NEW HORIZON COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC

Accredited by NAAC with 'A' Grade, Accredited by NBA

A MINI PROJECT REPORT

for

Mini Project using Python (20CSE59)

on

CARBON EMISSIONS

Submitted by

S REVANTH KUMAR

USN: 1NH19CS149, Sem-Sec: 5-C

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

CERTIFICATE

This is to certify that the mini project work titled

CARBON EMISSIONS

submitted in partial fulfillment of the degree of Bachelor of Engineering in Computer Science and Engineering by

S REVANTH KUMAR
USN:1NH19CS149

DURING

ODD SEMESTER 2021-2022

for

Course: Mini Project using Python-20CSE59

Signature of Reviewer Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner	*Signature with date*
1.	
2.	

ABSTRACT

Carbon emissions are changing our world completely, causing extreme weather events like tropical storms, wildfires, severe droughts and heat waves, negatively affecting crop production, causing disruption to animals' natural habitats, and more. The greenhouse gases are released into the atmosphere which contain co2, methane, nitrous oxide, ozone and water vapor, all of them help to contribute the sun's heat into the Earth, causing the temperature to rise. Carbon emissions are a type of greenhouse gases that occurs when co2 enters the air after a human activity or process like driving, burning of fuels, agriculture industry and more. The alarming fact it that carbon emissions can cause dramatic changes to the climate. The carbon emissions emitted now can stay in the atmosphere for thousands of years, affecting the planet all along. Carbon emissions can increase water-use efficiency in crops, these levels can also create imbalances in nitrogen and carbon, minimizing crops' necessary nutrients like iron, zinc, and protein. It can also have a major impact on humans, affecting the respiratory systems due to smog and pollution. Hence, spreading awareness about this issue and reducing the carbon emissions on a daily basis will have a positive impact on our future. The projects main idea is to let the user know the current state of a country by representing data in a graph. The user can see how countries have been emitting carbon over all these years. Next the user can check the how much carbon he/she is emitting on a yearly basis. It will also show ideas and ways for you to reduce carbon emissions in your life. By this the user has attained a certain level of understanding of the current global situation and the project will help in spreading awareness to the world and ensure we do something to reduce the emissions.

CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman, New Horizon Educational Institutions, for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal, New Horizon College of Engineering, for his constant support and encouragement.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and HOD, Department of Computer Science and Engineering, for her constant support.

I also express my gratitude to **Dr. Anidha Arulanandham** , Associate Professor, Department of Computer Science and Engineering, my mini project reviewer, for constantly monitoring the development of the project and setting up precise deadlines. Her / His valuable suggestions were the motivating factors in completing the work.

S REVANTH KUMAR

USN: 1NH19CS149

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Pollution is the introduction of contaminants into the natural environment that cause adverse change. The most important candidate in these polluting gases is carbon dioxide. Mainly caused due to the burning of fossil fuels like coal, oil extraction and more. Including these, burning biological waste like trees, solid waste also releases carbon dioxide. CO2 released stays in the atmosphere for the years to come, affecting the planet. The most dangerous problem that most humans overlook about carbon emissions is climate change. Change in climate change can cause devastating problems like storms, wildfires, droughts and severe heat waves. If this continues, our environment will change accordingly causing endangerment to the natural habitats for animals, which may lead to their extinction, including humans.

1.2 OBJECTIVES

Once developed, the application will provide services to

- Allow users to view carbon emissions,
  - From across the world over the years with a graph.
  - From an individual perspective, so as to understand where one stands in emitting co2.

- Bring awareness to the user with ways to reduce carbon emissions as an individual and as a country.

1.3 METHODOLOGY

Methodologies used in this project is python programming language with a database which is used to store values about the user and carbon emission values of countries across the years. When the user enters the particular country he needs to view, project will open a graph represented by actual data of the country across the years using matplotlib and SQLite database. Next, the user can calculate how much carbon they emit on a yearly basis. The project also shows them ways to avoid and reduce carbon emissions as an individual.

1.4 EXPECTED OUTCOMES

To spread awareness about carbon emissions, letting every citizen know the carbon levels and how to overcome the crisis by taking the right steps to reduce carbon footprints is the project's main idea. The project will give details about carbon emissions of the user during his daily life and to overcome or reduce the emissions. It will also provide insight on the carbon emissions in different countries all across the world.

1.4 HARDWARE AND SOFTWARE REQUIREMENTS

Processor	: Intel i3 or above / AMD Ryzen 3 or above
RAM	: 4GB
Hard Disk	: 256 GB or higher
Input device	: Standard
Operating system	: Windows
Compiler	: Pycharm or any IDE, SQL

CHAPTER 2

FUNDAMENTALS OF PYTHON

2.1 INTRODUCTION TO PYTHON

Python is high-level scripting language which may be used for an honestly anything like text processing, system administration and internet-related tasks. Unlike many similar languages, its core language is extremely small and straightforward to master, while allowing the addition of modules to perform a virtually limitless quite tasks. Python is a true object-oriented language, and is out there on quite a many platform. There’s even a python interpreter written entirely in Java, further enhancing python’s position as an impressive solution for internet-based problems.

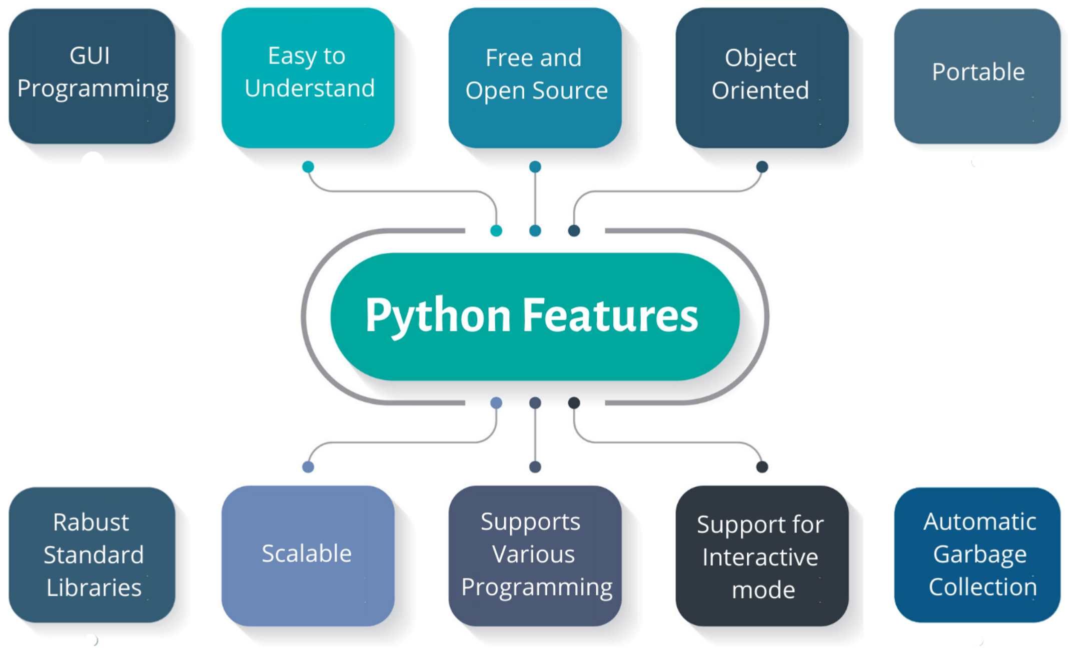

GUI Programming
Easy to Understand
Free and Open Source
Object Oriented
Portable
Python Features
Rabust Standard Libraries
Scalable
Supports Various Programming
Support for Interactive mode
Automatic Garbage Collection

Figure 2.1: Python features

2.2 Data Types and Structures

PRIMITIVE DATA STRUCTURES

a. Strings: Knowing the way to deal with textual data and their operators comes in handy when handling the string data type.

b. Float: The float () function converts the specified value into a floating-point number.

c. Integer: Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length.

d. Boolean: In programming you often need to know if an expression is True or False.You can evaluate any expression in Python, and get one of two answers, True or False.

