

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology (Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in

Information Technology

By

APARNA G
(Reg. No.37120010)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND
TECHNOLOGY (DEEMED TO BE
UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI -
600 119**

AUGUST 2020



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai -
600119 www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **APARNA.G (Reg.no.37120010)** who carried out the project entitled “**CARBON FOOTPRINT CALCULATOR**” under our supervision from June 2020 to Sept 2020.

Internal Guide

Dr. G. Mathivanan

Head of the Department

Dr. R. Subhashini, M.E...Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I, **APARNA G (REG. NO. 37120010)** hereby declare that the Project Report entitled “ **CARBON FOOTPRINT CALCULATOR** ” done by me under the guidance of **Dr. G. Mathivanan** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE:

G. Aparna

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr.SUBHASHINI M.E., Ph.D.**, Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Jeberson Retna Raj M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.



InsideSherpa

Inspiring and empowering
future professionals

Aparna Gopinath

Accenture Discovery Program

Certificate of Completion
September 2nd, 2020

Over the period of September 2020, Aparna Gopinath has completed practical task modules in:

Set Project Priorities
Assemble a plan
User Journey Redesign
Outcomes Analysis
Fix the errors

Choose your language
Prioritisation & Impact Assessment

Tom Brunskill
CEO, Co-Founder of
InsideSherpa

Enrolment Verification Code [ejcsYnqhSmQe9SSbk](#) | User Verification Code [CxL2dsfqB7ouTqJSS](#) | Issued by **InsideSherpa**



06/18/2020

Aparna G

has successfully completed

Crash Course on Python

an online non-credit course authorized by Google and offered through Coursera

Google

Google

**COURSE
CERTIFICATE**



Verify at coursera.org/verify/KSDKX5YVMUEK
Coursera has confirmed the identity of this individual and
their participation in the course.

ABSTRACT

A carbon footprint is the measure of total greenhouse gas emissions, caused by an individual or an organization, expressed as carbon dioxide equivalent. These greenhouse gases are released by our day-to-day activities including, but not limited to, transportation, electricity usage, the dress we wear and even the food we eat. We are taking buried carbon in the form of fossil fuels and releasing it to the atmosphere again. An increase in greenhouse gas emissions is the main cause for climate change. Many people are ignorant of this and keep adding more carbon to the atmosphere. To limit temperature rise to 1.5°C, we must reduce our greenhouse gas emissions 7.6% each year for the next decade.

Each individual must know how much carbon footprint they leave and know how much their activity affect the environment, so that they can be aware of their actions. In this project an easy-to-use carbon footprint calculator is developed. It can be used by anyone to calculate how much carbon they emit. It also provides smart suggestions based on their carbon emission and help them reduce their carbon footprint. Moving forward in technology is good, but we also want to start taking care of our environment, smartly.

TABLE OF CONTENTS

CH NO	TITLE NAME	PAGE NO
1.	INTRODUCTION	
	1.1 CLIMATE CHANGE	1
	1.2 GREEN HOUSE EFFECT	2
	1.3 CARBON FOOTPRINT	5
2.	AIM AND SCOPE	
	2.1 AIM OF CARBON FOOTPRINT CALCULATOR	6
	2.2 SCOPE OF CARBON FOOTPRINT CALCULATOR	8
3.	EXPERIMENTAL OR MATERIALS AND METHODS USED	
	3.1 TOOLS/LIBRARY	9
	3.2 STEPS TO BE INVOLVED	10
4.	RESULTS AND DISCUSSIONS	
	4.1 RESULT AND EXPLANATION	12
5.	SUMMARY AND CONCLUSION	
	5.1 CONCLUSION	17
	REFERENCES	18
	APPENDIX	19

LIST OF FIGURES

FIG.NO	NAME OF FIGURES	PAGE NO.
1.1	Hottest years on record	2
1.2	Green House Effect	3
1.3	Types of Green House Gases	4
1.4	Sources of Green House Gases	5
2.1	IPCC CO ₂ Concentration Pathways	7
3.1	CO ₂ Emission Factors for Different Types of Goods and Services	11
4.1	Result and Explanation	12

LIST OF ABBREVIATIONS

GHG **Green House Gas**

PPM **Parts Per Million**

IPCC **I**ntergovernmental **P**anel on **C**limate **C**hange

RCP **R**epresentative **C**oncentration **P**athway

CO2 **C**arbon **D**ioxide

GLOSSARY OF ITEMS

What is Carbon Footprint?

A carbon footprint is the measure of total greenhouse gas emissions, caused by an individual or an organization, expressed as carbon dioxide equivalent. These greenhouse gases are released by our day-to-day activities including, but not limited to, transportation, electricity usage, the dress we wear and even the food we eat.

What is Green House Effect?

The greenhouse effect is the way in which heat is trapped close to the surface of the earth by greenhouse gases.

What is IPCC?

The Intergovernmental Panel on Climate Change (IPCC) is a scientific body under the United Nations, which cover the "scientific, technical and social-economic information relevant to understanding the scientific basis of risk of human-induced climate change, its potential impacts and options for adaptation and mitigation."

What is RCP?

A Representative Concentration Pathway is a greenhouse gas concentration trajectory adopted by the IPCC.

CHAPTER 1

INTRODUCTION

1.1 CLIMATE CHANGE:

There exists an enormous pressure on the earth to protect its natural environment. Human activities, increase in human needs, and sophistication lead to deterioration of the natural environmental system. Climate change is one of the biggest threats to Humans. Millions of people around the world joined the Global climate strike. People tend to use the terms, Global warming and Climate change interchangeably, but global warming is just one aspect of climate change. Global warming refers to the rise in global temperature, while climate change includes its side effects. Climate change is about abnormal variations to the climate and their effects. The increase in the earth's average temperature is mainly due to the burning of fossil fuels. This increases the greenhouse gases and traps the sun's heat from escaping the atmosphere.

Burning of Fossil fuels is releasing carbon dioxide at a rate higher than the trees and oceans can take in. Before the industrial revolution in the 18th century, the atmospheric carbon level was about 275 parts per million. Now it has increased to 400 parts per million.

Ice sheets and glaciers are melting due to Global warming. And if the rise in temperature continues, the arctic will lose its summer ice by 2030. This causes sea level to rise and it would flood most of the coastal communities, submerging Pacific Islands, Mumbai, New Orleans. New York, and Osaka.

The increase in temperature helps the atmosphere to hold moisture, which makes weather more extreme causing major storms, floods, snowfall, and droughts. Water supply gets affected and food resources diminish. 2019 was Australia's hottest and driest year on record, making the bush fire season so devastating. More than 2 dozen people died. Nearly 3000 houses were damaged or destroyed. If earth becomes hotter, more bush fires and forest fires may occur.

<u>Hottest years on record</u>		
Rank	Year	°C over normal avg.
#1	2016	+0.99°
#2	2019	+0.95°
#3	2015	+0.93°
#4	2017	+0.91°
#5	2018	+0.83°
#6	2014	+0.74°
#7	2010	+0.72°
#8	2005	+0.67°
#8 (tied)	2013	+0.67°
#10	1998	+0.65°

Fig 1.1 Hottest years on record

The largest rainforest on earth, the Amazon's trees will start to die if the area's dry season lasts longer than 5-7 months. Right now, the dry season is just a few weeks less than this threshold. This leads to less conversion of carbon dioxide into oxygen, and the global temperature will increase further.

Coral reefs are at stake. Climate change has already killed half of the Great barrier reef's corals in the last 2 years. If the temperature of Oceans increases by 1.5 degrees Celsius, we could lose 70% of the coral reef by two thousand hundred, due to coral bleaching. These effects are just the tip of an iceberg. More devastating effects affecting food, health, weather, land, and water are awaiting us in the near future. This is all due to our ignorance.

1.2 GREEN HOUSE EFFECT:

A greenhouse is a structure with roof and walls made up of transparent materials like glass. These structures are used to grow plants in regulated climatic conditions. The glass roof and walls allow solar radiation to pass through and these radiations are absorbed by the ground and the contents inside the greenhouse. As there is no contact with the atmosphere the hot air cannot escape via convection. So the temperature inside the greenhouse increases. The warmer interior protects its content in the cold weather.

Just like a greenhouse, our atmosphere allows the heat from the sun through it. This heat is absorbed by the earth and excess heat is reflected towards the atmosphere. Radiatively active gases or greenhouse gases like water vapor, Carbon dioxide, methane, and ozone, radiate this energy in all directions. Most of this energy is directed towards the surface, warming it further. These gases act like the transparent roof of a greenhouse, keeping our planet warm, hence the name greenhouse effect.

