TU berlin CAMPUS EL GOUNA

Department of Energy Engineering

INTERDISCIPLINARY PROJECT NERKS

Presented by:

Group 1

Under the Supervision of

Professor Dr. Christoph Nytsch-Geusen

Christoph Banhardt M.Sc.

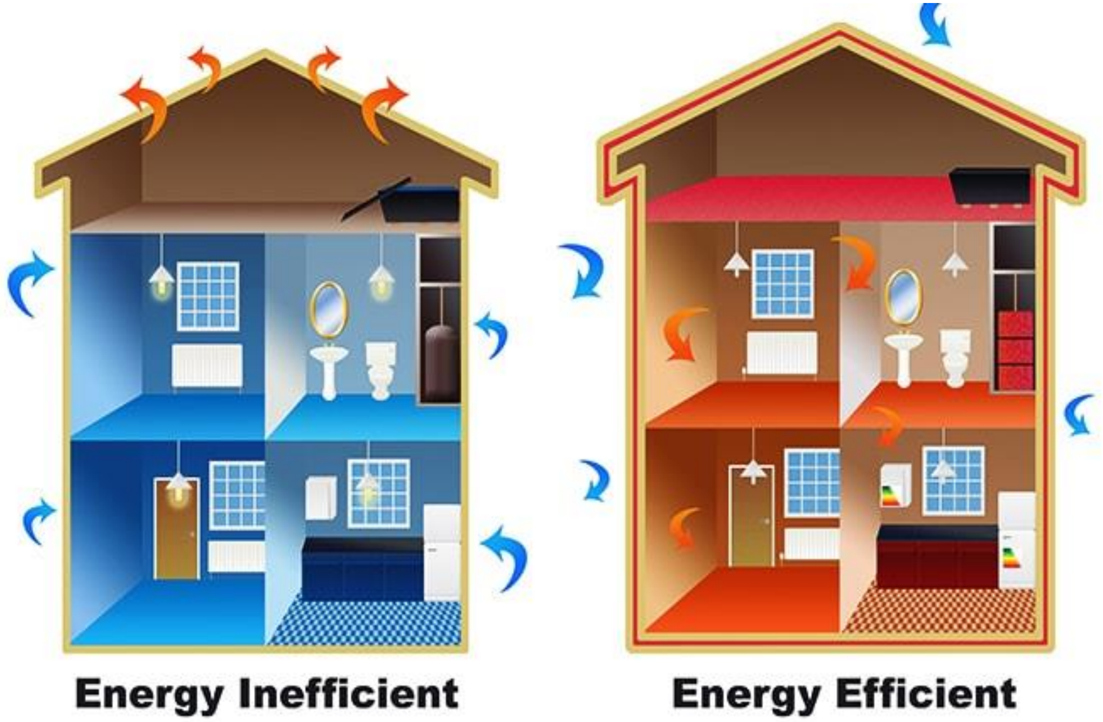
DATE- 31.07.2020

Agenda

- Motivation
- Introduction
- Description - Local Room and Building Situation
- Hardware and Software Technology
- Experimental Analysis
- User Interface
- Methodology – Process Flowchart
- Code Snippets
- Outputs
- Conclusion
- Future Potential Development

Motivation

Energy-saving buildings

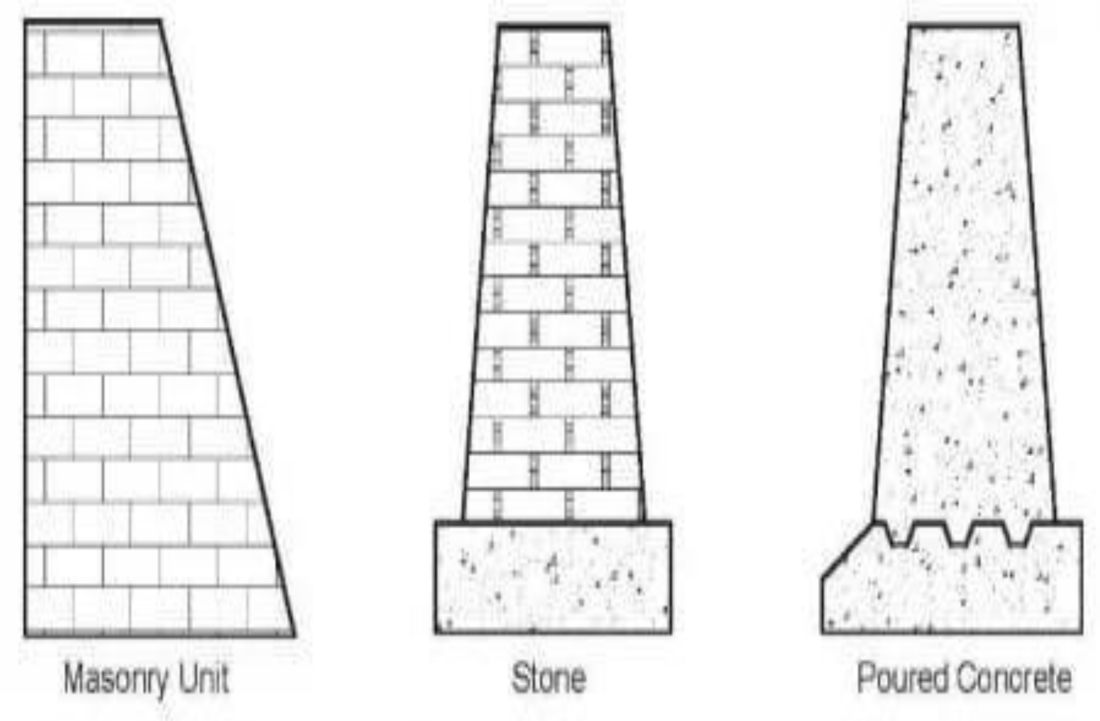


https://gosmartbricks.com/wp-content/uploads/2018/07/energy-inefficient-versus-energy-efficient-home-illustration.jpg
last visited: 30.07.2020

- Energy Engineer’s rule
- Thermal comfort
- AC/Heater
- Money saving

Introduction

General Idea



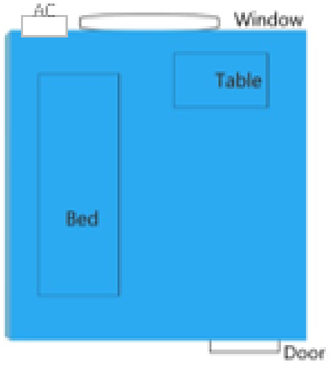
- Wall material
- Wall thickness
- Environmental conditions

https://theconstructor.org/wp-content/uploads/2018/09/Materials-used-for-gravity-retaining-wall-construction-450x202.jpg
last visited: 30.07.2020

Description, Local Room and Building Situation

Neelesh

https://www.google.com/maps/place/Jamals+luxor+apartments/@13.0479271,80.1202573,3a,75y,90t/data=!3m8!1e2!3m6!1sAF1QipNENyH-RjME3yK4ajWBYaxjXytcbk7gX73YJNY-!2e10!3e12!6shttps:%2F%2Flh5.googleusercontent.com%2Fp%2FAF1QipNENyH-RjME3yK4ajWBYaxjXytcbk7gX73YJNY-%3Dw203-h152-k-no!7i4160!8i3120!4m5!3m4!1s0x0:0xe58d4f93c7a1b22f!8m2!3d13.0481078!4d80.1199594

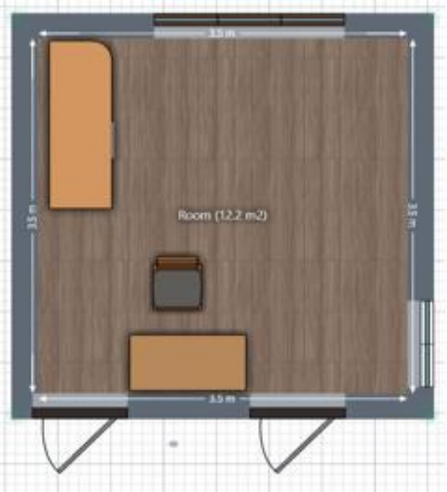


Room Layout

- Location – Chennai-South India.
- Room Dimension-2.5m(l)*2.5m(b)*3m(h)
- Construction material & thickness – Concrete and steel & 0.5m.
- Floor – 10th.

Shashank

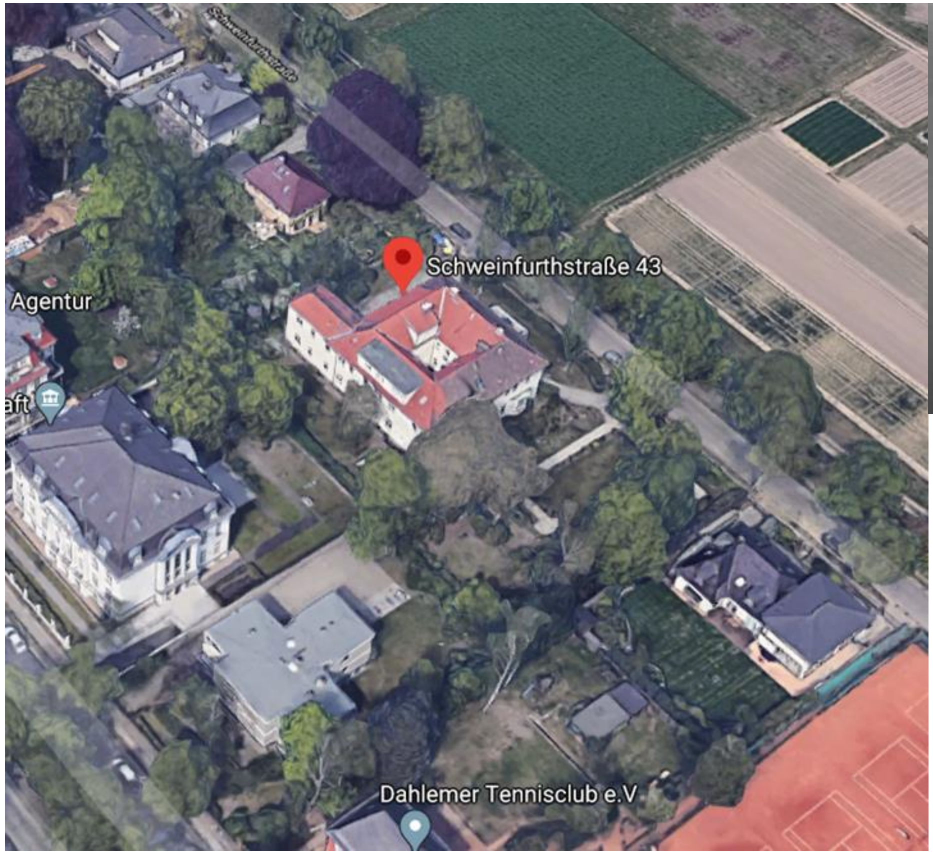
www.planner5d.com



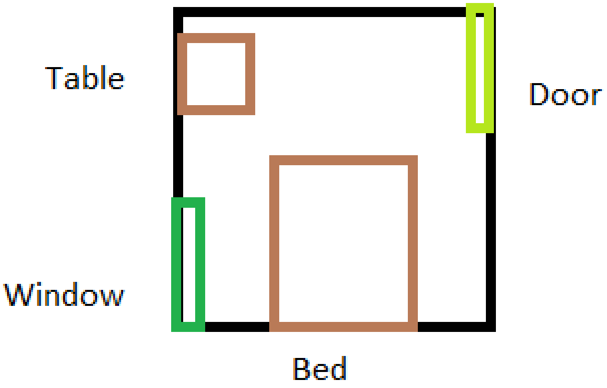
- Location – Dehradun-North India.
- Room Dimension- 3.9m(l)*3.9m(b)*3.6m(h)
- Construction material & thickness – Brick and Cement & 0.25m.
- Floor – 1st.

Description, Local Room and Building Situation

Rohit



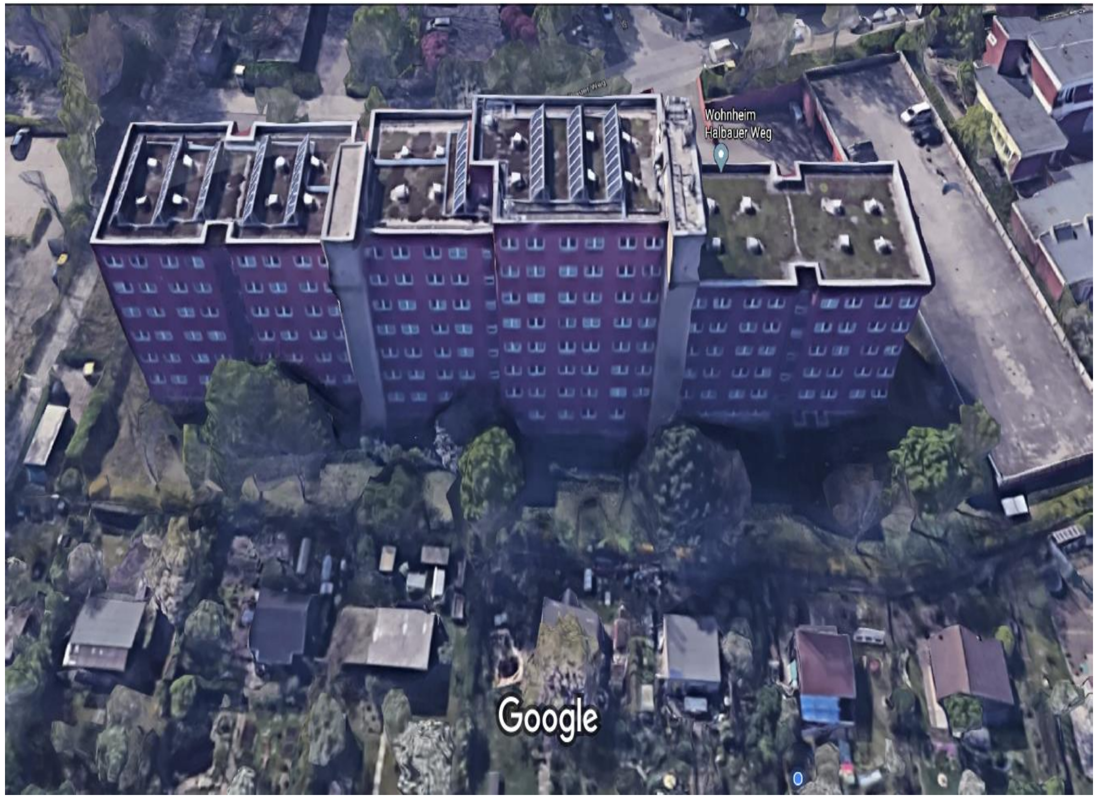
https://www.google.de/maps/place/Schweinfurthstra%C3%9Fe+43,+14195+Berlin/@52.4654669,13.2963241,53a,35y,39.56t/data=!3m1!1e3!4m5!3m4!1s0x47a85a76606a8e03:0x453f84ad3f5b70e1!8m2!3d52.4659782!4d13.2966038



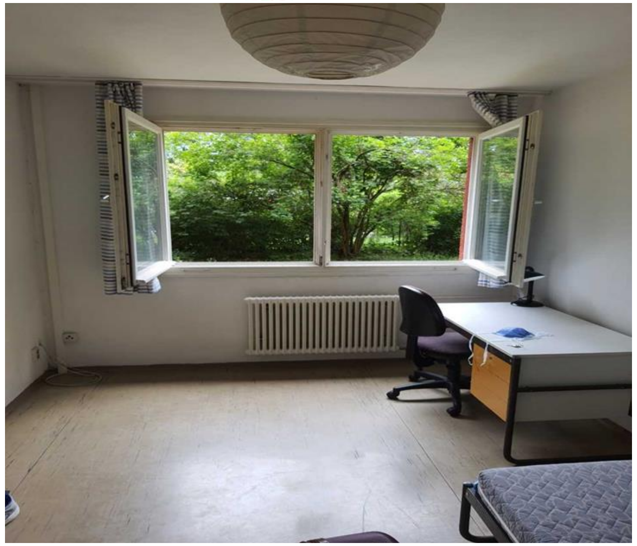
- Location – Berlin.
- Room Dimension- 3.5m(l)*3.5m(b)*3.6m(h)
- Construction material & thickness – Red Brick and Cement & 0.33m.
- Floor – 1st.

Description, Local Room and Building Situation

Eslam



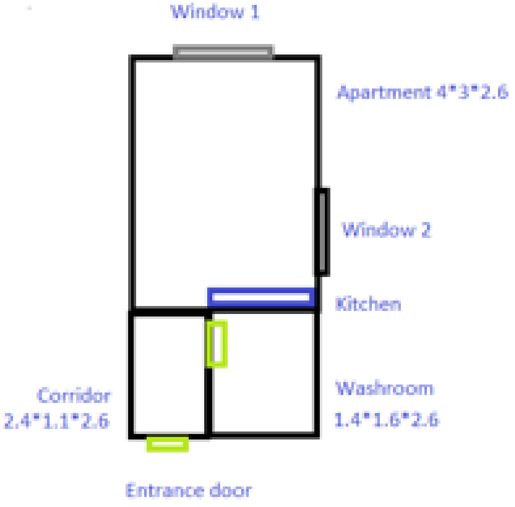
https://www.google.com/maps/@52.4303266,13.3467264,14z



- Location – Berlin.
- Room Dimension-4m(l)*3m(b)*3m(h)
- Construction material & thickness – Concrete, Steel & Gypsum board & 0.30m.
- Floor – Ground Floor.

Kunal

https://www.google.com/maps/@52.4303266,13.3467264,14z



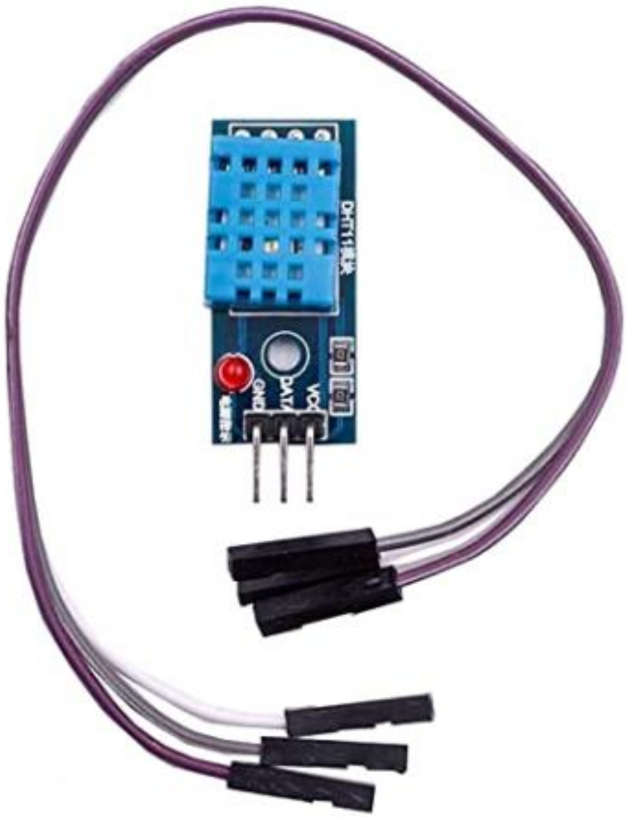
- Location – Berlin.
- Room Dimension-4m(l)*3m(b)*2.6m(h)
- Construction material & thickness – Concrete & Cement & 0.40m.
- Floor – 5th Floor.