NON-PRIMITIVE DATA TYPES

a. Set: Sets are used to store multiple items in a single variable. Set is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Tuple, and Dictionary, all with different qualities and usage. A set is a collection which is unordered, unchangeable, and unindexed. Set items are unchangeable, but you can remove items and add new items.

b. Lists and tuples (compound data types): One of the most commonly used and important data structures in Python are lists. A list is a collection of elements, and the collection can be of the same or varied data types. Understanding lists will eventually pave the way for computing algebraic equations and statistical models on your array of data.

Here are the concepts you should be familiar with:

- How multiple data types can be stored in a Python list.
- Indexing and slicing to access a specific element or sub-list of the list.

- Helper methods for sorting, reversing, deleting elements, copying, and appending.
- Nested lists — lists containing lists. For example, [1,2,3, [10,11]].
- Addition in a list.

c. Dictionaries: These are another sort of collection in Python. While lists are integer indexed hence dictionaries are address based. Dictionaries have key-value pairs, and keys are analogous to indexes in lists. There are many other data types in python like float, int, set, bool, list et cetera.

2.3 Conditionals, Loops, and Functions

a. Conditions and Branching: Python uses these Boolean variables to assess conditions. Whenever there is a comparison or evaluation, Boolean values are the resulting solution. Boolean operators (or, and, not). These are used to evaluate complex assertions together.

- or — One of the many comparisons should be true for the entire condition to be true.
- and — All of the comparisons should be true for the entire condition to be true.
- not — Checks for the opposite of the comparison specified.

b. Loops: Often you will need to try and do a repetitive task, and loops are going to be your ally to eliminate the overhead of code redundancy. You'll often got to iterate through each element of a list or dictionary, and loops are available handy for that. while and for are two sorts of loops.

c. List Comprehension: A sophisticated and succinct way of creating a list using and iterable followed by a for clause. For example, you can create a list of 9 cubes as shown in the example above using list comprehension.

d. Functions: While working on a big project, maintaining code becomes a real chore. If the program performs similar tasks many times, to manage the code conveniently functions are used. A function is a block of code that performs some operations on input file and provides you the specified output. Functions make the code more reliable, fast and easy to understand and code, it also reduces time.

2.4 Using External Libraries/Modules

One of the important reasons to use Python for data science is that the amazing community that develops high-quality packages for various domains and problems. Using external libraries and modules is an integral a part of performing on projects in Python.

These libraries and modules have defined classes, attributes, and methods that we will use to accomplish our tasks. for instance, the math library contains many mathematical functions that we will use to hold out our calculations.

2.5 Advantages of python

1. Easy to Read, Learn and Write:

Python is a high-level programming language that has English-like syntax. This makes it easier to read and understand the code. Python is really easy to pick up and learn, that is why a lot of people recommend Python to beginners. You need less lines of code to perform the same task as compared to other major languages like C/C++ and Java.

2. Improved Productivity:

Python is a very productive language. Due to the simplicity of Python, developers can focus on solving the problem. They don’t need to spend too much time in understanding the syntax or behavior of the programming language. You write less code and get more things done.

3. Interpreted Language:

Python is an interpreted language which means that Python directly executes the code line by line. In case of any error, it stops further execution and reports back the error which has occurred. Python shows only one error even if the program has multiple errors. This makes debugging easier.

4. Dynamically Typed”

Python doesn’t know the type of variable until we run the code. It automatically assigns the data type during execution. The programmer doesn’t need to worry about declaring variables and their data types.

5. Free and Open-Source

Python comes under the OSI approved open-source license. This makes it free to use and distribute. You can download the source code, modify it and even distribute your version of Python. This is useful for organizations that want to modify some specific behavior and use their version for development.

6. Vast Libraries Support

The standard library of Python is huge, you can find almost all the functions needed for your task. So, you don’t have to depend on external libraries.But even if you do, a Python package manager (pip) makes things easier to import other great packages from the Python package index (PyPi). It consists of over 200,000 packages.

7. Portability

In many languages like C/C++, you need to change your code to run the program on different platforms. That is not the same with Python. You only write once and run it anywhere.

2.6 Functions in Python

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing. As you already know, Python gives you many built-in functions like print(), etc. but you can also create your own functions. These functions are called user-defined functions.

Above shown is a function definition that consists of the following components.

1.Keyword def that marks the start of the function header.

2.A function name to uniquely identify the function. Function naming follows the same rules of writing identifiers in Python.

3.Parameters (arguments) through which we pass values to a function. They are optional.

4.A colon (:) to mark the end of the function header.

5.Optional documentation string (docstring) to describe what the function does.

6.One or more valid python statements that make up the function body. Statements must have the same indentation level (usually 4 spaces).

7.An optional return statement to return a value from the function.

CHAPTER 3

FUNDAMENTALS OF TKINTER

3.1 INTRODUCTION TO TKINTER

Tkinter ("Tk Interface") is python's standard cross-platform package for creating graphical user interfaces (GUIs). It provides access to an underlying Tcl interpreter with the Tk toolkit, which itself may be a cross-platform, multilanguage graphical interface library. Tkinter is not the only GUI library for python, but it's the one that comes standard. Additional GUI libraries which will be used with python include wxPython, PyQt, and kivy. Tkinter's greatest strength is its ubiquity and ease. It works out of the box on most platforms (Linux, OSX, Windows), and comes complete with a good range of widgets necessary for many common tasks (buttons, labels, drawing canvas, multiline text, etc). As a learning tool, tkinter has some features that are unique among GUI toolkits, like named fonts, bind tags, and variable tracing.

3.2 Tkinter library

Python has multiple options for developing/creating a GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most ordinarily used method. it's a typical Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is that the fastest and easiest method to make the GUI applications. Creating a GUI using tkinter is a simple task.

To create a tkinter app:

1. Importing the module – tkinter

2.Create the main window (container)

3.Add the desired number of widgets to the main window

4.Apply the event Trigger on the widgets.

• Tk(screenName=None, baseName=None, className=’Tk’, useTk=1): to make a main window, tkinter offers a way ‘Tk(screenName=None, baseName=None, className=’Tk’, useTk=1)’. to vary the name of the window, you'll change the className to the specified one.

• mainloop(): there's a way known by the name mainloop() is used when your application is prepared to run. mainloop() is an infinite loop accustomed to run the application, await an event to occur and process the event as long as the window isn't closed.

• tkinter also offers access to the geometric configuration of the widgets which may organize the widgets within the parent windows. There are mainly three geometry manager classes class.

• pack() method: It organizes the widgets in blocks before placing within the parent widget. grid() method: It organizes the widgets in grid (table-like structure) before placing within the parent widget. place() method: It organizes the widgets by placing them on specific positions directed by the programmer.

