Technical Manual

Osama Abou Hajar - C00220135.
4th Year Software Eng.
Submission Date: 20/04/2020

Supervisor Paul Barry

INSTITUTE of TECHNOLOGY CARLOW
Institiúid Teicneolaíochta Cheatharlach

Abstract

Emergency Info Hub (EIH), Is a central website that helps the emergency services to prepare for, respond to & recover from disaster, by providing all needed data for the targeted building (E.g. Number of people, area size and emergency exits).

The main objective of this project is giving the number of trapped people under rubbles or inside a building, by tracking their number using a simple movement sensor fitted on the main gate and face detection technology and save this number to the cloud to be used when a disaster happens

This document is the user manual provides system requirements and installation instructions. It also contains the code listings.

Table of Contents

EIH: Emergency Info Hub

The project link: https://www.e-hub.ie/

What is EIH?

Emergency Info Hub (EIH), is a fourth-year student project, has been designed to be a central website helps the emergency services to prepare for, respond to & recover from disaster, by providing all needed data for the targeted building (E.g. Number of people, area size and emergency exits). The main objective of this project is giving the number of trapped people under rubbles or inside a building, by tracking their number using a simple movement sensor fitted on the main gate and face detection technology and save this number to the cloud to be used when a disaster happens.

See the documentation for more details.

EIH - Structure:

The structure of the EIH GitHub repository is:

```
.
├── eih-front-end-webapp/     # The Fron-end folder and files which create the website to output the data.
├── eih-raspberrypi-body-detect/    # The Back-end folder and files where all the hardware work takeing place.
├── LICENSE                         # EIH LICENSE agreement for any use of the project.
└── README.md
```

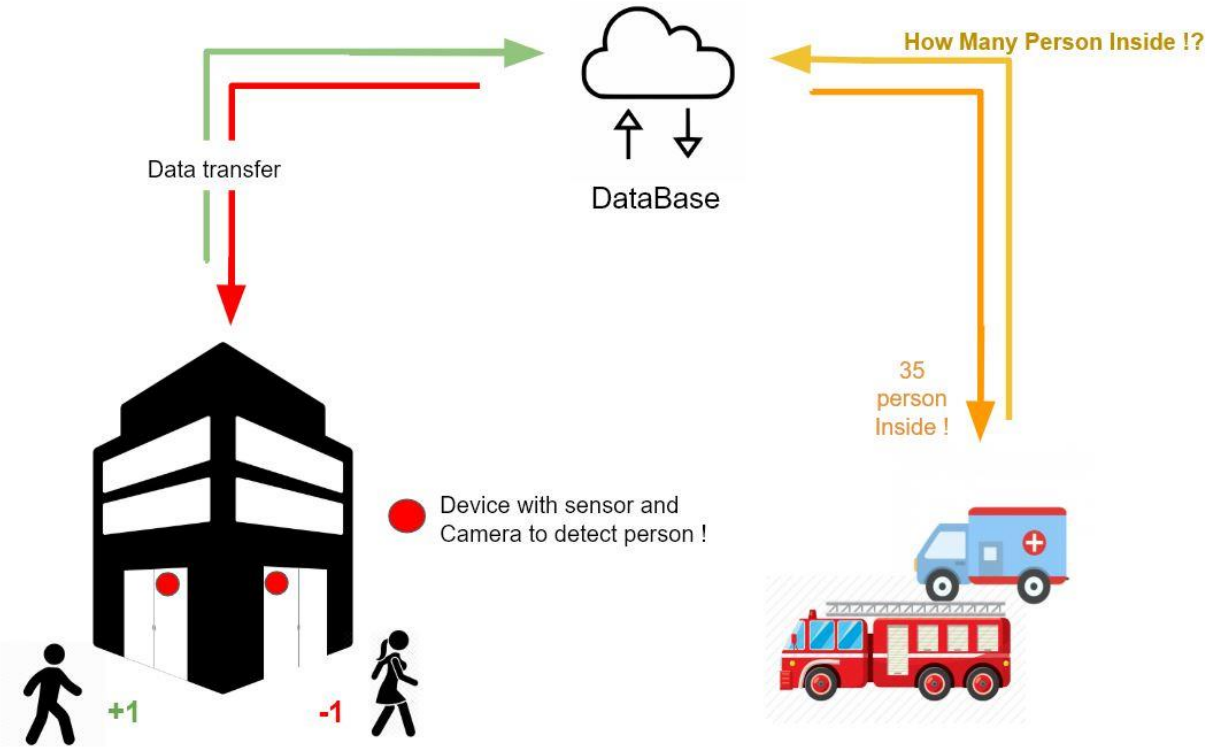
EIH Requirements

This project consists of two parts:

- Hardware: on this link (Hardware)
- Website: on this link (Website)

to run EIH you have to follow the inaction of the REAMDME.MD files for the Front-End and the Backend.

How EIH works?



It is shown in the diagram above:

- EIH briefly works as the following steps:
  - When Someone approaches the door, the sensors work, and the camera takes a picture of the people waking IN Or OUT
  - The Picture will be processed by a body-detect algorithm, to detect how many people in the picture.
  - The total number of people inside the building will be saved on the could (GOOGLE FIREBASE).
  - The Emergency Services will be able to reach this detail by the address search at any time.

EIH Story

The story of EIH started THE day when this photo was taken, back in 2012 during the Syrian war.

I was among those people running to help the persons stuck under rubbles.

During those moments the one and the only question echoing the ruins was: ANYBODY ALIVE DOWN HERE?

The hours, the minutes and even the seconds can count in finding someone alive and give them as second chance to live. Calling again and again... ANYBODY HERE? wondering every second, if we missed someone... if we left someone behind.

ANYBODY HERE? It the main reason I started EIH, BECAUSE: "EVERY LIFE MATTERS".

EIH - Front-End:

EIH front-end build using (FLASK) python framework, Jinja2 HTML, CSS, JavaScript web design languages.

The structure of the front-end is:

```
.
├── configuration/       # Configure the parameters and initial settings for some computer programs
├── static/              # Static web pages are HTML documents stored as files in the file system
├── templates/           # All the HTML files are stored inheriting the base, html page
├── app.py               # The main webApp python file where all python code and functions
├── .env                 # All the environment variables
├── requirements.txt     # EIH environment requirements.
├── test_flask.py        # Unit test, to make sure the DB connection is established and the webApp load successfully.
└── README.md            #
```

app.py

-- This is the main file for the webApp, all the function and the variables are getting processed, to run the file you need to have the following dependencies installed on your machine or the server-side for online deployment.

To run the app you need to have Python3 installed and flask, in addition to the (request), (firebase), (python-dotenv), (pyrebase) dependencies as the following:

```
curl "https://bootstrap.pypa.io/get-pip.py" -o "get-pip.py"
python get-pip.py
python3 get-pip.py
pip3 install flask
pip install requests
pip install python-firebase
pip3 install firebase
pip install python-dotenv
pip3 install pyrebase
```

This dependency will be responsible for all the processes and the connections between the Database and the WebApp.

.env File

-- The .env file contains all the environment variables, where all the Tokens, API_Keys, and other types of credentials can be stored as follows:

```
    $ export GOOGLE_API_KEY='GOOGLE-API-KEY-FOR-THE-CREATED-PROJECT-ON-THE-FIREBASE'
    $ export GOOGLE_MAP_API='GOOGLE-MAPS-API-KEY-TO-SHOW-THE-MAP-ON-RESULTS'
    $ export AUTH_DOMAIN='<YOU-PROJECT-NAME-ON-FIREBASE>.firebaseio.com'
    $ export DB_URL='https://<YOU-PROJECT-NAME-ON-FIREBASE>.firebaseio.com'
    $ export STOREG_BUKET='<YOU-PROJECT-NAME-ON-FIREBASE>.appspot.com'
    $ export SERVICE_ACCOUNT='<THE-PATH-TO-THE-DB-CONFIG-JSON-FILE>/projecteih-firebase-adminsdk-dmd9b-dfbc30ba25.json'
    $ export DB_FILE_URL='https://<YOU-PROJECT-NAME-ON-FIREBASE>.firebaseio.com/<THE-JSON-FILE-NAME-ON-FIREBASE>.json'
```

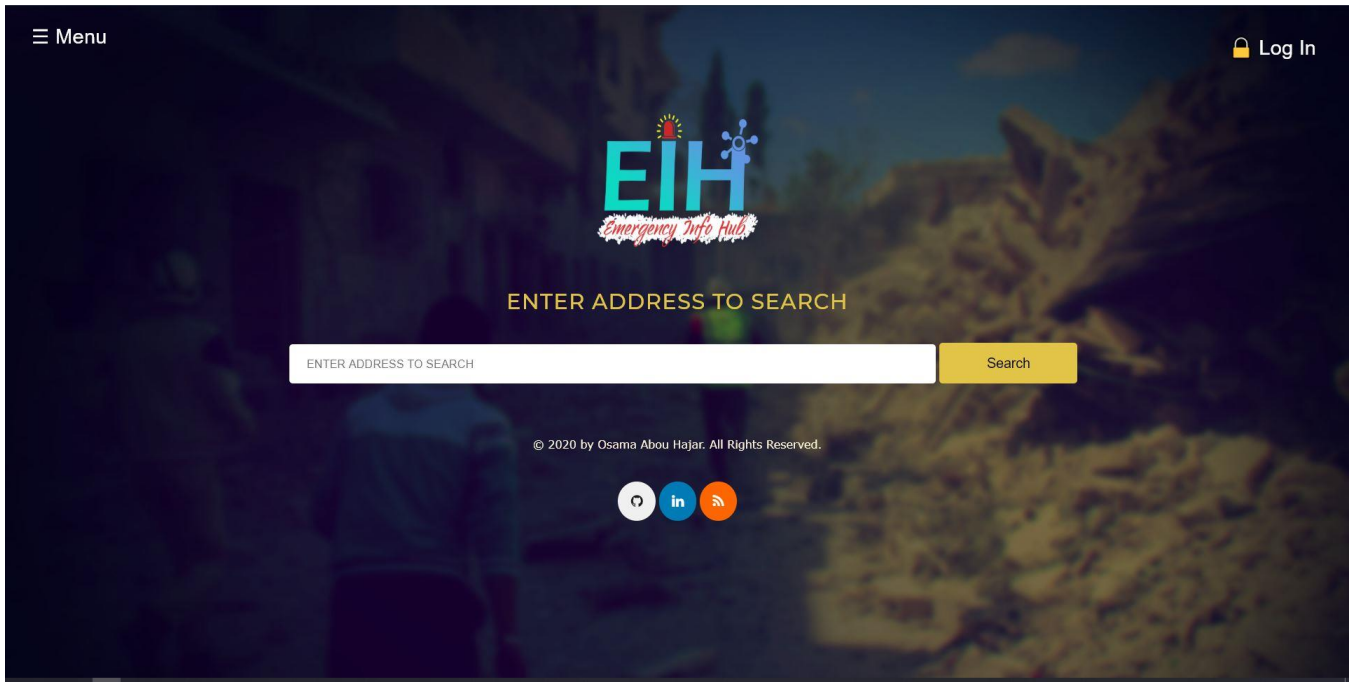
For more details about Google firebase and Google Address autocomplete and how to get this API_KEYS and Tokens, You have to register and read Google documentations

- (Goolge Firebase)
- (Goolge Maps API)

EIH - Screens & Code Functionality:

Search Screen:

It is the main screen of the project, it contains several components and functions to run.



1. **The Search input:** The search box has an integrated Address autocomplete, which creates compatibility to have one search mechanism. In this screen, the user can search for any Irish addresses The search box uses the Google Autocomplete API to give the user the ability to get the correct address and save their time during emergency moments. To be able to list the address from other countries than Ireland, the country code within the restrictions attribute in the files below should

- `templates/googlAutoCompleteGeneral.html`
- `templates/googlAutoComplete.html`

```
// Change the country code from 'IE' to the country where is EIH used
// to be able to list al locations with in this country
autocomplete.setComponentRestrictions({
    'country': 'IE'
});
```

For more information about the (Google Addresses Autocomplete) and the (countries code).