Microprocessor

https://www.amazon.de/gp/product/B06Y1ZPNMS/ref=ppx_yo_dt_b_asin_title_o00_s01?ie=UTF8&psc=1

- AZDelivery NodeMCU Lua Lolin V3 WiFi Parent.
- Performance ESP6266 Micro controller.
- Comfortable Prototyping with easy programming using a Lua script or Arduino code and breadboard compatible design (Sten 28 mm Pinlei distance).
- Dimensions (L x W x H): 58 x 31 x 13 mm

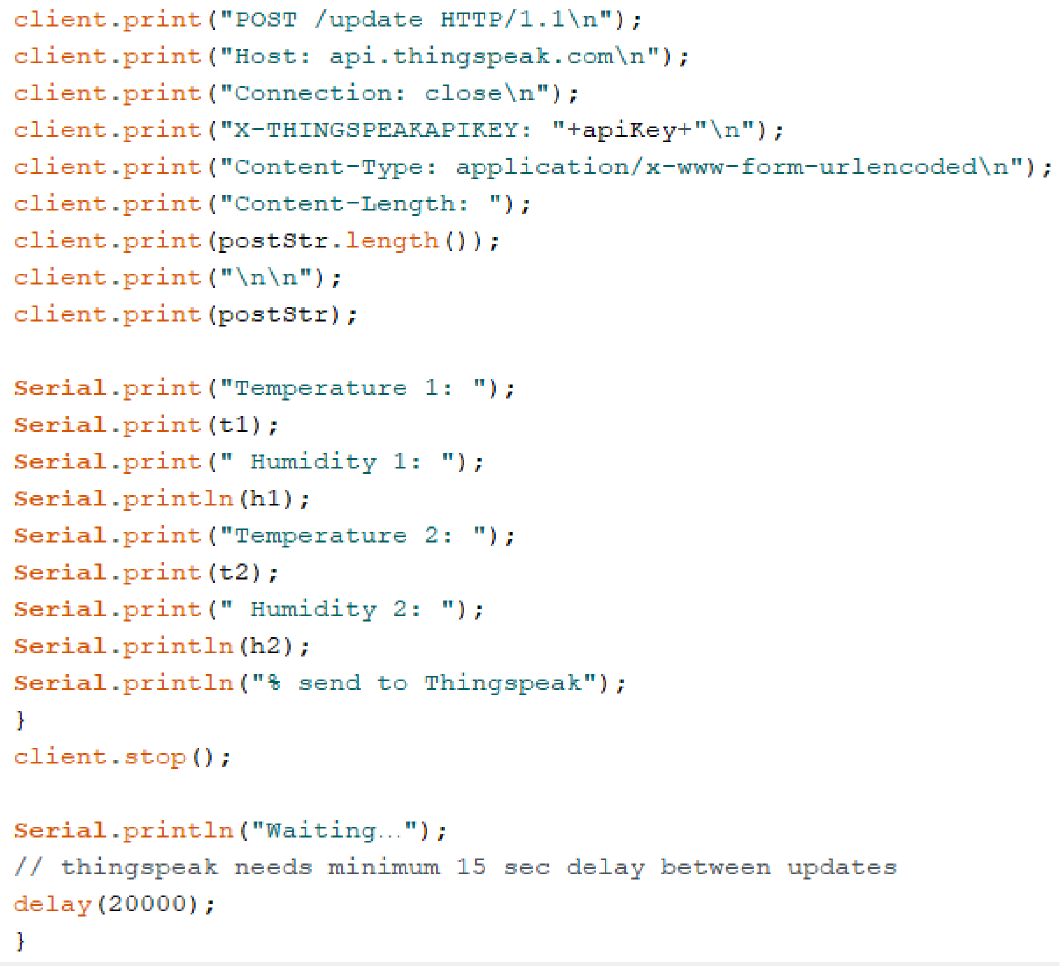
Sensor



https://www.amazon.de/gp/product/B07TXR5NQ6/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1

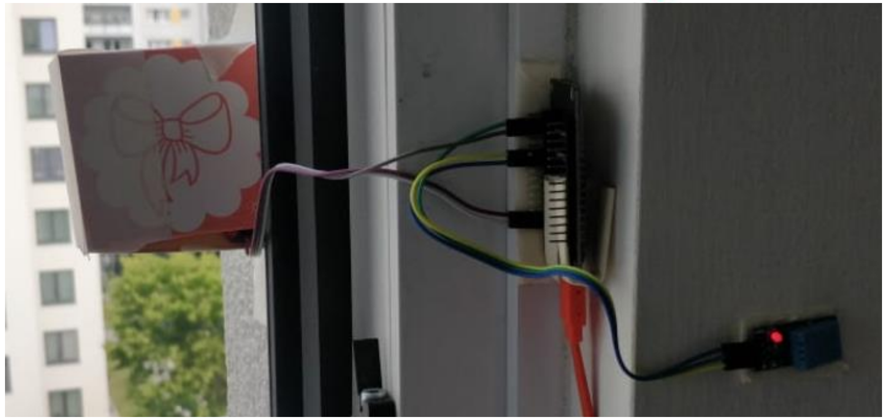
- AZDelivery DHT11/DHT22 Breakout Module Parent.
- The DHT11 is a reliable sensor for determining temperature and humidity.
- Since the sensor can be operated with 3.3V and 5V, it is suitable for connecting to Arduino, RN-Control, Raspberry Pi and
- The output is serially as a digital bit sequence.

Setup



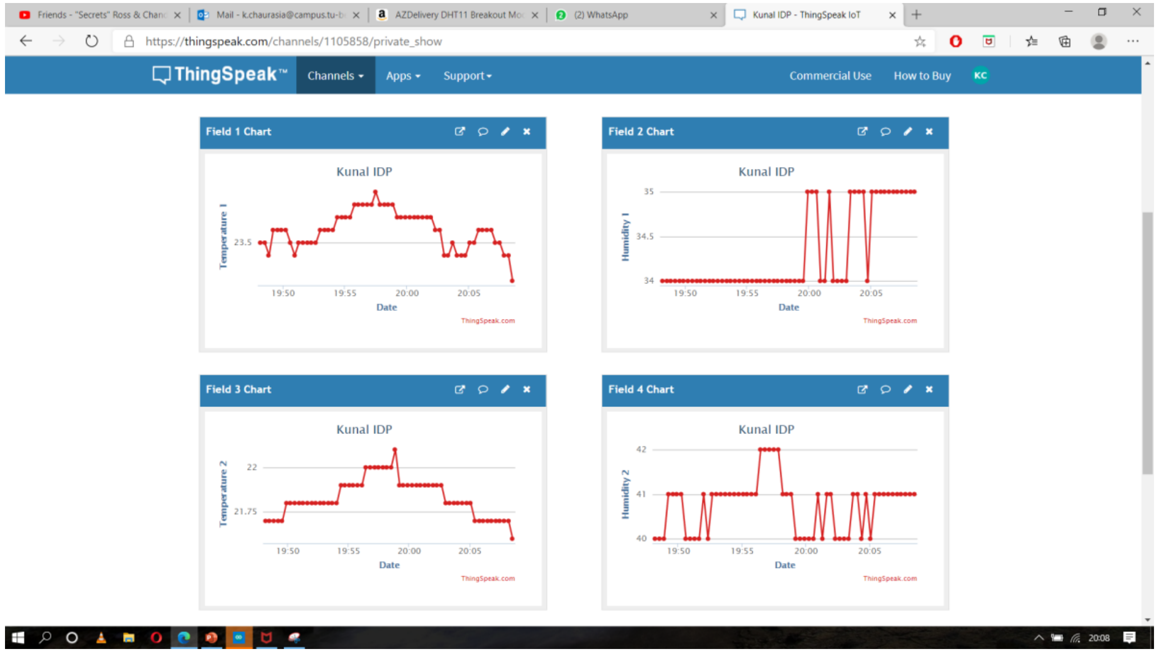
```
client.print("POST /update HTTP/1.1\n");
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(postStr.length());
client.print("\n\n");
client.print(postStr);

Serial.print("Temperature 1: ");
Serial.print(t1);
Serial.print(" Humidity 1: ");
Serial.println(h1);
Serial.print("Temperature 2: ");
Serial.print(t2);
Serial.print(" Humidity 2: ");
Serial.println(h2);
Serial.println("% send to Thingspeak");
}
client.stop();

Serial.println("Waiting...");
// thingspeak needs minimum 15 sec delay between updates
delay(20000);
}
```

Arduino Code



- Two DHT11 sensors were connected to the NodeMCU.
- Code was adjusted to connect the NodeMCU over wi-fi and send the data to client “ThinkSpeak”.
- ThinkSpeak recorded the data of Temperature and Humidity for both the Inside wall and Outside wall.

ThinkSpeak

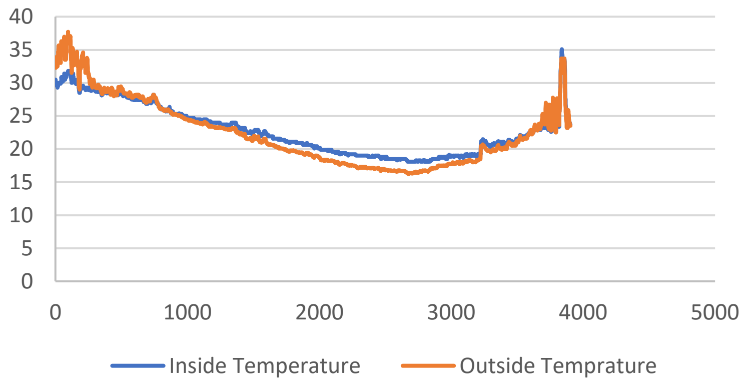


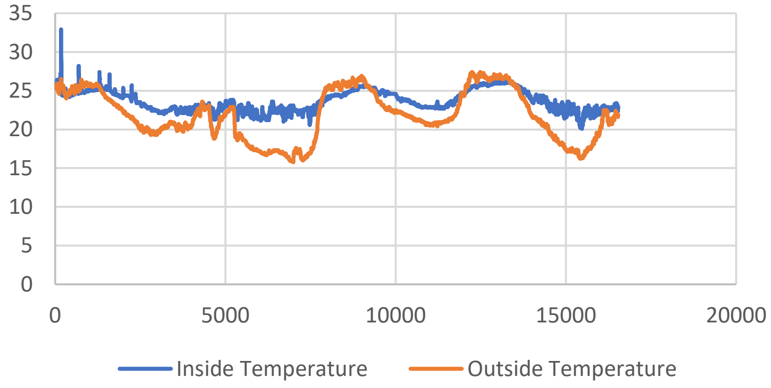
- ThinkSpeak Stores the Data send from the Microprocessor over wifi.
- The Graphs can be uploaded live directly to the website.
- ThinkSpeak lets you extract the data in .CSV, .JSON, .EML file.

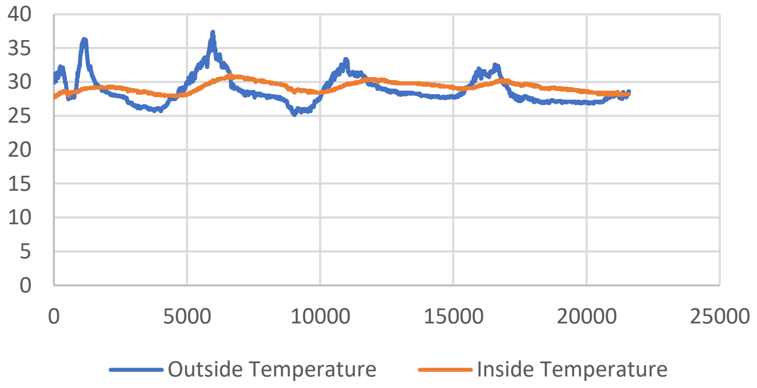
https://thingspeak.com/channels/1105858/private_show

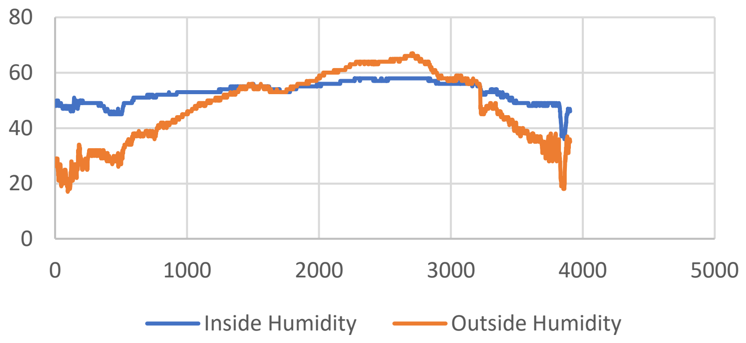
Experimental Analysis

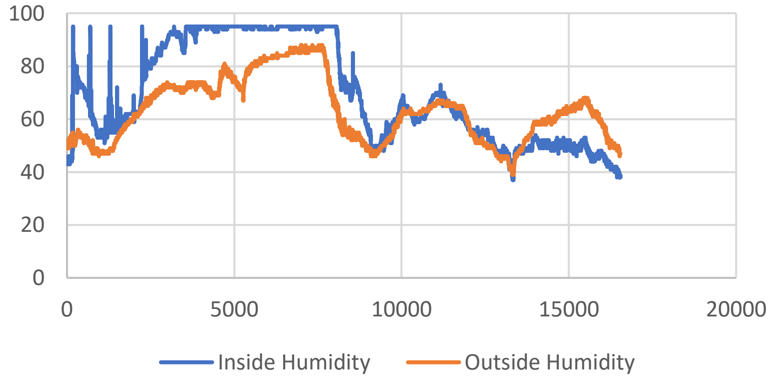
Result

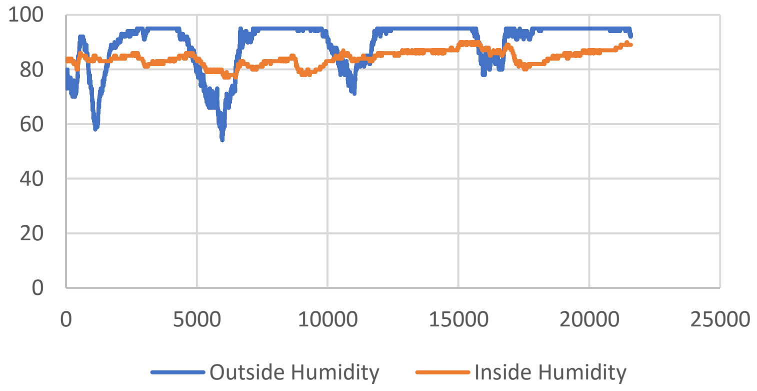

Temperature Variation (Rohit)


Temperature Variation (Kunal)

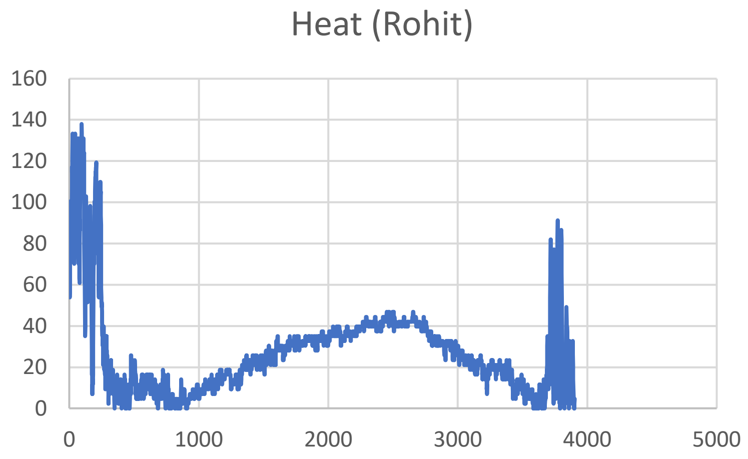

Temperature Variation(Shashank)

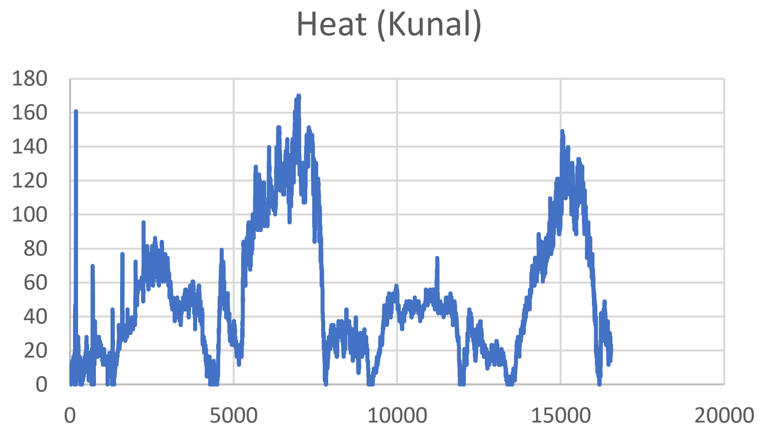

Humidity Variation (Rohit)

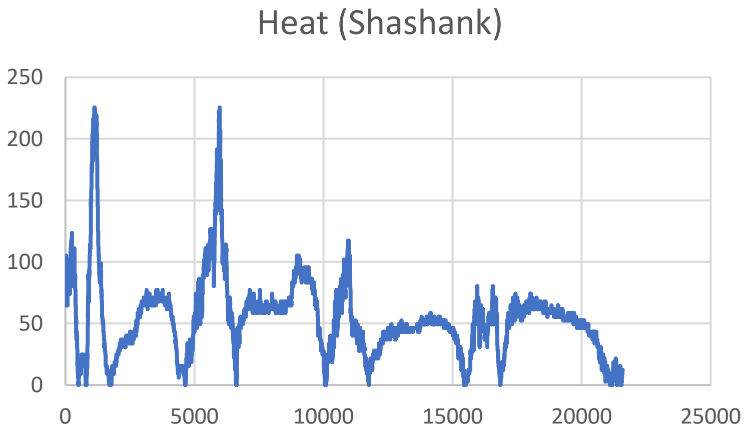

Humidity Variation (Kunal)

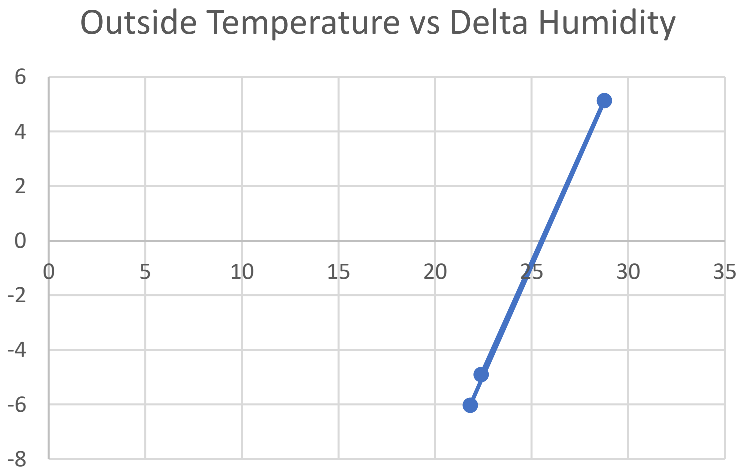

Humidity Variation (Shashank)

Result


Heat (Rohit)


Heat (Kunal)

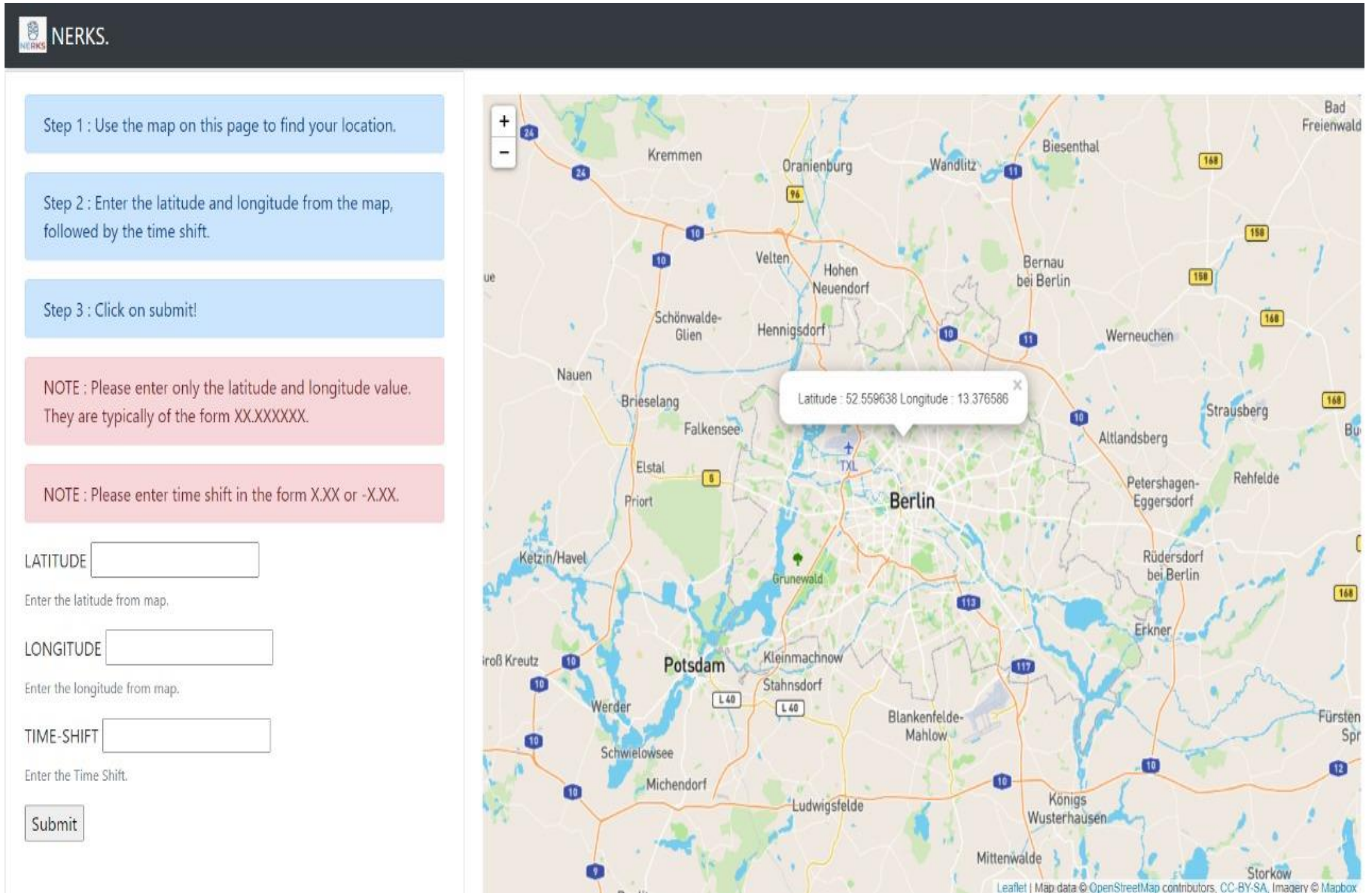

Heat (Shashank)


Outside Temperature vs Delta Humidity

- Delta Humidity relation with Outside Temperature.
- Relation of Delta temperature vs Thickness of wall is extrapolated from the data.
- The relation is used in the python code.