3.3 Python GUI Widgets

Widgets in tkinter are,

- Button
- Checkbutton
- Radiobutton
- Entry
- Frame
- Label

- Listbox
- Menu
- MenuButton
- Message
- Scale
- Scrollbar
- Text
- TopLevel
- SpinBox
- PannedWindow
- Canvas

3.4 Advantages of Tkinter

Layered approach:

The layered approach utilized in designing Tkinter gives Tkinter all of the benefits of the TK library. Therefore, at the time of creation, Tkinter inherited from the advantages of a GUI toolkit that had been given time to mature. This makes early versions of Tkinter tons more stable and reliable than if it had been rewritten from scratch. Moreover, the conversion from Tcl/Tk to Tkinter is basically trivial, so Tk programmers can learn to use Tkinter very easily.

Accessibility:

Learning Tkinter is extremely intuitive, and thus quick and painless. The Tkinter implementation hides the detailed and sophisticated calls in simple, intuitive methods. this is often a continuation of the Python way of thinking, since the language excels at quickly building prototypes. it's therefore expected that its preferred GUI library be implemented using an equivalent approach.

Portability:

Python scripts that use Tkinter don't require modifications to be ported from one platform to the opposite. Tkinter is out there for any platform that Python is implemented for, namely Microsoft Windows, X Windows, and Macintosh. this provides it an excellent advantage over most competing libraries, which are often restricted to at least one or two platforms. Moreover, Tkinter will provide the native look-and-feel of the precise platform it runs on.

Availability:

Tkinter is now included in any Python distribution. Therefore, no supplementary modules are required so as to run scripts using Tkinter.

CHAPTER 4

FUNDAMENTALS OF DBMS

4.1 INTRODUCTION

A database management system (DBMS) is a computerized system that enables users to make and maintain a database. The DBMS could be a general-purpose software that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. Defining a database involves specifying the data types, structures, and constraints of the data to be stored within the database. The database definition or descriptive information is additionally stored by the DBMS within the form of a database catalog or dictionary; it's called meta-data. Constructing the database is that the process of storing the data on some data-storage medium that's controlled by the DBMS. Manipulating a database includes functions like querying the database to retrieve specific data, updating the data-base to reflect changes within the mini-world, and generating reports from the info. Sharing a database allows multiple users and programs to access the database simultaneously.

4.2 CHARACTERISTICS

A number of characteristics distinguish the database approach from the much older approach of writing customized programs to access data stored in files. In traditional file processing, each user defines and implements the files needed for a specific software application as part of programming the application. For example, one user, the grade reporting office, may keep files on students and their grades. Programs to print a student’s transcript and to enter new grades are implemented as part of the application. A second user, the accounting office, may keep track of students’ fees and their payments. Although both users are interested in data about students, each user maintains separate files—and programs to manipulate these files—because each requires some data not available from the other user's files. This redundancy in defining and storing data results in wasted storage space and in redundant efforts to maintain common up-to-date data. In the database approach, a single repository maintains data that is defined once and then accessed by various users repeatedly through queries, transactions, and application programs. The main characteristics of the database approach versus the file-processing approach are the following:

■ Self-describing nature of a database system

■ Insulation between programs and data, and data abstraction

■ Support of multiple views of the data

■ Sharing of data and multiuser transaction processing

4.3 DATA MODELS

Data models show that how the data is connected and stored in the system. It shows the relationship between data. A Model is basically a conceptualization between attributes and entities. There were basically three main data models in DBMS that were Network, hierarchical, and relational. But these days, there a lots of data models that are given below. There are different types of the data models and now let see each of them in detail:

1. Flat data model
2. Entity relationship model
3. Relation model
4. Record base model
5. Network model
6. Hierarchical model

7. Object oriented data model

8. Object relation model

9. Semi structured model

10. Associative model

11. Context data model

4.4 THREE SCHEMA ARCHITECTURE

The purpose of the Three Schema architecture is so that:

• All users should be able to access same data

• A users view should be resistant to changes made in other views

• Users shouldn't ought to know physical database storage details

• DBA should be ready to change database storage structures without affecting the users view

• Internal structure of database should be unaffected by changes to physical aspects of storage

Physical Data Level

• Defines details of its physical storage structures.

• Deals with indices and RAM

• the internal schema uses a physical data model to explain the entire details of data storage and access paths for the database.

In other words, records, and blocks of data with links to every other are organized within the file for efficiency.

Conceptual Data Level

• Also referred to as Logical Level

• Hides details of the physical level

• Data is represented as a group of tables

• Describes the structure of the entire database for a community of user

External Data Level

• Specifies a view of the info within the application

• Tailored to the requirements of a specific user

• Portions of knowledge shouldn't be seen by some users

• Implements A level of security

4.5 ENTITY RELATIONSHIP (ER) MODEL

ER model describes data in terms of:

➔ **Entity type or set**

Collection (or set) of similar entities that have the same attributes.ER model defines entity sets, not individual entities. But entity sets described in terms of their attributes.