2. **The Search Button:** The search button will send a `POST` request to the `app.py` file to run the `display_form():` function. This function will be calling other functions to get the data and check from the database and check if the address is already registered and active or not.

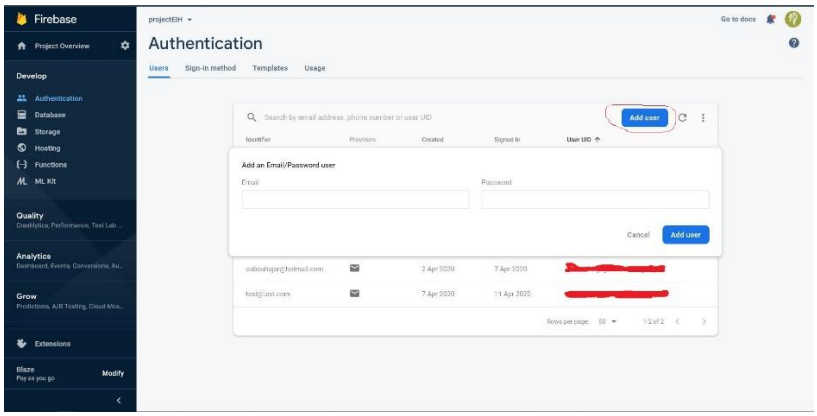
- if the functions return `True` the `retult.html` page will be displayed with all the data coming from the Database, in addition to Google Map box using the `maps2.html` file.

- if the functions return `False` the app will be redirected to the main page with alert no data for the searched address founded displayed.

Add Admin Screen:

The registration from the front-end is not an option, to gain access to the website the administration office, should grant you with the login credentials, that by registering your email and password to the user's authentications table in the firebase API side

as the following picture:



Locations Screen:

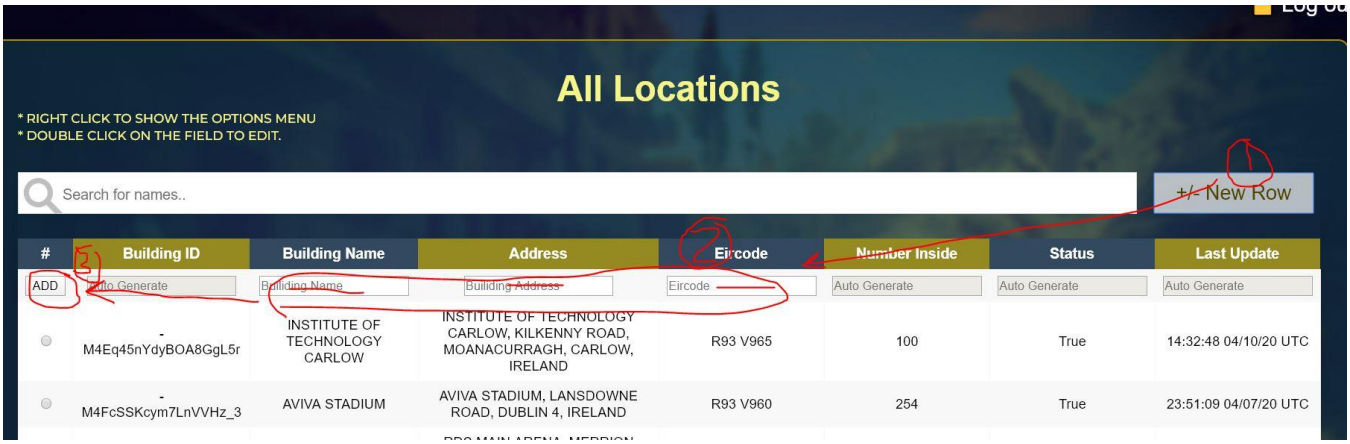
This screen will display all the locations within the databases and give a fully dynamic table to the admins to add, delete, update, and reset the number of people for each row.

- Public access all locations page:

- Admits access all locations page with the menu options on the right mouse click:

Add Building Feature

This is one of the most important parts to link the hardware to the database:
To add building, you have to login as Admin (If you have no Access you should contact the admin body create a credential to gain access). Then you follow the steps in the picture:



- Once you have registration completed the System will show you the Building registration ID, This ID should be used on the hardware side to link the backend to this actual building, as it shows in the picture.

if the Alert! The massage was telling the building is already in the system, that means the building should be activated again and the building ID is already assigned.

Update Building Feature

On the All locations page, the table supports the edit mode, double click on the row you want to edit and you can edit one row at the time, then you sleeted the radio bottom for the edited row and select the update option from the menu.

Reset Building Feature

To Reset a building, select the building row, and select the reset option from the table menu. The number of people inside the selected building will be reset to zero.

Delete Building Feature

The Delete option from the table menu will update the `active` value in the database to `False`.

```
{"active": False}
```

Activate Building Feature

The Activate option from the table menu will update the `active` value in the database to `True`.

```
{"active": True}
```

About Us Screen

The project description and the story of the project.

Case Study Screen

Redirect to an external link to show the case study for EIH as a final year project.

Contact Screen

A contact form to allow the users to contact the admins to give feedback or to report a problem.

EIH - API and Database Structure:

EIH data stored as an API on Google firebase real time database. The data are formatted as JSON with dictionary key and value.

```
data =
{
   '<BUILDING-ID>':
    {
        'active': '<BOOLEAN-VALUE>'
        "deviceId": '<STRING-SET-BY-THE-USERS-FROM-THE-BACK-END>'
        "known_name": '<THE-BUILDING-KNOWN-NAME>',
        "address": '<THE-ADDRESS-GENERATED-BY-GOOGLE-ADDRESS-AUTOCOMPLETE>',
        "eircode": '<THE-BUILDING-EIRCODE>',
        "numberOfPeopleINDetect": '<INTEGER-INITIALIZED-TO-ZERO>',
        "timeUpdated": '<AUTO-GENERATED-BY-THE-SERVER-TIME>'
    }
}
```

- **BUILDING-ID**: is auto-generated by the firebase with the push request, so when the user adds new building data the key will be generated and associated with this data.
- **ACTIVE**: Boolean data type, True to active and in work building, False for non-active or deleted address.
- DEVICE-ID: String data type, created by the user to differentiate between the hardware.
- **KNOWN NAME** is the known and public name for the address, which makes a better understanding by the reader than the proper address.
- **ADDRESS**: String datatype, the address will be autocompleted by the google autocomplete function, the address is considered the unique value to each address where the search will be depending on.
- **EIRCODE**: for the time of this project was done, the Eircode was not used by all the address in Ireland, which is hard to be used for the search and identity.
- **NUMBER-OF-PEOPLE-DETECTED**: this is the value where the number of people gets stored and updated, the initial value is zero when the building gets added.
- **TIME-UPDATED**: is the time when any changes happening to the data, this value will be assigned by the server time.

EIH - Flask Unit Test

- The file `test_flask.py` has been designed to ensure the connections to the database (firebase) has been established, and the webapp pages load successfully.

Run The Test file:

To run the test file you need to run the following command.

```
$ python3 test_flask.py -v
```

The test result will be all the tests passes.

In the case of implementing a new function to the App, you must write a new test match the structure of the tests.
For more details, you can referee to the comment within each file, or contact the developer by email.

EIH - Back-End

The structure of the back-end is:

```
.
├── cred/                   # Configure the parameters and initial settings for some computer programs
├── draft/                  # some files
├── images/                 # where the image got save to be processes by the body detect algorithm
├── detect.py               # the body detect algorithm
├── eihIR.py                # the main file for EIH back-end project
├── get-pip.py              # to install get
├── post-data-to-api.py     # The file post the data to the clould after the number get detected
├── requirments.txt         # all the for this project
├── setup.py                # to setup the project enviroument
└── README.md               #
```

Software Requirements - Installation:

To run the hardware part of EIH: 1- You need Python 3.5 or later to run mypy. You can have multiple Python versions (2.x and 3.x) installed on the same system without problems.

In Ubuntu, Mint, and Debian you can install Python 3 like this:

```
$ sudo apt-get install python3 python3-pip
```

For other Linux flavors, macOS and Windows, packages are available at

http://www.python.org/getit/

-- The file setup.py is made too easy to install all the requirements if you have Python 3 and pip3 installed on your device already, you can use the following command to set up the EIH project in full.

```
$ python setup.py
```

file setup.py

Once your setup is completed you will have the following dependencies installed on your system.

- **Dependencies**:
  - Requests
  - python-firebase
  - python-opencv
  - python-scipy
  - ipython
  - firebase
  - imutils

- python_jwt
- gcloud
- sseclient
- parse
- requests_toolbelt
- pyrebase

Export the Environment Variables

- once your setup.py done you have to set your environment variables to gain the access to your database credentials:

```
 $ export FIREBASE_DB_URL='https:<PROJECT-NAME>.firebaseio.com/<JSON-FILE-NAME>.json'

$ export REG_BUILIDING_ID='<THE-KEY-YOU-GET-AFTER-ADDING-THE-BUILDING-TO-THE-SYSTEM-FROM-THE-FRONT-END>'

$ export DEVICE_ID='<THE-DEVICE-ID-YOU-CHOSE>'

$ export PROJECT_API_KEY='<GOOGLE-FIREBASE-PRJECT-API-KEY>'
```

Hardware Requirements

You need the list of hardware parts to be connected:

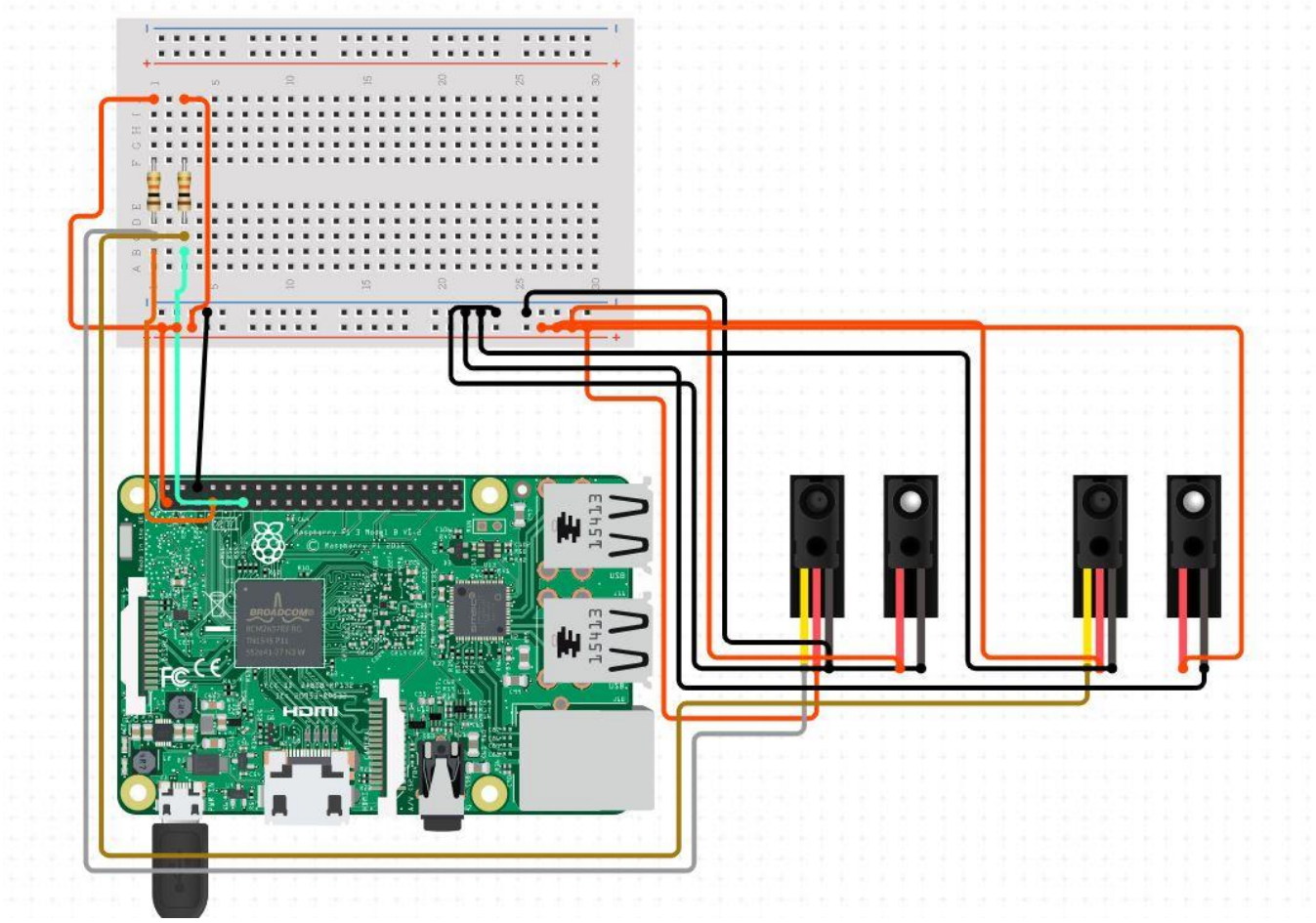
- Hardware List:
  - Raspberry Pi 4 with 8GB RAM && 8GB Memory Card
  - Raspberry Pi Camera
  - IR Break Beam Sensor LEDs