User Interface

Inputs



NERKS.

Step 1 : Use the map on this page to find your location.

Step 2 : Enter the latitude and longitude from the map, followed by the time shift.

Step 3 : Click on submit!

NOTE : Please enter only the latitude and longitude value. They are typically of the form XX.XXXXXX.

NOTE : Please enter time shift in the form X.XX or -X.XX.

LATITUDE

Enter the latitude from map.

LONGITUDE

Enter the longitude from map.

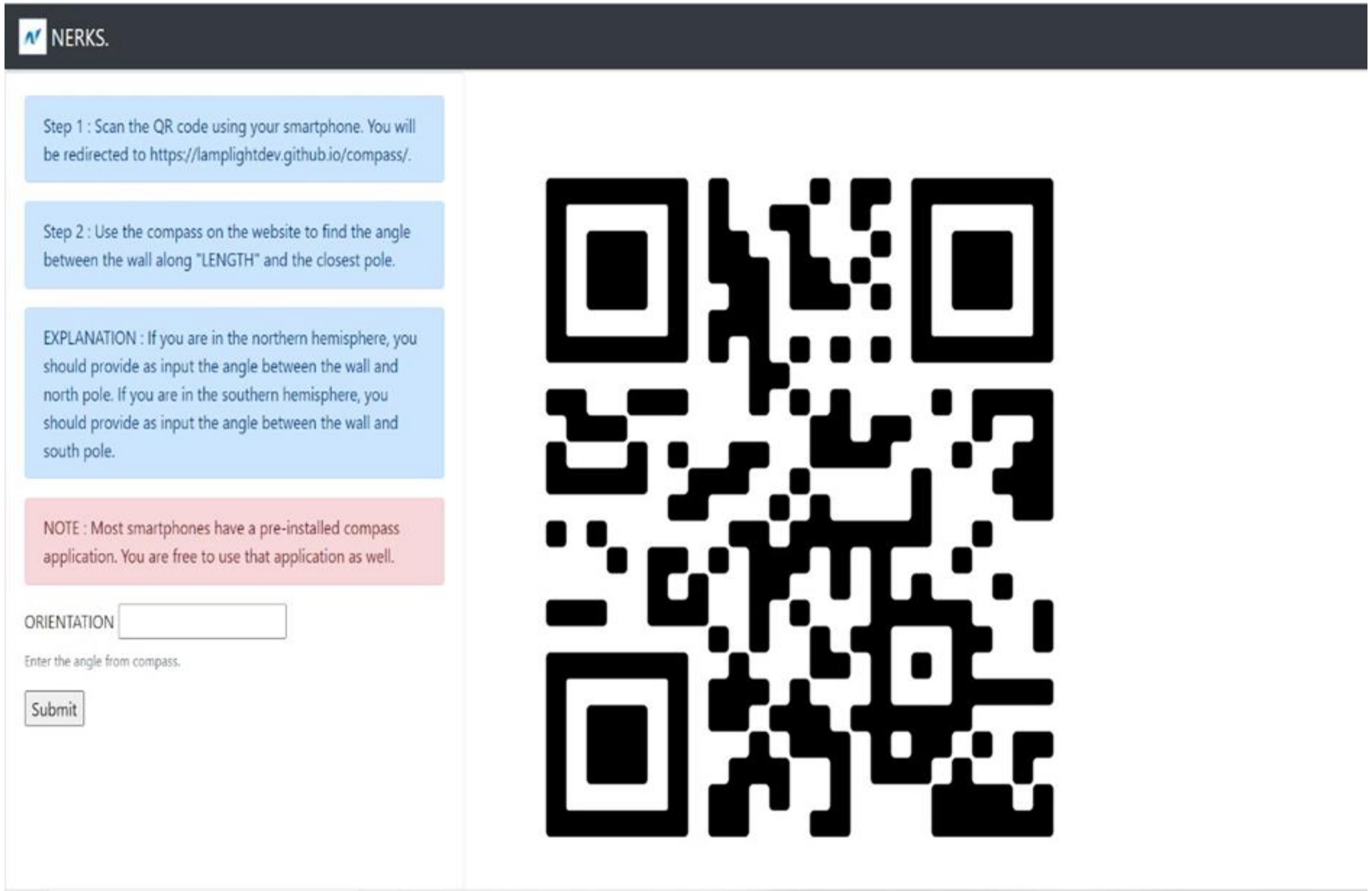
TIME-SHIFT

Enter the Time Shift.

Submit

Latitude : 52.559638 Longitude : 13.376586

Inputs



Step 1 : Scan the QR code using your smartphone. You will be redirected to https://lamplightdev.github.io/compass/.

Step 2 : Use the compass on the website to find the angle between the wall along "LENGTH" and the closest pole.

EXPLANATION : If you are in the northern hemisphere, you should provide as input the angle between the wall and north pole. If you are in the southern hemisphere, you should provide as input the angle between the wall and south pole.

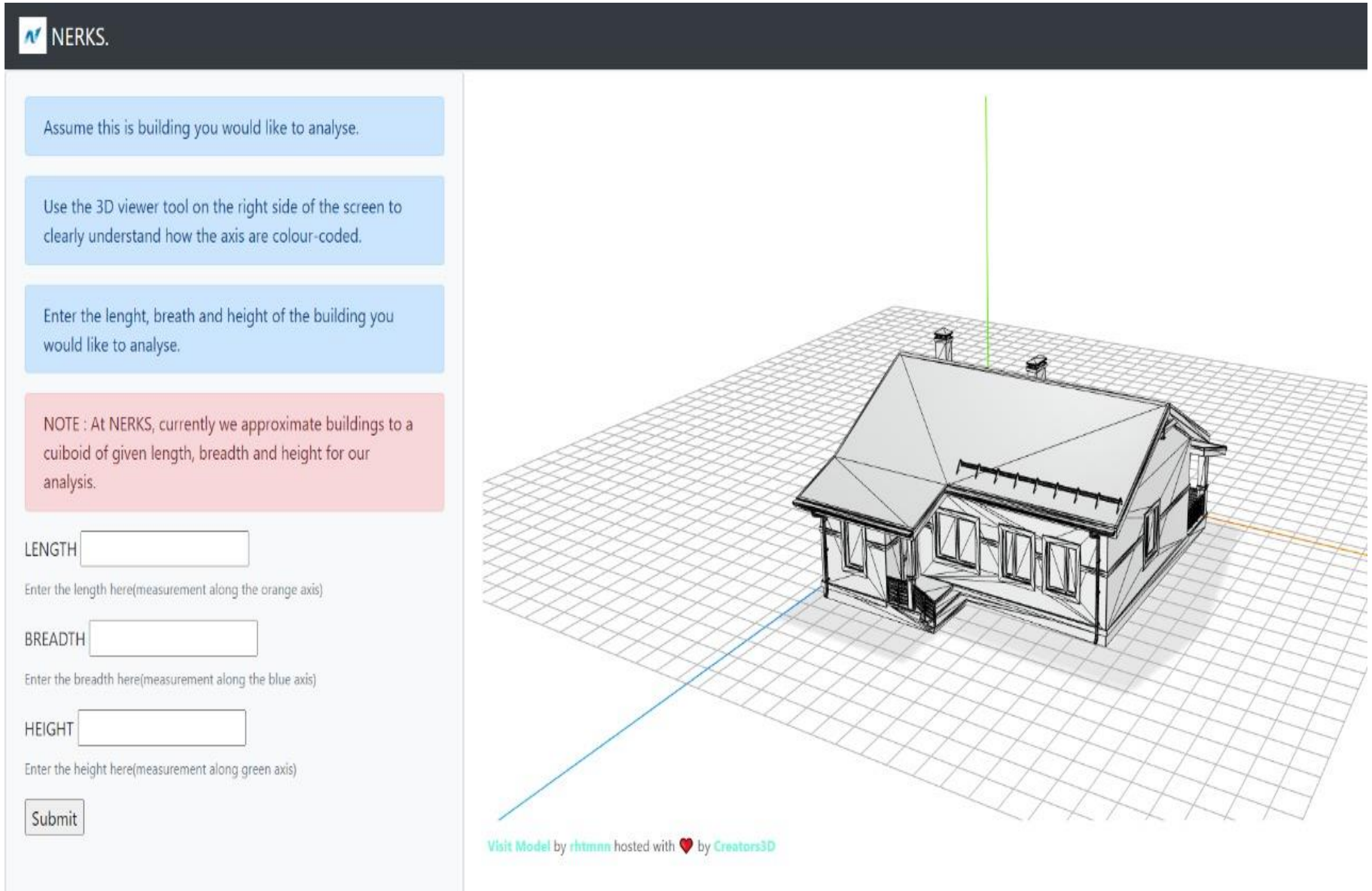
NOTE : Most smartphones have a pre-installed compass application. You are free to use that application as well.

ORIENTATION

Enter the angle from compass.

User Interface

Inputs



NERKS.

Assume this is building you would like to analyse.

Use the 3D viewer tool on the right side of the screen to clearly understand how the axis are colour-coded.

Enter the lenght, breath and height of the building you would like to analyse.

NOTE : At NERKS, currently we approximate buildings to a cuboid of given length, breadth and height for our analysis.

LENGTH

Enter the length here(measurement along the orange axis)

BREADTH

Enter the breadth here(measurement along the blue axis)

HEIGHT

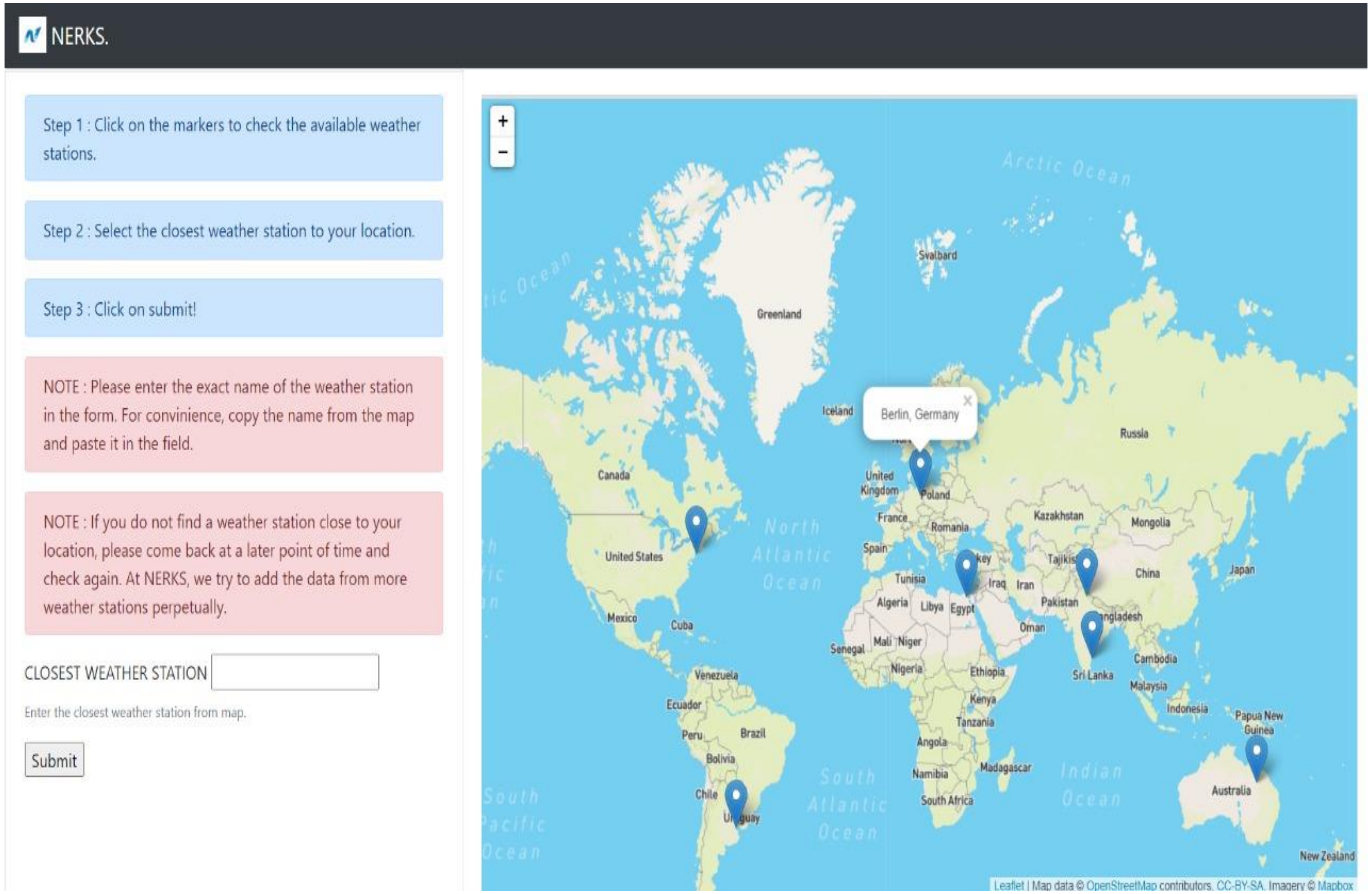
Enter the height here(measurement along green axis)

Submit

Visit Model by rhtmnn hosted with ♥ by Creators3D

User Interface

Inputs



NERKS.

Step 1 : Click on the markers to check the available weather stations.

Step 2 : Select the closest weather station to your location.

Step 3 : Click on submit!

NOTE : Please enter the exact name of the weather station in the form. For convinience, copy the name from the map and paste it in the field.

NOTE : If you do not find a weather station close to your location, please come back at a later point of time and check again. At NERKS, we try to add the data from more weather stations perpetually.

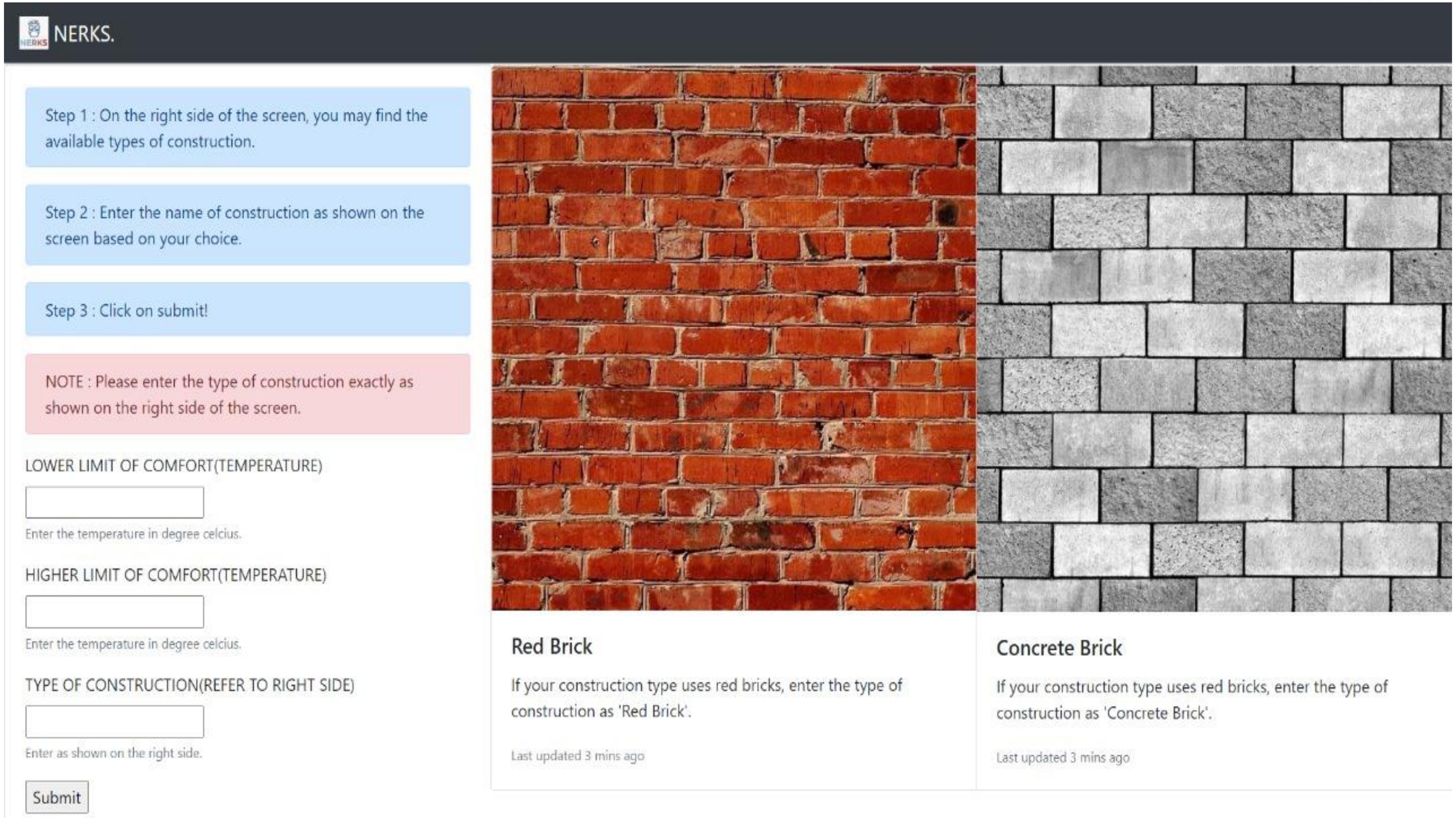
CLOSEST WEATHER STATION

Enter the closest weather station from map.