➔ **Types of attributes**

Simple (atomic) attributes

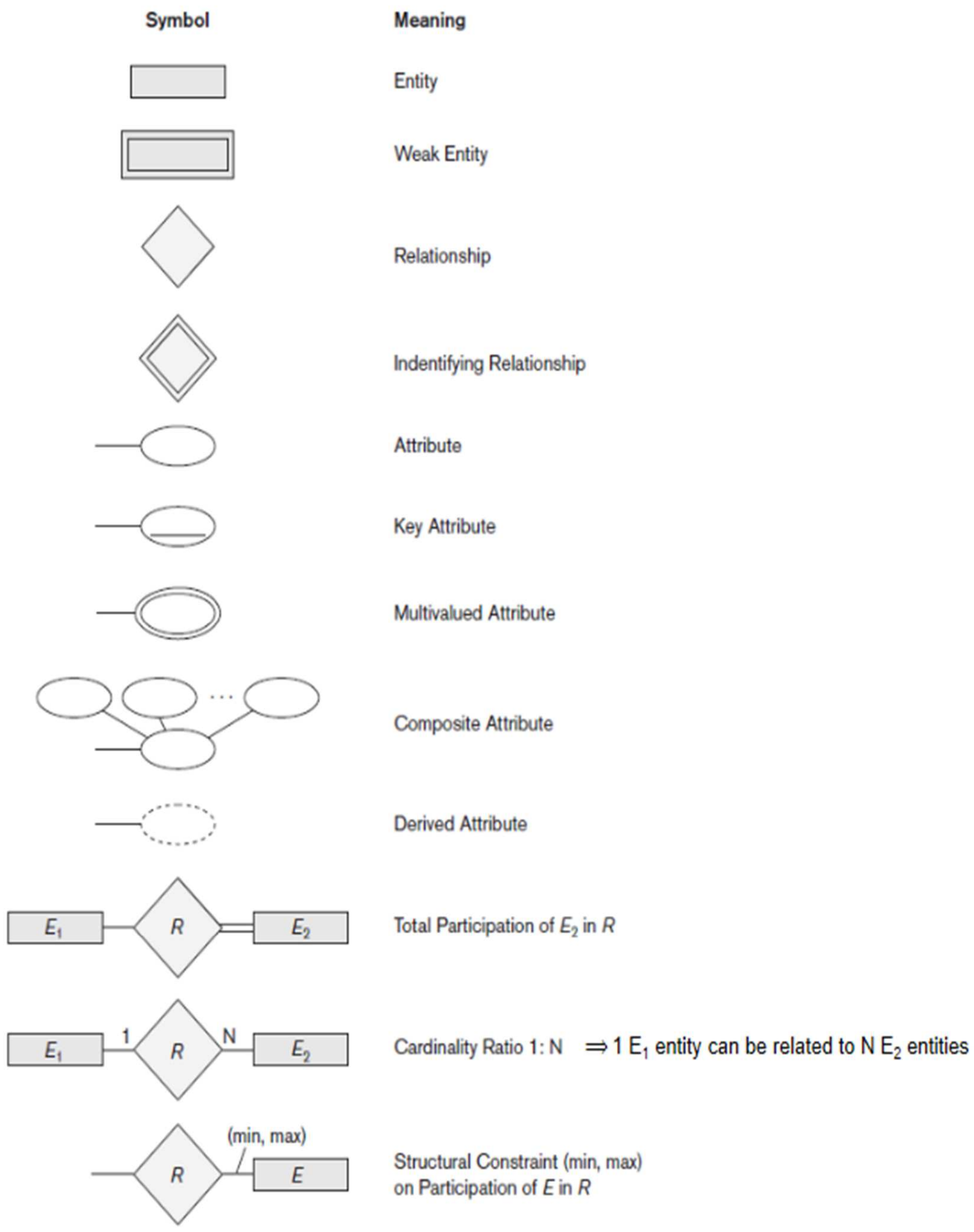
Composite attributes

Single-valued attributes

Multivalued attributes

Stored attributes

Derived attributes

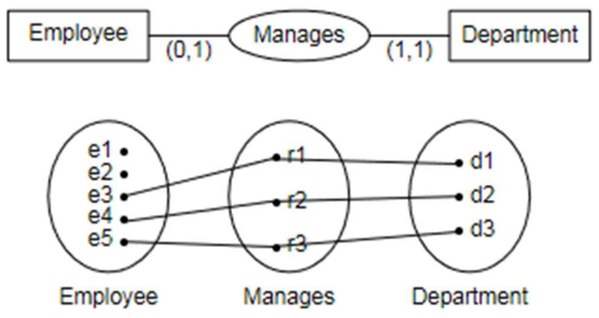

Symbol
Meaning
Entity
Weak Entity
Relationship
Indentifying Relationship
Attribute
Key Attribute
Multivalued Attribute
Composite Attribute
Derived Attribute
E1
R
E2
Total Participation of E2 in R
1
N
Cardinality Ratio 1: N ⇒ 1 E1 entity can be related to N E2 entities
(min, max)
E
Structural Constraint (min, max) on Participation of E in R

➔ **Relationships**

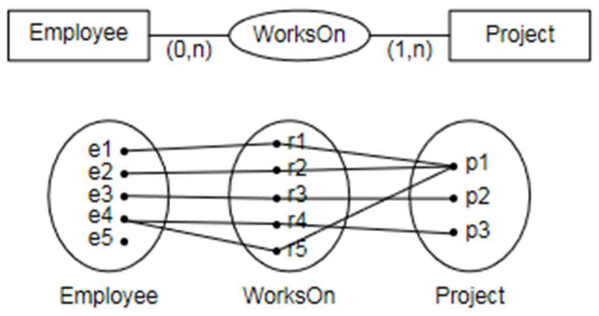
A relationship in a DBMS, is primarily the way two or more data sets are linked. This is so true for Relational Database Management Systems. One dataset may be then termed as the Foreign key and the ones linked to it may be termed as the Primary Key. There may be multiple Foreign and Primary keys linked to each other.

Types of relationships:

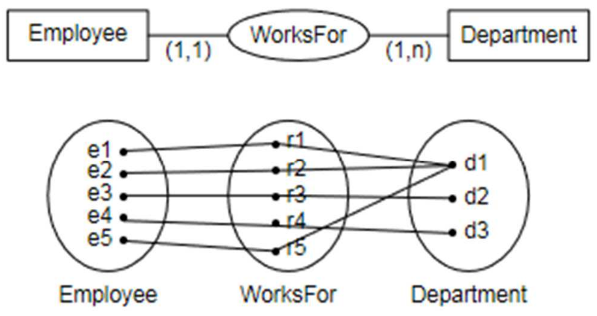
A one-to-one Relationship



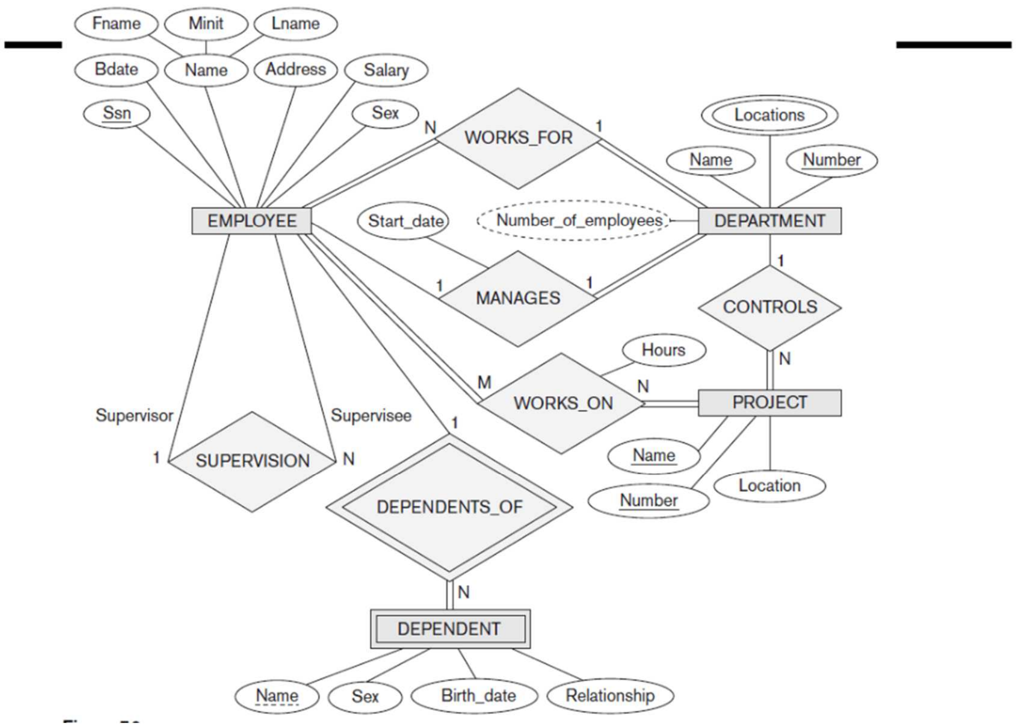
A many-to-many Relationship



A one-to-many Relationship



An ER Diagram:



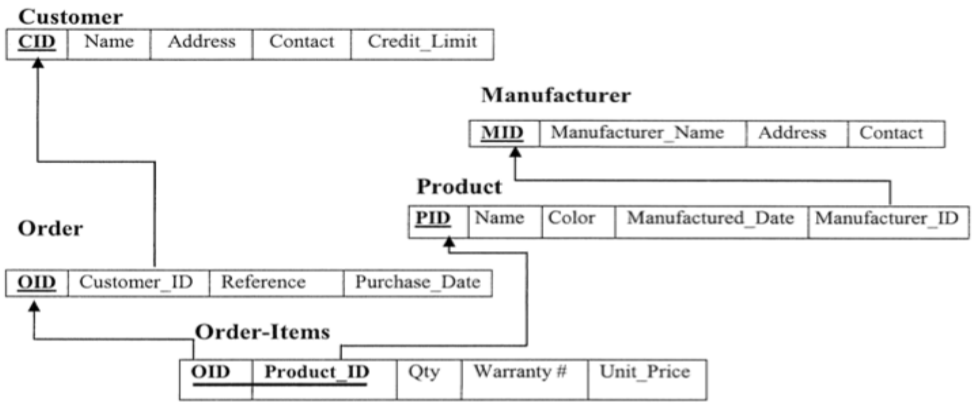
4.6 Relational Schema

Relation schema defines the design and structure of the relation like it consists of the relation name, set of attributes/field names/column names. Every attribute would have an associated domain.