Hardware Collaboration

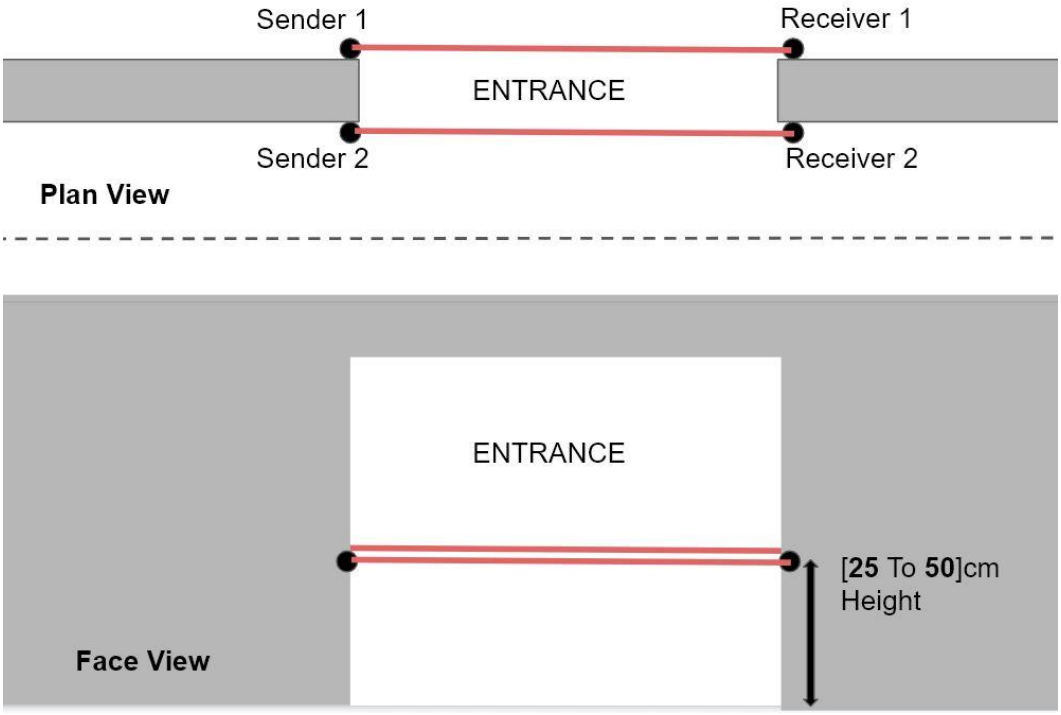
To have the best performance of your hardware the following instructions should be followed:

IR Break Beam Sensor LEDs

- Sensor one should be connected to GIPO 4 PIN(7)
- Sensor two should be connected to GIPO 17 PIN(11) As is shown in the picture below:

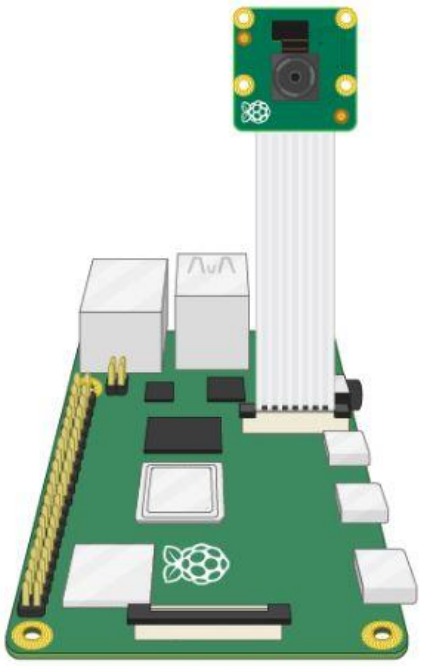


- Both sensors should be installed on both side of the door, with range [25cm] to [50cm] height from the ground taken the kids and adults height under consideration as shown in the picture below:



Raspberry Pi Camera

- the camera connects to the camera port on the Raspberry Pi as it is shown in the picture below.#

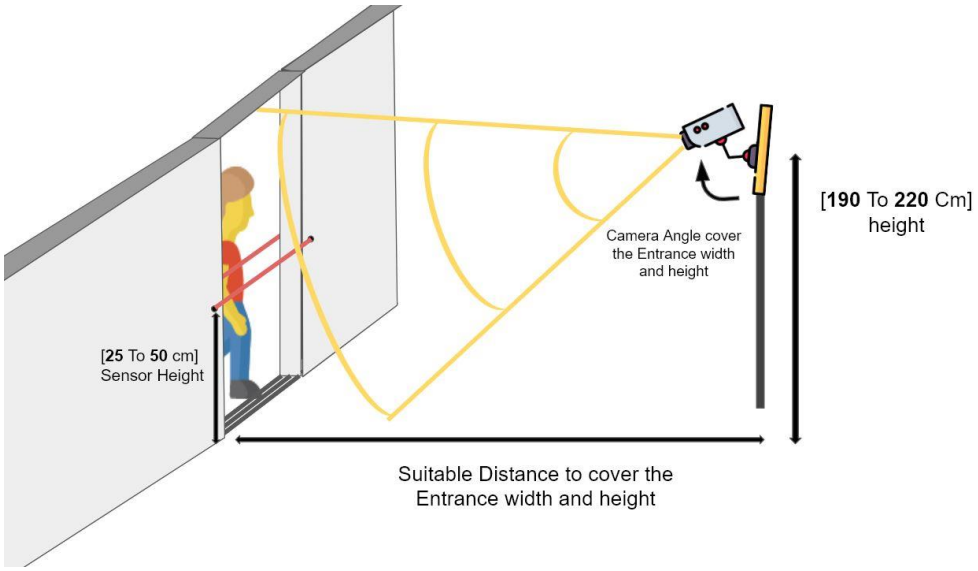


- Now you need to enable camera support using the **raspi-config** program you will have used when you first set up your Raspberry Pi:
    - Use the cursor keys to select and open Interfacing Options, and then select Camera and follow the prompt to enable the camera.
    - To test that the system is installed and working.

Try the following command:

```
$ raspistill -v -o test.jpg
```

- The camera should be installed closer to the midpoint of the entrance, with a height range [190cm] to [220cm], with distance and an angle cover the full Entrance width and height.



How To Run EIH Backend?

To Run EIH Project you do need the same software package to both directions **IN** or **OUT** and that by telling the system what direction to run with the argument passed as the following commands:

Before you run the project, This environment variables should be export and save to run the project.

```
    $ export FIREBASE_DB_URL='https://<YOUR-PROJECT-NAME>.firebaseio.com/<THE-JSON-FILE-NAME>.json'
    $ export REG_BUILIDING_ID='<THE-BUILDING-ID-FROM-FRONT-END-REGISTARTION>'
    $ export DEVICE_ID='<THE-DDEVICE-ID-YOU-CHOSE>'
    $ export PROJECT_API_KEY='<THE-FIREBASE-API-KEY>'
```

To get `REG_BUILIDING_ID` you have to follow the instructions on (the fron-end add new building)

To Run The Project for [OUT] Direction

```
$ python eihIR.py out
```

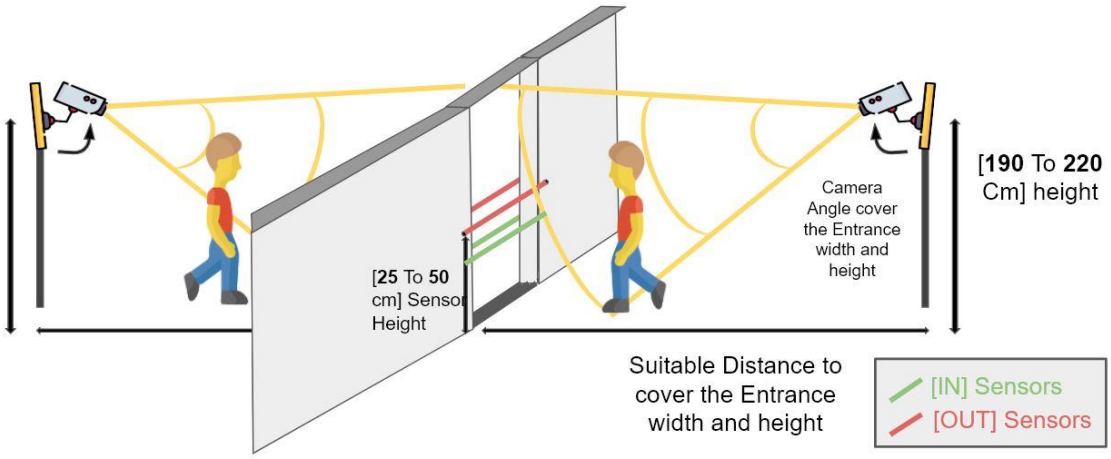
To Run The Project for [IN] Direction

```
$ python eihIR.py in
```

And this will change the calculations in the file post-data-to-api.py in the following code.

```
if args["numberOUT"] is not None:
    # substact the number coming from the API to the number coming from the RassPi
    newNumber = preNumber- currentNumberOUT
elif args["numberIN"] is not None:
    # add the number coming from the API to the number coming from the RassPi
    newNumber = preNumber + currentNumberIN
```

To Run the Project In The Both Direction, You need Two Hardware Set As It's Shown In The Picture