Submit

User Interface

Inputs



NERKS.

Step 1 : On the right side of the screen, you may find the available types of construction.

Step 2 : Enter the name of construction as shown on the screen based on your choice.

Step 3 : Click on submit!

NOTE : Please enter the type of construction exactly as shown on the right side of the screen.

LOWER LIMIT OF COMFORT(TEMPERATURE)

Enter the temperature in degree celcius.

HIGHER LIMIT OF COMFORT(TEMPERATURE)

Enter the temperature in degree celcius.

TYPE OF CONSTRUCTION(REFER TO RIGHT SIDE)

Enter as shown on the right side.

Red Brick

If your construction type uses red bricks, enter the type of construction as 'Red Brick'.

Last updated 3 mins ago

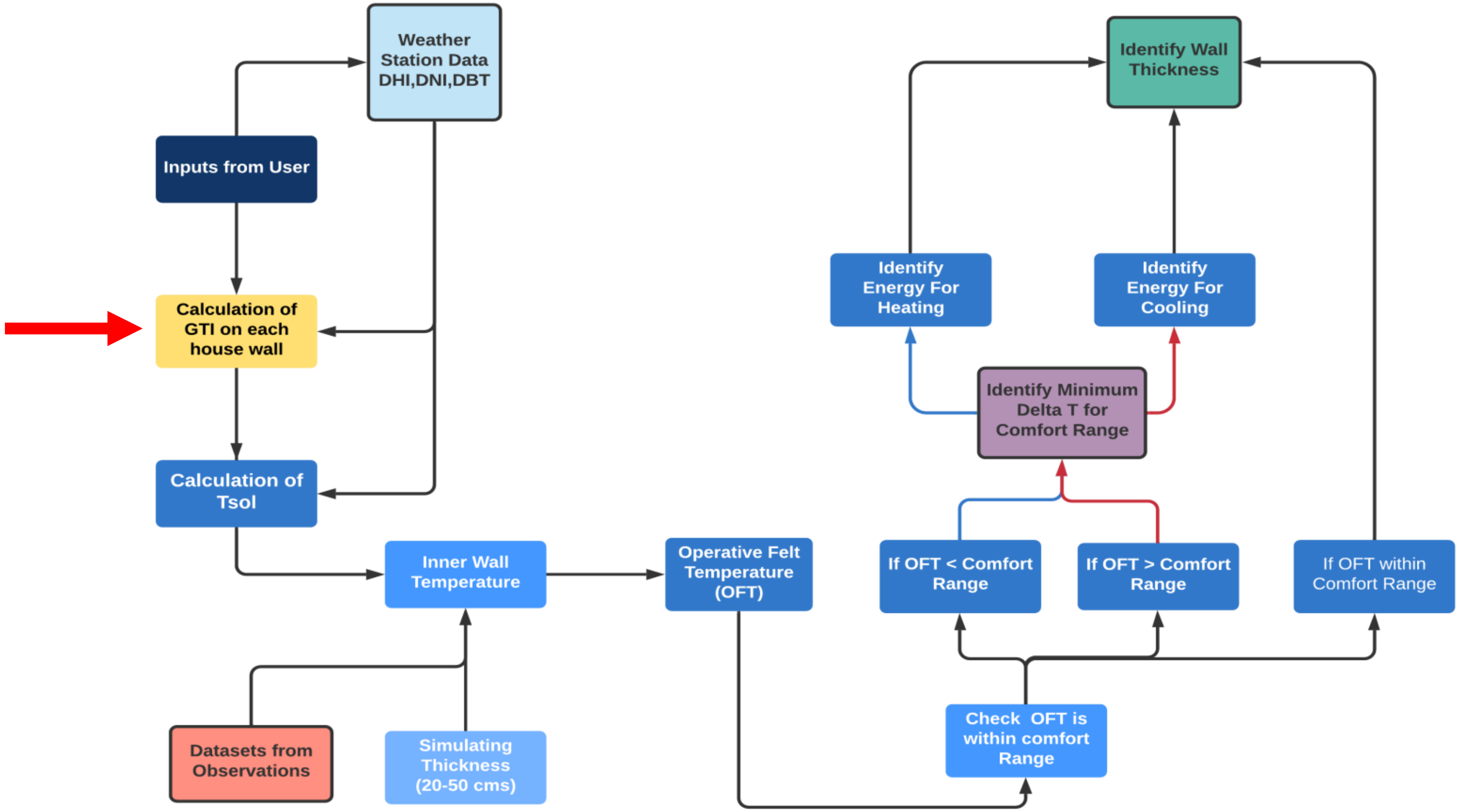
Concrete Brick

If your construction type uses red bricks, enter the type of construction as 'Concrete Brick'.

Last updated 3 mins ago

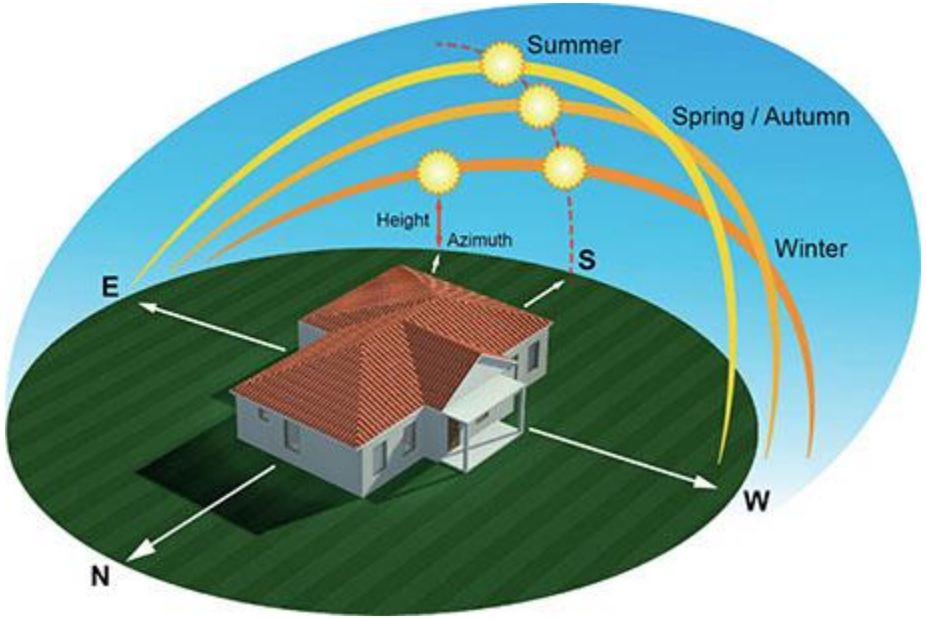
Methodology

Logic Flowchart


Weather Station Data DHI,DNI,DBT
Inputs from User
Calculation of GTI on each house wall
Calculation of Tsol
Inner Wall Temperature
Operative Felt Temperature (OFT)
Datasets from Observations
Simulating Thickness (20-50 cms)
Check OFT is within comfort Range
If OFT < Comfort Range
If OFT > Comfort Range
If OFT within Comfort Range
Identify Minimum Delta T for Comfort Range
Identify Energy For Heating
Identify Energy For Cooling
Identify Wall Thickness

Methodology

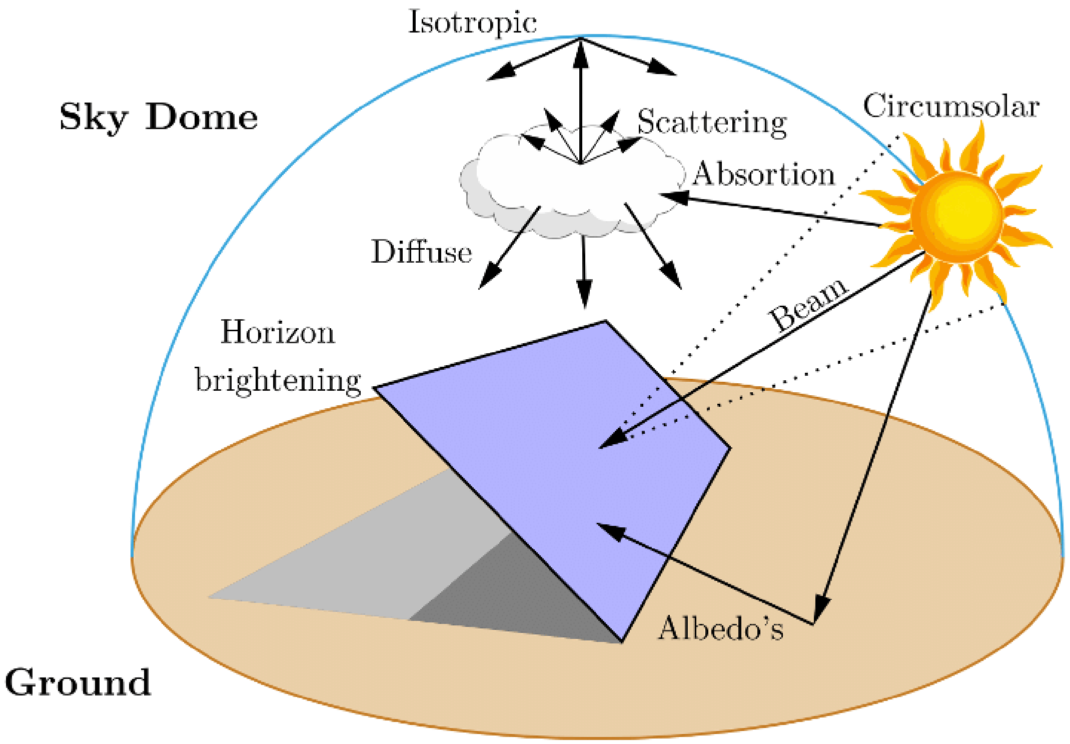
Calculation Of GTI on each House Wall



https://www.es-so-database.com/index.php/knowledge/appendices
Date last visited: 30/07/2020

- Based on : Input from user and data from Weather station
- Calculate solar position (Azimuth and Inclination angles) for everyday

Calculation Of GTI on each House Wall

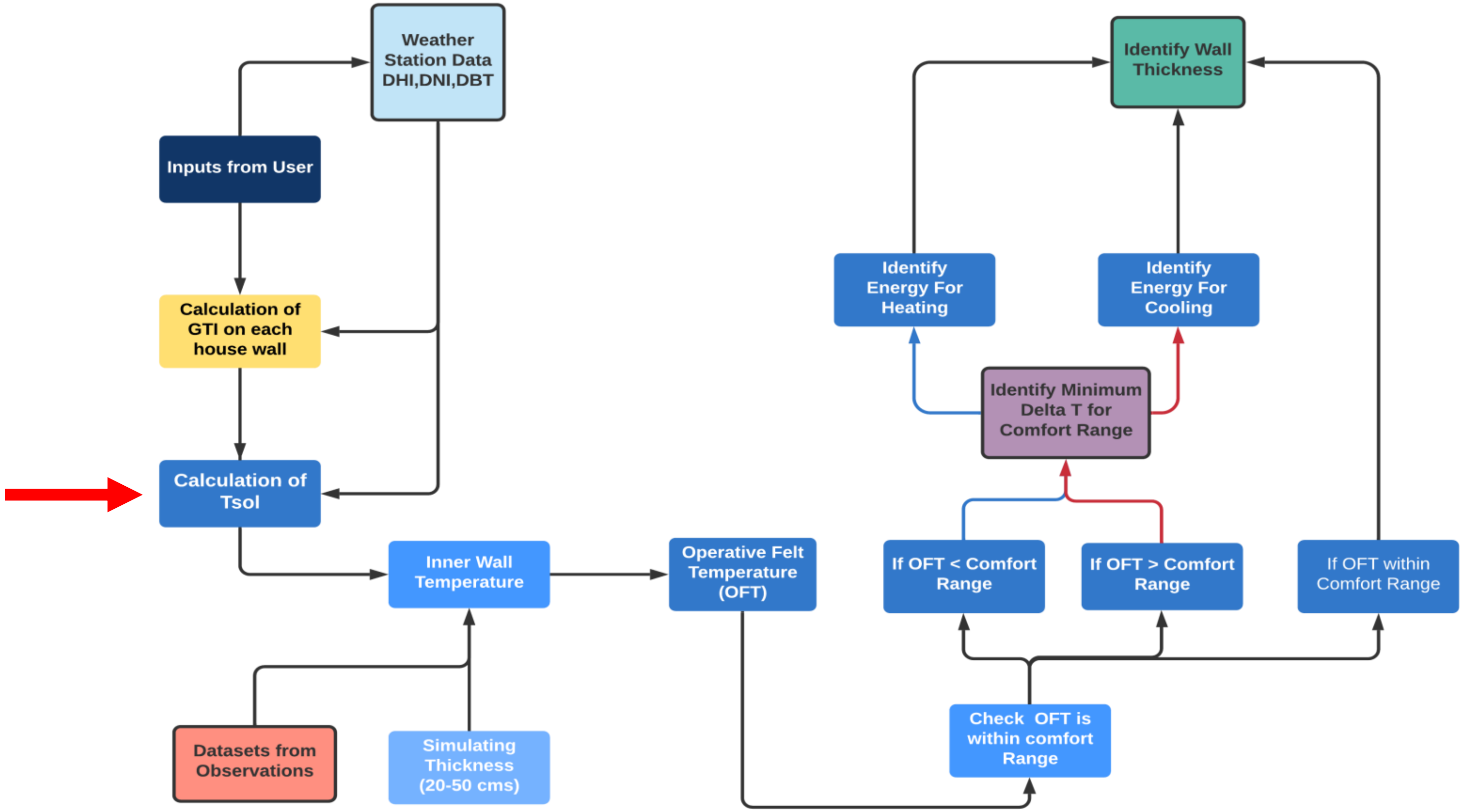


https://www.researchgate.net/figure/Solar-irradiance-components-on-a-tilted-surface_fig1_308613291
Date last visited: 30/07/2020

- Calculation of GTI – all walls of house
- Inclination angle of walls – 90^{O}; Inclination angle of roof – 0^{O}
- Input - Azimuth angle of one wall (Orientation)
- Azimuth angle of wall 2 = 90^{O} + wall1 ….

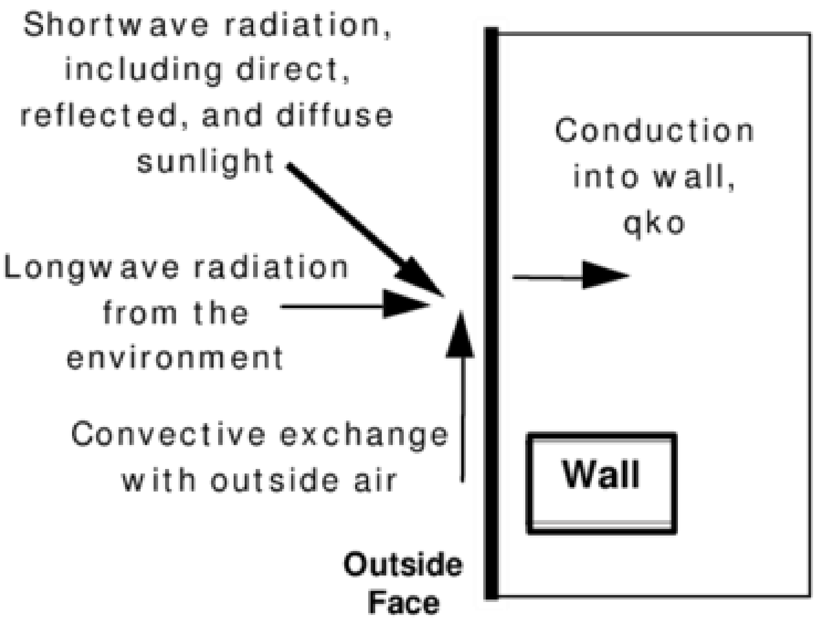
Methodology

Logic Flowchart


Weather Station Data DHI,DNI,DBT
Inputs from User
Calculation of GTI on each house wall
Calculation of Tsol
Inner Wall Temperature
Operative Felt Temperature (OFT)
Datasets from Observations
Simulating Thickness (20-50 cms)
Check OFT is within comfort Range
If OFT < Comfort Range
If OFT > Comfort Range
If OFT within Comfort Range
Identify Minimum Delta T for Comfort Range
Identify Energy For Heating
Identify Energy For Cooling
Identify Wall Thickness

Methodology

Calculation Of Sol-Air Temperature


Shortwave radiation, including direct, reflected, and diffuse sunlight
Longwave radiation from the environment
Convective exchange with outside air
Conduction into wall, qko
Wall
Outside Face

https://bigladdersoftware.com/epx/docs/8-0/engineering-reference/page-020.html
Date last visited: 30/07/2020

- **Tsol = Toutside + R * ($\alpha I_T - \varepsilon * I_1$)**

Toutside – Environment temp (Weather station)

α – absorptivity depends upon the material

ε – emissivity depends upon the material

I_T (in W/m^2) is GTI

I_1 (in W/m^2) is the intensity of long-wave radiation

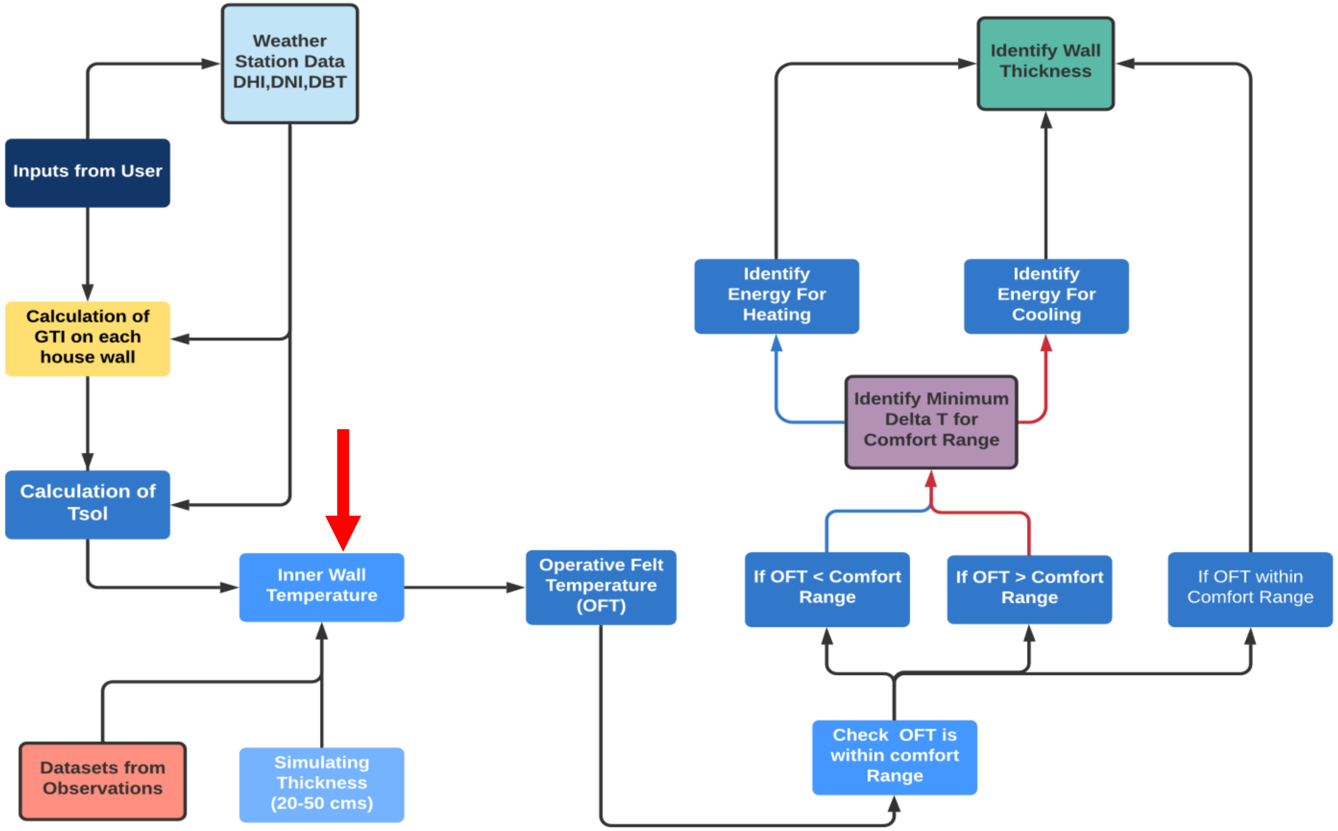
R = 1/h_o (h_o - Convective heat transfer coefficient)