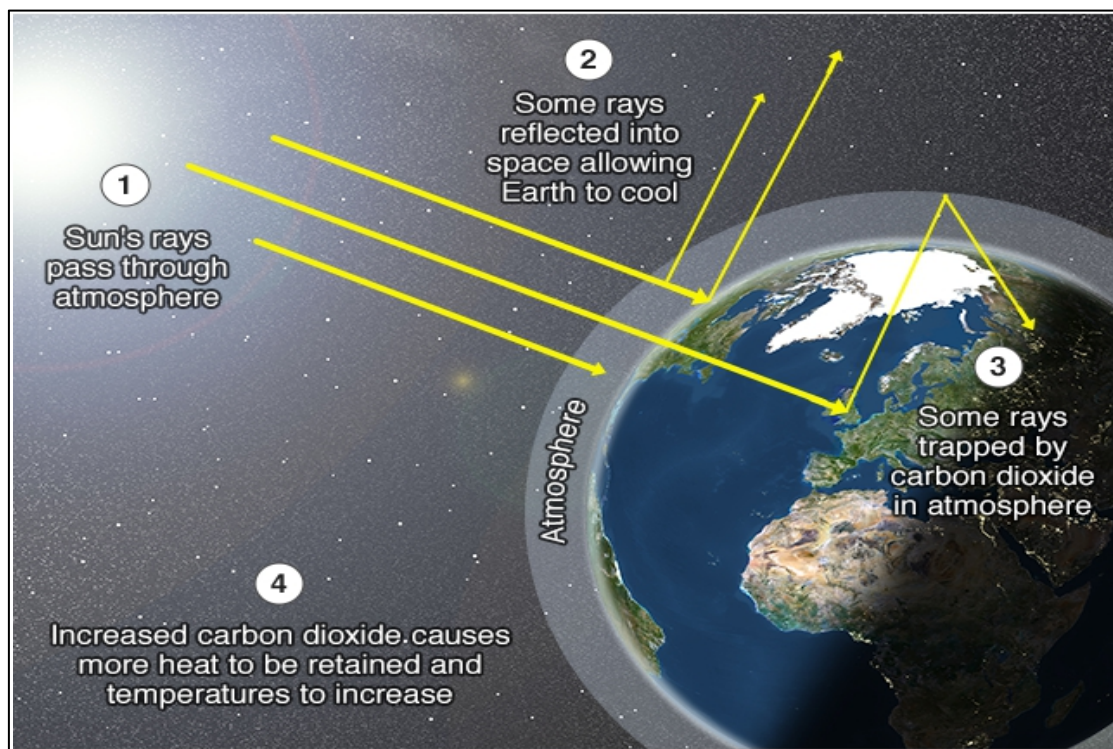


Fig 1.2 Green House Effect

But without the greenhouse effect, our planet would have an average temperature of -18°C , which is not very comfortable. The greenhouse effect keeps our earth warm and cozy at an average temperature of 14°C . It also allows the planet to maintain its water level on the surface by restricting the ice caps to the polar region. The greenhouse gases save us from the harmful low frequency and ultra-violet radiation that can damage our eyes and health.

1.2.1 Types of Green House Gases:

There are six different types of gases present in the atmosphere giving significant impact on the greenhouse effect:

- Carbon dioxide (CO₂)
- Methane (CH₄)
- Nitrous oxide (N₂O)
- Hydrofluorocarbons (HFCs)
- Perfluorocarbons (PFCs)
- Sulfur hexafluoride (SF₆)

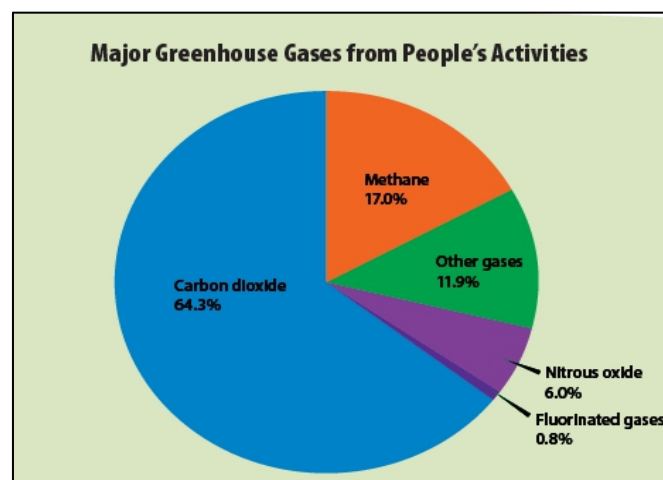


Fig 1.3 Types of Green House Gases

1.2.2 Sources of Green House Gases:

Many sectors emit GHGs that affect the natural environment. The following list gives some of these sectors that play a major role in the emission of GHG:

- Residential sector
- Industrial sector
- Commercial sector
- Transportation
- Agricultural sources
- Waste management

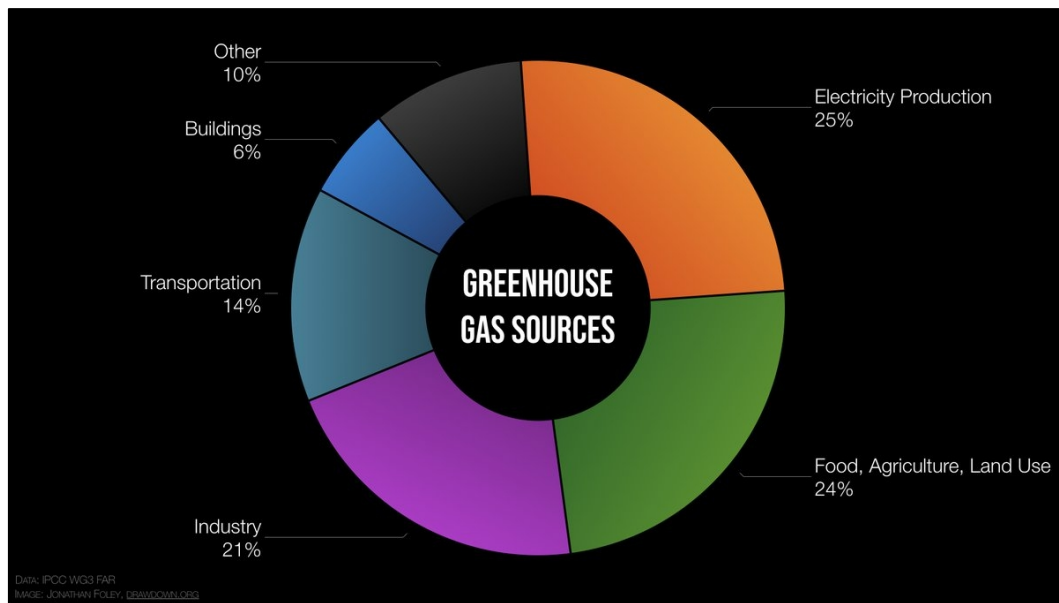


Fig 1.4 Sources of Green House Gases

1.3 CARBON FOOTPRINT:

A carbon footprint is the measure of total greenhouse gas emissions, caused by an individual or an organization, expressed as carbon dioxide equivalent. These greenhouse gases are released by our day-to-day activities including, but not limited to, transportation, electricity usage, the dress we wear and even the food we eat. We are taking buried carbon in the form of fossil fuels and releasing it to the atmosphere again. An increase in greenhouse gas emissions is the main cause for climate change. Many people are ignorant of this and keep adding more carbon to the atmosphere. To limit temperature rise to 1.5°C, we must reduce our greenhouse gas emissions 7.6% each year for the next decade. There are two types of carbon footprints:

1. Primary carbon footprint
2. Secondary carbon footprint

Primary carbon footprint is the result of direct emission of GHG due to combustion of fossil fuels, and this type of carbon footprint is in our direct control. Primary carbon footprint is the outcome of transportation, domestic energy consumption, etc. On the other hand, secondary carbon footprint is the result of indirect emission of GHG during the entire life cycle of different products. This may occur due to use of clothing, recreation and leisure goods, etc.

CHAPTER 2

AIM AND SCOPE

2.1 AIM OF CARBON FOOTPRINT CALCULATOR:

Due to human activities such as transportation, electricity production, and industries, an estimate of 43.1 billion tons of carbon dioxide was released in 2019. On average, about 1.16 million kilograms of carbon dioxide are released into the air every second. Increases in greenhouse gases have other effects apart from global warming including ocean acidification, ozone depletion, smog and air pollution, etc. Plant growth and nutrition levels are also affected by an increase in greenhouse gases. It also provides smart suggestions based on their carbon emission and help them reduce their carbon footprint. Moving forward in technology is good, but we also want to start taking care of our environment, smartly.

To limit temperature rise to 1.5°C, we must reduce our greenhouse gas emissions 7.6% each year for the next decade. Each individual must know how much carbon footprint they leave and know how much their activity affect the environment, so that they can be aware of their actions. In this project an easy-to-use carbon footprint calculator is developed. It can be used by anyone to calculate how much carbon they emit.

2.1.1 IPCC Report:

The Intergovernmental Panel on Climate Change (IPCC) is a scientific body under the United Nations, which cover the "scientific, technical and social-economic information relevant to understanding the scientific basis of risk of human-induced climate change, its potential impacts and options for adaptation and mitigation."

Below is the IPCC latest report in 2014 which predicts the future climate change, risks and impacts.

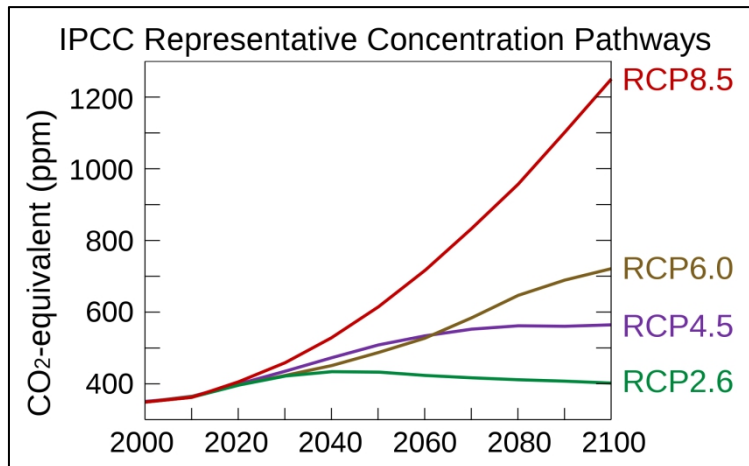


Fig 2.1 IPCC CO2 Concentration Pathways

The Representative Concentration Pathways (RCPs) describe 4 different scenarios based on different assumptions about population, economic growth, energy consumption and sources and land use over this century.