Project Code Front-End

1. Test_flask.py

```
# EIH Python unit test
# AUTHOR: OSAMA ABOU HAJAR
# Institute of Technology Carlow
# STUDENT ID: C002201315
# DATE: 20/04/2020
# Emergency Info Hub (EIH), Is a central website helps the emergency services to prepar
e for,
#      respond to & recover from disaster, by providing all needed data for the targete
d building
#      (E.g. Number of people, area size and emergency exits).
# The main objective of this project is giving the number of trapped people under rubbl
es or inside a building,
#  by tracking their number using a simple movement sensor fitted on the main gate and
face detection technology,
#  and save this number to the cloud to be used when a disaster happens.

### To run the rest you have to export the following env veriable
    #  export GOOGLE_API_KEY='GOOGLE-API-KEY-FOR-THE-CREATED-PROJECT-ON-THE-FIREBASE'
    #  export GOOGLE_MAP_API='GOOGLE-MAPS-API-KEY-TO-SHOW-THE-MAP-ON-RESULTS'
    #  export AUTH_DOMAIN='<YOU-PROJECT-NAME-ON-FIREBASE>.firebaseio.com'
    #  export DB_URL='https://<YOU-PROJECT-NAME-ON-FIREBASE>.firebaseio.com'
    #  export STOREG_BUKET='<YOU-PROJECT-NAME-ON-FIREBASE>.appspot.com'
    #  export SERVICE_ACCOUNT='<THE-PATH-TO-THE-DB-CONFIG-JSON-FILE>/projecteih-
firebase-adminsdk-dmd9b-dfbc30ba25.json'
    #  export DB_FILE_URL='https://<YOU-PROJECT-NAME-ON-FIREBASE>.firebaseio.com/<THE-
JSON-FILE-NAME-ON-FIREBASE>.json'


import unittest
import app
from unittest.mock import patch
from app import app as web
from flask import Flask, request, session
from flask import json, jsonify
import flask
import os
webF = flask.Flask(__name__)


class TestApp(unittest.TestCase):
```

```
    ##1 test the web load
    def test_main_page_load(self):
        tester = web.test_client()
        response = tester.get("/displayform", content_type="html/text")
        self.assertEqual(response.status_code, 200)

    ##2 test the all location page load
    def test_all_locations_page(self):
        tester = web.test_client()
        response = tester.get("/allLocations", content_type="html/text")
        self.assertEqual(response.status_code, 200)

    ##3 test the aboutus page load
    def test_about_us_page(self):
        tester = web.test_client()
        response = tester.get("/aboutus", content_type="html/text")
        self.assertEqual(response.status_code, 200)

    ##4 test the contact us page load
    def test_contact_us_page(self):
        tester = web.test_client()
        response = tester.get("/contactForm", content_type="html/text")
        self.assertEqual(response.status_code, 200)

    ##5 test the logout redirect to main page load
    def test_logout_page(self):
        tester = web.test_client()
        response = tester.get("/logout", follow_redirects=True)
        self.assertEqual(response.status_code, 200)

    ##6 test the display Results page load with the tested address in the database
    def test_search_results_page(self):
        tester = web.test_client()
        address = "Institute of Technology Carlow, Kilkenny Road, Moanacurragh, Carlow,
Ireland"
        response = tester.post(
            "/displayResults", data=dict(thelocation=address), follow_redirects=True
        )
        self.assertIn(b"People Inside", response.data)

    ##7 test fail login
    def test_fail_login(self):
        tester = web.test_client(self)
```

```
        userlogin = "test@test.com"
        password = "noPassword"
        response = tester.post(
            "/login",
            data=dict(loginInput=userlogin, loginPass=password),
            follow_redirects=True,
        )

        self.assertIn(b"Email Address OR Password", response.data)

    ##9 test to check the connections is good to the Database
    def test_get_data_from_db(self):
        result = app.get_all_data_from_firebase()
        self.assertIsNotNone(result)

    ##10 test the data type coming form the database
    def test_get_data_from_db_type(self):
        result = app.get_all_data_from_firebase()
        self.assertTrue(type(result) is dict)

    ##11 test database configuration
    def test_db_config(self):
        result = app.db_config()
        self.assertIsNotNone(result)

    # ##12 test delete row
    # def test_delete_row(self):
    #     self.assertIsNone(app.delete_row('RANDOM_ID'))

    # ##13 test active_row
    # def test_active_row(self):
    #     self.assertIsNone(app.active_row('RANDOM_ID'))

    # ##14 test delete row
    # def test_update_row(self):
    #     self.assertIsNone(app.update_row('RANDOM_ID',None))


if __name__ == "__main__":
    unittest.main()
```

2. App.py

```
from flask import Flask, render_template, request, redirect, url_for, session, jsonify
import requests
import pyrebase
import time
from datetime import timedelta
import datetime
import os
from dotenv import load_dotenv


## to load all the env vriables from the .env file
project_folder = os.path.expanduser("~/")  #
load_dotenv(os.path.join(project_folder, ".env"))

app = Flask(__name__)  # "dunder name".

DEBUG = True
## Configureation for flask
app.config.from_object(__name__)
## external configuration stores in external file
app.config.from_pyfile("configuration/myconfig.cfg")
## update flask config to set the life time of session to 15, if no actions the session wil
l be logout.
app.config.update(
    ## time out logged in session after minutes=15 time
    PERMANENT_SESSION_LIFETIME=timedelta(minutes=45),
)
## Generate a random String as secret key wach time we run the app for more secuity.
app.secret_key = os.urandom(32)

## Set up the DB url to be able to read the public data.
def get_all_data_from_firebase():
    db_file_url = os.environ["DB_FILE_URL"]
    r = requests.get(db_file_url)
    x = r.json()
    return x


## function to return the location has been searched by the user input on the main page.
def get_the_search_location():
    # to get the search address fully
    addressName = request.form["thelocation"].upper()
```

```
    return addressName


## function to split the address and return the first part as the surched name.
def get_the_name_form_search_text(addressName):
    ##split the address into list
    str_list = addressName.split(",")
    # display only the name from tha address.
    building_name_send = str_list[0]
    session["building_name_send"] = building_name_send
    return building_name_send


## function to congifure the database ## this information should be secret.
## all enviroument veriables are stored in other secret file
def db_config():
    config = {
        "apiKey": os.environ["GOOGLE_API_KEY"],
        "authDomain": os.environ["AUTH_DOMAIN"],
        "databaseURL": os.environ["DB_URL"],
        "storageBucket": os.environ["STOREG_BUKET"],
        "serviceAccount": os.environ["SERVICE_ACCOUNT"],
    }
    firebasePy = pyrebase.initialize_app(config)

    return firebasePy


## reset the locations selected to zero
def reset_to_zero(id_to_reset):
    db = db_config().database()
    db.child("locations").child(id_to_reset).update({"numberOfPeopleINDetect": 0})


## to add the new building detailes entered by the user.
## and return the new KEY after registration to by linked by the hardware.
def add_new_builiding(data):
    data_stored = get_all_data_from_firebase()
    if check_if_address_added_already(data_stored, data["address"]):
        session["address_found_in_DB"] = True
        return False
    else:
        db = db_config().database()
        id_generated = db.child("locations").push(data)
```

```
        session["id_generated"] = id_generated["name"]
        ## the Key returned is to be added to the hardware
        return id_generated


## to return false or True is the searched address is in the DB.
def check_if_searched_address_in_db(building_name_send):
    data_stored = get_all_data_from_firebase()
    return check_if_address_added_already(data_stored, get_the_search_location())


## ## to return false or True is the searched address is in the DB.
## set all sesssion when founded.
def check_if_address_added_already(data_stored, data):
    ## Error Pass is none data in the DB.
    if data_stored is None:
        pass
    else:
        ## To loop inside the json file coming back from the API.
        ## Key is the building id and the value its detailes.
        for key, value in data_stored.items():
            if str(value["address"]) == str(data):
                session["number_inside"] = str(value["numberOfPeopleINDetect"])
                session["building_Status"] = value['active']
                return True
    session["address_not_found_in_DB"] = False
    return False


## To delete the selected raw from the DB.
def delete_row(id_to_reset):
    db = db_config().database()
    db.child("locations").child(id_to_reset).update({"active": False})
## To activate the selected raw from the DB.
def active_row(id_to_reset):
    db = db_config().database()
    db.child("locations").child(id_to_reset).update({"active": True})


## To update the selected raw from the DB.
def update_row(id_to_update, data):
    db = db_config().database()
    x = db.child("locations").child(id_to_update).update(data)
```

```
## to check if the login was successful or not
def login_check():
    email = request.form["loginInput"]
    password = request.form["loginPass"]
    firebasePy = db_config()
    # Get a reference to the auth service
    auth = firebasePy.auth()
    # Log the user in
    try:
        user = auth.sign_in_with_email_and_password(email, password)
        session["logged_in_email"] = email
        session["loged_in"] = True
        session.permanent = True
        return True
    except IOError:
        ## Increment the session["login_attempts"] at each time the login failing.
        session["login_attempts"] += 1
        return False
    ## return the email and the password attepming to log in with.
    return email, password

# ## the application route.
# @app.before_request
# def before_request():
#     if not session:
#         session["login_attempts"] = 1
#         session["locked_status"] = False
#         session["request_ip_address_locked"] = ""

@app.route("/")
def route():
    ## redirevt to the main bage index in the display_form function.
    return redirect(url_for("display_form"))


@app.route("/logout")
def logout():
    ##clear all the session set already
    session.clear()
    # The key is secure enough, and each time you launch your system the key changes invali
dating all sessions.
    # app.secret_key = os.urandom(32)
    # return display_form()
```

```
    return redirect(url_for("display_form"))


@app.route("/displayform")
def display_form():
    ## for the first time running the webapp, if no session set aleady that mean
    ## this is the first time then the session get initialized.
    if not session:
        session["login_attempts"] = 1
        session["locked_status"] = False
        session["request_ip_address_locked"] = ""
    ## if the session were set already, and the ip was blocked.
    elif session["request_ip_address_locked"] == request.remote_addr:
        ## to check when the IP is blocke after the 5 time failing login attempts.
        ## and check if the time at lock out was greater than 5 minutes then re allow to lo
gin.
        if (
            session["locked_status"] is True
            and (time.time() - session["time_locked"]) > 500
        ):
            session["locked_status"] = False
            session.clear()
    ## return the GOOGLE_API_KEY saved in .env file to the index page. for more security.
    return render_template("index.html", API_KEY=os.environ["GOOGLE_API_KEY"], alert=sessio
n.pop('address_not_found_in_DB', None)  )


## route to display the results page.
##when the user enter the location in the search input
@app.route("/displayResults", methods=["POST"])
def displayResults():
    ### get the location searched by the user.
    addressName = get_the_search_location()
    ### split the name of the location from the first part of the address.
    building_name_send = get_the_name_form_search_text(addressName)
    ### if the addcess was found in the DB.
    found = check_if_searched_address_in_db(building_name_send)
    if not found:
        return redirect(url_for("display_form"))
    else:
        return render_template(
            "results.html",
            searched_text=addressName,
            building_name=building_name_send,
```

```
            total_numebr=session["number_inside"],
            building_Status=session["building_Status"],
            GOOGLE_MAP_API=os.environ['GOOGLE_MAP_API'],
        )


@app.route("/contactForm")
def contactForm():
    ### contact US form to be done by the organizations it selfs.
    ## if any one start using this project in the future they will be abke to set the conta
ct and methods they prefare.
    return render_template("contactUs.html")


@app.route("/emailMsg")
def emailMsg():
    ### once the contact msg has been sent than you page will be shown and redirect to the
main page in 5 seconds.
    return render_template("sentMsg.html")


## route to display all the data in the database.
@app.route("/allLocations")
def allLocations():
    ### to get all the data from the data set
    items_send = get_all_data_from_firebase()
    if items_send is None:
        ## if the data was none an empty dictionary will be sent.
        ### Error handeling.
        return render_template(
            "allLocations.html",
            items=dict(),
            x={},
            API_KEY=os.environ["GOOGLE_API_KEY"],
            alert=session.pop('address_not_found_in_DB', None)
        )
    else:
        return render_template(
            "allLocations.html",
            items=items_send,
            x=items_send,
            API_KEY=os.environ["GOOGLE_API_KEY"],
            alert=session.pop('address_not_found_in_DB', None)
        )
```

```
## login route
@app.route("/login", methods=["POST"])
def login():
    ### to check if the user attempts is 5.
    if session['login_attempts'] is not None:
        if session["login_attempts"] == 5:
            session["time_locked"] = time.time()
            session["request_ip_address_locked"] = request.remote_addr
            session["locked_status"] = True
            return redirect(url_for("route"))
    else:
        session['login_attempts'] = 1
    ## if attempts less than 5 then the login email,and password get checked.
    logPass = login_check()
    if logPass:
        return redirect(url_for("allLocations"))
    else:
        session["fail_login"] = True
        return redirect(url_for("display_form"))


@app.route("/resetLocation", methods=["POST"])
def resetLocation():
    ### getting the value of the radio button on the admin board>
    ### for all locations table.
    selected = request.form.get("selected_radio")
    ### if the user select the reset option
    if request.form.get("action") == "Reset":
        reset_to_zero(selected)
    ### if the user select the Delete option
    elif request.form.get("action") == "Delete":
        delete_row(selected)
    elif request.form.get("action") == "Activate":
        active_row(selected)
    ### if the user select the Update option
    elif request.form.get("action") == "Update":
        ### the data has been updated get collected and formatted
        ### then the data json sent tp update_row() fuction.
        data = {
            "known_name": request.form.get("known_name_update").upper(),
            "address": request.form.get("address_update").upper(),
            "eircode": request.form.get("eircode_update").upper(),
```

```
            "timeUpdated": time.strftime("%X %x %Z"),
        }
        update_row(selected, data)
    ### if the user select the Add option
    elif request.form.get("action") == "Add":
        ### the data has been added get collected and formatted
        ### then the data json sent tp add_new_builiding() fuction.
        data = {
            'active' : True,
            "deviceId": "none",
            "known_name": request.form.get("new_row_name").upper(),
            "address": request.form.get("new_row_address").upper(),
            "eircode": request.form.get("new_row_eircode").upper(),
            "numberOfPeopleINDetect": 0,
            "timeUpdated": time.strftime("%X %x %Z")
        }
        #to be display at after the registration compete
        session['new_row_name']=request.form.get("new_row_name").upper()
        add_new_builiding(data)
    return redirect(url_for("allLocations"))

## route to about Us bage
@app.route("/aboutus")
def aboutus():
    return render_template("aboutus.html",)


@app.after_request
def apply_caching(response):
    response.headers['X-Frame-Options'] = 'SAMEORIGIN'
    response.headers['X-XSS-Protection'] = '1; mode=block'
    response.headers['Strict-Transport-Security'] = 'max-age=31536000; includeSubDomains'
    # response.headers['Content-Security-Policy'] = "default-src 'self'"
    response.headers['X-Content-Type-Options'] = 'nosniff'
    return response


# the application start
if __name__ == "__main__":
    app.run()
```