R_wall=0.05 m^2K/W

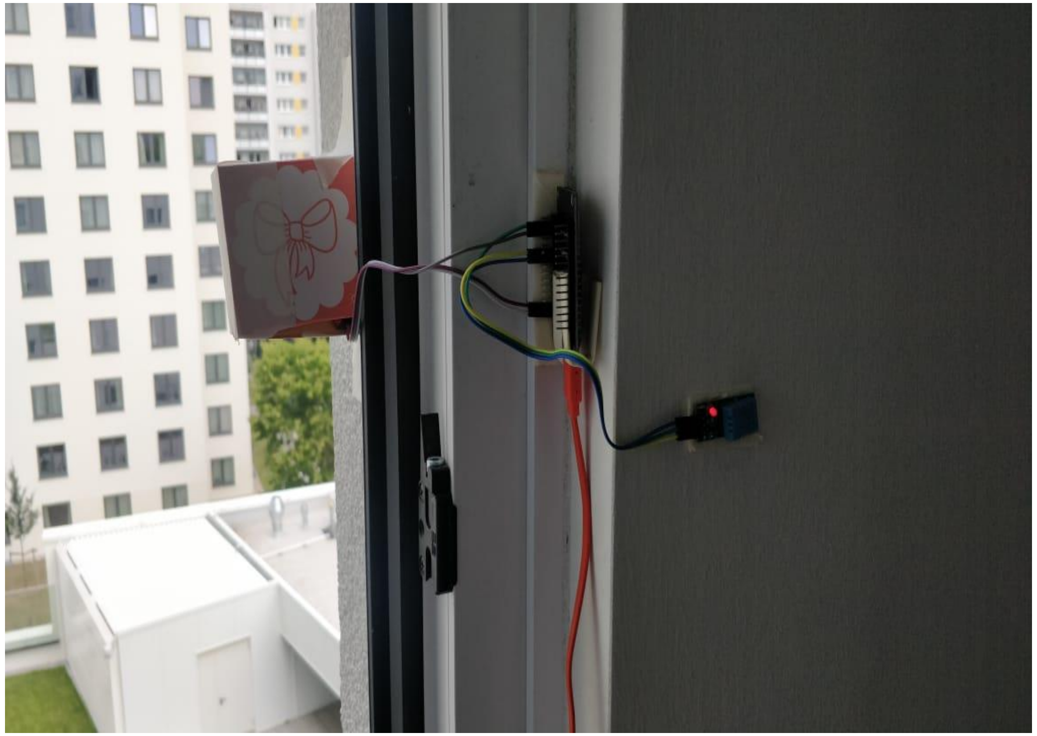
R_roof=0.04 m^2K/W

Methodology

Logic Flowchart


Weather Station Data DHI,DNI,DBT
Inputs from User
Calculation of GTI on each house wall
Calculation of Tsol
Inner Wall Temperature
Datasets from Observations
Simulating Thickness (20-50 cms)
Operative Felt Temperature (OFT)
Check OFT is within comfort Range
If OFT < Comfort Range
If OFT > Comfort Range
If OFT within Comfort Range
Identify Minimum Delta T for Comfort Range
Identify Energy For Heating
Identify Energy For Cooling
Identify Wall Thickness

Calculation of Inner Wall Temperature



- $T_{\text{inside wall}}$=T_{sol} + ((ΔT * wall thickness)/100)

  (No radiation on outside walls e.g. Night/cloudy)
- $T_{\text{inside wall}}$ =Tsol - ((ΔT * wall thickness)/100) (Radiation on outside walls)

For concrete – ΔT is given by

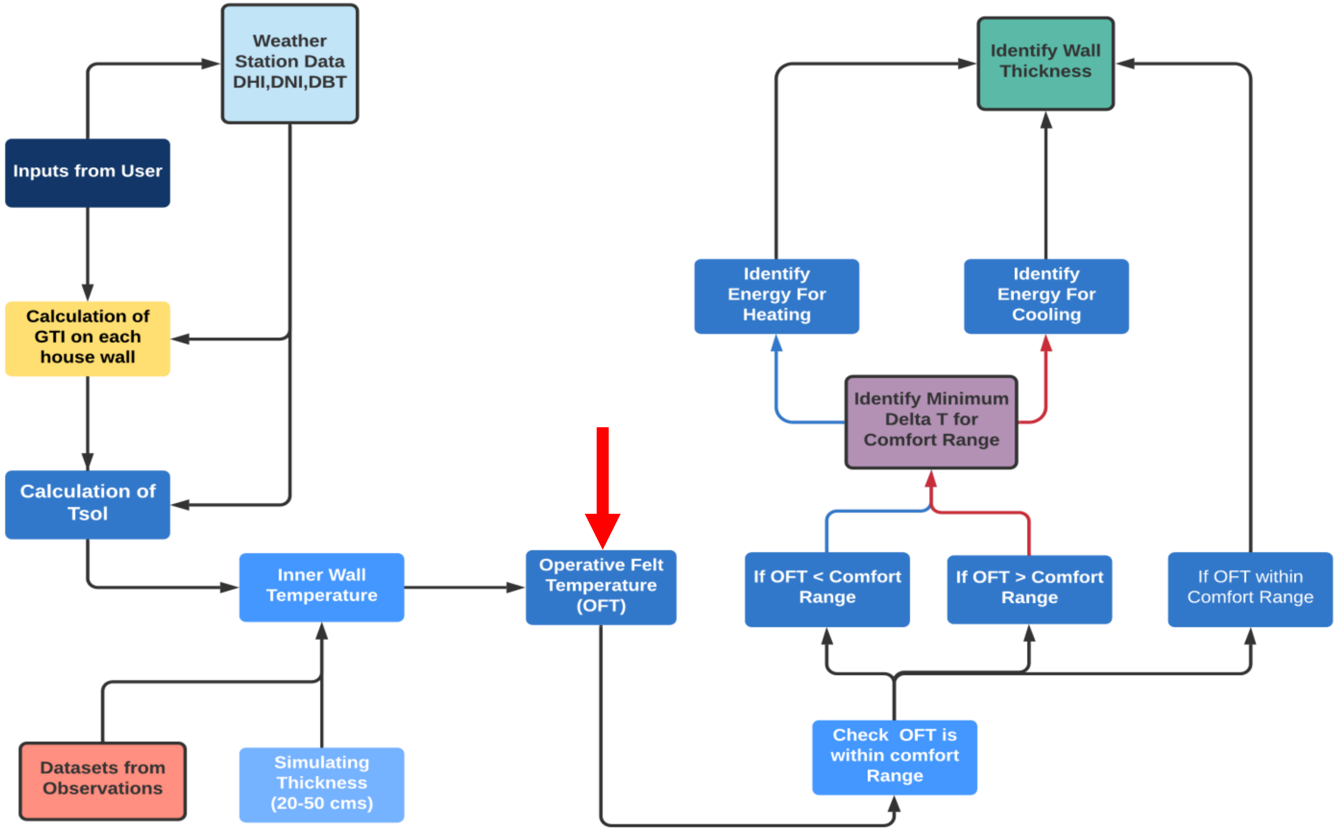
** ΔT (avg)/Wall thickness=1.8826/0.4= 4.7065 (From our experimental data)

For brick – ΔT is given by

** ΔT (avg)/Wall thickness=1.93/0.25= 7.72 (From our experimental data)

Methodology

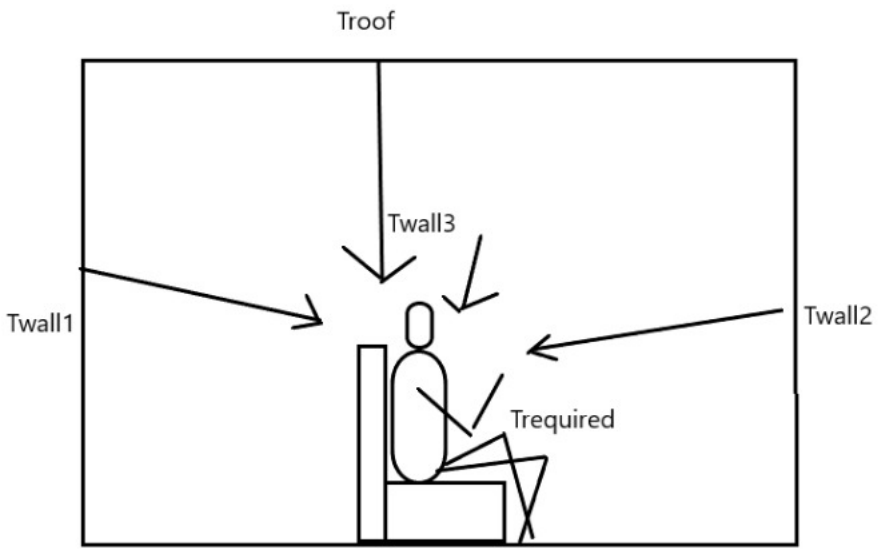
Logic Flowchart



Weather Station Data DHI,DNI,DBT
Inputs from User
Calculation of GTI on each house wall
Calculation of Tsol
Inner Wall Temperature
Datasets from Observations
Simulating Thickness (20-50 cms)
Operative Felt Temperature (OFT)
Check OFT is within comfort Range
If OFT < Comfort Range
If OFT > Comfort Range
If OFT within Comfort Range
Identify Minimum Delta T for Comfort Range
Identify Energy For Heating
Identify Energy For Cooling
Identify Wall Thickness

Methodology

Calculation of Operative Felt Temperature (OFT)

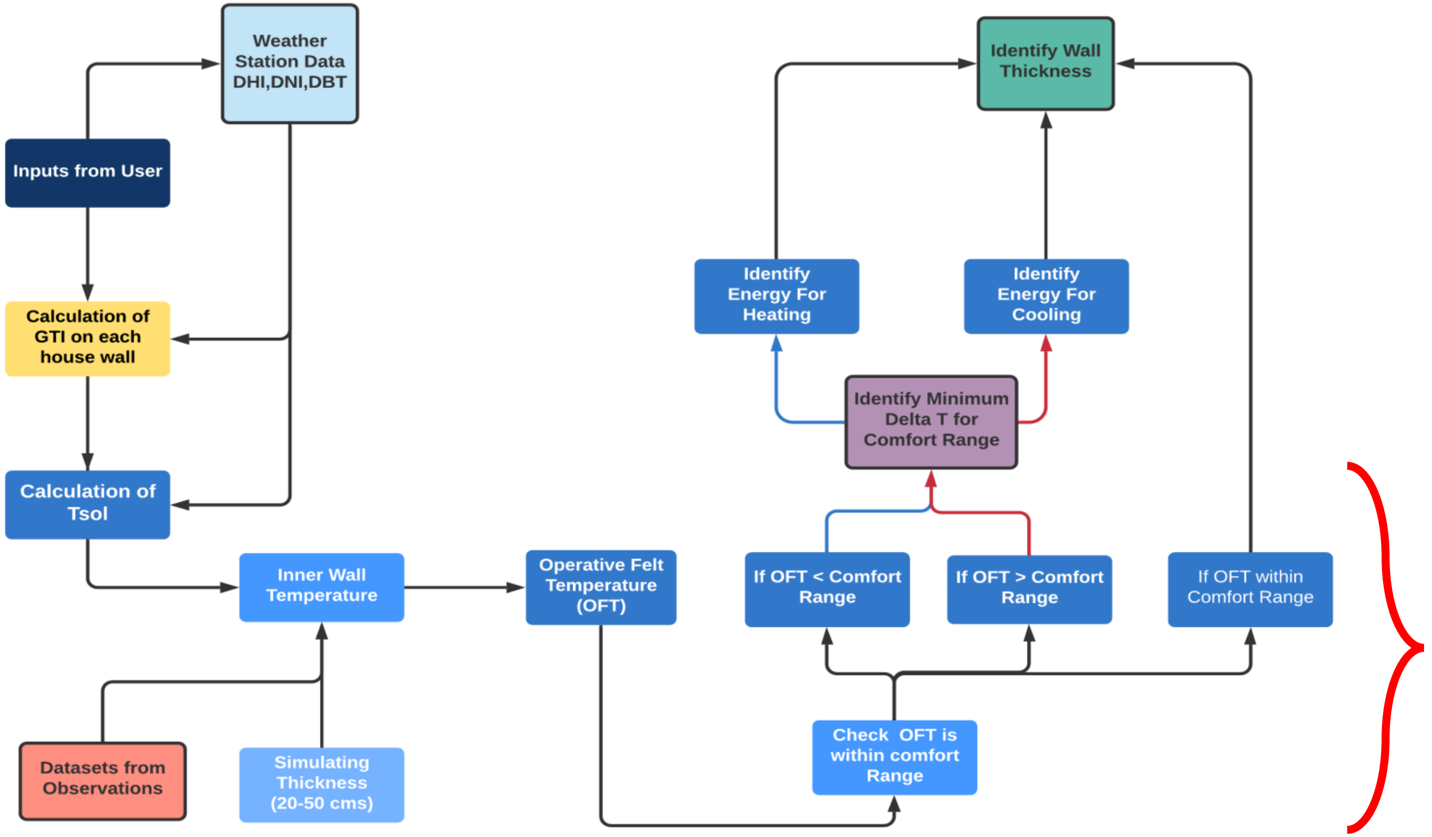


- Operative Felt Temperature – Wall Radiation + Surrounding Air Temperature
- Trequired = [$\phi_{w1 \rightarrow m}$ * T w1 + $\phi_{w2 \rightarrow m}$ * T w2 + $\phi_{w3 \rightarrow m}$ * T w3 + $\phi_{w4 \rightarrow m}$ * T w4 + $\phi_{wroof \rightarrow m}$ * T roof)+dbtair]/2
- Tw – Inner Wall Temperature
- ϕ = view factor

$\phi_{w1 \rightarrow m}$ = (A1/(A1+A2+A3+A4+A5+A6))

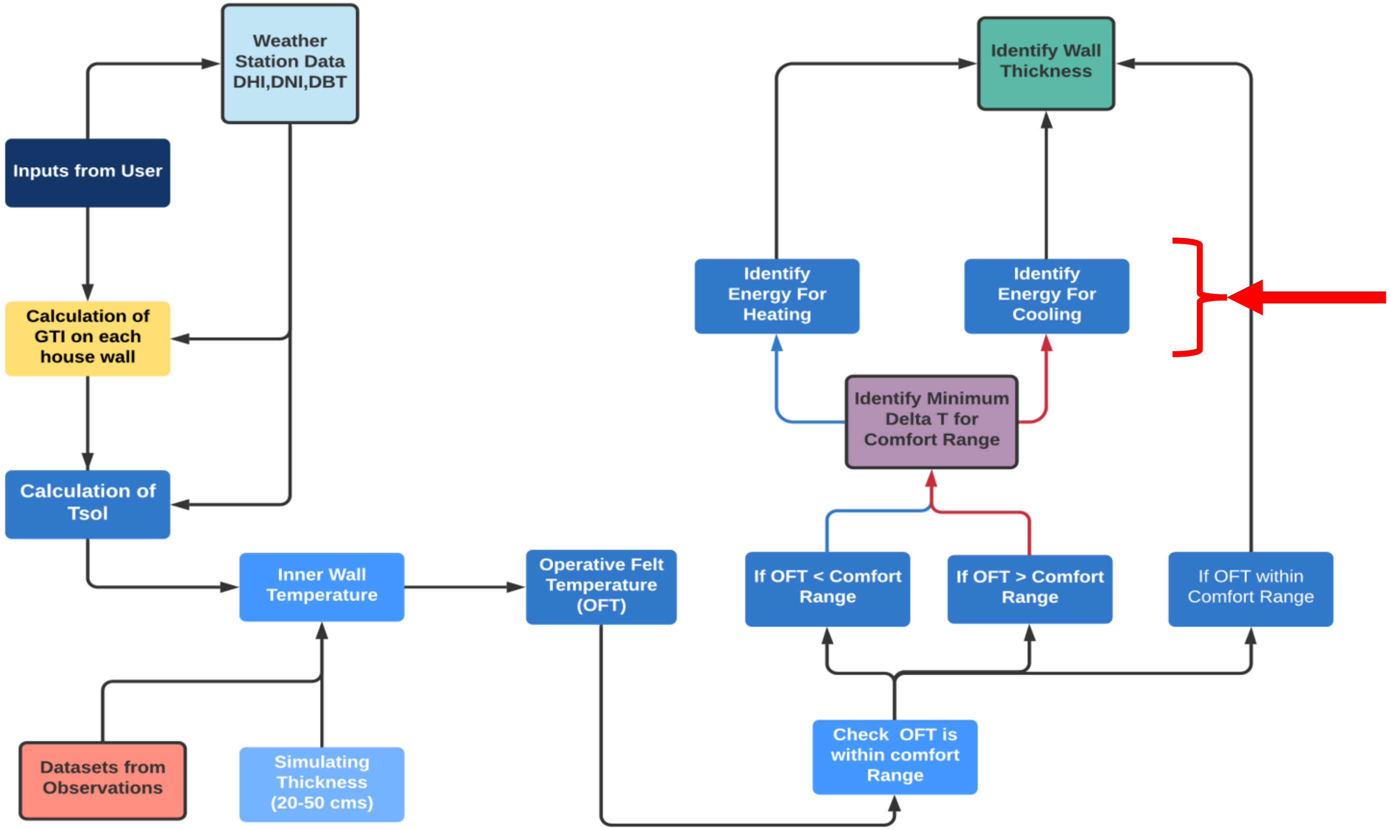
Methodology

Logic Flowchart


Weather Station Data DHI,DNI,DBT
Inputs from User
Calculation of GTI on each house wall
Calculation of Tsol
Inner Wall Temperature
Datasets from Observations
Simulating Thickness (20-50 cms)
Operative Felt Temperature (OFT)
Check OFT is within comfort Range
If OFT < Comfort Range
If OFT > Comfort Range
If OFT within Comfort Range
Identify Minimum Delta T for Comfort Range
Identify Energy For Heating
Identify Energy For Cooling
Identify Wall Thickness

Methodology

Logic Flowchart


Weather Station Data DHI,DNI,DBT
Inputs from User
Calculation of GTI on each house wall
Calculation of Tsol
Inner Wall Temperature
Datasets from Observations
Simulating Thickness (20-50 cms)
Operative Felt Temperature (OFT)
Check OFT is within comfort Range
If OFT < Comfort Range
If OFT > Comfort Range
If OFT within Comfort Range
Identify Minimum Delta T for Comfort Range
Identify Energy For Heating
Identify Energy For Cooling
Identify Wall Thickness

Methodology

Energy Calculation

- Heating Load = $\text{Density}_{\text{air}}$ * $\text{Volume}_{\text{air (room)}}$ * $\text{Cp}_{\text{ air}}$ * (Min Comfort Temp – Actual Room Temp)
- Cooling Load = $\text{Density}_{\text{air}}$ * $\text{Volume}_{\text{air (room)}}$ * $\text{Cp}_{\text{ air}}$ * (Actual Room Temp – Max Comfort Temp)