RCP 2.6

This scenario might be described as the best case for limiting anthropogenic climate change. It requires a major turnaround in climate policies and a start to concerted action in the next few years in all countries, both developing and developed. Global CO2 emissions peak by 2020 and decline to around zero by 2080. Concentrations in the atmosphere peak at around 440 ppm in mid century and then start slowly declining.

RCP 4.5

Emissions peak around mid century at around 50% higher than 2000 levels and then decline rapidly over 30 years and then stabilize at half of 2000 levels. CO2 concentration continues on trend to about 520 ppm in 2070 and continues to increase but more slowly.

RCP 6

In this scenario, emissions double by 2060 and then dramatically fall but remain well above current levels. CO2 concentration continues increasing, though at a slower rate in the latter parts of the century, reaching 620 ppm by 2100.

RCP 8.5

This is the nightmare scenario in which emissions continue to increase rapidly through the early and mid parts of the century. By 2100 annual emissions have stabilized at just under 30 gigatonnes of carbon compared to around 8 gigatonnes in 2000. Concentrations of CO₂ in the atmosphere accelerate and reach 950 ppm by 2100 and continue increasing for another 100 years.

2.2 SCOPE OF CARBON FOOTPRINT CALCULATOR:

This Carbon footprint calculator was built focusing on individual use. It can be used by everyone to calculate their carbon emission and also to get an idea how to reduce their carbon footprint. As more and more companies are trying to make their business environmentally sustainable, this app can also be used in both small and large businesses to calculate their monthly and annual Greenhouse gas emission from their Electricity usage, vehicles, flight trips and even canteen. This might reduce their expense as well, by reducing the food wastage, electricity usage and the fuel usage.

CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS USED

3.1 TOOLS/LIBRARY

This project is done using python. In this project the following libraries and tools were used to calculate carbon footprint:

1. Tkinter
2. GeoPy
3. Matplotlib

Tkinter:

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python. The name Tkinter comes from Tk interface. In this project Tkinter plays a major role as it is used for creating interface.

GeoPy:

GeoPy is a Python library that makes geographical calculations easier for the users. GeoPy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources. Here GeoPy is used for calculating the distance between two places using the coordinates of that place.

Matplotlib:

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkit like Tkinter. pyplot is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a

figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. In this project we use matplotlib to draw a graph result.

3.2 Steps to be involved:

Step 1- Data collection:

- **Electricity** : Collect data on your annual electricity bills. You can find number of power units (In India, one unit = 1KWh of electricity) consumed in your home from the monthly electricity bills issues by State Electricity Board/ Distribution/Collection companies. Take monthly consumed units and then multiply them by 12 (No of months in a year) for calculating annual emission.
- **Petrol/Diesel** : To find the number of liters of petrol/diesel you used in your car/motorcycle, We can divide the distance travelled by the mileage of the vehicle.
- **LPG** : Generally one LPG cylinder has around 14 kg of liquefied petroleum gas. Multiply number of cylinders used in a year by 14 and add the resulted value in the calculation.
- **Food** : Enter the amount of meat or chicken or any other food with high carbon footprint in kg and multiply it with the emission factor.

Step 2 – Calculation Methodology:

- **Electricity** : Input value (in KWh) X 0.708 (Emission Factor) = Output value in (Kg of CO₂)
- **Petrol** : Input value(In Litres) X 2.31 (Emission Factor) = Output value in (Kg of CO₂)
- **Diesel** : Input value(In Litres) X 2.68 (Emission Factor) = Output value in (Kg of CO₂)
- **LPG** : Input value(In Kg) X 2.79 (Emission Factor) = Output value in (Kg of CO₂)
- **Food** : Input value(In Kg) X Respective Emission Factor = Output value in (Kg of CO₂)

- Your Carbon Footprint : Addition of all the above values = Output value in (Kg of CO₂)
- Divide final value with 1000 so that you get total carbon footprint in ton of CO₂.
- Final Carbon footprint should be in tons of CO₂ (tCO₂.)

Items/Sector	CO₂ emission factor
Electricity	0.708
Diesel	2.68
Petrol	2.31
LPG	2.79
Beef	27
Cheese	13.5
Chicken	6.9
Fish	6.1

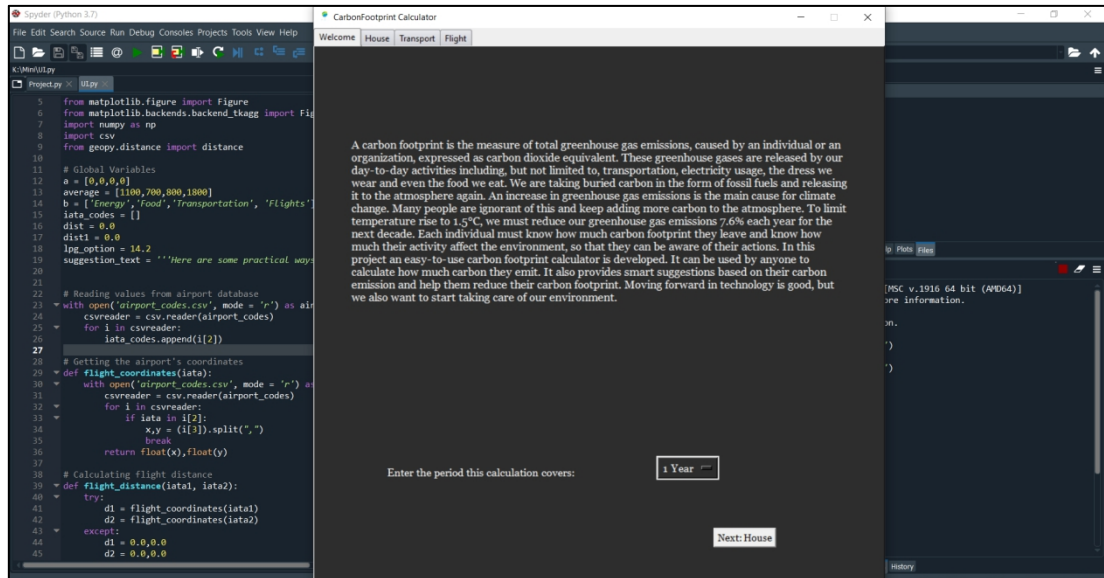
Fig 3.1 CO₂ Emission Factors for Different Types of Goods and Services