1. A specific characteristic, that bears the same real-world concept, may appear in more than one relationship with the same or a different name. For example, in Employees relation, Employee Id (EmpId) is represented in Vouchers as AuthBy and PrepBy.

2. The specific real-world concept that appears more than once in a relationship should be represented by different names. For example, an employee is represented as subordinate or junior by using EmpId and as a superior or senior by using SuperId, in the employee's relation.

3. The integrity constraints that are specified on database schema shall apply to every database state of that schema.



Domain constraint, Key constraint, Entity integrity constraint, and Referential integrity constraint are the four different constraints of the relational databases. Let us now discuss them in detail.

1. Domain constraint: The value of each characteristic of a relationship needs to be an indivisible value and we need to draw it out of the possible values corresponding to its domain. Thus, the value of a characteristic needs to adjust to the data type corresponding to the domain.

2. Key constraints and Null values: Each data record that relates to a tuple of a relation in a table needs to be distinct. Thus, this implies that no two rows or tuples in relation or table can have the same combination of values for their entire data item. Every relation has a super key by default and depicts uniqueness constraints. It is a combination of all the characteristics. Sometimes a relation can have more than just one key. Each such key is a candidate key. Out of these, we need to define one of the keys as the Primary Key.

3. Entity integrity constraint: This constraint states that primary key value cannot be null as it defines the individual tuple in a relation. A null value indicates the failure to identify such tuples and thus it means that they are duplicates.

4. Referential integrity constraint: It is specified to maintain the consistency among the tuples of two or more relations.

CHAPTER 5

FUNDAMENTALS OF SQL

5.1 INTRODUCTION

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in relational database. SQL is the standard language for Relation Database System. All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix and SQL Server use SQL as standard database language.

Also, they are using different dialects, such as:

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc.

5.2 ADVANTAGES OF SQL

- Allows users to access data in relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

5.3 SQL COMMANDS

SQL Commands: The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP.

DML (Data Manipulation Language)

DML statements affect records in a table. These are basic operations we perform on data such as selecting a few records from a table, inserting new records, deleting unnecessary records, and updating/modifying existing records.

SELECT – select records from a table

INSERT – insert new records

UPDATE – update/Modify existing records

DELETE – delete existing records

DDL (Data Definition Language)

DDL statements are used to alter/modify a database or table structure and schema. These statements handle the design and storage of database objects.

CREATE – create a new Table, database, schema

ALTER – alter existing table, column description

DROP – delete existing objects from database

DCL (Data Control Language)

DCL statements control the level of access that users have on database objects.

GRANT – allows users to read/write on certain database objects

REVOKE – keeps users from read/write permission on database objects

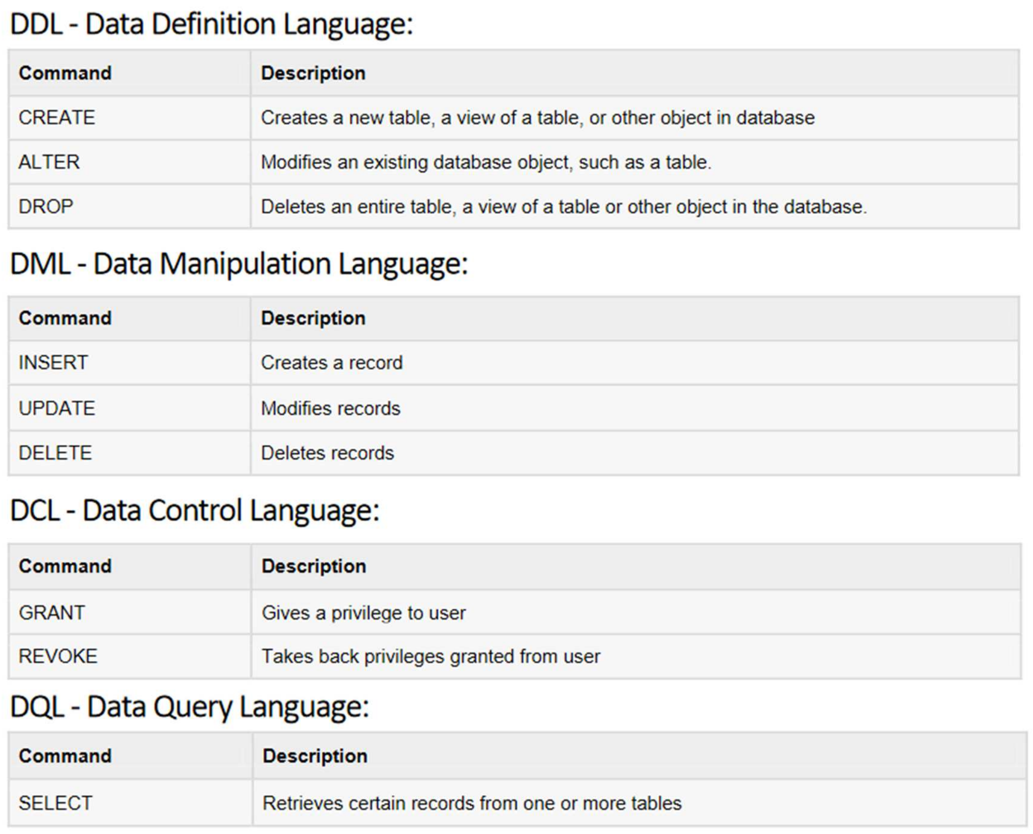
TCL (Transaction Control Language)

TCL statements allow you to control and manage transactions to maintain the integrity of data within SQL statements.

BEGIN Transaction – opens a transaction

COMMIT Transaction – commits a transaction

ROLLBACK Transaction – ROLLBACK a transaction in case of any error



DDL - Data Definition Language:

Command	Description
CREATE	Creates a new table, a view of a table, or other object in database
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other object in the database.

DML - Data Manipulation Language:

Command	Description
INSERT	Creates a record
UPDATE	Modifies records
DELETE	Deletes records

DCL - Data Control Language:

Command	Description
GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user

DQL - Data Query Language:

Command	Description
SELECT	Retrieves certain records from one or more tables

CHAPTER 6

DESIGN

6.1 DESIGN GOALS

The project has been designed in an easy user accessible. To spread awareness about carbon emissions, letting every citizen know the carbon levels and how to overcome the crisis by taking the right steps to reduce carbon footprints is the project’s main idea. The project will give details about carbon emissions of the user during his daily life and to overcome or reduce the emissions. It will also provide insight on the carbon emissions in different countries all across the world. This mini project has ensured that the user has an interactive and explorable environment. The interface is user friendly, simple to understand and has tried to ensure that there are no bugs.

6.2 ALGORITHM

STEP 1: START

STEP 2: Enter the country name for displaying carbon emissions

STEP 3: GUI window

STEP 4: Enter number of family members

STEP 5: Enter avg calories

STEP 6: Select the utilities that you use on a daily basis.