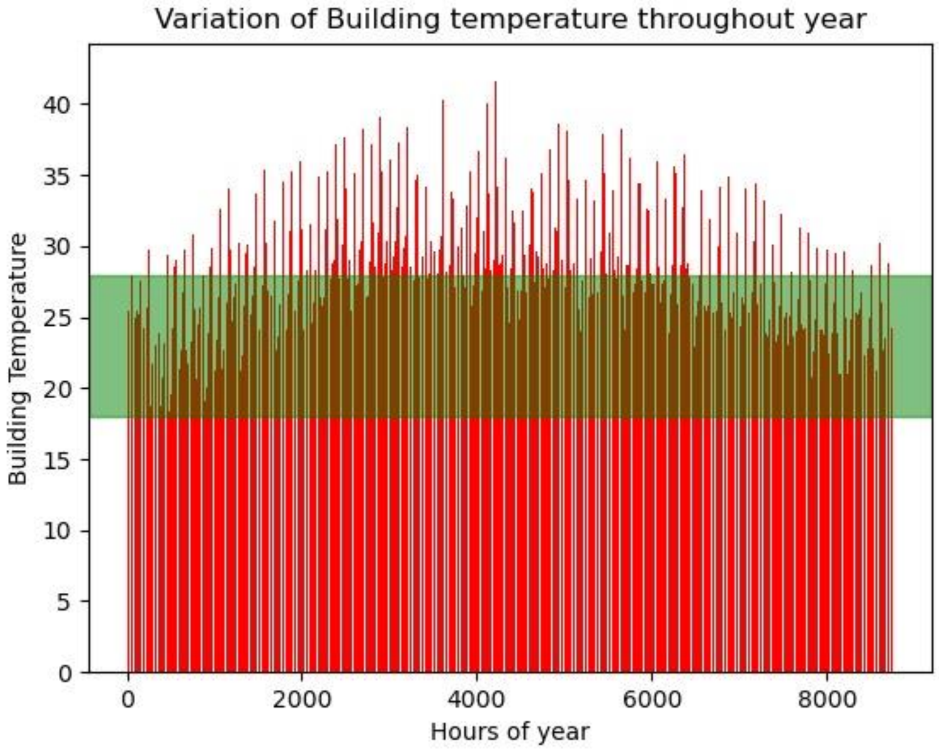
❑ Air density=1.225 [kg/m3]

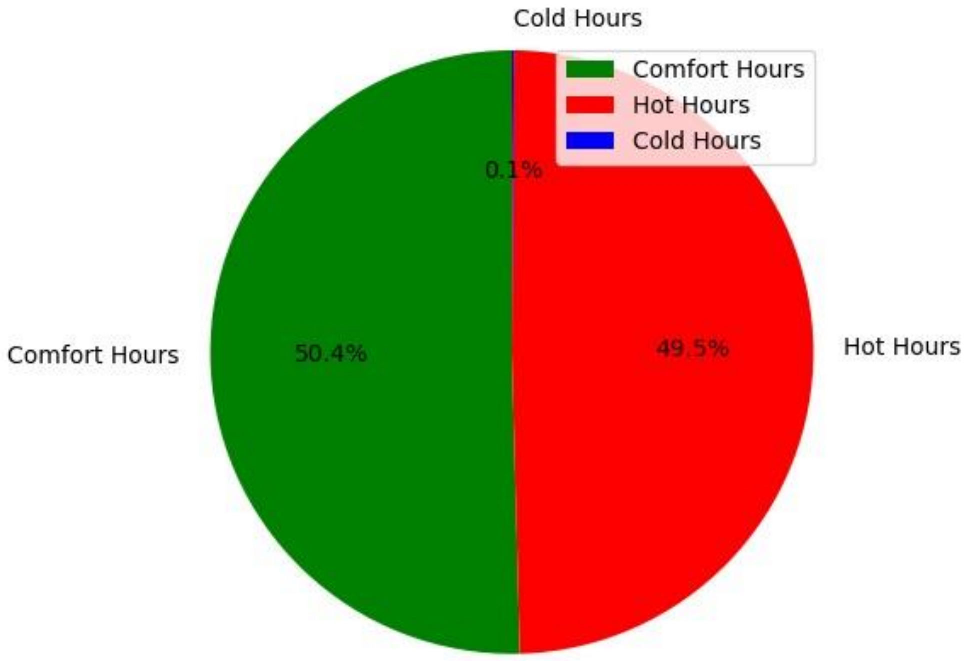
❑ Cp(air)=1 [kJ/kg]

https://econaur.com/energy-conservation-building-code-a-step-to-increase-energy-efficient-buildings/
Date last visited: 30/07/2020

Output

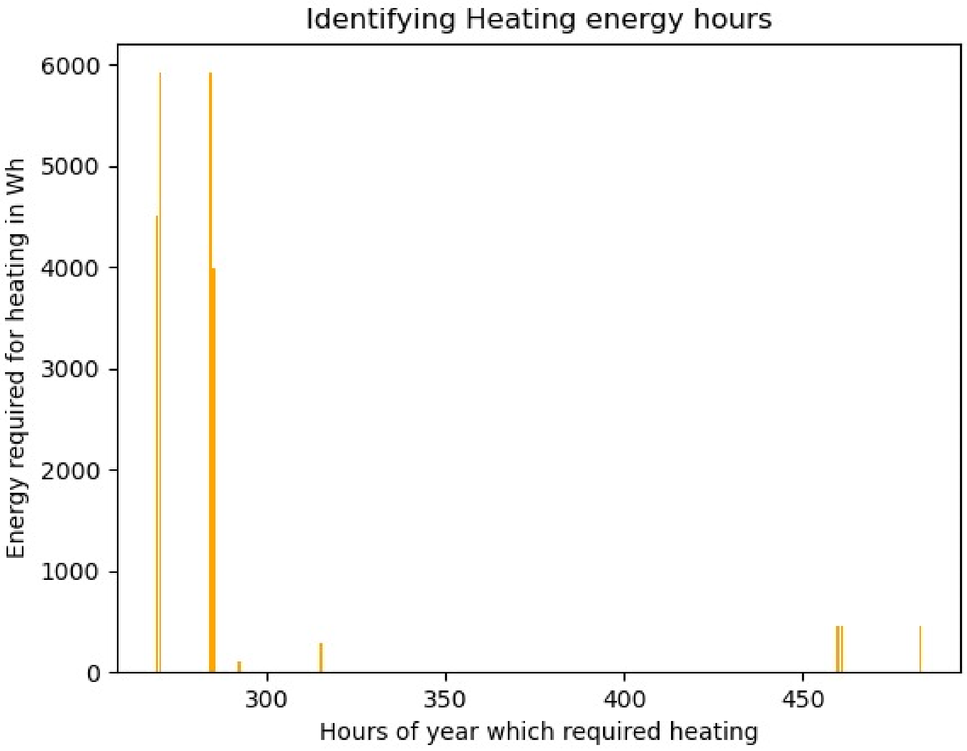
Visual Representation

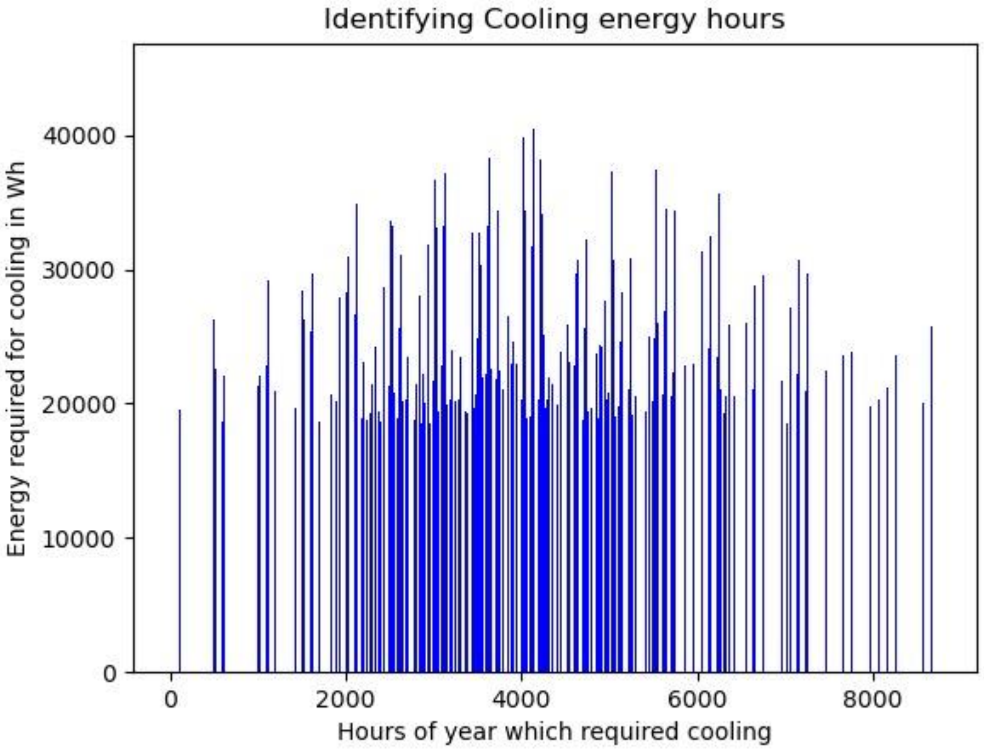




Output

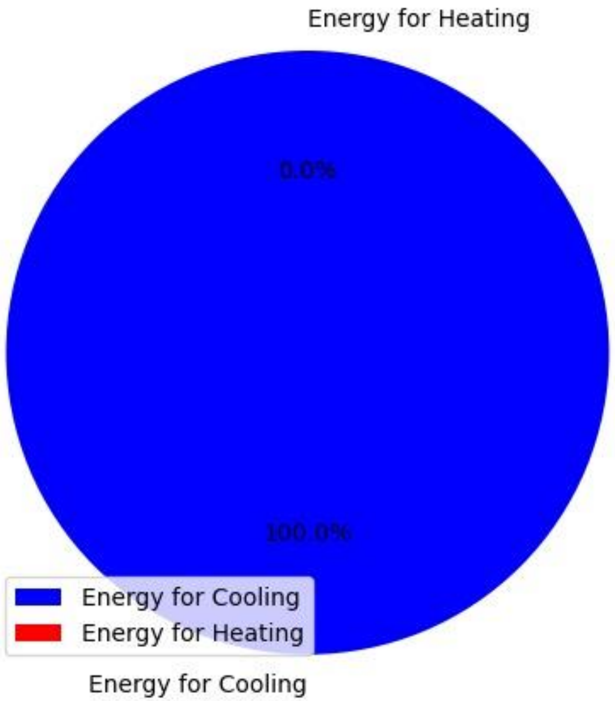
Visual Representation

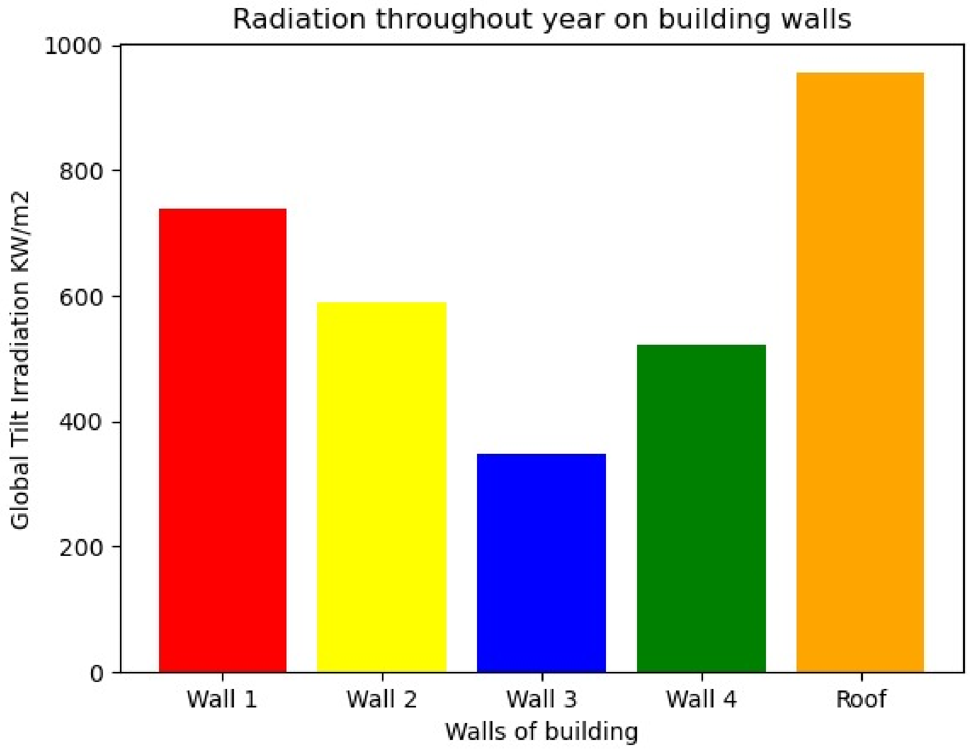




Output

Visual Representation



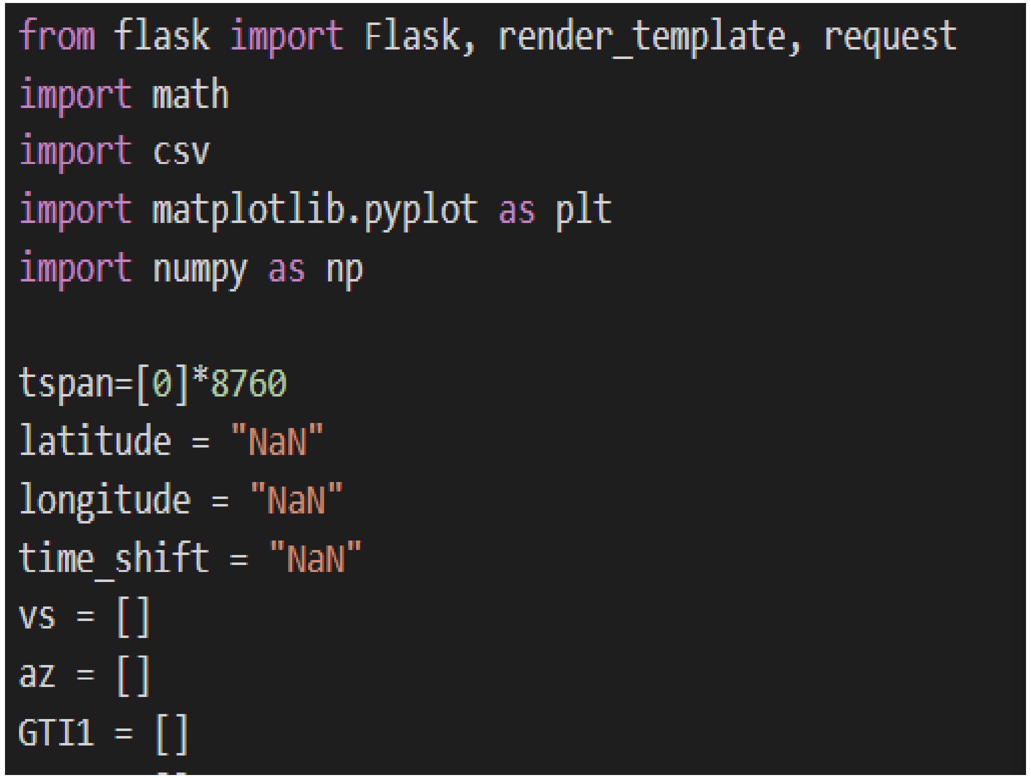


Demo

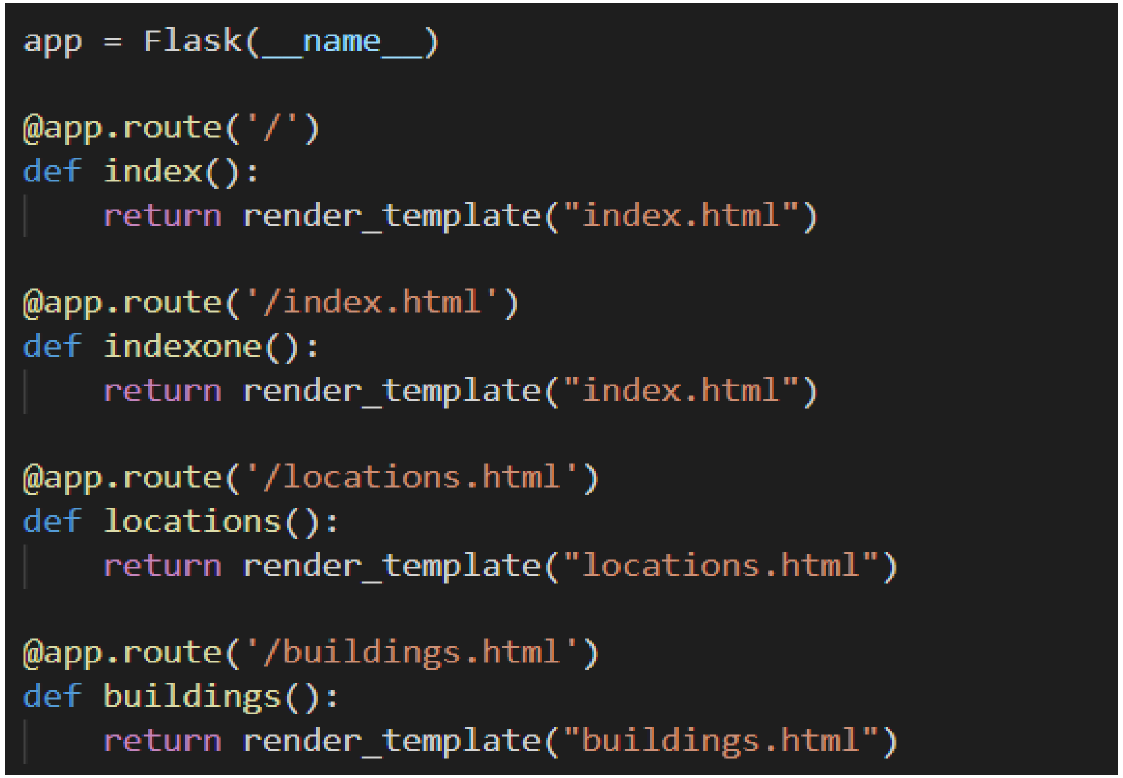
Live Demo!

