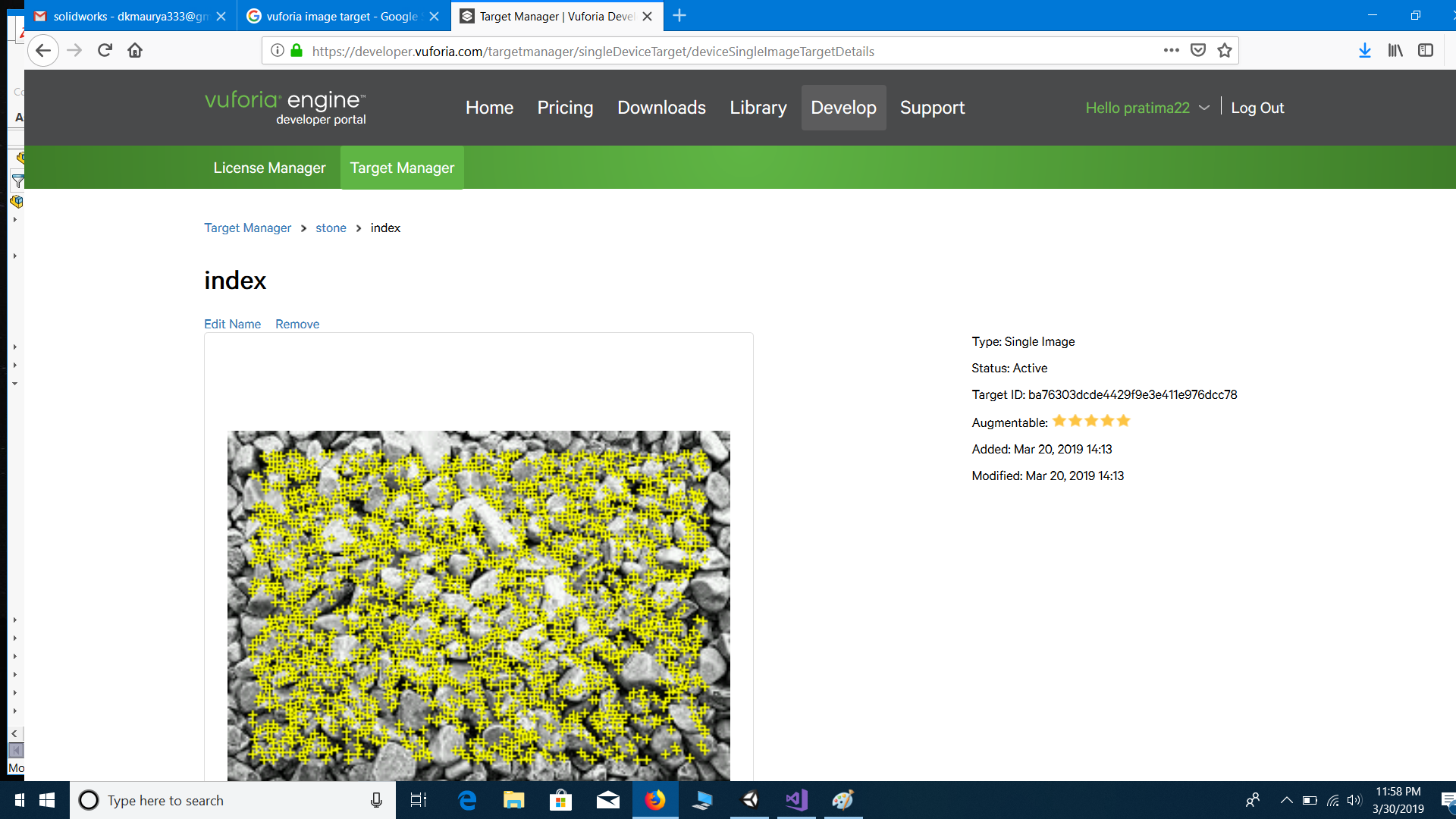


- Click on "Add Database" and enter a unique database name in the pop-up window. Keep "device" option selected and click "Create"button.

- Click on the database name from the list of databases from the refreshed page and click on "Add Target" button. Select "Single Image" and browse the target image from computer. Now the target image will be uploaded and shown on the list of targets.