STEP 7: Displays Total expenditure from utilities

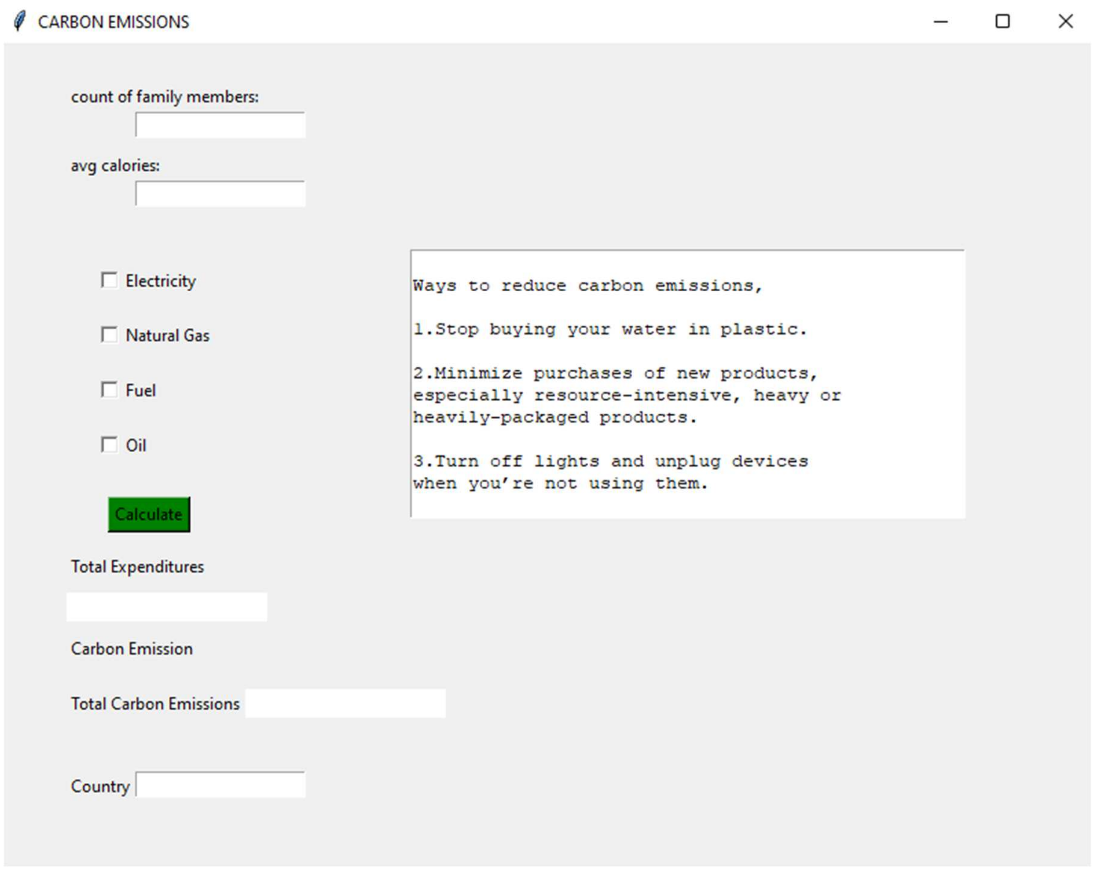
STEP 8: Displays carbon emissions for the number of family members.

STEP 9: Displays total carbon emissions

STEP 10: Shows ways to reduce carbon emissions

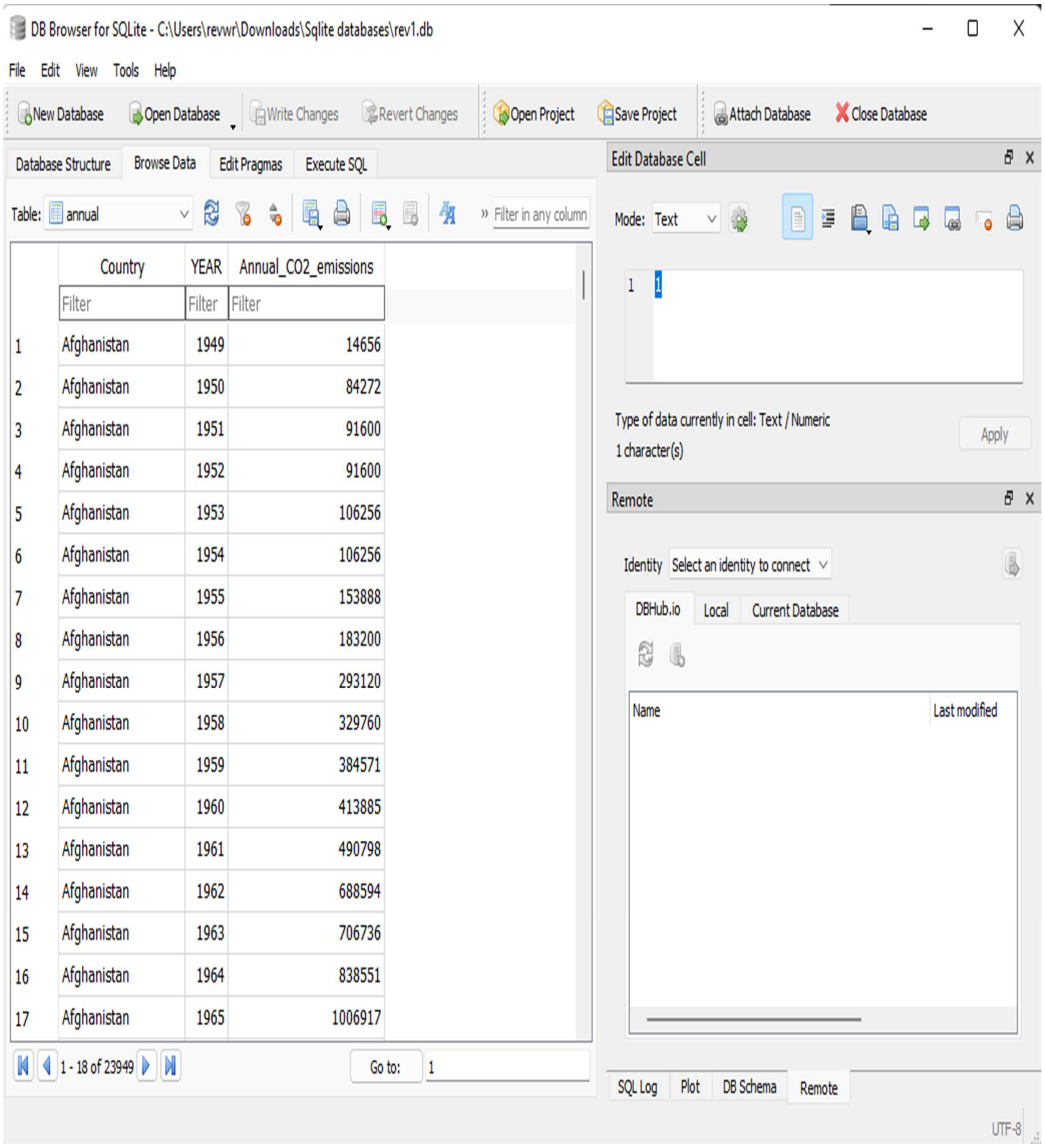
STEP 11: STOP

6.3 GUI STRUCTURE



```
Ways to reduce carbon emissions,

1.Stop buying your water in plastic.

2.Minimize purchases of new products,
especially resource-intensive, heavy or
heavily-packaged products.

3.Turn off lights and unplug devices
when you're not using them.
```

6.4 DATABASE



DB Browser for SQLite - C:\Users\revwr\Downloads\Sqlite databases\rev1.db

Table: annual

	Country	YEAR	Annual_CO2_emissions
1	Afghanistan	1949	14656
2	Afghanistan	1950	84272
3	Afghanistan	1951	91600
4	Afghanistan	1952	91600
5	Afghanistan	1953	106256
6	Afghanistan	1954	106256
7	Afghanistan	1955	153888
8	Afghanistan	1956	183200
9	Afghanistan	1957	293120
10	Afghanistan	1958	329760
11	Afghanistan	1959	384571
12	Afghanistan	1960	413885
13	Afghanistan	1961	490798
14	Afghanistan	1962	688594
15	Afghanistan	1963	706736
16	Afghanistan	1964	838551
17	Afghanistan	1965	1006917