Python Flask

Code Snippets



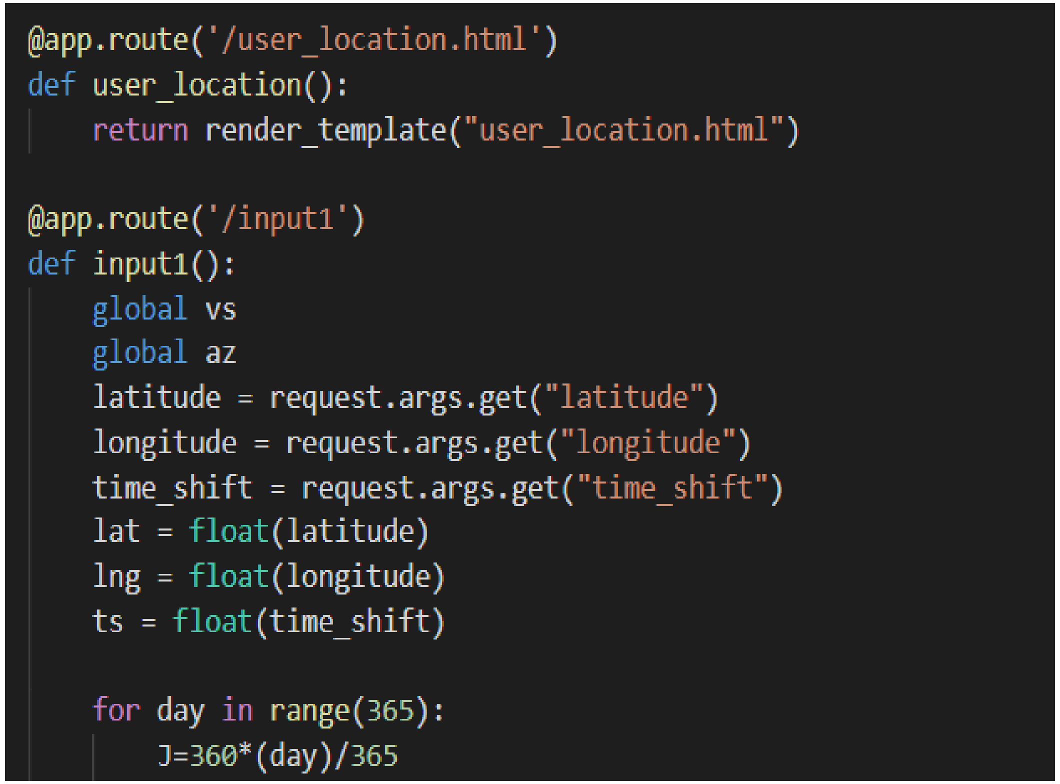
```
from flask import Flask, render_template, request
import math
import csv
import matplotlib.pyplot as plt
import numpy as np

tspan=[0]*8760
latitude = "NaN"
longitude = "NaN"
time_shift = "NaN"
vs = []
az = []
GTI1 = []
```



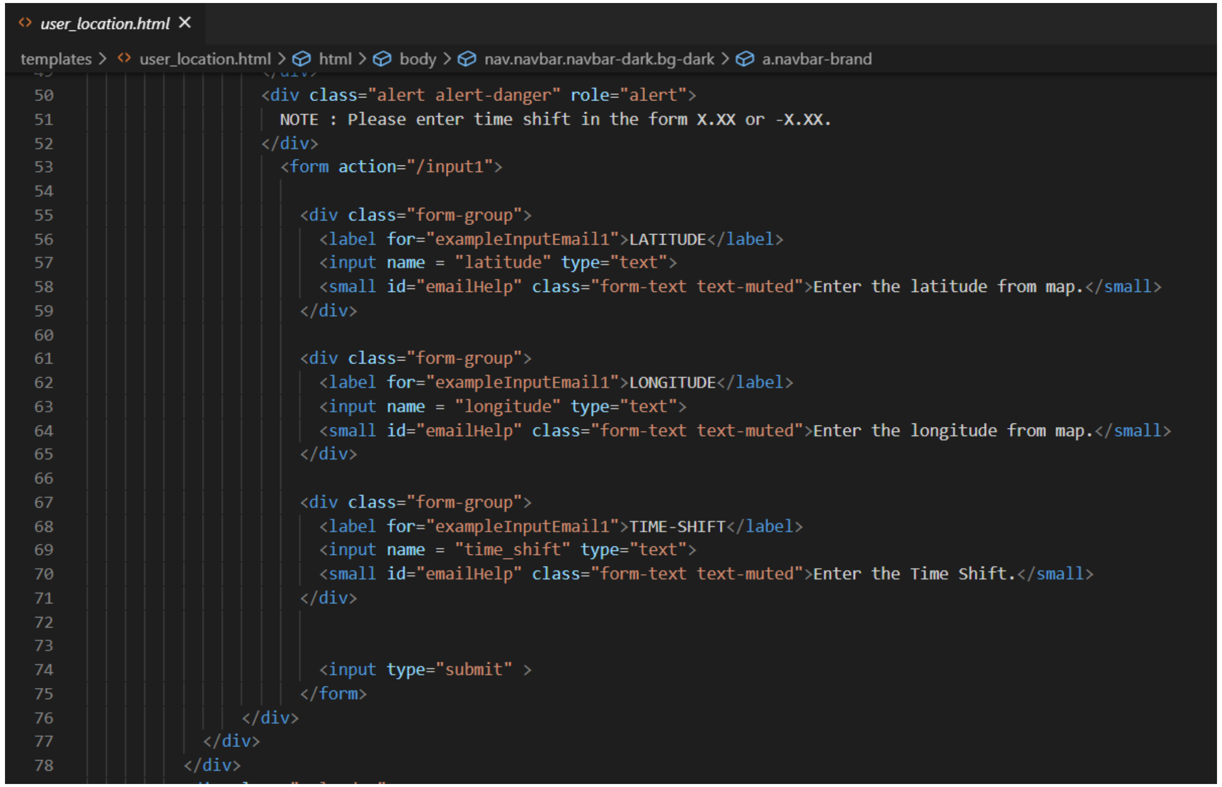
```
app = Flask(__name__)

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

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

@app.route('/locations.html')
def locations():
    return render_template("locations.html")

@app.route('/buildings.html')
def buildings():
    return render_template("buildings.html")
```

Python Flask

Code Snippets



```
@app.route('/user_location.html')
def user_location():
    return render_template("user_location.html")

@app.route('/input1')
def input1():
    global vs
    global az
    latitude = request.args.get("latitude")
    longitude = request.args.get("longitude")
    time_shift = request.args.get("time_shift")
    lat = float(latitude)
    lng = float(longitude)
    ts = float(time_shift)

    for day in range(365):
        J=360*(day)/365
```

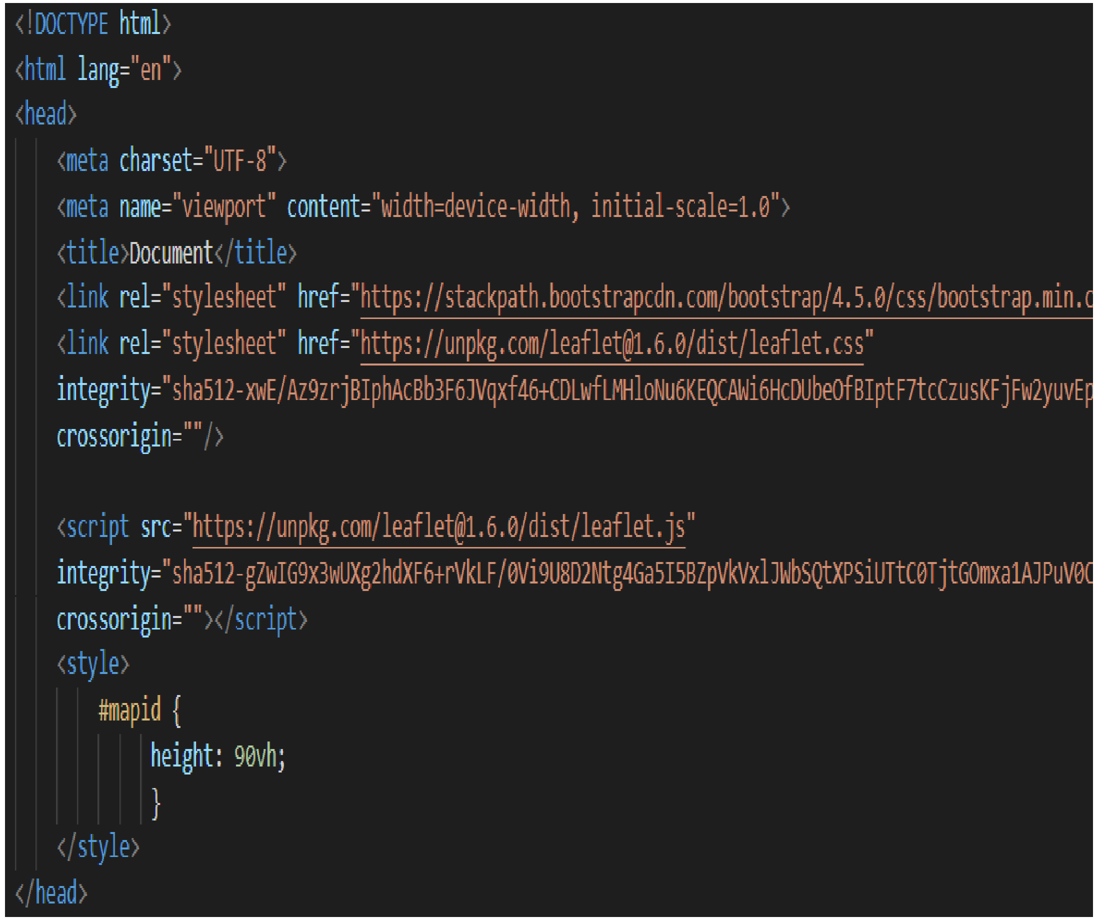


user_location.html ×
templates > user_location.html > html > body > nav.navbar.navbar-dark.bg-dark > a.navbar-brand

```
    <div class="alert alert-danger" role="alert">
     NOTE : Please enter time shift in the form X.XX or -X.XX.
    </div>
     <form action="/input1">

      <div class="form-group">
       <label for="exampleInputEmail1">LATITUDE</label>
       <input name = "latitude" type="text">
       <small id="emailHelp" class="form-text text-muted">Enter the latitude from map.</small>
      </div>

      <div class="form-group">
       <label for="exampleInputEmail1">LONGITUDE</label>
       <input name = "longitude" type="text">
       <small id="emailHelp" class="form-text text-muted">Enter the longitude from map.</small>
      </div>

      <div class="form-group">
       <label for="exampleInputEmail1">TIME-SHIFT</label>
       <input name = "time_shift" type="text">
       <small id="emailHelp" class="form-text text-muted">Enter the Time Shift.</small>
      </div>


       <input type="submit" >
      </form>
   </div>
 </div>
</div>
```

CSS / Javascript

Code Snippets



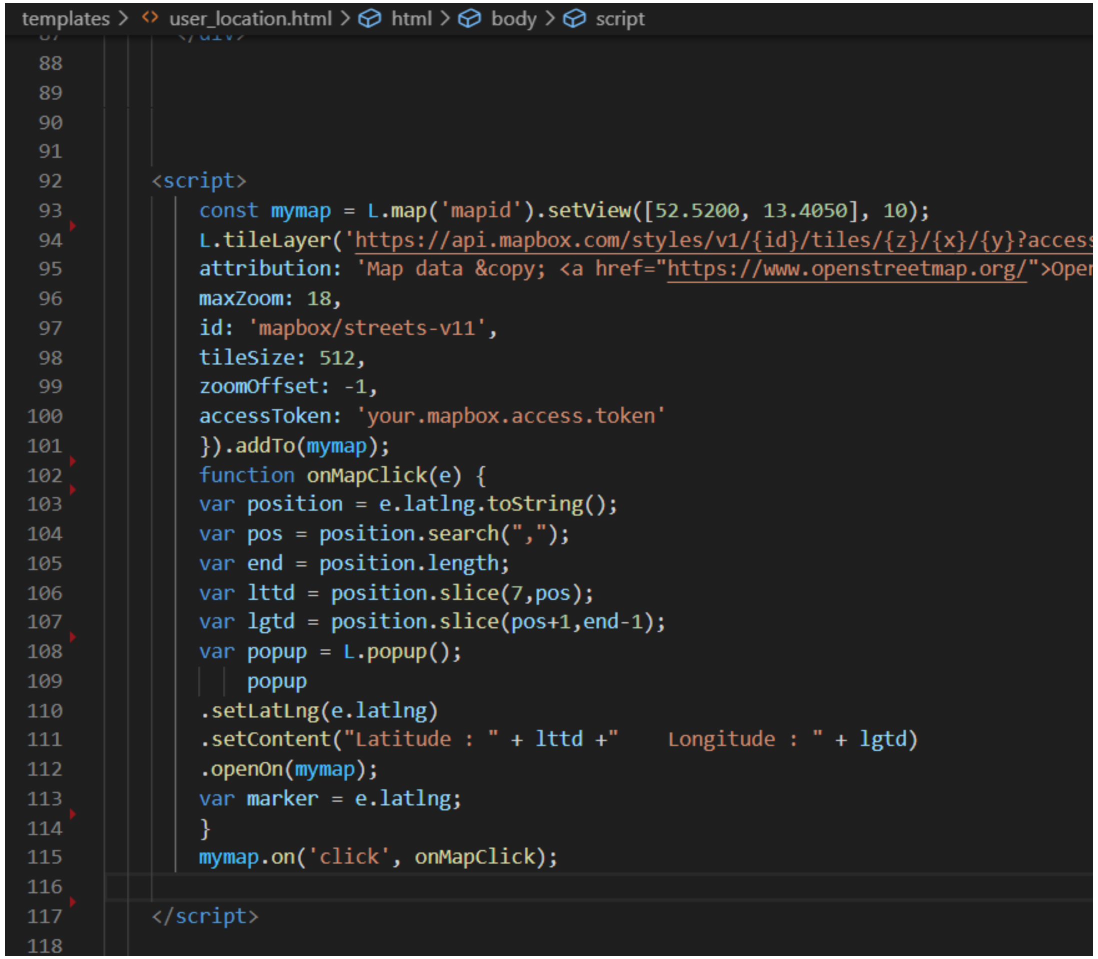
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.c
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css"
    integrity="sha512-xwE/Az9zrjBIphAcBb3F6JVqxf46+CDLwfLMHloNu6KEQCAWi6HcDUbeOfBIptF7tcCzusKFjFw2yuvEp
    crossorigin=""/>

    <script src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"
    integrity="sha512-gZwIG9x3wUXg2hdXF6+rVkLF/0Vi9U8D2Ntg4Ga5I5BZpVkVxlJWbSQtXPSiUTtC0TjtGOmxa1AJPuV0C
    crossorigin=""></script>
    <style>
        #mapid {
            height: 90vh;
            }
    </style>
</head>
```

- Leafletjs : open source JS library for interactive maps.

- Bootstrap

Javascript

Code Snippets

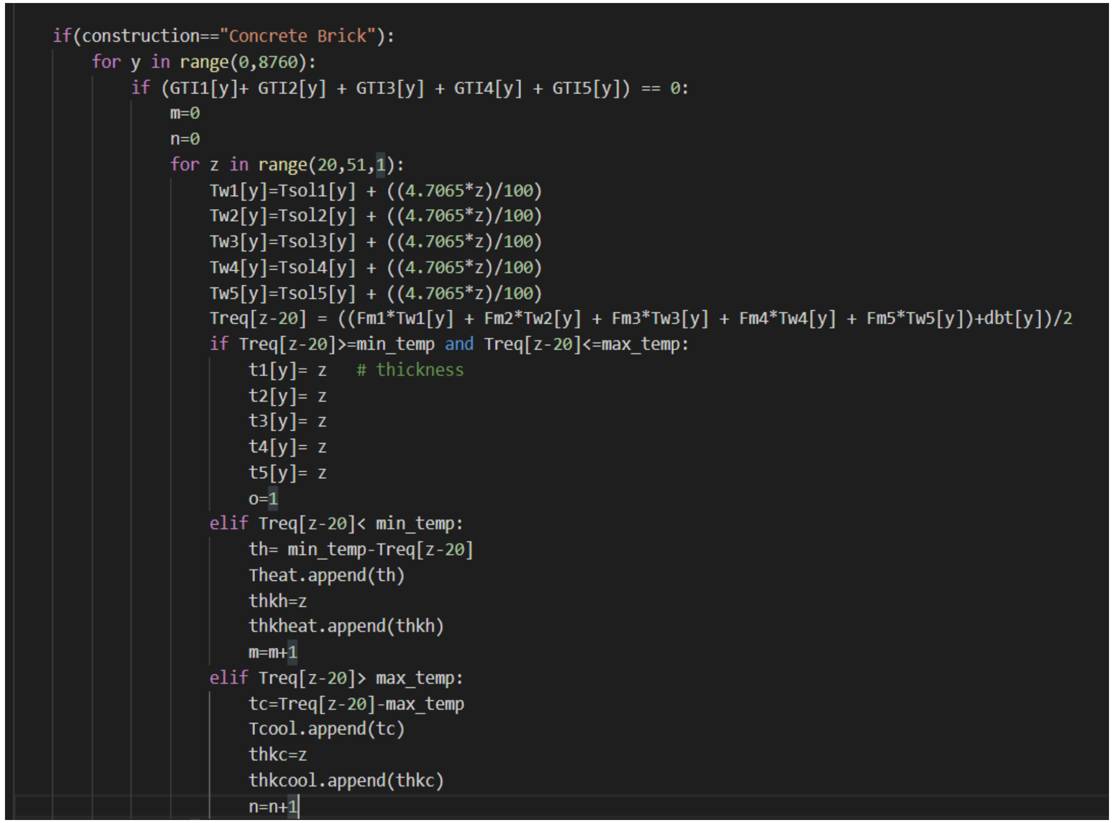


```
<script>
  const mymap = L.map('mapid').setView([52.5200, 13.4050], 10);
  L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access
  attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">Oper
  maxZoom: 18,
  id: 'mapbox/streets-v11',
  tileSize: 512,
  zoomOffset: -1,
  accessToken: 'your.mapbox.access.token'
  }).addTo(mymap);
  function onMapClick(e) {
  var position = e.latlng.toString();
  var pos = position.search(",");
  var end = position.length;
  var lttd = position.slice(7,pos);
  var lgtd = position.slice(pos+1,end-1);
  var popup = L.popup();
      popup
  .setLatLng(e.latlng)
  .setContent("Latitude : " + lttd +"    Longitude : " + lgtd)
  .openOn(mymap);
  var marker = e.latlng;
  }
  mymap.on('click', onMapClick);