# **Selected image.**

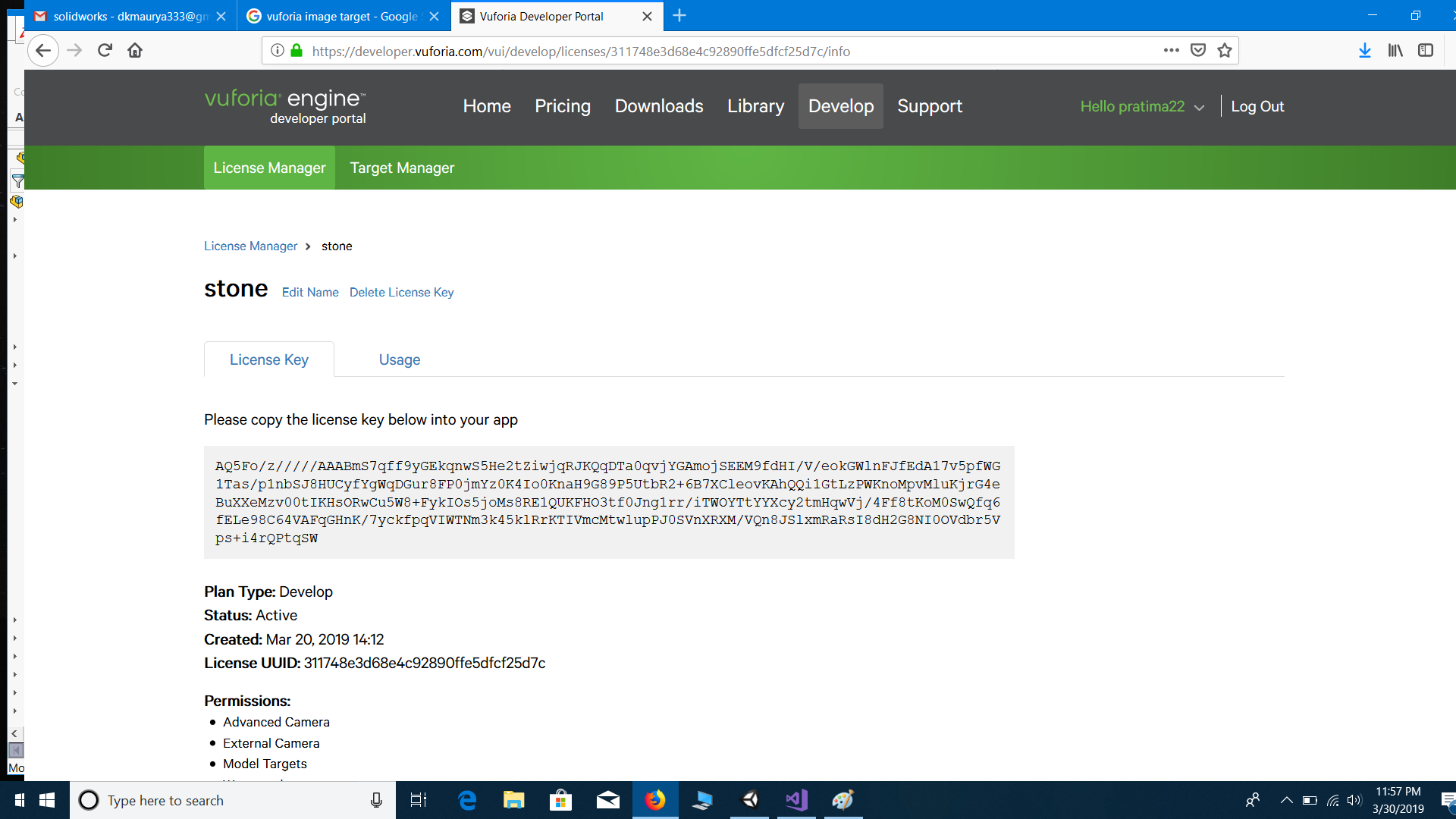




- Click on "Download Database" button and select the "Unity Editor" as the development platform in the pop-up window. Click on "Download" button. The database will be compiled and downloaded on the computer.

- Click on "Downloads" in the navigation bar and download Vuforia for Unity.

- Click on "Develop" in the navigation bar and tap "License Manager". Click on "Add License Key". Enter an application name, select mobile from devices, select starter under license key and click on "Next" button. Confirm to create the license key. On refreshing the page,the newly created license key will be displayed in the list.



- Click on the license key name and copy the license key from the web page.

- Open Unity Editor . Create a New project and save it to a folder on the computer.

- In the Unity Editor , "ARCamera" must be selected under "Hierarchy" on the left sidebar. Click on "Assets"-> "Import Package" ->"Custom Package" and browse the Vuforia Package. Once the package is imported, again follow"Assets"->"ImportPackage"-> "Custom   Package" and browse target image. Similarly, import the database earlier downloaded.

- Copy the license key and paste to "App License Key" under "Vuforia Behaviour" in the sidebar.Add <database\_name>

- Select "Image Target" under "Hierarchy" on the left sidebar and select database name under "Image Target"Canvas". Select "World Space" under "Render Mode" in "Canvas" option in the right sidebar and click twice on "Canvas" under Hierarchy" on the left sidebar.

- Enter an appropriate position, rotation and scale values and then click on "Add Component".    Select "Image". Click on target image from database and set "Texture Mode" . Click on "Apply" and drag the image from  Database under "Project" to "Source Image" under "Canvas Renderer". Change color and transparency level.

- Click on "Canvas" in the "Hierarchy" window and right click and select "UI" -> "Button". Change position, scale, color and transparency     setting. Right Click on "Button" and Click "UI" ->  "Text" to add text on the button. Select transform, font and paragraph properties as desired.

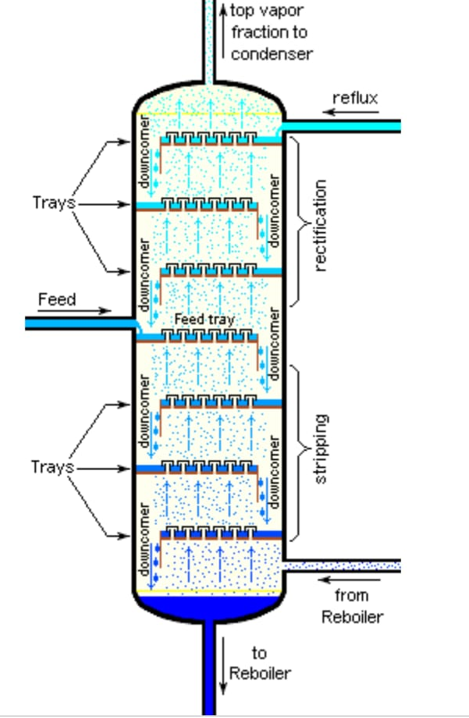


Solidworks

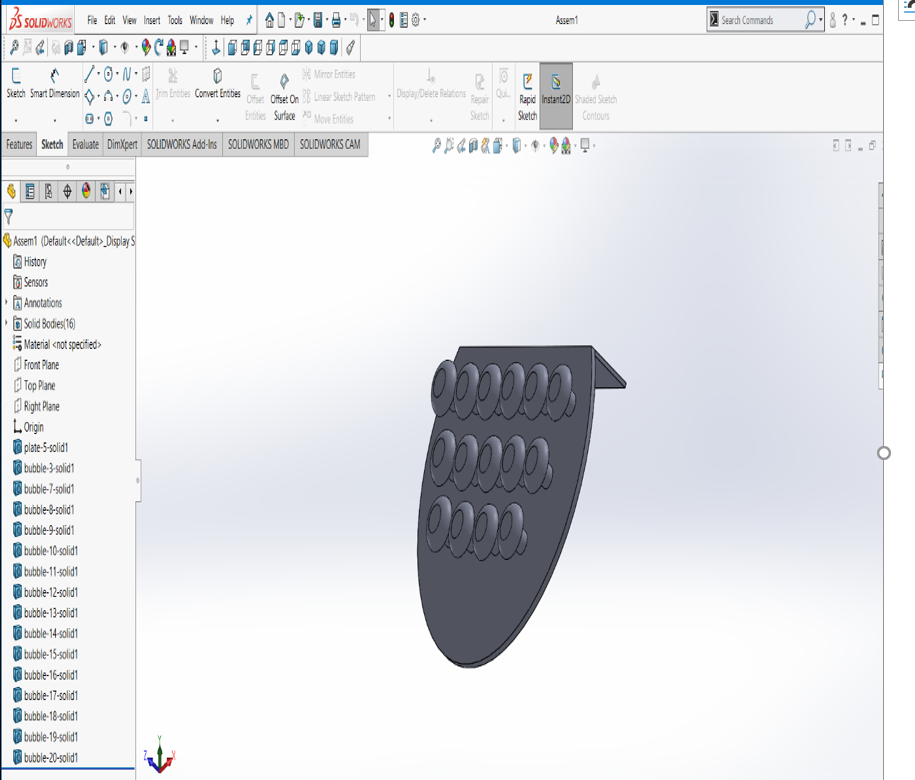
**SOLIDWORKS** is a very productive 3D CAD software tool, with its integrated analytical tools and design automation to help stimulate physical behaviour such as kinematics, dynamics, stress, deflection, vibration, temperatures or fluid flow to suit all types of design.