1 - 18 of 23949 Go to: 1

Edit Database Cell — Mode: Text

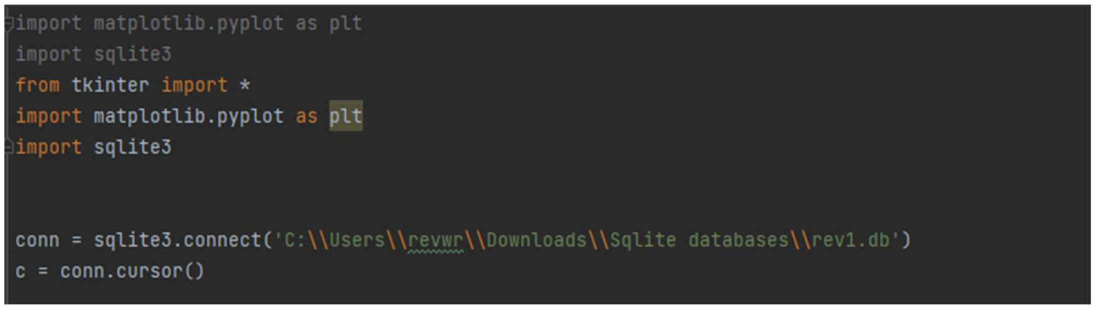
Type of data currently in cell: Text / Numeric

1 character(s)

CHAPTER 7

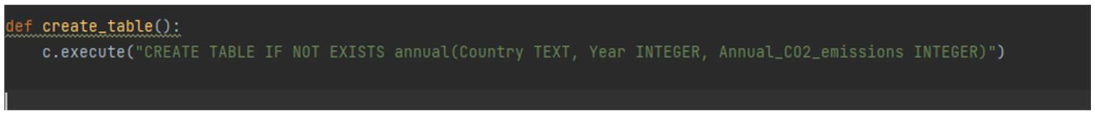
IMPLEMENTATION

7.1 Connecting to database



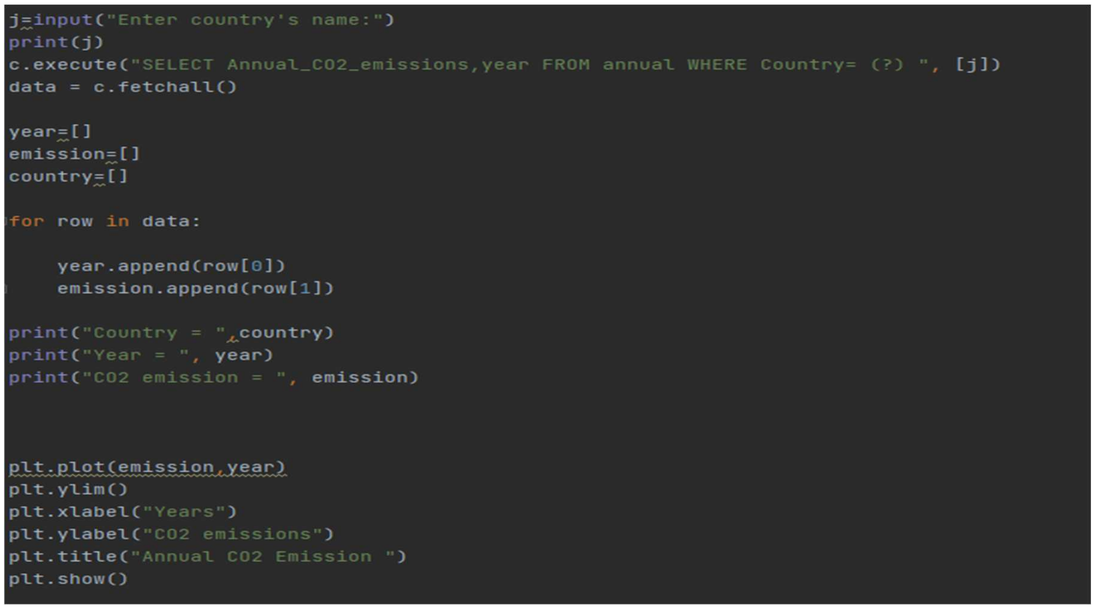
```
import matplotlib.pyplot as plt
import sqlite3
from tkinter import *
import matplotlib.pyplot as plt
import sqlite3


conn = sqlite3.connect('C:\\Users\\revwr\\Downloads\\Sqlite databases\\rev1.db')
c = conn.cursor()
```

7.2 Creating table



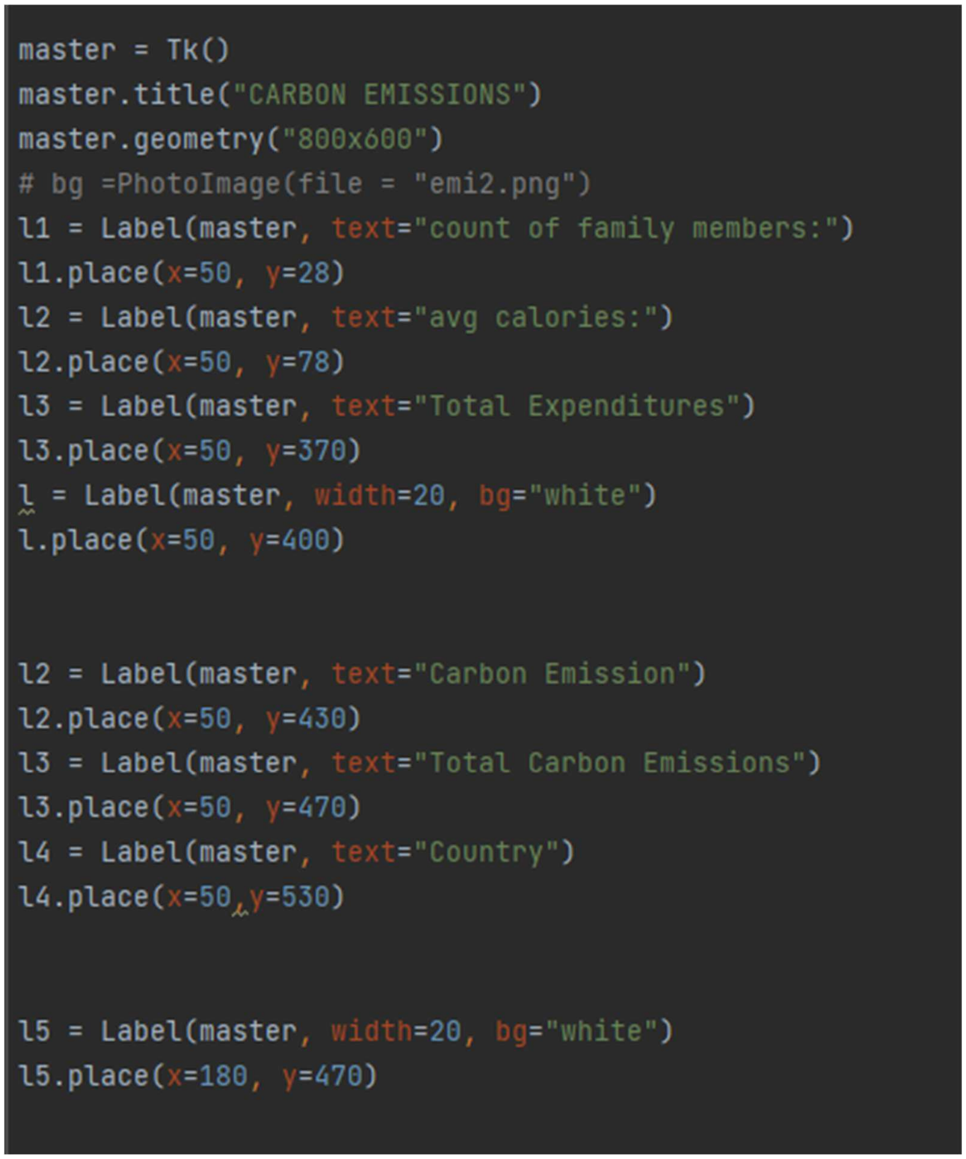
```
def create_table():
    c.execute("CREATE TABLE IF NOT EXISTS annual(Country TEXT, Year INTEGER, Annual_CO2_emissions INTEGER)")
```