CHAPTER 4

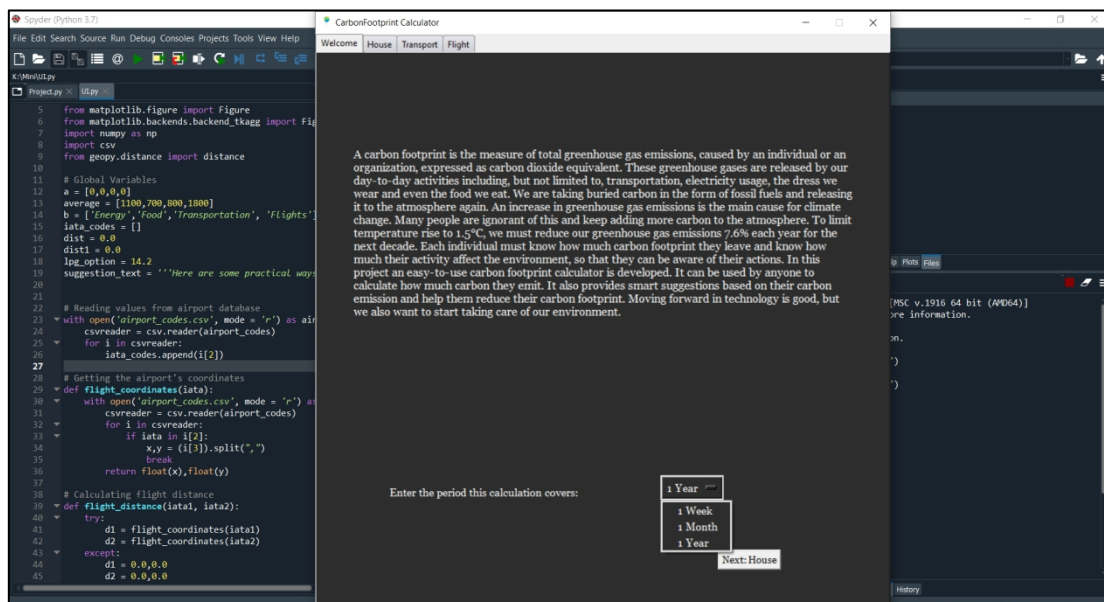
RESULT AND DISCUSSION

4.1 RESULT AND EXPLANATION :

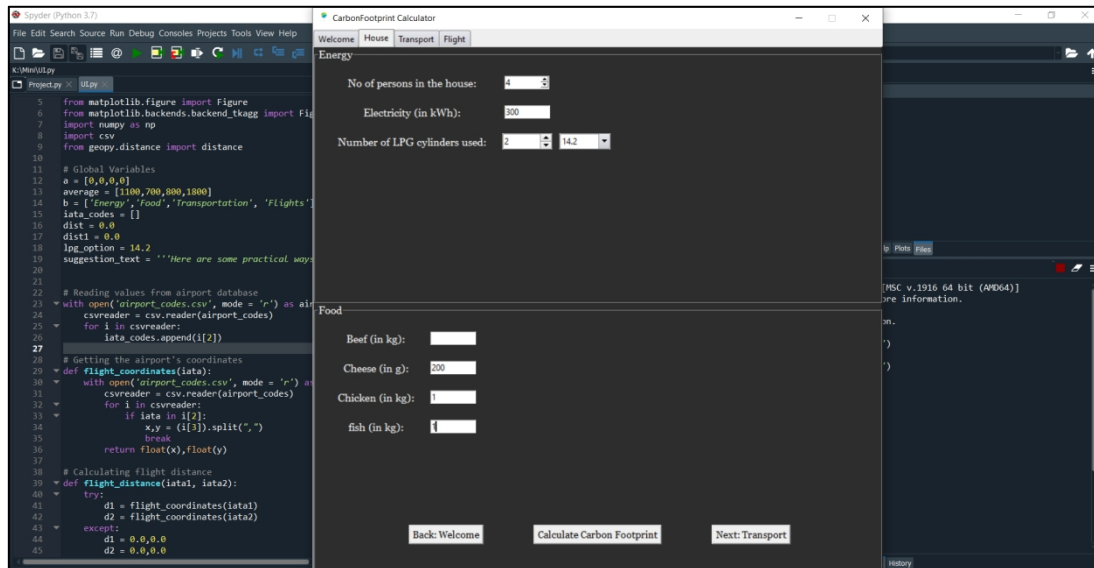
- The first section is the homepage with an introduction.



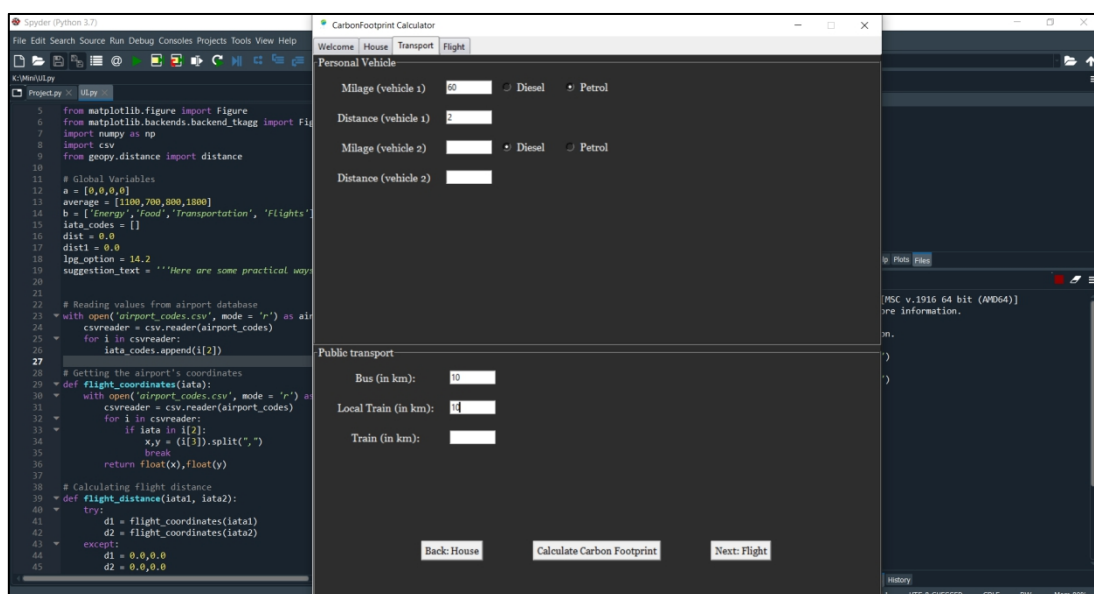
- We can limit our time period of calculation for 1 week, 1 month or 1 year.



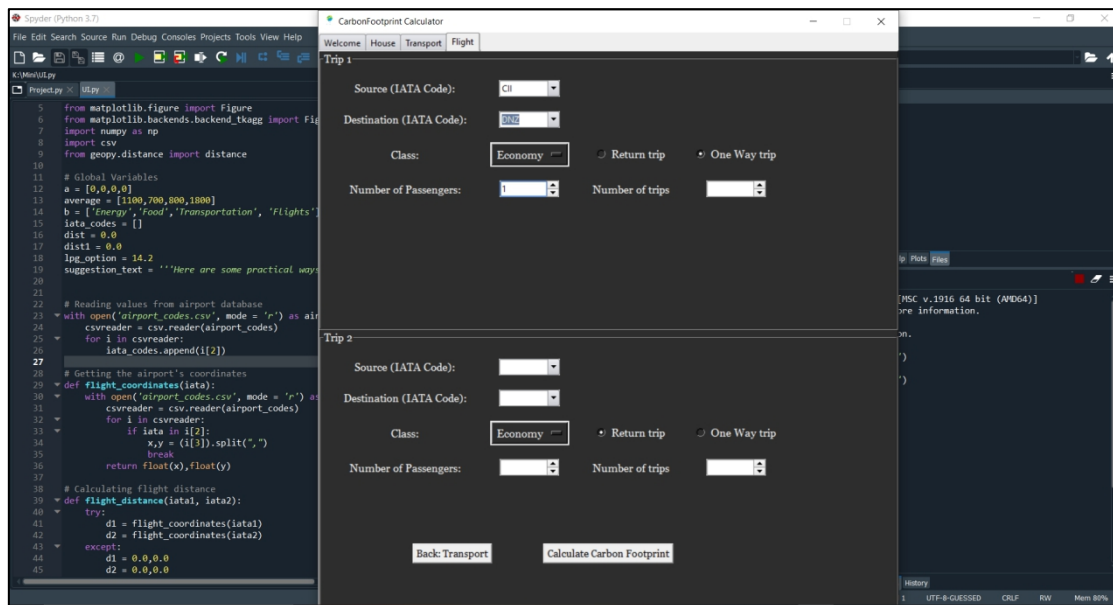
- The next section will be for energy sources and food. Here limited food options are given. Other daily needed food are not included. We have to enter the respective amount of values we consumed for respective resources.



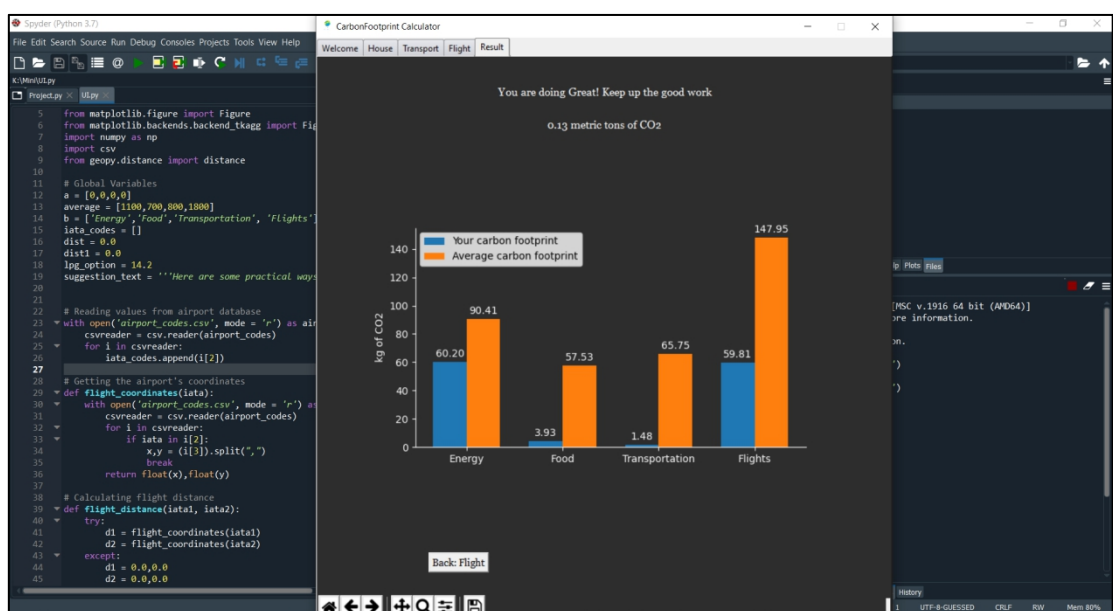
- Next section is for transportation. We need to enter respective values if we used any of them. Else we can skip them by not filling them.