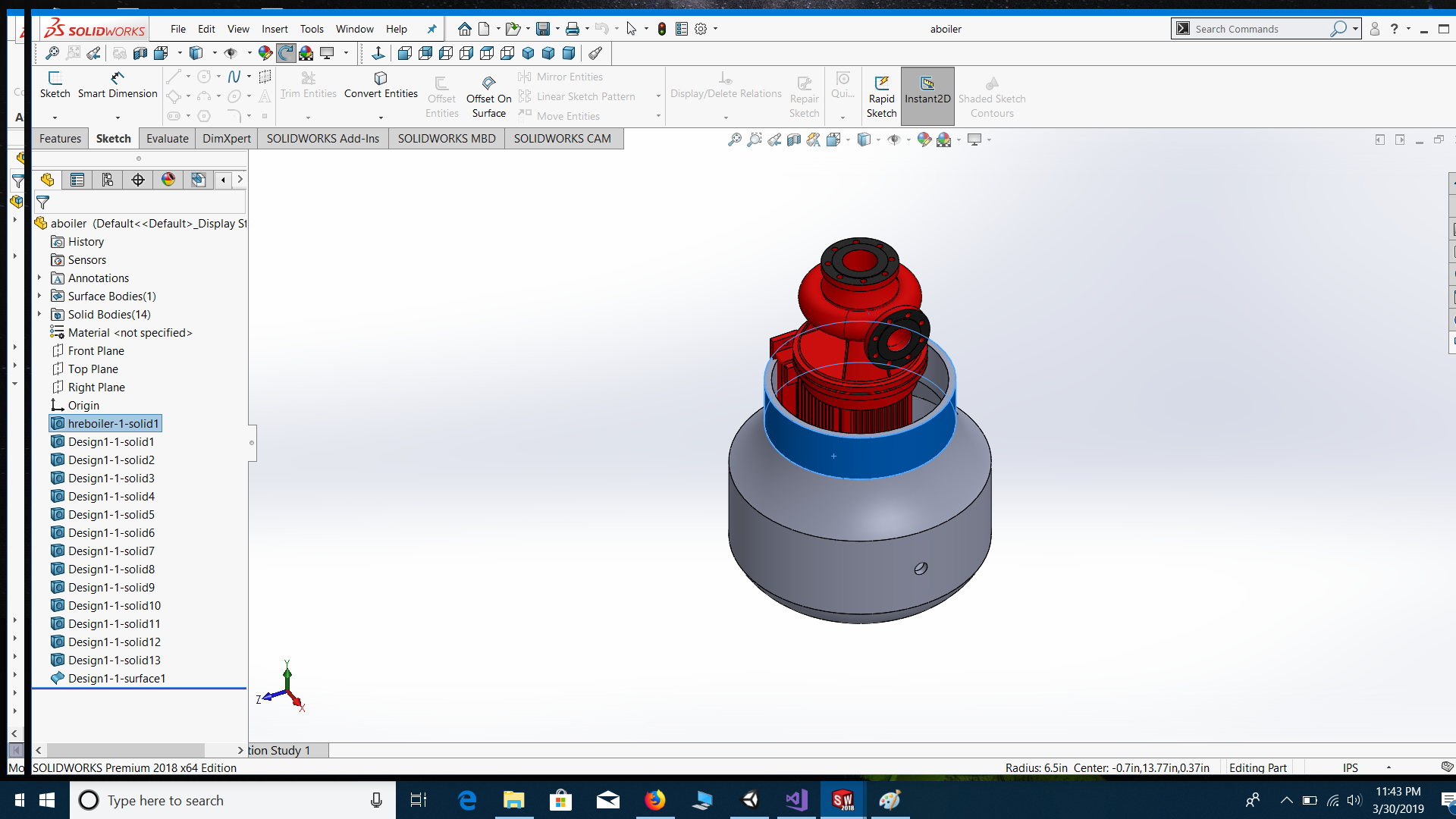
FRACTIONATING TOWER

A **fractionating column** is an essential item used in distillation of liquid mixtures so as to separate the mixture into its component parts, or fractions, based on the differences in volatilities. **Fractionating** columns are used in small scale laboratory distillations as well as for large scale industrial distillations.