3. base.html

```
<!-- EIH HTML CODE -->
<!-- AUTHOR: OSAMA ABOU HAJAR -->
<!-- Institute of Technology Carlow -->
<!-- STUDENT ID: C002201315 -->
<!-- DATE: 20/04/2020 -->
<!-
- Emergency Info Hub (EIH), Is a central website helps the emergency services to prepar
e for,
     respond to & recover from disaster, by providing all needed data for the targeted
building
     (E.g. Number of people, area size and emergency exits).
The main objective of this project is giving the number of trapped people under rubbles
 or inside a building,
 by tracking their number using a simple movement sensor fitted on the main gate and fa
ce detection technology,
 and save this number to the cloud to be used when a disaster happens.

 -->
<!doctype html>
<html>

<head>
    <title>EIH - EVERY LIFE MATTERS
    </title>
    <link href="https://fonts.googleapis.com/css?family=Montserrat:100,200,300,400,500,
600,700,800,900" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></sc
ript>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"><
/script>
    <link rel="stylesheet" type="text/css" href="../static/EIH.css">
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    </script>
    <link rel="icon" type="image/png" href="static/logo.png">

</head>

<body>
    <!-- TO INCLUDE THE MENU AND LOGIN PAGES ALL OVER THE WEBSITE -->
    {% include 'loginForm.html' %}{% include 'nav.html' %}
```

```
    <h1>{{ the_title }}</h1>

    {% block body %} {% endblock %}

    <div class="footer"> &copy; 2020 by Osama Abou Hajar. All Rights Reserved.</div>
    <div class="socialMedia">
        <a href="https://github.com/OAbouHajar" class="fa fa-
github" target="_blank"></a>
        <a href="https://www.linkedin.com/in/osamaabouhajar/" class="fa fa-
linkedin" target="_blank"></a>
        <a href="http://glasnost.itcarlow.ie/~softeng4/C00220135/index.html" class="fa
fa-rss" target="blank"></a>
    </div>
</body>

</html>
```

4. Alllocations.html

```
{% extends "base.html" %} {% block body %}

<div class="page2">
    {% if session["address_found_in_DB"] is defined %}
    <div id="alert" class="alert alert-danger alert-dismissible fade in">
        <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
        <strong>Alert!</strong> The Entered address is in Our DataBase.
    </div>
    {% elif session["id_generated"] is defined %}
    <div id="alert" class="alert alert-okay alert-dismissible fade in">
        <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
        <strong>Alert!</strong> the Builiding {{ session['new_row_name'] }} registation
 key is : <strong>'{{session['id_generated']}}'</strong> Add this key to your hardware
to connect.
    </div>
    {% endif %}
    <div class="resultAllLocations">
        <h2>All Locations</h2>
        {% if session["loged_in"] is defined %}
        <span class="tableinstraction">* Right Click To Show The Options Menu</span>
        <br>
        <span class="tableinstraction">* Double Click On The Field To Edit.</span> {% e
ndif %}
        <div class="table-wrapper">
            <table class="fl-table" id="allLocation">
                <thead>
                    <tr>
                        {% if session["loged_in"] is defined %}

                        <th>#</th>
                        <th>Building ID</th>
                        {% endif %}
                        <th>Building Name</th>
                        <th>Address</th>
                        <th>Eircode</th>
                        <th>Number Inside</th>
                        <th>Status</th>
                        <th>Last Update</th>
                    </tr>
                </thead>
                <tbody>
```

```
                    <form name="all" id="all_data" action='/resetLocation' method='POST
'>
                        {% if session["loged_in"] is not defined %}
                        <input type="text" id="myInput" onkeyup="search_all()" placehol
der="Search for names.." title="Type in a name"> {% elif session["loged_in"] is defined
 %}
                        <input type="text" id="myInput" onkeyup="search()" placeholder=
"Search for names.." title="Type in a name">
                        <div class="buttonsContainer">
                            <!--
<input type="submit" name="action" value="Reset" class=" buttonreset" onclick="return c
heckbox_check();"></input>-->
                            <!--
<input type="submit" name="action" value="Update" class=" buttonreset" onclick="return
storeChangesInSession();"></input>-->
                            <!--
<input type="submit" name="action" value="Delete" class=" buttonreset" onclick="return
checkbox_check();"></input>-->
                            <input type="button" id="add_buttom" value="+/- New Row" cl
ass=" buttonreset" onclick="return add_row()"></input>
                        </div>
                        <div id="contextMenu" class="dropdown clearfix">
                            <div class="dropdown-menu" role="menu" aria-
labelledby="dropdownMenu" style="display:block;position:static;margin-bottom:5px;">
                                <input type="submit" name="action" value="Reset" class=
" buttonreset2" onclick="return checkbox_check();"></input>
                                <br>
                                <input type="submit" name="action" class=" buttonreset2
" tabindex="-1" href="/resetLocation" value="Add" onclick="return add_new();"></input>
                                <br>
                                <input type="submit" name="action" class=" buttonreset2
" tabindex="-
1" href="/resetLocation" value="Delete" onclick="return checkbox_check();"></input>
                                <br>
                                <input type="submit" name="action" class=" buttonreset2
" tabindex="-
1" href="/resetLocation" value="Activate" onclick="return checkbox_check();"></input>
                                <br>
                                <input type="submit" name="action" class=" buttonreset2
" tabindex="-
1" href="/resetLocation" value="Update" onclick="return storeChangesInSession();"></inp
ut>
```

```
                                <input type="button" id="add_buttom" value="+/- New Row
" class=" buttonreset2" tabindex="-1" onclick="return add_row()"></input>

                            </div>
                        </div>
                        {% endif %}
                        <tr id="myDIV" style="display: none">
                            <td><button type="submit" name="action" id="new_form_id" va
lue="Add" class=" buttonAdd" onclick="return add_new();">ADD</button>
                            </td>
                            <td><input type="text" disabled placeholder="Auto Generate"
></input>
                            </td>
                            <td><input type="text" name="new_row_name" id="new_row_name
" placeholder="Builiding Name"></input>
                            </td>
                            <td> {% include 'googlAutoCompleteGeneral.html' %} </td>
                            <td><input name="new_row_eircode" id="new_row_eircode" type
="text" placeholder="Eircode"></input>
                            </td>
                            </input>
                            <td><input type="text" disabled placeholder="Auto Generate"
 /></td>
                            <td><input type="text" disabled placeholder="Auto Generate"
 /></td>
                            <td><input type="datetime" disabled placeholder="Auto Gener
ate" required></td>
                            </input>
                        </tr>
                        {% for k, v in x.items(): %}
                        <tr>
                            {% if session["loged_in"] is defined %}

                            <td>
                                <input type="radio" name="reset_checkbox" id="{{ k }}"
value="{{ k }}">
                            </td>
                            <td id="{{k}}">{{ k }}</td>
                            {% endif %}
                            <td name="known_name" id="{{k}}_name" contenteditable>{{ v.
known_name }}</td>
                            <input type="hidden" name="known_name_update" id="known_nam
e_update" />
```

```
                            <td name="address" id="{{k}}_address" value="{{ v.address }
}" contenteditable>{{ v.address }}</td>
                            <input type="hidden" name="address_update" id="address_upda
te" />
                            <td name="eircode" id="{{k}}_eircode" value="{{ v.eircode }
}" contenteditable>{{ v.eircode }}</td>
                            <input type="hidden" name="eircode_update" id="eircode_upda
te" />
                            <td name="numberOfPeopleINDetect" id="numberOfPeopleINDetec
t" value="{{ v.numberOfPeopleINDetect }}">{{ v.numberOfPeopleINDetect }}</td>
                            <input type="hidden" name="numberOfPeopleINDetect" value="{
{ v.numberOfPeopleINDetect }}" />
                            <td name="active" id="{{k}}_active" value="{{ v.active }}">
{{ v.active }}</td>
                            <input type="hidden" name="active" value="{{ v.active }}" /
>
                            <td name="timeUpdated" id="timeUpdated" value="{{ v.timeUpd
ated }}">{{ v.timeUpdated }}</td>
                        </tr>
                        {% endfor %}
                        <input type="hidden" name="selected_radio" id='selected_radio'
/>

                    </form>
                    <br>
                </tbody>
            </table>
        </div>
        <!--{% if session["loged_in"] is defined %}-->
        <!--
<span class="tableinstraction">*Double click on the field to edit.</span> {% endif %}--
>
        <form action='/displayform'>
            <button class=" button button3 buttonBack"> Home </button>
        </form>

    </div>
</div>



<script>
    function checkbox_check() {
```

```
        var ele = document.getElementsByName('reset_checkbox');
        var founded = false;
        for (i = 0; i < ele.length; i++) {
            if (ele[i].checked) {
                rad = ele[i].value;
                founded = true
            }
        }
        if (founded == false) {
            alert('select one option pelase!')
            return false;

        }
        var rad;
        document.getElementById("selected_radio").value = rad;
        return confirm('Do you to comtinue the process?');

    }

    function add_row() {
        var x = document.getElementById("myDIV");
        if (x.style.display === "none") {
            x.style.display = "table-row";
        } else {
            x.style.display = "none";
        }

    }

    function storeChangesInSession() {


        var ele = document.getElementsByName('reset_checkbox');
        var founded = false;
        for (i = 0; i < ele.length; i++) {
            if (ele[i].checked) {
                rad = ele[i].value;
                founded = true
            }
        }
        if (founded == false) {
            alert('select one option pelase!')
            return false;
```

```
    }
    var rad;
    document.getElementById("selected_radio").value = rad;
    var name = document.getElementById(rad + '_name').innerHTML;
    var address = document.getElementById(rad + '_address').innerHTML;
    var eircode = document.getElementById(rad + '_eircode').innerHTML;
    // var active = document.getElementById(rad + '_active').innerHTML;

    document.getElementById("known_name_update").value = name;
    document.getElementById("address_update").value = address;
    document.getElementById("eircode_update").value = eircode;
    // document.getElementById("eircode_update").value = active;

    return confirm('Do you to comtinue the process?');


}

function add_new() {
    var name = document.getElementById('new_row_name').value;
    var address = document.getElementById('pac-input').value;
    var eircode = document.getElementById('new_row_eircode').value;

    if (name == "" || address == "" || eircode == "") {
        alert('Please fill all the feilds to add!');
        return false;
    } else {
        return confirm('Do you to comtinue the process?');
    }

}
$("form").keypress(function(e) {
    //Enter key
    if (e.which == 13) {
        return false;
    }
});

function search_all() {
    var input, filter, table, tr, td, i, txtValue;
    input = document.getElementById("myInput");
    filter = input.value.toUpperCase();
```

```
        table = document.getElementById("allLocation");
        tr = table.getElementsByTagName("tr");
        for (i = 2; i < tr.length; i++) {
            td = tr[i].getElementsByTagName("td")[0];
            if (td) {
                txtValue = td.textContent || td.innerText;
                if (txtValue.toUpperCase().indexOf(filter) > -1) {
                    tr[i].style.display = "";
                } else {
                    tr[i].style.display = "none";
                }
            }
        }
    }

    function search() {
        var input, filter, table, tr, td, i, txtValue;
        input = document.getElementById("myInput");
        filter = input.value.toUpperCase();
        table = document.getElementById("allLocation");
        tr = table.getElementsByTagName("tr");
        for (i = 2; i < tr.length; i++) {
            td = tr[i].getElementsByTagName("td")[2];
            if (td) {
                txtValue = td.textContent || td.innerText;
                if (txtValue.toUpperCase().indexOf(filter) > -1) {
                    tr[i].style.display = "";
                } else {
                    tr[i].style.display = "none";
                }
            }
        }
    }


    $(function() {

        var $contextMenu = $("#contextMenu");

        $("body").on("contextmenu", "table tr", function(e) {
            $contextMenu.css({
                display: "block",
                left: e.pageX,
```

```
                top: e.pageY
            });
            return false;
        });

        $('html').click(function() {
            $contextMenu.hide();
        });

        $("#contextMenu li a").click(function(e) {
            var f = $(this);
        });

    });
</script>
{% endblock %}
```