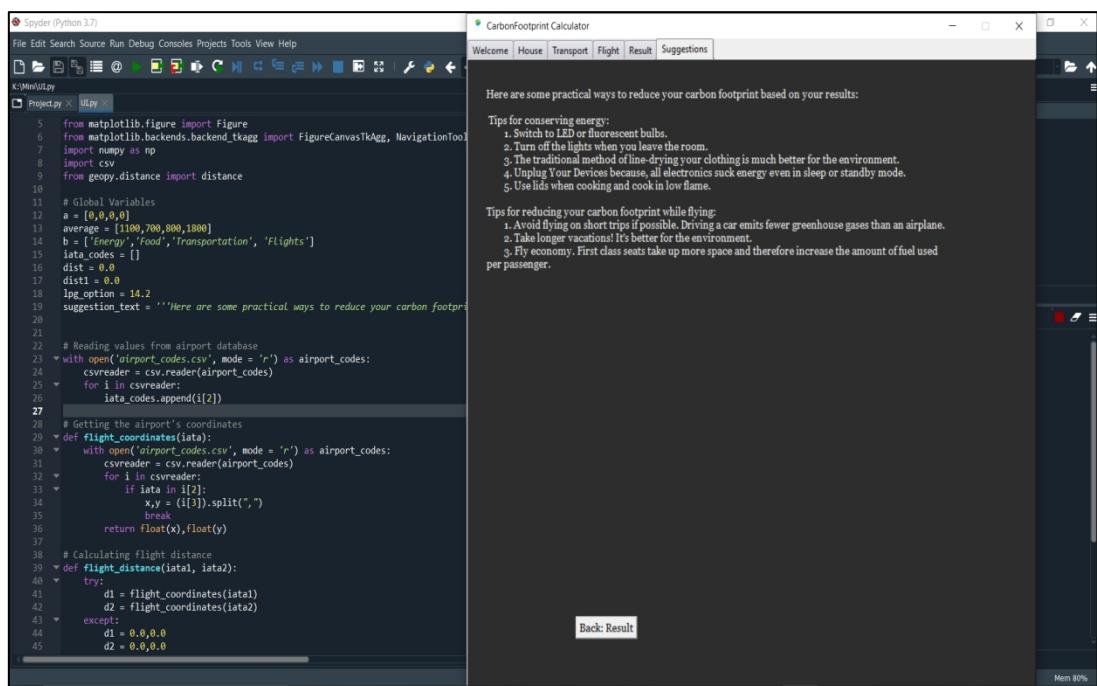
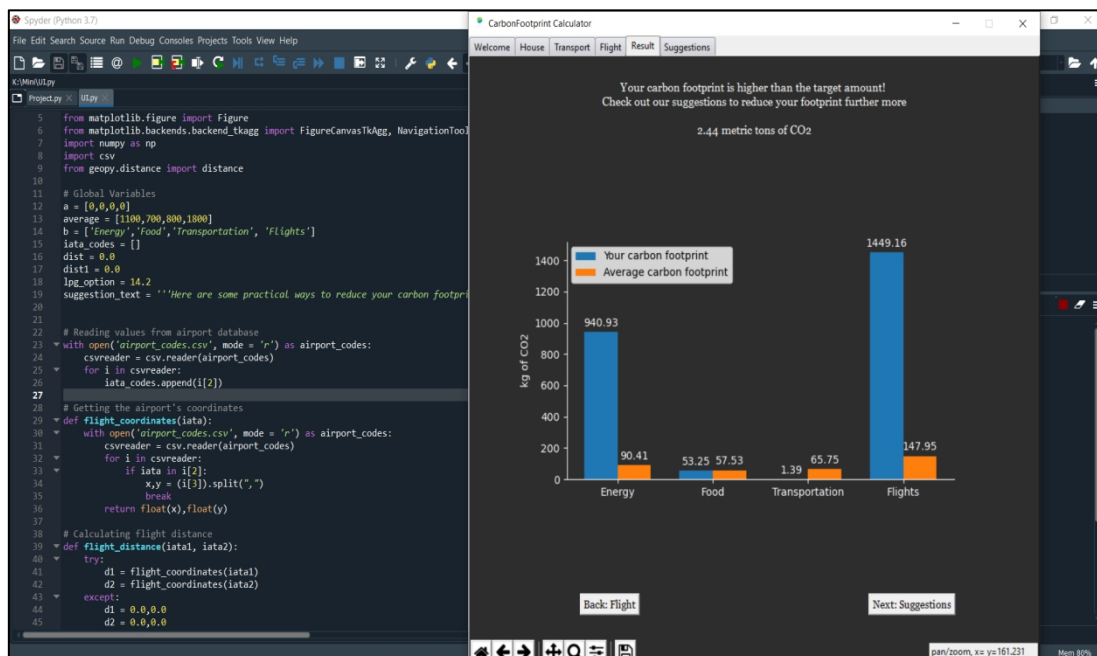
- Next section is for flight. Here we need to enter the airport codes which will be available in the flight ticket for source and destination.
- For this program we have used a csv file with airport codes in it.
- For every section there will be a separate button to calculate your carbon footprint without going to next section.



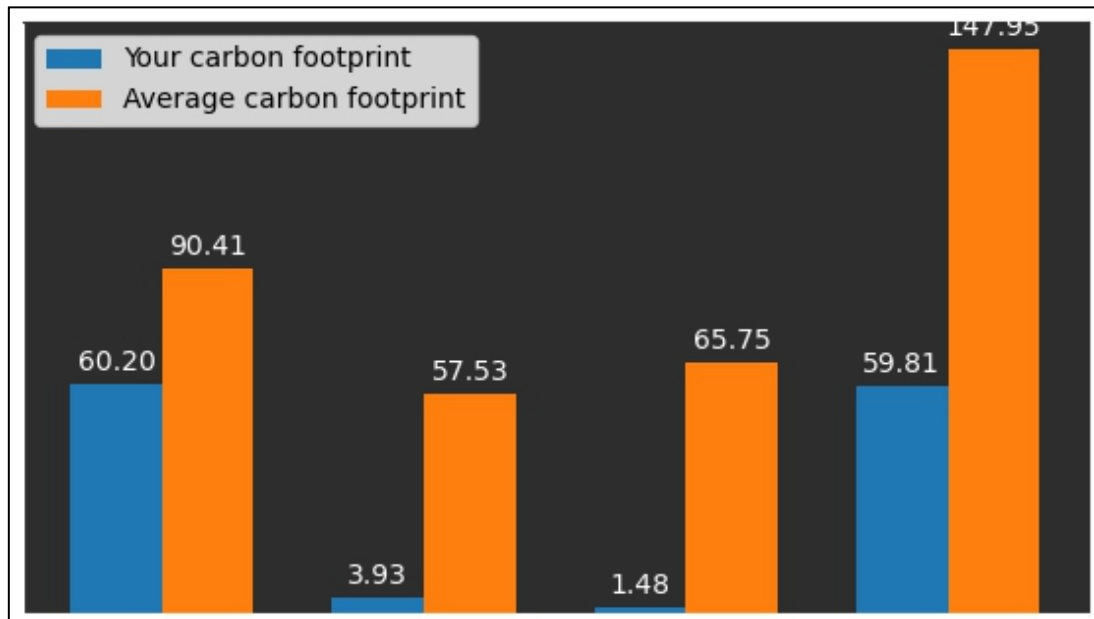
- And if the total average carbon emission is lesser than the maximum limit, then the below will be the result with a graphical representation of the result.



- If the result exceeds the limit, then there will be an extra suggestion tab after the result section which will help you to reduce your carbon footprint by giving few suggestions.



- In the result section there will be a tool bar for zooming your result, saving your result graph, etc.



CHAPTER 5

SUMMARY AND CONCLUSION

5.1 CONCLUSION:

Carbon footprint has emerged as a strong mode of GHG expression. Various GHGs are CO₂, CH₄, N₂O, and CFC, and their emissions may result in serious problems such as heat waves and other extreme events (cyclones floods, storms, wildfires), changes in patterns of infectious disease, reduced food yields and freshwater supplies, impaired functioning of ecosystems, displacement of vulnerable populations, and loss of livelihoods. While earlier studies focused only on CO₂ emissions as the guidelines and suggested inclusion of all the important GHGs in calculation, carbon footprint started becoming synonymous to a comprehensive GHG account, over the life cycle stages of any product or activity. No definition, however, has yet been accepted coherently as is clear from the fact that there are different selection of gases and tiers among studies. However, as carbon footprint reports are increasing in response to legal or business requirements, most of the calculations are following the GHG protocol worldwide. Since it has been extended to cover natural system as well, it becomes essential to deal with the unavoidable emissions. Carbon footprint has been commercialized and is being utilized by organizations to count their carbon and adopt measures to cut down emissions. This business sense has taken carbon consciousness to the households through numerous online calculators and has helped in making the civil society aware of how much their activities are contributing to global warming. Ironically, there is no check on such carbon calculating facilities and they lack coherence and transparency. Since carbon footprint is associated with money transactions and has been found to influence businesses, legal guidelines are necessary to monitor these calculations. Carbon footprint must be harnessed as a strong tool to promote GHG emission reductions among businesses, events, and civil society and should be included as indicator of sustainable development.