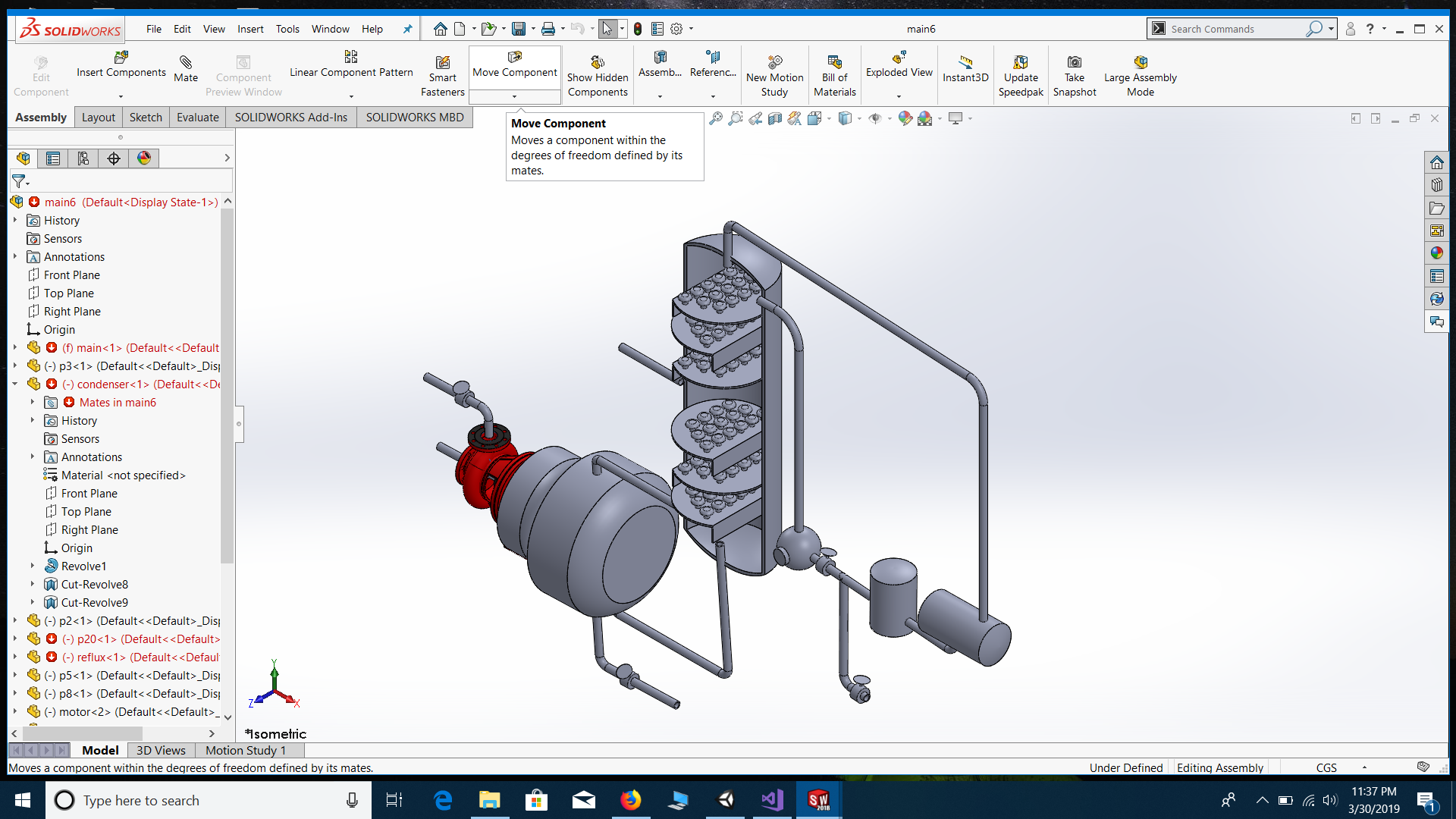


1.**Initial learning stage of solidworks**

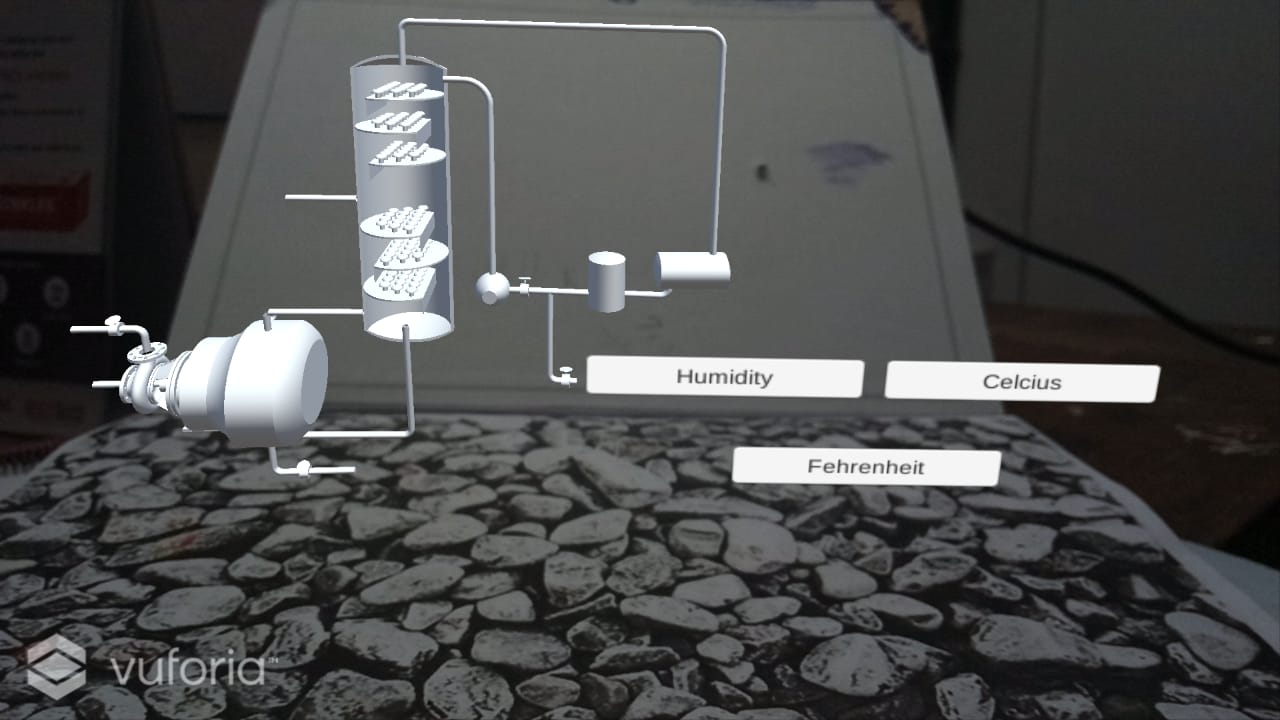




2**. The finalised outlook of solidworks**