5. Index.html

```
{% extends "base.html" %} {% block body %} {% if session['fail_login'] is sameas true %
}

<body onload="loginForm()">
    {% endif %}

    <div class="page1">
        <div class="center" id="p1">
            <!-- <img src="static/logo.png" class="center logo"></img> -->
            <div class="center logo"></div>
            <h4>Enter Address to search</h4>
            <form action='/displayResults' method='POST'>
                {% include 'googlAutoComplete.html' %}
            </form>
        </div>
    </div>
    <!-- To HIDE OR SHOW THE ALERT -->
    {% if alert is sameas false %}
    <div id="alert" class="alert alert-danger alert-dismissible fade in">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close" onclick="deleteItems()">&times;</a>
        <strong>Alert!</strong> The Entered Address <b>{{session['building_name_send']}
}</b> Has No Data In Our DataBase.
    </div>

    {% endif %} {% endblock %}

    <script>
        $("form").keypress(function(e) {
            //Enter key
            if (e.which == 13) {
                return false;
            }
        });
    </script>
```

6. loginForm.html

```
{% extends "base.html" %} {% block body %} {% if session['fail_login'] is sameas true %
}

<body onload="loginForm()">
    {% endif %}

    <div class="page1">
        <div class="center" id="p1">
            <!-- <img src="static/logo.png" class="center logo"></img> -->
            <div class="center logo"></div>
            <h4>Enter Address to search</h4>
            <form action='/displayResults' method='POST'>
                {% include 'googlAutoComplete.html' %}
            </form>
        </div>
    </div>
    <!-- To HIDE OR SHOW THE ALERT -->
    {% if alert is sameas false %}
    <div id="alert" class="alert alert-danger alert-dismissible fade in">
        <a href="#" class="close" data-dismiss="alert" aria-
label="close" onclick="deleteItems()">&times;</a>
        <strong>Alert!</strong> The Entered Address <b>{{session['building_name_send']}
}</b> Has No Data In Our DataBase.
    </div>

    {% endif %} {% endblock %}

    <script>
        $("form").keypress(function(e) {
            //Enter key
            if (e.which == 13) {
                return false;
            }
        });
    </script>
```

7. Maps2.html

```
<!DOCTYPE html>
<html>

<head>
    <style>
        /* Set the size of the div element that contains the map */

        #map {
            display: flex;
            align-items: center;
            justify-content: center;
            margin: auto;
            height: 200px;
            border-radius: 10px/* The height is 400 pixels */
            /* The width is the width of the web page */
        }
    </style>
</head>

<body onload="initialize()">
    <div id="map"></div>
    <!-- <div>
        <input id="address" type="textbox" value="Sydney, NSW">
        <input type="button" value="Encode" onclick="codeAddress()">
    </div> -->
</body>
<script>
    var geocoder;
    var map;

    function initialize() {
        geocoder = new google.maps.Geocoder();
        var latlng = new google.maps.LatLng(-34.397, 150.644);
        var mapOptions = {
            zoom: 15,
            center: latlng
        }
        map = new google.maps.Map(document.getElementById('map'), mapOptions);
        codeAddress();
    }

    function codeAddress() {
```

```
        var address = document.getElementById('resultsLable').innerHTML;

        geocoder.geocode({
            'address': address
        }, function(results, status) {
            if (status == 'OK') {
                map.setCenter(results[0].geometry.location);
                var marker = new google.maps.Marker({
                    map: map,
                    position: results[0].geometry.location
                });
            } else {
                // alert('Geocode was not successful for the following reason: ' + stat
us);
            }
        });
    }
</script>
<script async defer src="https://maps.googleapis.com/maps/api/js?key={{ GOOGLE_MAP_API
}}&callback=initMap">
</script>

</html>
```

8. Nav.html

```
<!DOCTYPE html>
<html>

<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
        body {
            font-family: 'Lato', sans-serif;
        }

        .overlay {
            height: 100%;
            width: 0;
            position: fixed;
            z-index: 1;
            top: 0;
            left: 0;
            background-color: rgb(0, 0, 0);
            background-color: rgba(0, 0, 0, 0.9);
            overflow-x: hidden;
            transition: 0.5s;
        }

        .overlay-content {
            position: relative;
            top: 25%;
            width: 100%;
            text-align: center;
            margin-top: 30px;
        }

        .overlay a {
            padding: 8px;
            text-decoration: none;
            font-size: 36px;
            color: #f7f588;
            display: block;
            transition: 0.3s;
        }

        .overlay a:hover,
        .overlay a:focus {
```

```
            color: rgb(143, 255, 255);
        }

        .overlay .closebtn {
            position: absolute;
            top: 20px;
            right: 45px;
            font-size: 60px;
        }

        @media screen and (max-height: 450px) {
            .overlay a {
                font-size: 20px
            }
            .overlay .closebtn {
                font-size: 40px;
                top: 15px;
                right: 35px;
            }
        }

        @media only screen and (max-width: 768px) {
            .menu {
                position: relative;
                top: 4%;
                left: 2%;
                color: #ffffff;
                font-size: 24px;
                cursor: pointer;
            }
        }

        .menu {
            position: relative;
            top: 18px;
            left: 2%;
            color: #ffffff;
            font-size: 24px;
            cursor: pointer;
        }
    </style>
</head>
```

```
<body>

    <div id="myNav" class="overlay">
        <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>
        <div class="overlay-content">
            <a href="/displayform">Home</a>
            <a href="/allLocations">Locations</a>
            <a href="/aboutus">About</a>
            <a href="http://glasnost.itcarlow.ie/~softeng4/C00220135/index.html" target
="_blank">Case Study</a>
            <a href="/contactForm">Contact</a> {% if session["loged_in"] is defined %}
            <a href="/logout">Log Out</a> {% endif %}
        </div>
    </div>

    <span class="menu" onclick="openNav()">&#9776; Menu</span>

    <script>
        // to show or hide the Menu.
        function openNav() {
            document.getElementById("myNav").style.width = "100%";
        }

        function closeNav() {
            document.getElementById("myNav").style.width = "0%";
        }
    </script>

</body>

</html>
```

9. Result.html

```
{% extends "base.html" %} {% block body %}

<div class="page2">
    <div class="result">
        <span class="searchResults"> The Search results For : </span>
        <span id="resultsLable" class=" resultsLable ">  {{searched_text}}</span>
        <br><br><br>
        <label>Building Name:  </label>
        <span id="address" class=" resultsLable "> {{ building_name }} </span>
        <hr>
        <label>People Inside:   </label>
        <span class="resultsLable ">  {{ total_numebr }} Person</span>
        <hr>
        <label>Building Status :</label>
        <span class="resultsLable ">{%if building_Status %}Active {% else %}NOT Acitve
{% endif %}</span>
        <hr> <br> {% include 'maps2.html' %}

    </div>

    <form action='/'>
        <button class=" button button3 buttonBack"> Home </button>
    </form>
</div>




{% endblock %}
```

10. Aboutus.html

```
{% extends "base.html" %} {% block body %}

<div class="pageus">
    <div class="center" id="pus">
        <!-- <img src="static/logo.png" class="center logo"></img> -->
        <div class="center logo"></div>
        <h4>EMERGENCY INFO HUB</h4>
        <p>Emergency Info Hub (EIH), Is a central website helps the emergency services
to prepare for, respond to & recover from disaster, by providing all needed data for th
e targeted building (E.g. Number of people, area size and emergency exits).
            <br> <br>The main objective of this project is giving the number of trapped
 people under rubbles or inside a building, by tracking their number using a simple mov
ement sensor fitted on the main gate and face detection technology, and save
            this number to the cloud to be used when a disaster happens.
        </p>
        <a id="thestory"></a>
        <hr class="style-story">
        <div class="zoom">
        </div>
        <p>
            The story of EIH started at that day when this photo was taken (the website
 background), back to 2012 during the Syrian war,
            <br> when I was among those guys running to help the people stuck under rub
bles.
            <br>during this moments the one and the only question you will hear is: <st
rong> ANYBODY HERE? </strong>
            <br><br>The hours, the minutes and even the Seconds might be a reason to gi
ve someone a new life. calling again and again...
            <strong>ANYBODY HERE?</strong>, feeling bad if any one left behind with no
help.
            <br>
            <br>
            <strong>ANYBODY HERE? </strong> it's the reason to start, and here where we
 are: EIH - "EVERY LIFE MATTERS".
        </p>
        <hr class="style-eight">
        <div class="profile_image">
        </div>
        <h1>Osama Abou Hajar</h1>
        <h2>Software Development Engineer</h2>
        <h2>Institute of Technology Carlow</h2>
        <h5>Supervisor: Paul Barry</h5>
```

```
    </div>
</div>
{% endblock %}
```

11. Contactus.html

```
{% extends "base.html" %} {% block body %}

<!-- Form Started -->
<div class="page2">
    <div class="contact">
        <form action="/emailMsg">
            <div>
                <label><i class="fa fa-user" aria-hidden="true"></i> Name</label>
                <input type="text" name="name" class="form-
control" placeholder="Enter Name" required>
            </div>
            <div class="form-group">
                <label><i class="fa fa-envelope" aria-hidden="true"></i> Email</label>
                <input type="email" name="email" class="form-
control" placeholder="Enter Email" required>
            </div>
            <div class="form-group">
                <label><i class="fa fa-comment" aria-hidden="true"></i> Message</label>
                <textarea rows="2" name="message" class="form-
control" placeholder="Type Your Message" required></textarea>
            </div>

            <br><br><br>
            <button type="submit" value="Submit" class=" buttonContactUs button3 "> Sen
d </button>
            <button value="Reset" class=" buttonContactUs button4  " href='/'> Cancel <
/button>
        </form>
    </div>
</div>

<!-- Form Ended -->
{% endblock %}
```