REFERENCES

- <https://www.greenmatters.com/p/how-is-carbon-footprint-measured>
- <https://mossy.earth/methodologies/carbon-footprint-calculator>
- <https://medium.com/@davidfurphy/what-on-earth-is-an-rcp-bbb206ddee26>
- April 2015, In book: Handbook of Sustainable Apparel Production (pp.141-166)
Edition: First Chapter: 7 Publisher: CRC Press Editors: S. Muthu
https://www.researchgate.net/publication/276193965_Carbon_Footprint_of_Textile_and_Clothing_Products
- <https://www.youtube.com/channel/UCYeA1x-IL1FzWc-AQb-dUuA/featured>

APPENDIX

SOURCE CODE:

```
from tkinter import *
import tkinter as tk
import tkinter.ttk as ttk
from ttkthemes import ThemedStyle
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,
NavigationToolbar2Tk
import numpy as np
import csv
from geopy.distance import distance

# Global Variables
a = [0,0,0,0]
average = [1100,700,800,1800]
b = ['Energy','Food','Transportation', 'Flights']
iata_codes = []
dist = 0.0
dist1 = 0.0
lpg_option = 14.2
suggestion_text = ""Here are some practical ways to reduce your carbon footprint
based on your results: ""

# Reading values from airport database
with open('airport_codes.csv', mode = 'r') as airport_codes:
    csvreader = csv.reader(airport_codes)
    for i in csvreader:
        iata_codes.append(i[2])
```

Getting the airport's coordinates

```
def flight_coordinates(iata):  
    with open('airport_codes.csv', mode = 'r') as airport_codes:  
        csvreader = csv.reader(airport_codes)  
        for i in csvreader:  
            if iata in i[2]:  
                x,y = (i[3]).split(",")  
                break  
    return float(x),float(y)
```

Calculating flight distance

```
def flight_distance(iata1, iata2):  
    try:  
        d1 = flight_coordinates(iata1)  
        d2 = flight_coordinates(iata2)  
    except:  
        d1 = 0.0,0.0  
        d2 = 0.0,0.0  
    return distance(d1,d2).km
```

Getting value from the flight combobox

```
def flight_func(event):  
    try:  
        src_option, dest_option = src_combo.get(), dest_combo.get()  
        src_option1, dest_option1 = src_combo2.get(), dest_combo2.get()  
    except:  
        src_option, dest_option = 0,0  
        src_option1, dest_option1 = 0,0  
    global dist, dist1  
    dist = round(flight_distance(src_option, dest_option),2)  
    dist1 = round(flight_distance(src_option1, dest_option1),2)
```

Getting value from LPG Combobox

```
def lpg_func(event):
```

```
lpg = lpg_combo.get()
global lpg_option
lpg_option = float(lpg)*1.55537*1.8
```

Calculations to be done when the calculate button is pressed

```
def button_pressed():
```

```
    try:
        x = float(trip_var.get())*int(trip_spin.get())*int(passenger_spin.get())
```

```
    except:
```

```
        x = 1
```

```
    try:
```

```
        y = float(trip_var2.get())*int(trip_spin2.get())*int(passenger_spin2.get())
```

```
    except:
```

```
        y = 1
```

```
duration = (duration_value[duration_list.index(duration_choice.get())])/365
```

```
avg = [round(i * duration, 2) for i in average]
```

```
a[0] = (calc(elec_entry.get(), 0.708, people_entry.get())
```

```
        + calc(lpg_spin.get(), lpg_option, people_entry.get()))
```

```
a[1] = (calc(beef_entry.get(), 27, people_entry.get())
```

```
        + calc(cheese_entry.get(), 13.5/1000, people_entry.get())
```

```
        + calc(chicken_entry.get(), 6.9, people_entry.get())
```

```
        + calc(fish_entry.get(), 6.1, people_entry.get()))
```

```
a[2] = (calc(bus_entry.get(),0.10312,1)
```

```
        + calc(local_train_entry.get(),0.036939994,1)
```

```
        + calc(train_entry.get(),0.00497,1)
```

```
        + calc(dist1_entry.get(),veh1_var.get(),mile1_entry.get())
```

```
        + calc(dist2_entry.get(),veh2_var.get(),mile2_entry.get()))
```

```
flight_value = class_value[class_name.index(class_choice.get())]
```

```
flight_value1 = class_value[class_name.index(class_choice2.get())]
```

```

a[3] = calc(dist, 0.158*1.891*flight_value*x, 1) + calc(dist1,
0.158*1.891*flight_value1*y, 1)
final_text.set("{:.2f}".format(sum(a)/1000) + " metric tons of CO2")

```

```

result_button = Button(flight, text = "Next: Result", command = selec4, font =
'Georgia 10').place(relx = 0.7, rely = 0.9, anchor = 'w')

```

```

flight_button = Button(result, text = "Back: Flight", command = selec3, font =
'Georgia 10').place(relx = 0.3, rely = 0.9, anchor = 'e')

```

```

result1_button = Button(suggestion, text = "Back: Result", command = selec4,
font = 'Georgia 10').place(relx = 0.3, rely = 0.9, anchor = 'e')

```

```

f = Figure(figsize=(7,4), dpi=100)
f.patch.set_facecolor('#2D2D2D')
graph = f.add_subplot(111)
x = np.arange(len(b))
width = 0.35
rects1 = graph.bar(x - width/2, a, width, label='Your carbon footprint')
rects2 = graph.bar(x + width/2, avg, width, label='Average carbon footprint')
graph.spines["bottom"].set_color("white")
graph.spines["left"].set_color("white")
graph.spines["top"].set_color("#2D2D2D")
graph.spines["right"].set_color("#2D2D2D")
graph.set_facecolor('#2D2D2D')
graph.set_xticks(x)
graph.set_xticklabels(b)
graph.tick_params(axis='x', colors='white')
graph.tick_params(axis='y', colors='white')
graph.set_ylabel('kg of CO2', color = 'white')
graph.legend()

```

```

def autolabel(rects):
    for rect in rects:
        height = rect.get_height()
        graph.annotate('{:.2f}'.format(height),

```

```

xy=(rect.get_x() + rect.get_width() / 2, height),
xytext=(0, 3),
textcoords="offset points",
ha='center', va='bottom', color='white')

autolabel(rects1)
autolabel(rects2)

```

```

result_bar = FigureCanvasTkAgg(f, result)
result_bar.get_tk_widget().place(relx = 0.5, rely = 0.5, anchor = 'center')
result_bar.draw()

```

```

toolbar = NavigationToolbar2Tk(result_bar, result)
toolbar.update()
toolbar.config(background='#2D2D2D')
result_bar.get_tk_widget().place(relx = 0.5, rely = 0.5, anchor = 'center')

```

```

maxi = b[a.index(max(a))]
#result_text.set("Your " + maxi + " emission is too high!. Try to reduce it :)")
tabControl.select(4)

```

```

a_temp,b_temp = [],[]
for i in range(len(a)):
    if (a[i] > avg[i]):
        a_temp.append(a[i])
        b_temp.append(b[i])

```

```

final_suggestion = suggest(b_temp)

```

```

suggestion_message = tk.Message(suggestion,text = final_suggestion, bg =
'#2D2D2D', fg = 'white', width = 650, font = 'Georgia 10').grid(row = 0, column = 0,
columnspan = 3, padx = 20, pady = 30)

```

```

if (sum(a)/1000 <= 2):
    if(len(a_temp) == 0):
        result_text.set("\nYou are doing Great! Keep up the good work")
    else:
        result_text.set("\nGood job! Check out our suggestions to reduce
your footprint further more")
        tabControl.add(suggestion, text = 'Suggestions')
        Suggestion_button = Button(result, text = "Next: Suggestions",
command = selec5, font = 'Georgia 10').place(relx = 0.7, rely = 0.9, anchor = 'w')
    else:
        result_text.set("\nYour carbon footprint is higher than the target amount!
\nCheck out our suggestions to reduce your footprint further more")
        tabControl.add(suggestion, text = 'Suggestions')
        Suggestion_button = Button(result, text = "Next: Suggestions", command
= selec5, font = 'Georgia 10').place(relx = 0.7, rely = 0.9, anchor = 'w')
        suggest(b_temp)

```

```

def suggest(x):
    global suggestion_text
    suggestion_text = "Here are some practical ways to reduce your carbon
footprint based on your results: "
    if ('Energy' in x):
        suggestion_text = suggestion_text + "\n\n Tips for conserving energy:
1. Switch to LED or fluorescent bulbs.
2. Turn off the lights when you leave the room.
3. The traditional method of line-drying your clothing is much better for the
environment.
4. Unplug Your Devices because, all electronics suck energy even in
sleep or standby mode.
5. Use lids when cooking and cook in low flame."
    if ('Food' in x):