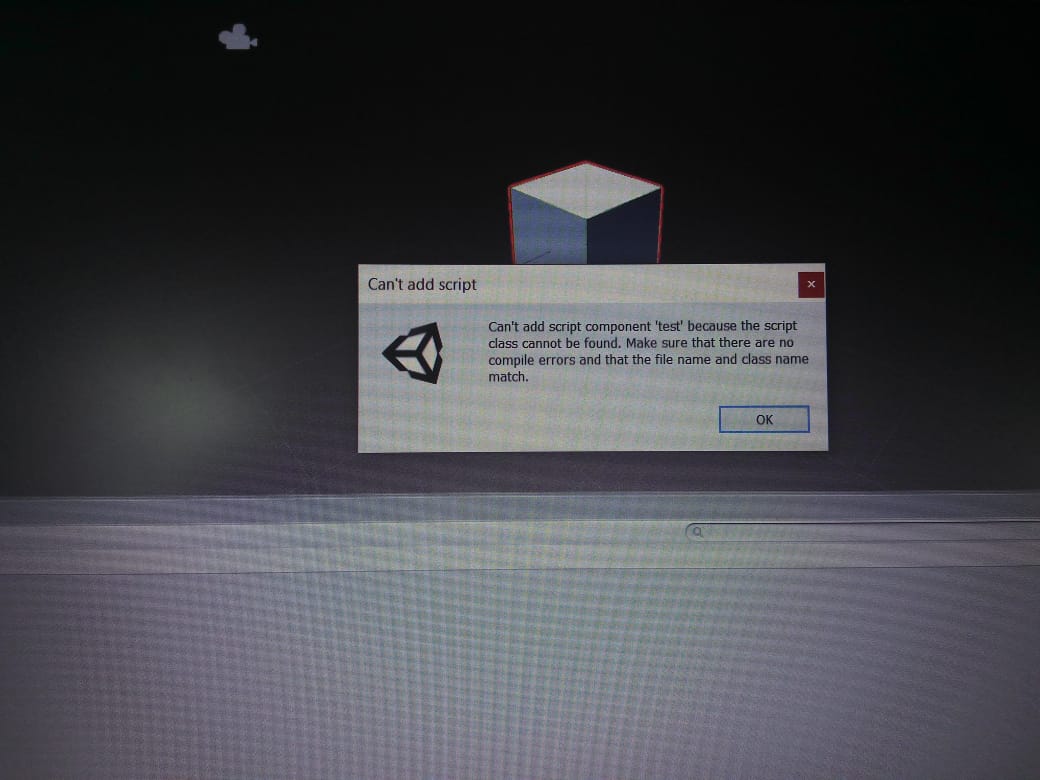


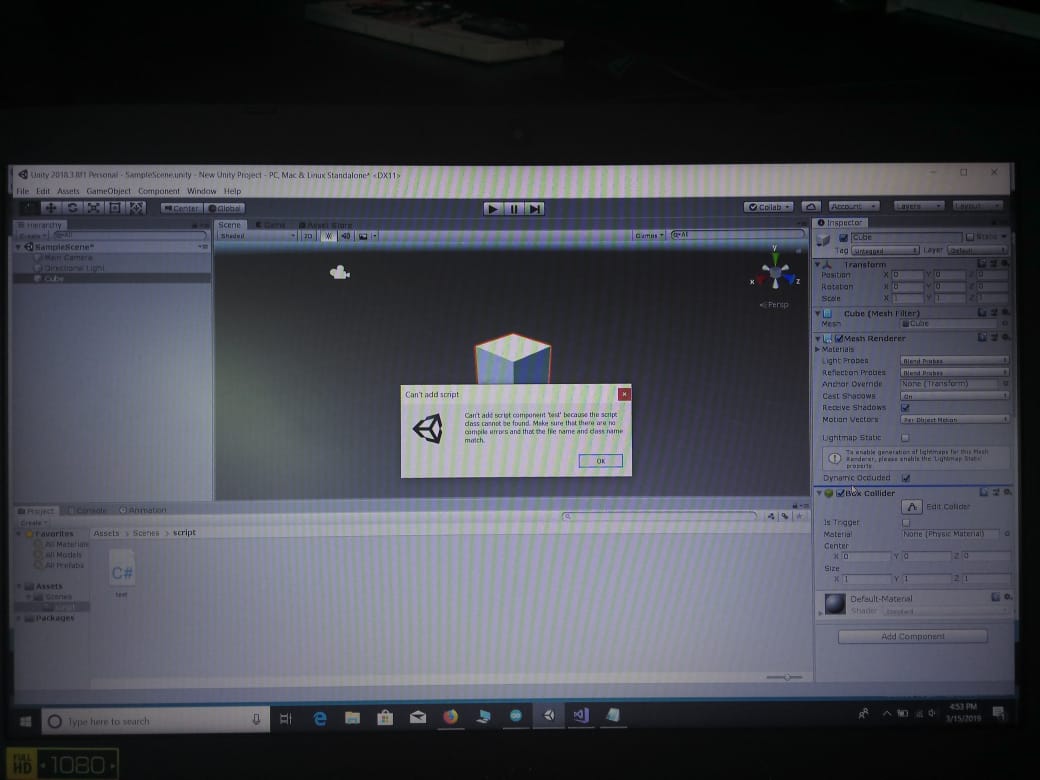
3**. Importing CAD model in unity**



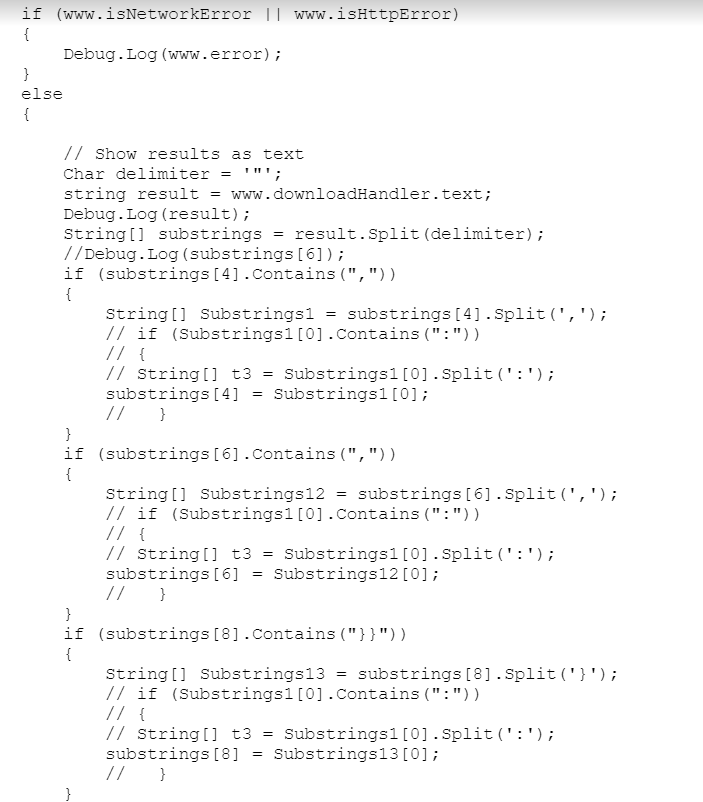
Problem encountered and tackled.