12. snetMsg.html

```
{% extends "base.html" %} {% block body %}
<!-- this page will be displayed after the contact us form filled and sent
can be used for many cases such as display the request ID.....etc. -->
<!-- Form Started -->
<div class="page2">
    <div class="contact">
        <H1>Thanks!!</H1>
        <BR>
        <h2>YOUR MESSAGE HAS BEEN SENT,
            <br>WEILL BE BACK TO AS SOON AS POSSIBLE</h2>
        <h3>You Will Be Redirected To The Main Page In 5 Seconds</h3>

    </div>
</div>
<script>
    setTimeout(function() {
        window.location.href = '/';
    }, 5000);
</script>

<!-- Form Ended -->
{% endblock %}
```

13. EIH.css

```
html,
body {
    width: 100%;
    height: 100%;
    margin: auto;
    background-image: url(/static/home-bg.png);
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
}

#contextMenu {
    position: absolute;
    display: none;
}

.buttonreset2 {
    border: none;
    padding: 9px 15px;
    text-align: left;
    text-decoration: none;
    font-size: 19px;
    width: 57%;
    background-color: #e2c449;
    color: #000000;
    border-bottom: 1px solid black;
}

.buttonreset2:hover {
    background-color: #173242;
    color: white;
}

.buttonsContainer {
    display: contents;
}


/* Text between <hr> tag */


/* Glyph, by Harry Roberts */
```

```
hr.style-eight {
    overflow: visible;
    padding: 0;
    border: none;
    border-top: medium double #307477;
    color: #307477;
    text-align: center;
    width: 80%;
}

hr.style-eight:after {
    content: "The Developer";
    color: #337e82;
    display: inline-block;
    position: relative;
    top: -0.7em;
    font-size: 1.5em;
    padding: 0 0.25em;
    background: #ffffffcc;
}

hr.style-story {
    overflow: visible;
    padding: 0;
    border: none;
    border-top: medium double #307477;
    color: #307477;
    text-align: center;
    width: 80%;
}

hr.style-story:after {
    content: "The Story";
    color: #337e82;
    display: inline-block;
    position: relative;
    top: -0.7em;
    font-size: 1.5em;
    padding: 0 0.25em;
    background: #ffffffcc;
}
```

```
.zoom {
    padding: 91px;
    background-position: center center;
    background-image: url(/static/the_story.jpg);
    transition: transform .2s;
    width: 61%;
    height: 137px;
    margin: 0 auto;
    border-radius: 10px;
}

.zoom:hover {
    /*transform: scale(1.5);*/
    /* (150% zoom - Note: if the zoom is too large, it will go outside of the viewport)
 */
    height: 309px;
    content: "Marret Alnoman, Syrian 2012";
}

#myInput {
    background-image: url(/static/searchicon.png);
    background-position: 0px -0px;
    background-repeat: no-repeat;
    background-size: 5%;
    width: 82%;
    font-size: 16px;
    padding: 14px 6px 11px 48px;
    border: 1px solid #ddd;
    margin-bottom: 4px;
    margin-right: 1%;
}

#allLocation tr:hover {
    background-color: #e7f1ff;
}

input[name=name] {
    padding: 15px 20px;
    margin-left: 20%;
    box-sizing: border-box;
    border: 2px yellow;
    border-radius: 4px;
    width: 50%;
```

```
    display: inline-block;
}

input[name=thelocation],
input[type=email] {
    padding: 15px 20px;
    margin-left: 20%;
    box-sizing: border-box;
    border: 2px yellow;
    border-radius: 4px;
    width: 50%;
    display: inline-block;
}

input[name=loginInput],
input[name=loginPass] {
    padding: 12px 20px;
    box-sizing: border-box;
    border: 2px yellow;
    border-radius: 4px;
    width: 37%;
    margin-bottom: 3%;
}

.userlogin {
    color: #fff;
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    margin-left: 4%;
    font-size: -webkit-xxx-large;
    margin-bottom: 6%;
    margin-top: 4%;
}

.loginStyle {
    margin-left: 36%;
}

textarea {
    padding: 12px 20px;
    margin-left: 16%;
    box-sizing: border-box;
    border: 2px yellow;
    border-radius: 4px;
```

```
    width: 50%;
    height: 100%;
    display: inline-block;
    max-width: 500px;
}

.logo {
    height: 185px;
    width: 200px;
    display: block;
    margin: auto;
    content: url(/static/logo.png);
}

.button {
    background-color: #4CAF50;
    /* Green */
    border: none;
    color: white;
    padding: 14px 55px;
    text-align: center;
    text-decoration: none;
    font-size: 16px;
    -webkit-transition-duration: 0.4s;
    /* Safari */
    transition-duration: 0.4s;
    cursor: pointer;
    border-radius: 4px;
}

.buttonContactUs {
    /* Green */
    border: none;
    color: white;
    padding: 12px 55px;
    text-align: center;
    text-decoration: none;
    font-size: 16px;
    -webkit-transition-duration: 0.4s;
    /* Safari */
    transition-duration: 0.4s;
    cursor: pointer;
    border-radius: 12px;
```

```
}

.buttonLogOUT {
    border: none;
    color: white;
    padding: 12px 55px;
    text-align: center;
    text-decoration: none;
    font-size: 16px;
    -webkit-transition-duration: 0.4s;
    transition-duration: 0.4s;
    cursor: pointer;
    border-radius: 12px;
    margin-left: 11%;
}

.buttonLogIN {
    border: none;
    color: white;
    padding: 12px 55px;
    text-align: center;
    text-decoration: none;
    font-size: 16px;
    -webkit-transition-duration: 0.4s;
    transition-duration: 0.4s;
    cursor: pointer;
    border-radius: 12px;
    margin-left: 11%;
}

.buttonreset {
    border: none;
    color: white;
    padding: 9px 24px;
    text-align: center;
    text-decoration: none;
    font-size: 22px;
    -webkit-transition-duration: 0.4s;
    transition-duration: 0.4s;
    cursor: pointer;
    border-radius: 0px;
    margin-bottom: 1%;
    margin-right: 4.5px;
```

```
    background-color: #ffffffb0;
    color: #4e4600;
}

.buttonAddsimple {
    padding: 7px 25px 10px 24px;
    text-align: center;
    text-decoration: none;
    -webkit-transition-duration: 0.4s;
    transition-duration: 0.4s;
    cursor: pointer;
    border-radius: 0px;
    margin-bottom: 1%;
    margin-right: 4.5px;
    background-color: #ffffffb0;
    color: #4e4600;
    left: 86%;
    position: sticky;
    width: 11%;
    font-size: 25px;
    border: none;
}

.buttonAdd:disabled {
    background-color: #8a8787;
}

.buttonreset:hover {
    background-color: #324960;
    color: rgb(255, 255, 255);
}

.button3 {
    background-color: #e2c449;
    color: black;
    /* border: 2px solid #9c9c7b; */
}

.button4 {
    background-color: #ec6161;
    border: 2px solid #9c9c7b;
}
```

```
.Back {
    background-color: #f7f588;
    color: black;
    border: 2px solid #9c9c7b;
}

.buttonBack {
    display: block;
    margin: auto;
}

.buttonAlignContactFrom {
    display: inline;
    margin: auto;
}

.button3:hover {
    background-color: #9c9c7b;
    color: black;
}

.button4:hover {
    background-color: #9c9c7b;
    color: black;
}

.center {
    display: block;
    margin: auto;
}

h1 {
    text-align: center;
    color: aliceblue;
    font-size: 52px;
    margin: auto;
}

h2 {
    text-align: center;
    color: #f7f588;
    margin: auto;
    font-size: 39px;
```

```
}

h3 {
    text-align: center;
    color: #860b0b;
    margin: auto;
    padding-top: 77px;
}

.tableinstraction {
    /* text-align: center; */
    /* letter-spacing: 1px; */
    font-size: 14px;
    font-family: 'Montserrat', sans-serif;
    color: #f6f288;
    text-transform: uppercase;
    font-weight: 500;
}

.footer {
    color: cornsilk;
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    font-size: small;
    display: flex;
    align-items: center;
    justify-content: center;
}

#p1 h4 {
    text-align: center;
    letter-spacing: 1px;
    font-size: 25px;
    font-family: 'Montserrat', sans-serif;
    color: #e2c449;
    text-transform: uppercase;
    font-weight: 500;
}

#pus h4 {
    text-align: center;
    letter-spacing: 1px;
    font-size: 56px;
    font-family: 'Montserrat', sans-serif;
```

```
    color: #f4af0b;
    text-transform: uppercase;
    font-weight: 500;
    margin-top: 2%;
    margin-bottom: 2%;
}

#pus h5 {
    text-align: center;
    letter-spacing: 1px;
    font-size: 15px;
    font-family: 'Montserrat', sans-serif;
    color: #f4af0b;
    text-transform: uppercase;
    font-weight: 500;
    margin-top: 1%;
    margin-bottom: 1%;
}

#pus h1 {
    text-align: center;
    letter-spacing: 1px;
    font-size: 27px;
    font-family: 'Montserrat', sans-serif;
    color: #f4af0b;
    text-transform: uppercase;
    font-weight: 500;
    margin-top: 2px;
    margin-bottom: 8px;
    color: #e0bb62;
}

#pus h2 {
    font-size: 15px;
    color: #6cd1d6;
    background-color: #ffffff54;
    font-family: 'Montserrat', sans-serif;
    padding: 1px 5px;
    width: 360px;
}

.page1 {
    padding: 5% 2% 4% 2%;
```

```
    margin: auto;
}

.pageus {
    padding: 3% 2% 4% 2%;
    margin: auto;
}

p {
    color: white;
    padding: 2% 15% 4% 15%;
    font-size: 17px;
    text-align: center;
}

.page2 {
    padding: 10% 2% 4% 2%;
    margin: auto;
}

.result {
    display: block;
    margin: auto;
    background-color: rgba(4, 29, 53, 0.58);
    border-radius: 12px;
    padding: 2% 2%;
    width: 80%;
    margin-bottom: 2%;
    border: 2px solid #b39004;
}

.result label {
    color: aliceblue;
    margin-right: 25%;
    font-size: x-large;
}

.resultsLable {
    color: #eac119;
    font-size: 19px;
}

.loginLable {
```

```
    color: rgb(243, 217, 23);
    font-size: x-large;
}

.searchResults {
    color: rgb(251, 88, 74);
    font-size: x-large;
}


/* Contact Form */

.contact {
    display: block;
    margin: auto;
    background-color: rgba(153, 182, 200, 0.678);
    border: 2px solid red;
    padding: 10px;
    border: 2px solid #f7f588;
    padding: 10px;
    border-radius: 15px;
    padding: 2% 2%;
    width: 60%;
    margin-bottom: 2%;
}

.contact label {
    color: aliceblue;
    margin-right: 10%;
    font-size: x-large;
}

.profile_image {
    margin: auto;
    display: table;
    align-items: center;
    content: url(/static/me.png);
}


/*socila Media*/

.socialMedia {
    display: flex;
```

```
    align-items: center;
    justify-content: center;
    margin-top: 2%;
    margin-bottom: 2%;
}

.fa {
    padding: 15px;
    font-size: 30px;
    width: 5;
    text-align: center;
    text-decoration: none;
    margin: 5px 2px;
    border-radius: 50%;
}

.fa:hover {
    opacity: 0.7;
}

.fa-google {
    background: #dd4b39;
    color: white;
}

.fa-linkedin {
    background: #007bb5;
    color: white;
}

.fa-github {
    background: #f0f0f0;
    color: rgba(0, 0, 0, 0.89)
}

.fa-rss {
    background: #ff6600;
    color: white;
}

.alert {
    padding: 15px;
    margin-left: 20%;
```

```
    margin-right: 20%;
    margin-top: -2%;
    margin-bottom: 3%;
    border: 1px solid transparent;
    border-radius: 4px;
    color: #8e1212;
}

.alertFailLog {
    padding: 15px;
    padding-left: 35px;
    margin-left: 0%;
    /* margin-right: 62%; */
    margin-top: 0%;
    margin-bottom: 0%;
    border: 1px solid transparent;
    border-radius: 4px;
    color: #8e1212;
    width: 30%;
}

.alert-dismissable .close,
.alert-dismissible .close {
    position: relative;
    top: -12px;
    right: 1px;
    color: white;
}

.fade.in {
    opacity: 1;
}

.alert-danger {
    color: #ffffff;
    background-color: #ad1d1da6;
    border-color: #dec86e;
}

.alert-okay {
    color: #961818;
    background-color: #a5c131cc;
    border-color: #251e02;
```

```
}

.close {
    float: right;
    font-size: 35px;
    font-weight: 700;
    line-height: 1;
    color: #000;
    text-shadow: 0 1px 0 #fff;
    filter: alpha(opacity=20);
    opacity: .2;
}

a {
    background-color: transparent;
}


/*for Mobile*/

@media only screen and (max-width: 768px) {
    /* For mobile phones: */
    html,
    body {
        background-position: bottom;
    }
    input[type=text],
    input[type=email],
    textarea {
        margin-left: 0%;
        width: 95%;
    }
    [class*="button"] {
        padding: 10px 35px;
        font-size: 16px;
        margin-left: 32%;
        margin-top: 4%;
    }
    .logo {
        height: 130px;
        width: 140px;
    }
    .profile_image {
```

```
        width: 40%;
    }
    [class*="fa"] {
        padding: 7px;
        font-size: 15px;
        width: 3;
    }
    [class*="footer"] {
        margin-top: 15%;
        font-size: 11px;
    }
    #p1 h4 {
        font-size: 14px;
        padding-top: 10px;
    }
    .page1 {
        padding: 25% 2% 4% 2%;
    }
    .pageus {
        padding: 25% 2% 4% 2%;
    }
    .page2 {
        padding: 25% 2% 4% 2%;
    }
    .result {
        width: 90%;
    }
    .result label {
        margin-right: 3%;
        font-size: large;
    }
    .searchResults {
        font-size: large;
    }
    .buttonBack {
        display: block;
        margin: auto;
        font-size: large;
    }
    .buttonreset {
        margin-left: 1%;
        padding: 13px 28px;
        font-size: 22px;
```

```
}
.buttonAddsimple {
    margin-left: 1%;
    margin-top: auto;
    width: 78%;
}
#myInput {
    background-image: url(/static/searchicon.png);
    background-position: 0px -0px;
    background-repeat: no-repeat;
    background-size: 18%;
    width: 83%;
    font-size: 16px;
    /* padding: 14px 6px 11px 48px; */
    border: 1px solid #ddd;
    margin-bottom: 6px;
    margin-left: 0.5%;
}
.userlogin {
    color: #fff;
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    margin-left: 4%;
    font-size: 50px;
    margin-bottom: 6%;
    margin-top: 25%;
    margin-left: -2%;
}
input[name=loginInput],
input[name=loginPass] {
    width: 85%;
}
.loginLable {
    color: rgb(243, 217, 23);
    font-size: x-large;
}
.loginStyle {
    margin-left: 8%;
}
.buttonContactUs {
    margin-left: 3%;
    width: -webkit-fill-available;
}
.alert {
```

```
        padding: 15px;
        margin-left: 8%;
        margin-right: 6%;
        margin-top: 3%;
        margin-bottom: 3%;
        border: 1px solid #fffbfb00;
        border-radius: 3px;
        color: #ffffff;
    }
    .socialMedia {
        display: flex;
        align-items: center;
        justify-content: center;
        margin-top: 2%;
        margin-bottom: 15%;
    }
    #pus h4 {
        font-size: 37px;
    }
    #pus h1 {
        font-size: 20px;
    }
    #pus h2 {
        font-size: 12px;
        width: 81%;
    }
    #pus h5 {
        font-size: 12px;
    }
    p {
        font-size: 14px;
        padding: 10% 5%;
    }
    .zoom {
        width: 46%;
    }
}


/* Table Styles */

.resultAllLocations {
    display: block;
```

```
    margin: auto;
    background-color: rgba(26, 103, 115, 0.34);
    border-radius: 12px;
    padding: 2% 2%;
    width: auto;
    margin-bottom: 2%;
    border: 2px solid #b39004;
}

.table-wrapper {
    margin: 31px 0px 31px;
    box-shadow: 0px 35px 50px rgba( 0, 0, 0, 0.2);
}

.fl-table {
    border-radius: 5px;
    font-size: 16px;
    font-weight: normal;
    border: #fff;
    border-collapse: collapse;
    width: 100%;
    max-width: 100%;
    background-color: white;
    margin-top: 1%;
}

.fl-table td,
.fl-table th {
    text-align: center;
    padding: 8px;
}

.fl-table td {
    border-right: 1px solid #f8f8f8;
    font-size: 15px;
}

.fl-table thead th {
    color: #ffffff;
    background: rgb(149, 138, 33);
}

.fl-table thead th:nth-child(odd) {
```

```
    color: #ffffff;
    background: #324960;
}

.fl-table tr:nth-child(even) {
    background: #F8F8F8;
}


/* Responsive */

@media (max-width: 767px) {
    .fl-table {
        display: block;
        width: 100%;
    }
    .table-wrapper:before {
        content: "Scroll horizontally >";
        display: block;
        text-align: right;
        font-size: 11px;
        color: white;
        padding: 0 0 10px;
    }
    .fl-table thead,
    .fl-table tbody,
    .fl-table thead th {
        display: block;
    }
    .fl-table thead th:last-child {
        border-bottom: none;
    }
    .fl-table thead {
        float: left;
    }
    .fl-table tbody {
        width: auto;
        position: relative;
        overflow-x: auto;
    }
    .fl-table td,
    .fl-table th {
        padding: 20px .625em .625em .625em;
```

```
        height: 60px;
        vertical-align: middle;
        box-sizing: border-box;
        overflow-x: hidden;
        overflow-y: auto;
        width: 120px;
        font-size: 13px;
        text-overflow: ellipsis;
    }
    .fl-table thead th {
        text-align: left;
        border-bottom: 1px solid #f7f7f9;
    }
    .fl-table tbody tr {
        display: table-cell;
    }
    .fl-table tbody tr:nth-child(odd) {
        background: none;
    }
    .fl-table tr:nth-child(even) {
        background: transparent;
    }
    .fl-table tr td:nth-child(odd) {
        background: #F8F8F8;
        border-right: 1px solid #E6E4E4;
    }
    .fl-table tr td:nth-child(even) {
        border-right: 1px solid #E6E4E4;
    }
    .fl-table tbody td {
        display: block;
        text-align: center;
    }
}


/*     log in form */
```