```

```
suggestion_text = suggestion_text + ""\n\nTips for reducing your carbon footprint through dietary and lifestyle choices:
```

- ```
 1. Go vegan at least once a week
 2. Limiting your meat and/or dairy consumption can make a huge difference.
 3. Reduce your food waste, Don't over-serve food.
 4. Save and eat leftovers and Store food in the right places""
```

```
if ('Transportation' in x):
```

```
 suggestion_text = suggestion_text + ""\n\nTips for conserving fuel:
 1. Try walking, taking a bus, train, or bicycle to your destination whenever possible.
 2. If you are traveling alone, carpool with your friends or neighbors.
 3. Look after tyres. Check tyre pressures regularly
 4. The less braking and acceleration, the less fuel used.
 5. When idling for longer periods of time, shut off the engine.""
```

```
if ('Flights' in x):
```

```
 suggestion_text = suggestion_text + ""\n\nTips for reducing your carbon footprint while flying:
 1. Avoid flying on short trips if possible. Driving a car emits fewer greenhouse gases than an airplane.
 2. Take longer vacations! It's better for the environment.
 3. Fly economy. First class seats take up more space and therefore increase the amount of fuel used per passenger.""
 return suggestion_text
```

```
Functions for switching tabs
```

```
def selec0():
```

```
 tabControl.select(0)
```

```
def selec1():
```

```
 tabControl.select(1)
```

```
def selec2():
```

```
 tabControl.select(2)
```

```
def selec3():
```

```
 tabControl.select(3)
```

```

def selec4():
 tabControl.select(4)
def selec5():
 tabControl.select(5)

Function for backend calculation
def calc(var, val, people):
 try:
 carbon = (float(var) * float(val)) / float(people)
 except:
 carbon = 0
 return round(carbon,2)

Graphical user interface
root = tk.Tk()
style = ThemedStyle(root)
style.set_theme("plastik")
style.configure('TButton', background = '#C5D961', foreground = '#2D2D2D',
focusthickness=3, focuscolor='none')
style.map('TButton', background=[('active','#C5D961')])
root.title("CarbonFootprint Calculator")
root.iconbitmap('K:/Mini/logo.ico')
root.geometry("800x800")
root.resizable(False, False)
root['bg'] = '#2D2D2D'

Creating tabs
tabControl = ttk.Notebook(root)
welcome = Frame(tabControl, bg = '#2D2D2D')
house = Frame(tabControl, bg = '#2D2D2D')
transport = Frame(tabControl, bg = '#2D2D2D')
flight = Frame(tabControl, bg = '#2D2D2D')
result = Frame(tabControl, bg = '#2D2D2D')
suggestion = Frame(tabControl, bg = '#2D2D2D')

```

```
Adding tabs
```

```
tabControl.add(welcome, text = 'Welcome')
```

```
tabControl.add(house, text = 'House')
```

```
tabControl.add(transport, text = 'Transport')
```

```
tabControl.add(flight, text = 'Flight')
```

```
tabControl.add(result, text = 'Result')
```

```
tabControl.hide(result)
```

```
style.configure("Tab", focuscolor=style.configure(".")["background"])
```

```
tabControl.pack(expand = 1, fill = "both")
```

```
Welcome tab (1st tab)
```

```
welcome_message = tk.Message(welcome, text = "A carbon footprint is the
measure of total greenhouse gas emissions, caused by an individual or an
organization, expressed as carbon dioxide equivalent. These greenhouse gases
are released by our day-to-day activities including, but not limited to, transportation,
electricity usage, the dress we wear and even the food we eat. We are taking
buried carbon in the form of fossil fuels and releasing it to the atmosphere again.
An increase in greenhouse gas emissions is the main cause for climate change.
Many people are ignorant of this and keep adding more carbon to the atmosphere.
To limit temperature rise to 1.5°C, we must reduce our greenhouse gas emissions
7.6% each year for the next decade. Each individual must know how much carbon
footprint they leave and know how much their activity affect the environment, so
that they can be aware of their actions. In this project an easy-to-use carbon
footprint calculator is developed. It can be used by anyone to calculate how much
carbon they emit. It also provides smart suggestions based on their carbon
emission and help them reduce their carbon footprint. Moving forward in technology
is good, but we also want to start taking care of our environment.", bg = '#2D2D2D',
fg = 'white', font = 'Georgia 12', width = 700).place(relx = 0.5, rely = 0.32, anchor =
'center')
```

```

duration_label = tk.Label(welcome, text = 'Enter the period this calculation covers:',
padx = 20, pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').place(relx =
0.1, rely = 0.78, anchor = 'w')
duration_choice = StringVar(welcome)
duration_list = ['1 Week', '1 Month', '1 Year']
duration_value = [7,30,365]
duration_option = OptionMenu(welcome, duration_choice, *duration_list)
duration_option.config(bg = '#2D2D2D',fg = 'white', activebackground = '#2D2D2D',
activeforeground = 'white', font = 'Georgia 11')
duration_option['menu'].config(bg = '#2D2D2D',fg = 'white', activeforeground =
'#2D2D2D', activebackground = 'white', font = 'Georgia 11')
duration_choice.set(duration_list[2])
duration_option.place(relx = 0.6, rely = 0.75)

```

```

house1_button = Button(welcome, text = "Next: House", command = selec1, font =
'Georgia 10').place(relx = 0.7, rely = 0.9, anchor = 'w')

```

# House tab (2nd tab)

```

house1 = LabelFrame(house, text="Energy", bg = '#2D2D2D', fg = 'white', padx =
10, pady = 10, font = 'Georgia 11')
house1.pack(fill="both", expand="yes")

```

```

people_label = tk.Label(house1, text = "No of persons in the house:", padx = 20,
pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=0, column=0)
people_entry = tk.Spinbox(house1,from_ = 1, to = 10, width = 8)
people_entry.grid(row = 0, column=1)

```

```

elec_label = tk.Label(house1, text = "Electricity (in kWh):", padx = 20, pady =10, bg
= '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=1, column=0)
elec_entry = Entry(house1, width = 10)
elec_entry.grid(row = 1, column=1)

```

```

lpg_label = tk.Label(house1, text = "Number of LPG cylinders used:", padx = 20,
pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=2, column=0)
lpg_spin = ttk.Spinbox(house1,from_ = 1, to = 10, width = 8)
lpg_spin.grid(row = 2, column=1)
lpg_combo = ttk.Combobox(house1, value= [14.2,5,19,47.5], width = 8)
lpg_combo.bind("<<ComboboxSelected>>", lpg_func)
lpg_combo.set(14.2)
lpg_combo.grid(row = 2, column = 2, padx = 10)

```

```

house2 = LabelFrame(house, text="Food", bg = '#2D2D2D', fg = 'white', padx = 10,
pady = 10, font = 'Georgia 11')
house2.pack(fill="both", expand="yes")

```

```

beef_label = tk.Label(house2, text = "Beef (in kg):", padx = 20, pady =10, bg =
'#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=0, column=0)
beef_entry = Entry(house2, width = 10)
beef_entry.grid(row = 0, column=1)

```

```

cheese_label = tk.Label(house2, text = "Cheese (in g):", padx = 20, pady =10, bg =
'#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=1, column=0)
cheese_entry = Entry(house2, width = 10)
cheese_entry.grid(row = 1, column=1)

```

```

chicken_label = tk.Label(house2, text = "Chicken (in kg):", padx = 20, pady =10, bg
= '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=2, column=0)
chicken_entry = Entry(house2, width = 10)
chicken_entry.grid(row = 2, column=1)

```

```

fish_label = tk.Label(house2, text = "fish (in kg):", padx = 20, pady =10, bg =
'#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=3, column=0)
fish_entry = Entry(house2, width = 10)
fish_entry.grid(row = 3, column=1)

```

```
welcome_button = Button(house, text = "Back: Welcome", command = selec0, font
= 'Georgia 10').place(relx = 0.3, rely = 0.9, anchor = 'e')
final_button = Button(house, text = "Calculate Carbon Footprint", command =
button_pressed, font = 'Georgia 10').place(relx = 0.5, rely = 0.9, anchor = 'center')
transport_button = Button(house, text = "Next: Transport", command = selec2, font
= 'Georgia 10').place(relx = 0.7, rely = 0.9, anchor = 'w')
```

```
Transport tab (3rd tab)
```

```
labelframe1 = LabelFrame(transport, text="Personal Vehicle", bg = '#2D2D2D', fg =
'white', padx = 10, pady =5, font = 'Georgia 11')
labelframe1.pack(fill="both", expand="yes")
```

```
mile1_label = tk.Label(labelframe1, text = "Milage (vehicle 1)", padx = 20, pady =10,
bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=0, column=0)
mile1_entry = Entry(labelframe1, width = 10)
mile1_entry.grid(row = 0, column=1)
```

```
veh1_var = DoubleVar()
veh1_var.set(2.68)
veh1_R1 = Radiobutton(labelframe1, text="Diesel",variable = veh1_var,
value=2.68, bg = '#2D2D2D', fg = 'white', activebackground =
'#2D2D2D',activeforeground= 'white',selectcolor = '#2D2D2D', font = 'Georgia 11')
veh1_R1.grid(row = 0, column = 2, padx = 10)
veh1_R2 = Radiobutton(labelframe1, text="Petrol",variable = veh1_var, value=2.31,
bg = '#2D2D2D', fg = 'white', activebackground = '#2D2D2D',activeforeground=
'white',selectcolor = '#2D2D2D', font = 'Georgia 11')
veh1_R2.grid(row = 0, column = 3, padx = 10)
```

```
dist1_label = tk.Label(labelframe1, text = "Distance (vehicle 1)", padx = 20, pady
=10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=1, column=0)
dist1_entry = Entry(labelframe1, width = 10)
dist1_entry.grid(row = 1, column=1)
```

```

mile2_label = tk.Label(labelframe1, text = "Milage (vehicle 2)", padx = 20, pady =10,
bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=2, column=0)
mile2_entry = Entry(labelframe1, width = 10)
mile2_entry.grid(row = 2, column=1)