1.we have faced some problem while dealing with unity.



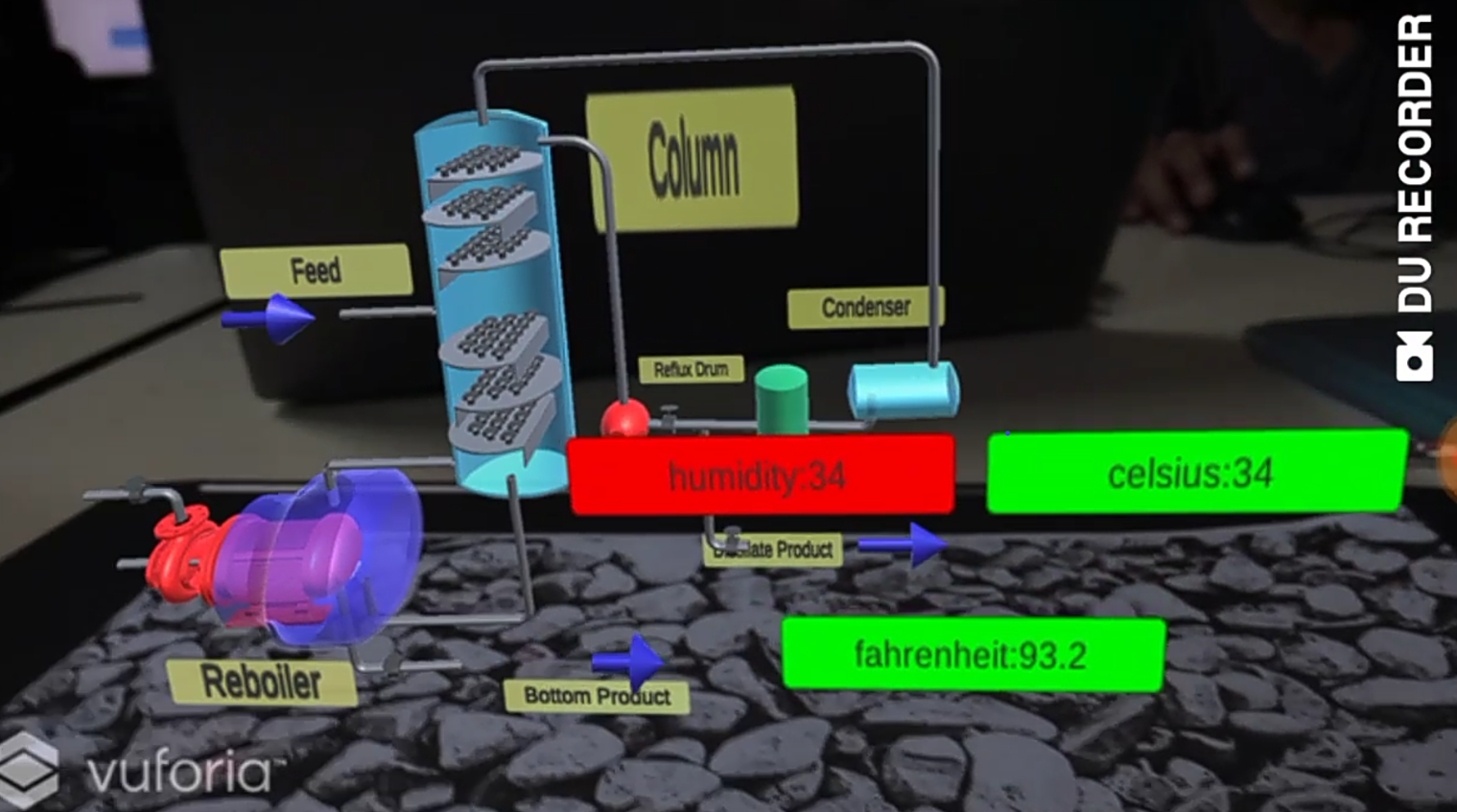


2.we have come across a problem where delimiting of strings was a difficult part but, we have solve the problem by applying some knowledge of c# programming i.e, By providing following condition:

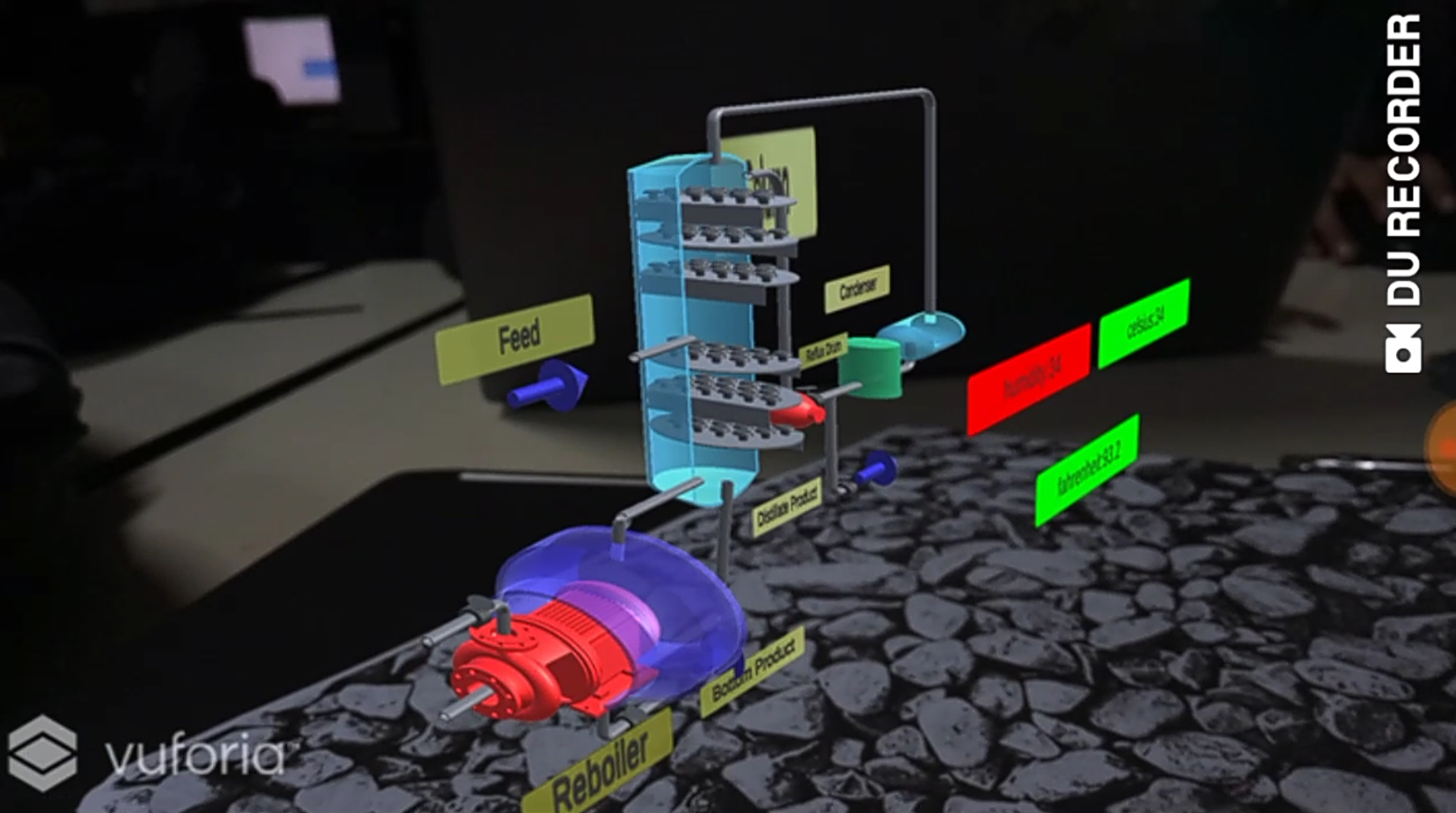


RESULT OF THE PROJECT

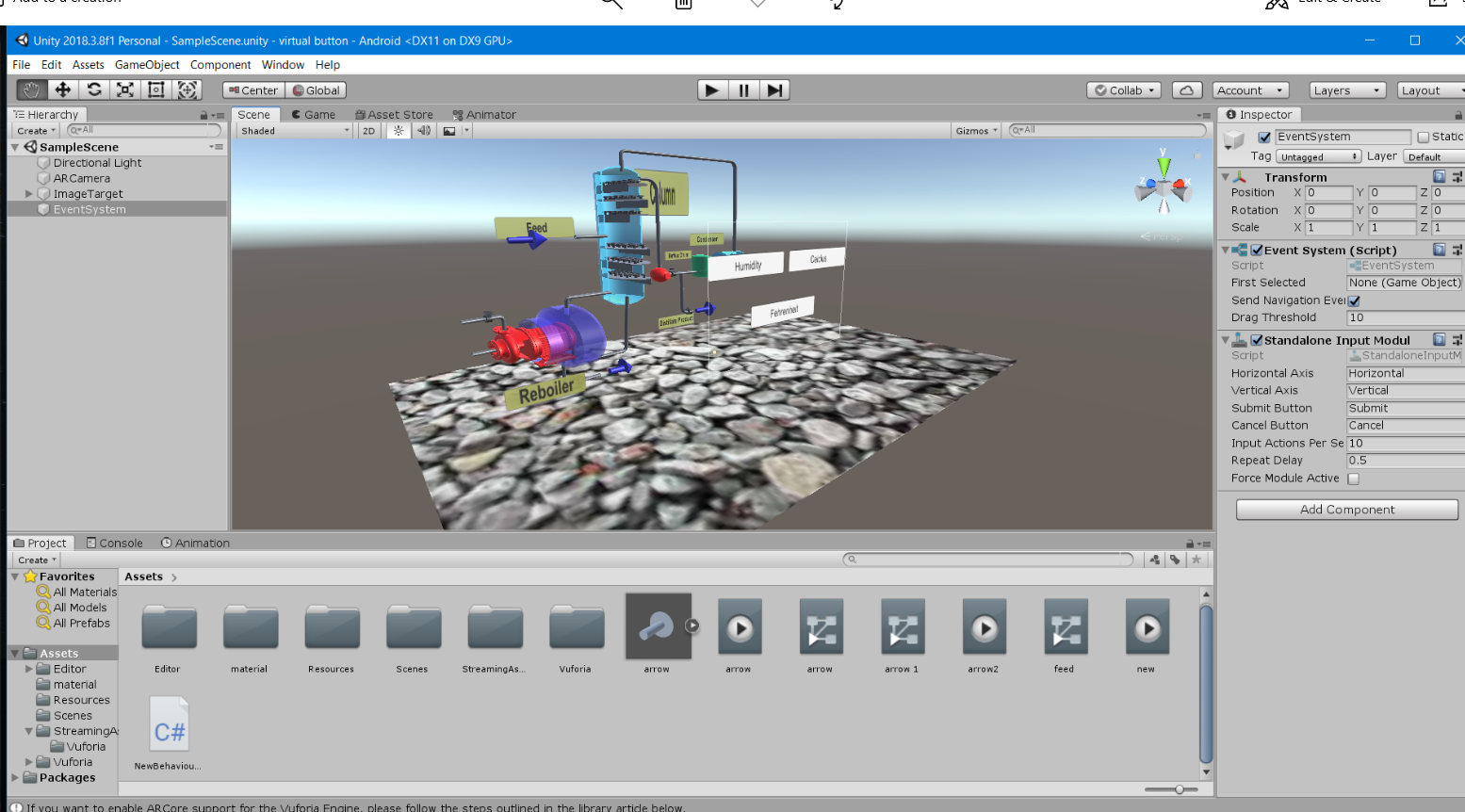
* The result of the project is to read the values of temperature in unity and the result will be produced according to the provided condition given in the code.
* If the value of temp is above 32 deg Celsius color of the button will be changed from green to red.
* The finilised result can be highlighted through the below output:



**(Side view of the model)**



**The overall output of the project.**



**Final Arduino Code**

//#include <ESP8266WiFi.h>

#include <ThingerESP8266.h>

#include "DHT.h"

#define USERNAME "Dheeraj333"

#define DEVICE\_ID "ESP8266"

#define DEVICE\_CREDENTIAL "SpkkB%01wuWR"

#define SSID "Lenevo" //Wifi hotspot name

#define SSID\_PASSWORD "94158232801" // Wifi connection password

#define DHTPIN D1

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

ThingerESP8266 thing(USERNAME, DEVICE\_ID, DEVICE\_CREDENTIAL);

void setup() {

pinMode(LED\_BUILTIN, OUTPUT);

dht.begin();

thing.add\_wifi(SSID, SSID\_PASSWORD);

// digital pin control example (i.e. turning on/off a light, a relay, configuring a parameter, etc)

//thing["led"] <<[](pson& in){ digitalWrite(BUILTIN\_LED, in? LOW : HIGH);};

// resource output example (i.e. reading a sensor value)

//thing["millis"] >>[](pson& out) {out=millis(); };

// more details at http://docs.thinger.io/arduino/

// thing["in\_out"]=[](pson& in, pson& out){

// out["sum"]=(long)in["value1"]+(long)in["value2"];

//out["mult"]=(long)in["value1"]\*(long)in["value2"];

//define a new resource for reading from DHT11 sensor

/\*thing["dht11"] >> [](pson& out){

out["humidity"] = dht.readHumidity();

out["celsius"] = dht.readTemperature();

out["fahrenheit"] = dht.readTemperature(true);

};\*/

}

void loop() {

//define a new resource for reading from DHT11 sensor

thing["dht11"] >> [](pson& out){

out["humidity"] = dht.readHumidity();

out["celsius"] = dht.readTemperature();

out["fahrenheit"] = dht.readTemperature(true);

};

thing.handle();

}

**Final Unity Code**

using System.Collections;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.Networking;

using System.Text.RegularExpressions;

using System;

[System.Serializable]

public class NewBehaviourScript : MonoBehaviour

{

public Button button,buttonc,buttonf;

public Text cel;

public Text hum;

public Text far;

void Update()

{

StartCoroutine(GetText());

}

IEnumerator GetText()

{

string url = "https://api.thinger.io/v2/users/Dheeraj333/devices/ESP8266/dht11";

string authorization = "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXYiOiJFU1A4MjY2IiwiaWF0IjoxNTUzNzgwODQwLCJqdGkiOiI1YzljZDA2ODI3ZTQyZjg3YWUxYzYyNTUiLCJ1c3IiOiJEaGVlcmFqMzMzIn0.\_qEcNGnnJxIIULqyeYevEwJXEdHCoP1STsyAym48Q6c";

UnityWebRequest www = UnityWebRequest.Get(url);

{

www.SetRequestHeader("AUTHORIZATION", authorization);

yield return www.SendWebRequest();

if (www.isNetworkError || www.isHttpError)

{

Debug.Log(www.error);

}

else

{

Char delimiter = '"';

string result = www.downloadHandler.text;

Debug.Log(result);

String[] substrings = result.Split(delimiter);

if (substrings[4].Contains(","))

{

String[] Substrings1 = substrings[4].Split(',');

substrings[4] = Substrings1[0];

}

if (substrings[6].Contains(","))

{

String[] Substrings12 = substrings[6].Split(',');

substrings[6] = Substrings12[0];

}

if (substrings[8].Contains("}}"))

{

String[] Substrings13 = substrings[8].Split('}');

substrings[8] = Substrings13[0];

}

string s1 = substrings[5];

string s2 = s1 + substrings[6];

cel.text = s2.ToString();

string s3 = substrings[3];

string s4 = s3 + substrings[4];

hum.text = s4.ToString();

string s5 = substrings[7];

string s6 = s5 + substrings[8];

far.text = s6.ToString();

if (s4.Contains(":20") || s4.Contains(":21") || s4.Contains(":22") || s4.Contains(":23") || s4.Contains(":24") || s4.Contains(":25") || s4.Contains(":26") || s4.Contains(":27") || s4.Contains(":28") || s4.Contains(":29") || s4.Contains(":30")||s4.Contains(":31")||s4.Contains(":32"))

{

TurnGreen();

}

else

TurnRed();

if (s2.Contains(":24")||s2.Contains(":25")||s2.Contains(":26")||s2.Contains(":27")||s2.Contains(":28")||s2.Contains(":29")||s2.Contains(":30")|| s2.Contains(":31")||s2.Contains(":32")||s2.Contains(":33")||s2.Contains(":34"))

{

TurnGreenc();

TurnGreenf();

}

else

{

TurnRedc();

TurnRedf();

}

void TurnGreenc()

{

ColorBlock colors = buttonc.colors;

colors.normalColor = Color.green;

colors.highlightedColor = new Color32(0, 246, 0, 255);

buttonc.colors = colors;

}

void TurnRedc()

{

ColorBlock colors = buttonc.colors;

colors.normalColor = Color.red;

colors.highlightedColor = new Color32(255, 100, 100, 255);

buttonc.colors = colors;

}

void TurnGreenf()

{

ColorBlock colors = buttonf.colors;

colors.normalColor = Color.green;

colors.highlightedColor = new Color32(0, 246, 0, 255);

buttonf.colors = colors;

}

void TurnRedf()

{

ColorBlock colors = buttonf.colors;

colors.normalColor = Color.red;

colors.highlightedColor = new Color32(255, 100, 100, 255);

buttonf.colors = colors;

}

void TurnGreen()

{

ColorBlock colors = button.colors;

colors.normalColor = Color.green;

colors.highlightedColor = new Color32(0, 246, 0, 255);

button.colors = colors;

}

void TurnRed()

{

ColorBlock colors = button.colors;

colors.normalColor = Color.red;

colors.highlightedColor = new Color32(255, 100, 100, 255);

button.colors = colors;

}

}

}

}

}