Project Code Back-End

14. eihIR.py

```
import RPi.GPIO as GPIO
import subprocess
import argparse
import os
import pdb
import sys
from picamera import PiCamera
from datetime import datetime
import time

## Camera Configuration
camera = PiCamera()
camera.resolution = (400,200)


## Sensors Configuration
BEAM_PIN1 = 4  ##sensor Number !
BEAM_PIN2 = 17 ##sensor Number 2


##current working directory
path = os.getcwd()
# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
listArg =  sys.argv[1:]
direction = listArg[0]


def ir1First(channel):
    print('one First')
    print('WAIT TWO >>>>')
    ## ignoring further edges for 1000ms and wait for one again
    channel = GPIO.wait_for_edge(BEAM_PIN2, GPIO.RISING, timeout=1000)
    ## when the sensor 2 beam break then we start the process
    if channel is None:
        print('Timeout occurred')
    else:
        ## if sensor two was detected the camera run to take picture
        print('CAMERA', channel)
        print ('*** PICTURE TAKEN ***')
```

```
        camera.capture("{}/images/pic.png".format(path))
        #timeNow = str(datetime.now().strftime("%H,%M,%S"));
        #camera.capture("/home/pi/year4/projectEIH/body-
detect/images/pic"+timeNow+".png")
        print ('++++ BODY DETECTING ++++ ')
        command = "python {}/detect.py --images {}/images/ -
d {}".format(path,path, direction )
        subprocess.call([command], shell=True)




## set the pins mode
GPIO.setmode(GPIO.BCM)
## set up the sensors to the rass pi
GPIO.setup(BEAM_PIN2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(BEAM_PIN1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
## If any beam break happened for the sensor 1 callback the funstion to check sensor 2
# ignoring further edges for 200ms for switch bounce handling
GPIO.add_event_detect(BEAM_PIN1, GPIO.RISING, callback=ir1First, bouncetime=200)

#exit the program when user press enter.
message = input("Press enter to quit\n\n")
GPIO.cleanup()
```

15. Post-data-to-api.py

```
from firebase import firebase
import requests
import argparse
import time
import os
import pyrebase


def get_stored_number_of_people_from_db(reg_id):
    r = requests.get('https://projecteih.firebaseio.com/locations.json')
    x= r.json()
    return x[reg_id]['numberOfPeopleINDetect']


def db_config():
    path = os.getcwd()

    config = {
    "apiKey": "AIzaSyBa_oAgm7dmE-sFGGm8XG7HYs0gWxVFyJ8",
    "authDomain": "projecteih.firebaseio.com",
    "databaseURL": "https://projecteih.firebaseio.com",
    "storageBucket": "projecteih.appspot.com",
    "serviceAccount": "{}/cred/projecteih-firebase-adminsdk-dmd9b-
dfbc30ba25.json".format(path )
    }
    firebasePy = pyrebase.initialize_app(config)

    return firebasePy


def update_with_the_new_number(id_to_reset, new_number):
    db = db_config().database()
    x= db.child("locations").child(id_to_reset).update({'numberOfPeopleINDetect': new_n
umber})
    return x

## get the previouse number on the API
url = os.environ['FIREBASE_DB_URL']
reg_id = os.environ['REG_BUILIDING_ID']
device_id = os.environ['DEVICE_ID']
```

```
## read the argument sent from line 55 the detect.py file with the number of new people
 detected
ap = argparse.ArgumentParser()
ap.add_argument("-n", "--numberOUT", type=int,
help="Current number of people OUT to send to the API")
ap.add_argument("-n2", "--numberIN", type=int,
help="Current number of people IN to send to the API")
args = vars(ap.parse_args())
currentNumberOUT = args["numberOUT"]
currentNumberIN = args["numberIN"]

## get the previous number from the DB
preNumber = get_stored_number_of_people_from_db(reg_id)

if args["numberOUT"] is not None:
    ## substact the number coming from the API to the number coming from the RassPi
    newNumber = preNumber- currentNumberOUT
elif args["numberIN"] is not None:
    ## add the number coming from the API to the number coming from the RassPi
    newNumber = preNumber + currentNumberIN

## send the new number to the update with the building ID
update_with_the_new_number(reg_id, newNumber)
```

16. setup.py

```
import argparse
import os
import pdb
import sys

print("##########################################################################################
#################")
print("###################################     SETUP START       ######################
#################")
print("##########################################################################################
#################")
##installing python
os.system('sudo apt-get update')
os.system('sudo apt-get install python3.6')
## Installing pip
os.system('curl "https://bootstrap.pypa.io/get-pip.py" -o "get-pip.py"')
os.system('python get-pip.py')
os.system('python3 get-pip.py')      # For specific python version
## Install all python dependency
os.system('sudo pip install requests')
os.system('sudo pip install python-firebase')
os.system('pip3 install firebase')
os.system('pip3 install imutils')
os.system('pip3 install python_jwt')
os.system('pip3 install gcloud')
os.system('pip3 install sseclient')
os.system('pip3 install parse')
os.system('pip3 install requests_toolbelt')
os.system('pip3 install flask')
os.system('pip3 install Crypto')
os.system('pip install pyrebase')
os.system('pip3 install pyrebase')

print("##########################################################################################
#################")
print("###################################     SETUP DONE        ######################
#################")
print("##########################################################################################
#################")
```