```

```

veh2_var = DoubleVar()
veh2_var.set(2.68)
veh2_R1 = Radiobutton(labelframe1, text="Diesel",variable = veh2_var,
value=2.68, bg = '#2D2D2D', fg = 'white', activebackground =
'#2D2D2D',activeforeground= 'white',selectcolor = '#2D2D2D', font = 'Georgia 11')
veh2_R1.grid(row = 2, column = 2, padx = 10)
veh2_R2 = Radiobutton(labelframe1, text="Petrol",variable = veh2_var, value=2.31,
bg = '#2D2D2D', fg = 'white', activebackground = '#2D2D2D',activeforeground=
'white',selectcolor = '#2D2D2D', font = 'Georgia 11')
veh2_R2.grid(row = 2, column = 3, padx = 10)

```

```

dist2_label = tk.Label(labelframe1, text = "Distance (vehicle 2)", padx = 20, pady
=10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=3, column=0)
dist2_entry = Entry(labelframe1, width = 10)
dist2_entry.grid(row = 3, column=1)

```

```

labelframe2 = LabelFrame(transport, text="Public transport", bg = '#2D2D2D', fg =
'white', padx = 10, pady =5, font = 'Georgia 11')
labelframe2.pack(fill="both", expand="yes")

```

```

bus_label = tk.Label(labelframe2, text = "Bus (in km):", padx = 20, pady =10, bg =
'#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=0, column=0)
bus_entry = Entry(labelframe2, width = 10)
bus_entry.grid(row = 0, column=1)

```

```

local_train_label = tk.Label(labelframe2, text = "Local Train (in km):", padx = 20,
pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=1, column=0)
local_train_entry = Entry(labelframe2, width = 10)
local_train_entry.grid(row = 1, column=1)

```

```
train_label = tk.Label(labelframe2, text = "Train (in km):", padx = 20, pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=2, column=0)
```

```
train_entry = Entry(labelframe2, width = 10)
```

```
train_entry.grid(row = 2, column=1)
```

```
house_button = Button(transport, text = "Back: House", command = selec1, font = 'Georgia 10').place(relx = 0.3, rely = 0.9, anchor = 'e')
```

```
final_button = Button(transport, text = "Calculate Carbon Footprint", command = button_pressed, font = 'Georgia 10').place(relx = 0.5, rely = 0.9, anchor = 'center')
```

```
flight_button = Button(transport, text = "Next: Flight", command = selec3, font = 'Georgia 10').place(relx = 0.7, rely = 0.9, anchor = 'w')
```

```
Flight tab (4th Tab)
```

```
flight1 = LabelFrame(flight, text="Trip 1", bg = '#2D2D2D', fg = 'white', padx = 10, pady = 10, font = 'Georgia 11')
```

```
flight1.pack(fill="both", expand='yes')
```

```
src_label = tk.Label(flight1, text = "Source (IATA Code):", padx = 20, pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=0, column=0)
```

```
src_combo = ttk.Combobox(flight1, value = iata_codes[1:], width = 10)
```

```
src_combo.bind("<<ComboboxSelected>>", flight_func)
```

```
src_combo.grid(row = 0, column = 1, padx = 10, pady = 10)
```

```
dest_label = tk.Label(flight1, text = "Destination (IATA Code):", padx = 20, pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=1, column=0)
```

```
dest_combo = ttk.Combobox(flight1, value = iata_codes[1:], width = 10)
```

```
dest_combo.bind("<<ComboboxSelected>>", flight_func)
```

```
dest_combo.grid(row = 1, column = 1, padx = 10, pady = 10)
```

```
class_choice = StringVar(flight1)
```

```
class_label = tk.Label(flight1, text = "Class:", padx = 20, pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=2, column=0)
```

```
class_name = ['Economy', 'Business', 'First Class']
```

```

class_value = [1,2.890,3.987]
flight_class = OptionMenu(flight1, class_choice, *class_name)
flight_class.config(bg = '#2D2D2D',fg = 'white', activebackground = '#2D2D2D',
activeforeground = 'white', font = 'Georgia 11')
flight_class['menu'].config(bg = '#2D2D2D',fg = 'white', activeforeground =
'#2D2D2D', activebackground = 'white', font = 'Georgia 11')
class_choice.set(class_name[0])
flight_class.grid(row = 2, column = 1, padx = 10, pady = 10)

```

```

trip_var = DoubleVar()
trip_var.set(1)
return_trip = Radiobutton(flight1, text="Return trip",variable = trip_var, value=1, bg
= '#2D2D2D', fg = 'white', activebackground = '#2D2D2D',activeforeground=
'white',selectcolor = '#2D2D2D', font = 'Georgia 11')
return_trip.grid(row = 2, column = 2, padx = 10)
one_way_trip = Radiobutton(flight1, text="One Way trip",variable = trip_var,
value=0.5, bg = '#2D2D2D', fg = 'white', activebackground =
'#2D2D2D',activeforeground= 'white',selectcolor = '#2D2D2D', font = 'Georgia 11')
one_way_trip.grid(row = 2, column = 3, padx = 10)

```

```

passenger_label = tk.Label(flight1, text = "Number of Passengers:", padx = 20,
pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=3, column=0)
passenger_spin = ttk.Spinbox(flight1,from_ = 1, to = 10, width = 10)
passenger_spin.grid(row = 3, column=1)

```

```

trip_label = tk.Label(flight1, text = "Number of trips", padx = 20, pady =10, bg =
'#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=3, column=2)
trip_spin = ttk.Spinbox(flight1,from_ = 1, to = 10, width = 10)
trip_spin.grid(row = 3, column=3)

```

```

flight2 = LabelFrame(flight, text="Trip 2", bg = '#2D2D2D', fg = 'white', padx = 10,
pady = 10, font = 'Georgia 11')
flight2.pack(fill="both", expand='yes')

```

```

src_label2 = tk.Label(flight2, text = "Source (IATA Code):", padx = 20, pady =10, bg
= '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=0, column=0)
src_combo2 = ttk.Combobox(flight2, value = iata_codes[1:], width = 10)
src_combo2.bind("<<ComboboxSelected>>", flight_func)
src_combo2.grid(row = 0, column = 1, padx = 10, pady = 10)

```

```

dest_label2 = tk.Label(flight2, text = "Destination (IATA Code):", padx = 20, pady
=10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=1, column=0)
dest_combo2 = ttk.Combobox(flight2, value = iata_codes[1:], width = 10)
dest_combo2.bind("<<ComboboxSelected>>", flight_func)
dest_combo2.grid(row = 1, column = 1, padx = 10, pady = 10)

```

```

class_choice2 = StringVar(flight2)
class_label2 = tk.Label(flight2, text = "Class:", padx = 20, pady =10, bg =
'#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=2, column=0)
flight_class2 = OptionMenu(flight2, class_choice2, *class_name)
flight_class2.config(bg = '#2D2D2D',fg = 'white', activebackground = '#2D2D2D',
activeforeground = 'white', font = 'Georgia 11')
flight_class2['menu'].config(bg = '#2D2D2D',fg = 'white', activeforeground =
'#2D2D2D', activebackground = 'white', font = 'Georgia 11')
class_choice2.set(class_name[0])
flight_class2.grid(row = 2, column = 1, padx = 10, pady = 10)

```

```

trip_var2 = DoubleVar()
trip_var2.set(1)
return_trip2 = Radiobutton(flight2, text="Return trip",variable = trip_var2, value=1,
bg = '#2D2D2D', fg = 'white', activebackground = '#2D2D2D',activeforeground=
'white',selectcolor = '#2D2D2D', font = 'Georgia 11')
return_trip2.grid(row = 2, column = 2, padx = 10)
one_way_trip2 = Radiobutton(flight2, text="One Way trip",variable = trip_var2,
value=0.5, bg = '#2D2D2D', fg = 'white', activebackground =
'#2D2D2D',activeforeground= 'white',selectcolor = '#2D2D2D', font = 'Georgia 11')
one_way_trip2.grid(row = 2, column = 3, padx = 10)

```

```

passenger_label2 = tk.Label(flight2, text = "Number of Passengers:", padx = 20,
pady =10, bg = '#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=3, column=0)
passenger_spin2 = ttk.Spinbox(flight2,from_ = 1, to = 10, width = 10)
passenger_spin2.grid(row = 3, column=1)

```

```

trip_label2 = tk.Label(flight2, text = "Number of trips", padx = 20, pady =10, bg =
'#2D2D2D', fg = 'white', font = 'Georgia 11').grid(row=3, column=2)
trip_spin2 = ttk.Spinbox(flight2,from_ = 1, to = 10, width = 10)
trip_spin2.grid(row = 3, column=3)

```

```

transport_button_button = Button(flight, text = "Back: Transport", command =
selec2, font = 'Georgia 10').place(relx = 0.3, rely = 0.9, anchor = 'e')
final_button = Button(flight, text = "Calculate Carbon Footprint", command =
button_pressed, font = 'Georgia 10').place(relx = 0.5, rely = 0.9, anchor = 'center')

```

```

Result tab (5th tab)

```

```

result_text = tk.StringVar()
result_text.set("Enter the values to calculate your Carbon footprint")
final_text = tk.StringVar()
final_text.set("Calculate Carbon Footprint")

```

```

result_message = tk.Label(result, textvariable = result_text, bg = '#2D2D2D', fg =
'white', font = 'Georgia 11').place(relx = 0.5, rely = 0.05, anchor = 'center')
final_message = tk.Label(result, textvariable = final_text, bg = '#2D2D2D', fg =
'white', font = 'Georgia 11').place(relx = 0.5, rely = 0.12, anchor = 'center')

```

```

root.mainloop()

```