</script>
```

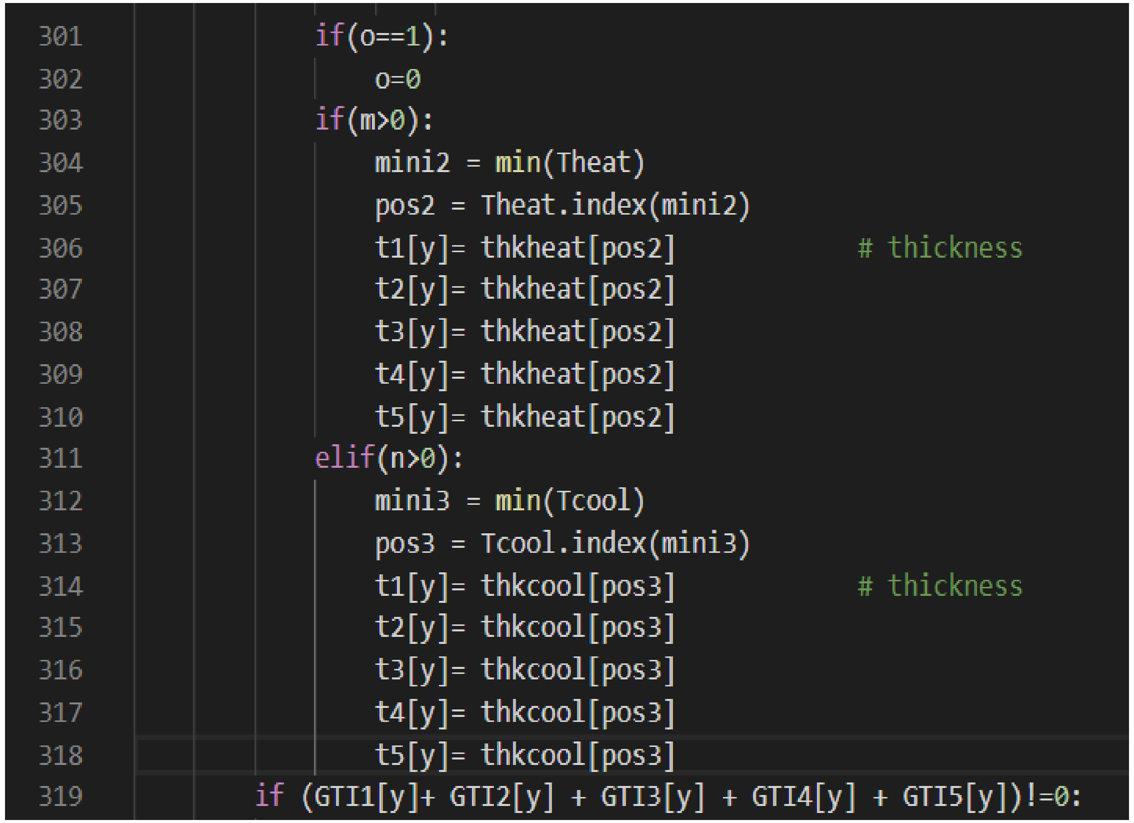
- Python at the base
- HTML at the forefront
- Javascript simply makes it “AWESOME”

Python

Code Snippets



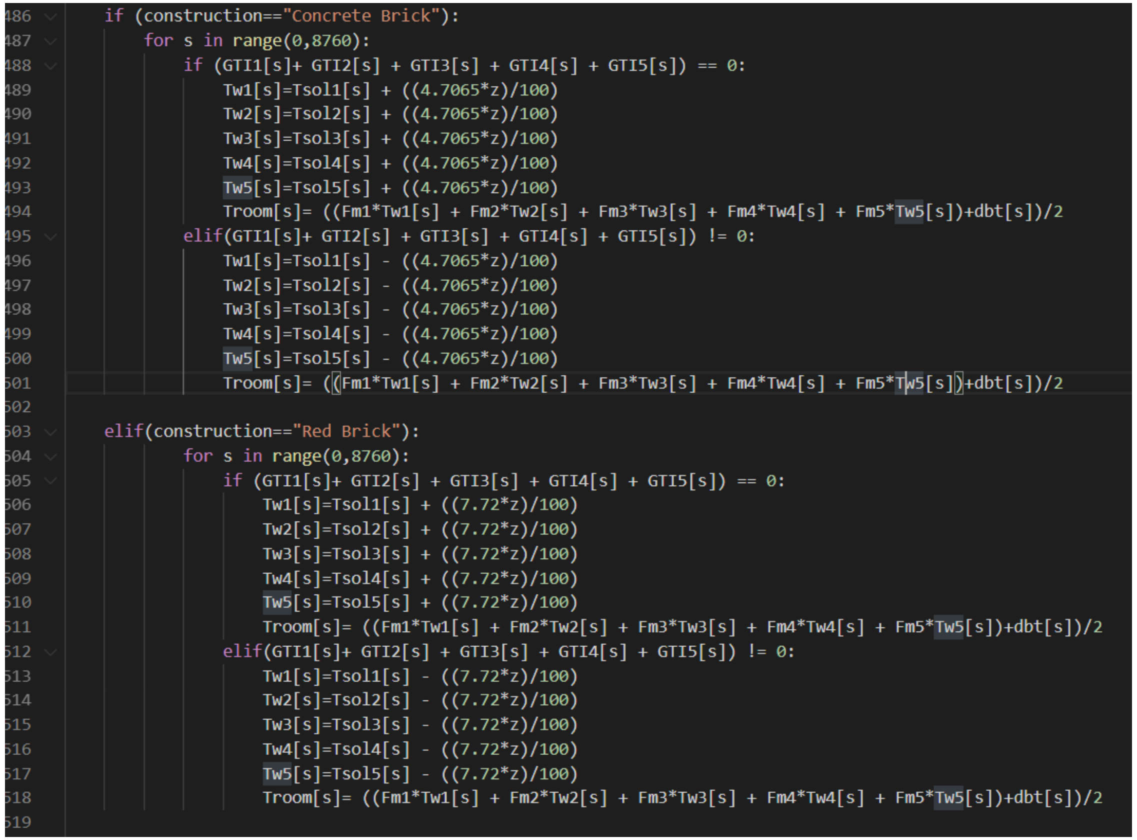
```python
    if(construction=="Concrete Brick"):
        for y in range(0,8760):
            if (GTI1[y]+ GTI2[y] + GTI3[y] + GTI4[y] + GTI5[y]) == 0:
                m=0
                n=0
                for z in range(20,51,1):
                    Tw1[y]=Tsol1[y] + ((4.7065*z)/100)
                    Tw2[y]=Tsol2[y] + ((4.7065*z)/100)
                    Tw3[y]=Tsol3[y] + ((4.7065*z)/100)
                    Tw4[y]=Tsol4[y] + ((4.7065*z)/100)
                    Tw5[y]=Tsol5[y] + ((4.7065*z)/100)
                    Treq[z-20] = ((Fm1*Tw1[y] + Fm2*Tw2[y] + Fm3*Tw3[y] + Fm4*Tw4[y] + Fm5*Tw5[y])+dbt[y])/2
                    if Treq[z-20]>=min_temp and Treq[z-20]<=max_temp:
                        t1[y]= z   # thickness
                        t2[y]= z
                        t3[y]= z
                        t4[y]= z
                        t5[y]= z
                        o=1
                    elif Treq[z-20]< min_temp:
                        th= min_temp-Treq[z-20]
                        Theat.append(th)
                        thkh=z
                        thkheat.append(thkh)
                        m=m+1
                    elif Treq[z-20]> max_temp:
                        tc=Treq[z-20]-max_temp
                        Tcool.append(tc)
                        thkc=z
                        thkcool.append(thkc)
                        n=n+1
```



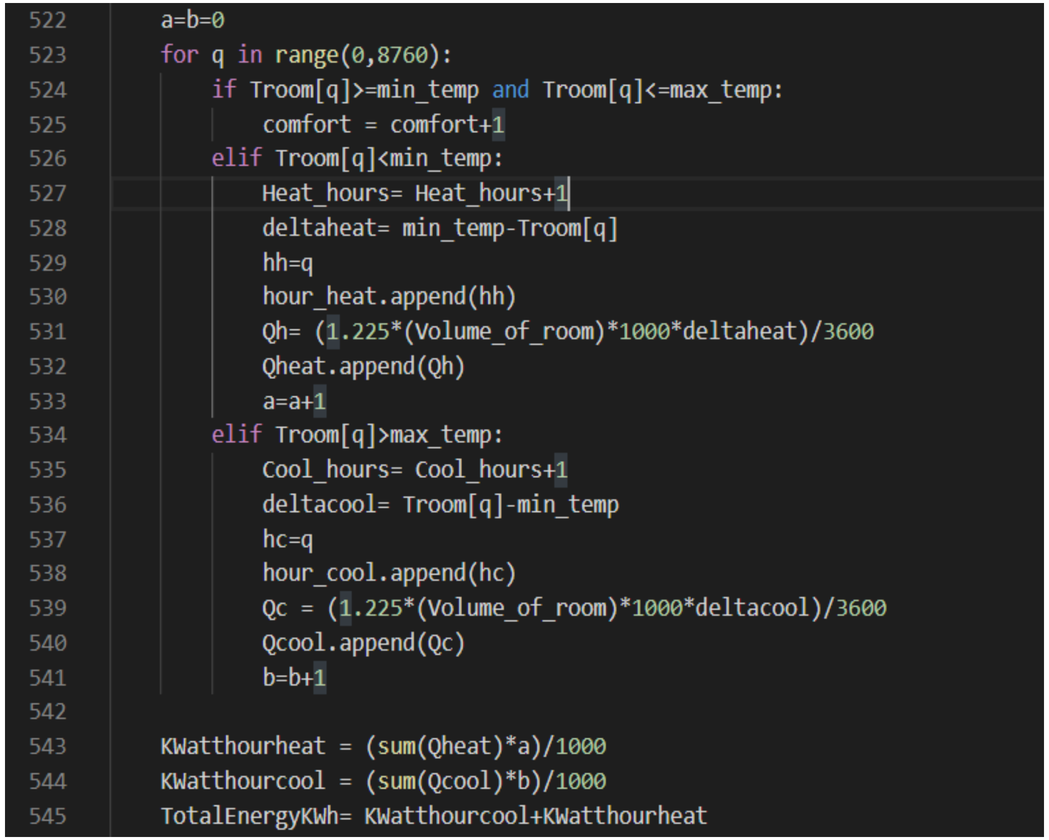
```python
            if(o==1):
                o=0
            if(m>0):
                mini2 = min(Theat)
                pos2 = Theat.index(mini2)
                t1[y]= thkheat[pos2]              # thickness
                t2[y]= thkheat[pos2]
                t3[y]= thkheat[pos2]
                t4[y]= thkheat[pos2]
                t5[y]= thkheat[pos2]
            elif(n>0):
                mini3 = min(Tcool)
                pos3 = Tcool.index(mini3)
                t1[y]= thkcool[pos3]              # thickness
                t2[y]= thkcool[pos3]
                t3[y]= thkcool[pos3]
                t4[y]= thkcool[pos3]
                t5[y]= thkcool[pos3]
        if (GTI1[y]+ GTI2[y] + GTI3[y] + GTI4[y] + GTI5[y])!=0:
```

Python

Code Snippets



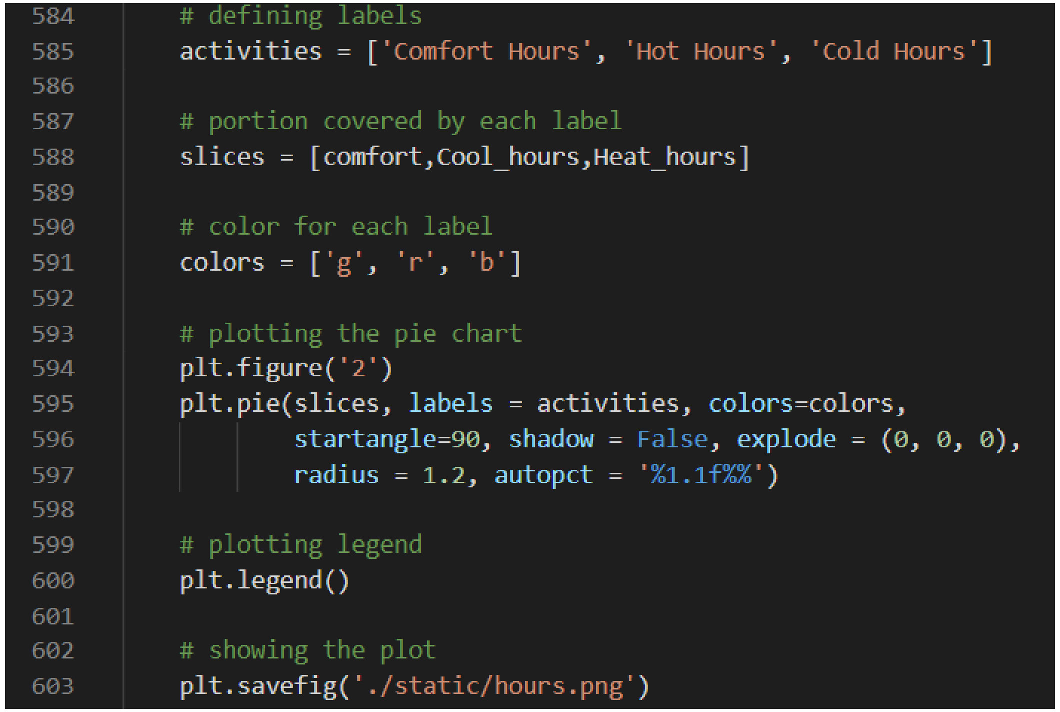
```
    if (construction=="Concrete Brick"):
        for s in range(0,8760):
            if (GTI1[s]+ GTI2[s] + GTI3[s] + GTI4[s] + GTI5[s]) == 0:
                Tw1[s]=Tsol1[s] + ((4.7065*z)/100)
                Tw2[s]=Tsol2[s] + ((4.7065*z)/100)
                Tw3[s]=Tsol3[s] + ((4.7065*z)/100)
                Tw4[s]=Tsol4[s] + ((4.7065*z)/100)
                Tw5[s]=Tsol5[s] + ((4.7065*z)/100)
                Troom[s]= ((Fm1*Tw1[s] + Fm2*Tw2[s] + Fm3*Tw3[s] + Fm4*Tw4[s] + Fm5*Tw5[s])+dbt[s])/2
            elif(GTI1[s]+ GTI2[s] + GTI3[s] + GTI4[s] + GTI5[s]) != 0:
                Tw1[s]=Tsol1[s] - ((4.7065*z)/100)
                Tw2[s]=Tsol2[s] - ((4.7065*z)/100)
                Tw3[s]=Tsol3[s] - ((4.7065*z)/100)
                Tw4[s]=Tsol4[s] - ((4.7065*z)/100)
                Tw5[s]=Tsol5[s] - ((4.7065*z)/100)
                Troom[s]= ((Fm1*Tw1[s] + Fm2*Tw2[s] + Fm3*Tw3[s] + Fm4*Tw4[s] + Fm5*Tw5[s])+dbt[s])/2

    elif(construction=="Red Brick"):
            for s in range(0,8760):
                if (GTI1[s]+ GTI2[s] + GTI3[s] + GTI4[s] + GTI5[s]) == 0:
                    Tw1[s]=Tsol1[s] + ((7.72*z)/100)
                    Tw2[s]=Tsol2[s] + ((7.72*z)/100)
                    Tw3[s]=Tsol3[s] + ((7.72*z)/100)
                    Tw4[s]=Tsol4[s] + ((7.72*z)/100)
                    Tw5[s]=Tsol5[s] + ((7.72*z)/100)
                    Troom[s]= ((Fm1*Tw1[s] + Fm2*Tw2[s] + Fm3*Tw3[s] + Fm4*Tw4[s] + Fm5*Tw5[s])+dbt[s])/2
                elif(GTI1[s]+ GTI2[s] + GTI3[s] + GTI4[s] + GTI5[s]) != 0:
                    Tw1[s]=Tsol1[s] - ((7.72*z)/100)
                    Tw2[s]=Tsol2[s] - ((7.72*z)/100)
                    Tw3[s]=Tsol3[s] - ((7.72*z)/100)
                    Tw4[s]=Tsol4[s] - ((7.72*z)/100)
                    Tw5[s]=Tsol5[s] - ((7.72*z)/100)
                    Troom[s]= ((Fm1*Tw1[s] + Fm2*Tw2[s] + Fm3*Tw3[s] + Fm4*Tw4[s] + Fm5*Tw5[s])+dbt[s])/2

```



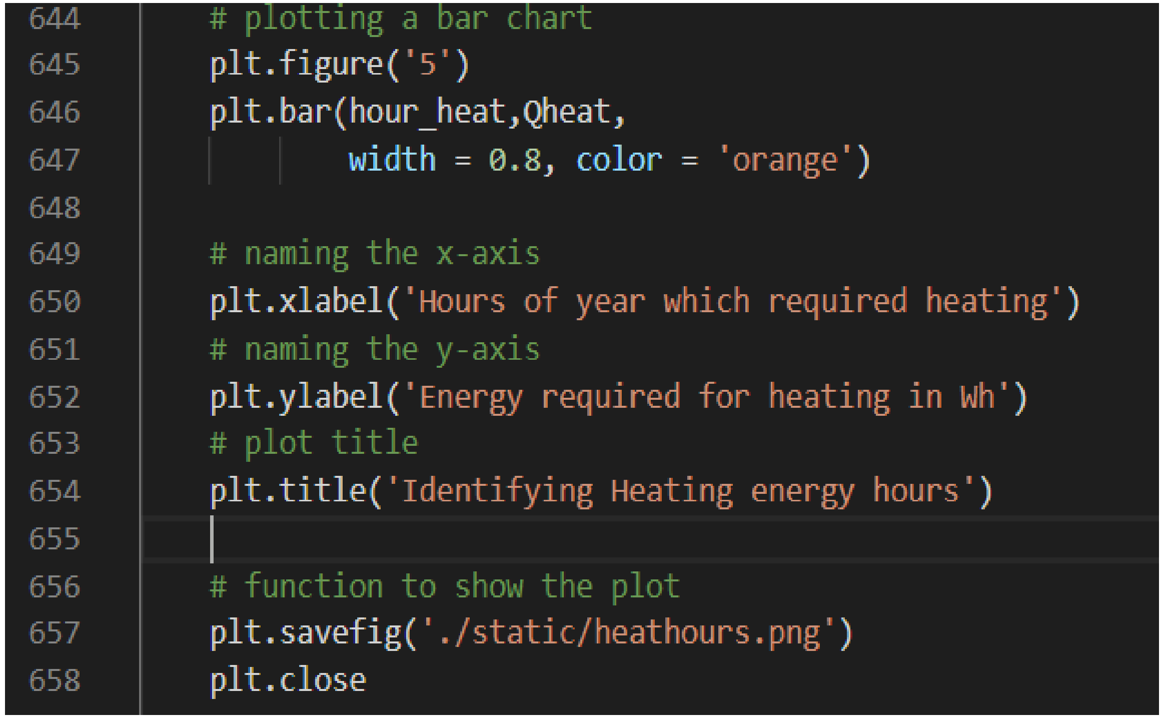
```
    a=b=0
    for q in range(0,8760):
        if Troom[q]>=min_temp and Troom[q]<=max_temp:
            comfort = comfort+1
        elif Troom[q]<min_temp:
            Heat_hours= Heat_hours+1
            deltaheat= min_temp-Troom[q]
            hh=q
            hour_heat.append(hh)
            Qh= (1.225*(Volume_of_room)*1000*deltaheat)/3600
            Qheat.append(Qh)
            a=a+1
        elif Troom[q]>max_temp:
            Cool_hours= Cool_hours+1
            deltacool= Troom[q]-min_temp
            hc=q
            hour_cool.append(hc)
            Qc = (1.225*(Volume_of_room)*1000*deltacool)/3600
            Qcool.append(Qc)
            b=b+1

    KWatthourheat = (sum(Qheat)*a)/1000
    KWatthourcool = (sum(Qcool)*b)/1000
    TotalEnergyKWh= KWatthourcool+KWatthourheat
```

Python

Code Snippets



```python
    # defining labels
    activities = ['Comfort Hours', 'Hot Hours', 'Cold Hours']

    # portion covered by each label
    slices = [comfort,Cool_hours,Heat_hours]

    # color for each label
    colors = ['g', 'r', 'b']

    # plotting the pie chart
    plt.figure('2')
    plt.pie(slices, labels = activities, colors=colors,
            startangle=90, shadow = False, explode = (0, 0, 0),
            radius = 1.2, autopct = '%1.1f%%')

    # plotting legend
    plt.legend()

    # showing the plot
    plt.savefig('./static/hours.png')
```



```python
    # plotting a bar chart
    plt.figure('5')
    plt.bar(hour_heat,Qheat,
            width = 0.8, color = 'orange')

    # naming the x-axis
    plt.xlabel('Hours of year which required heating')
    # naming the y-axis
    plt.ylabel('Energy required for heating in Wh')
    # plot title
    plt.title('Identifying Heating energy hours')

    # function to show the plot
    plt.savefig('./static/heathours.png')
    plt.close
```

Summary – Interpretation of results

- Analysis of Thermal Comfort
- Freedom in construction material choice
- Provide basis – Financial Estimation of Energy Costs
- Financial Estimation of Building Costs
- Better Orientation

Conclusion

Future Potential Development

- Better Precision Sensors
- Collection of data for longer time – for accuracy
- More time – Improve UI
- Funds – To purchase recent weather data for any location on globe
- Better platform - Reduced latency

Conclusion

Future Potential Development – Tackling Assumptions

- No humidity (only thermal comfort considered)
- No person inside the room
- The floor is thermally insulated
- Constant convection coefficient (wind velocity neglected)
- No shading effects
- Paint effect is neglected
- Minimum allowed thickness=20cm, maximum allowed thickness=50cm

References

- *IVHE GuideBook A* – Institution of Ventilating and heating Engineers, London,1970
- F Kreith, *Principles of Heat Transfer*, 3rd Heat transfer,Feffer and Simons,London,1974
- P W O’Callagahan,S.D. Probert ,*Sol Air Temperature, School of Mechanical Engineering, Cranfield University*
- *ASHRAE*
- Mackey and Wright*, Sol air Temperature(1946)*
- https://www.w3schools.com/python/default.asp
- https://www.w3schools.com/html/default.asp
- IDP lecture slides

Summary – IDP

- Zoom – Long virtual Meetings
- GitHub – Sharing codes
- IOT – Arduino and Thingspeak
- Lacks Physical classroom experiences
- Different Time Zones