7.3 Plotting graph

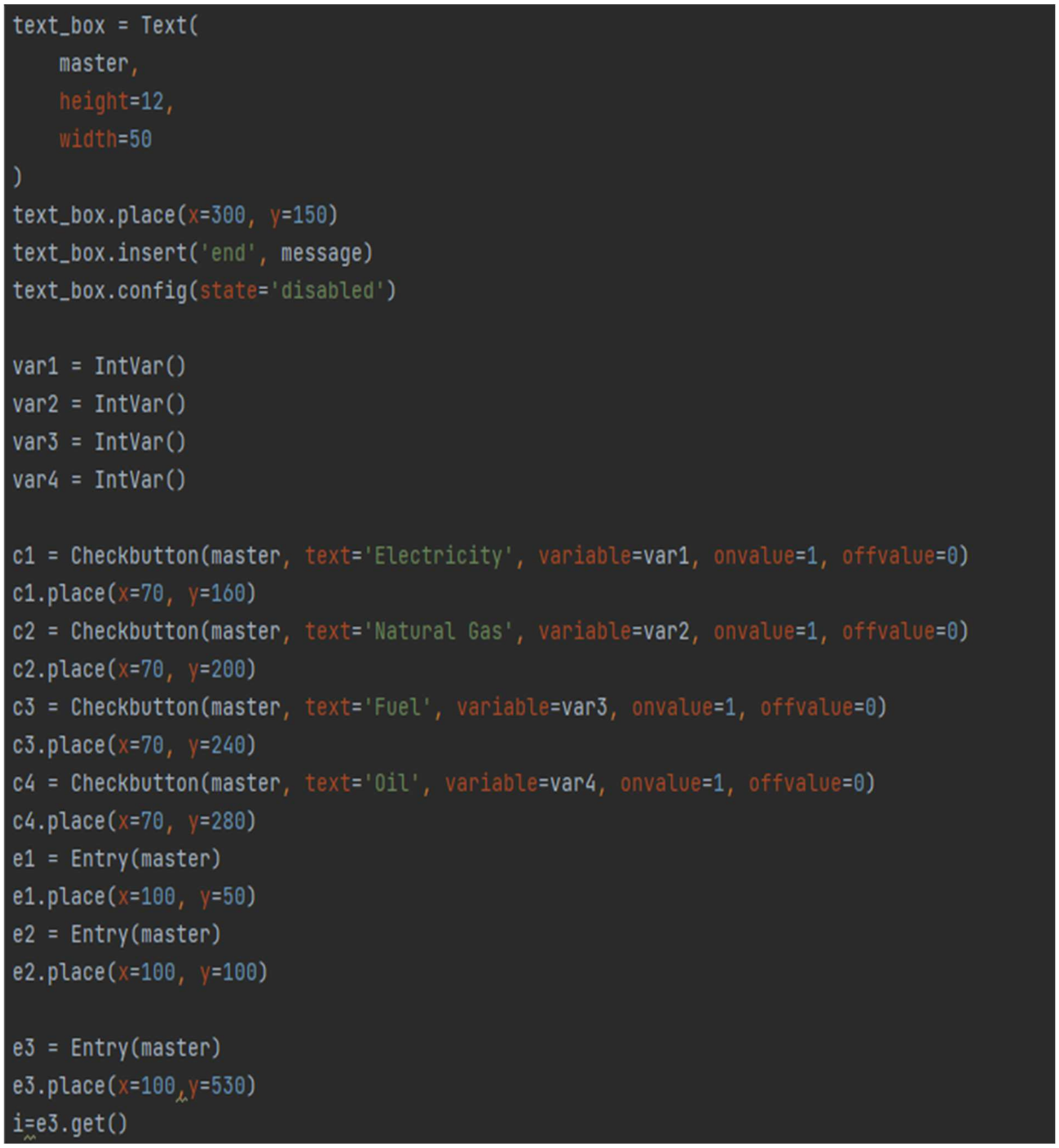


```
j=input("Enter country's name:")
print(j)
c.execute("SELECT Annual_CO2_emissions,year FROM annual WHERE Country= (?) ", [j])
data = c.fetchall()

year=[]
emission=[]
country=[]

for row in data:

    year.append(row[0])
    emission.append(row[1])

print("Country = ",country)
print("Year = ", year)
print("CO2 emission = ", emission)



plt.plot(emission,year)
plt.ylim()
plt.xlabel("Years")
plt.ylabel("CO2 emissions")
plt.title("Annual CO2 Emission ")
plt.show()
```

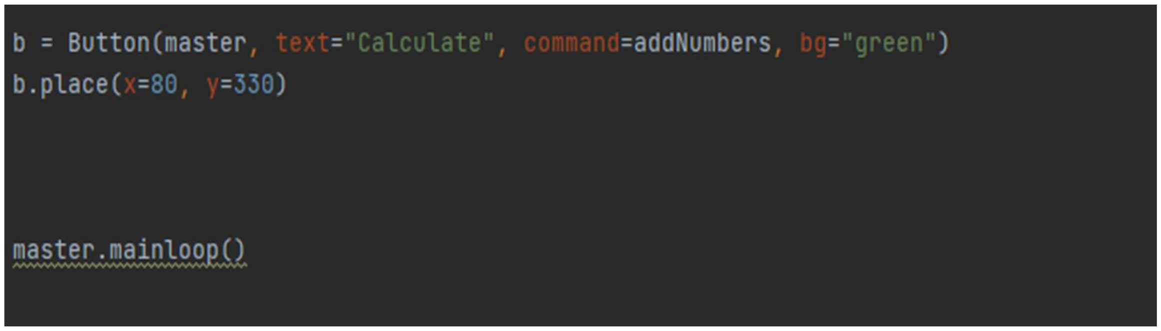
7.4 GUI



```
master = Tk()
master.title("CARBON EMISSIONS")
master.geometry("800x600")
# bg =PhotoImage(file = "emi2.png")
l1 = Label(master, text="count of family members:")
l1.place(x=50, y=28)
l2 = Label(master, text="avg calories:")
l2.place(x=50, y=78)
l3 = Label(master, text="Total Expenditures")
l3.place(x=50, y=370)
l = Label(master, width=20, bg="white")
l.place(x=50, y=400)


l2 = Label(master, text="Carbon Emission")
l2.place(x=50, y=430)
l3 = Label(master, text="Total Carbon Emissions")
l3.place(x=50, y=470)
l4 = Label(master, text="Country")
l4.place(x=50,y=530)


l5 = Label(master, width=20, bg="white")
l5.place(x=180, y=470)
```

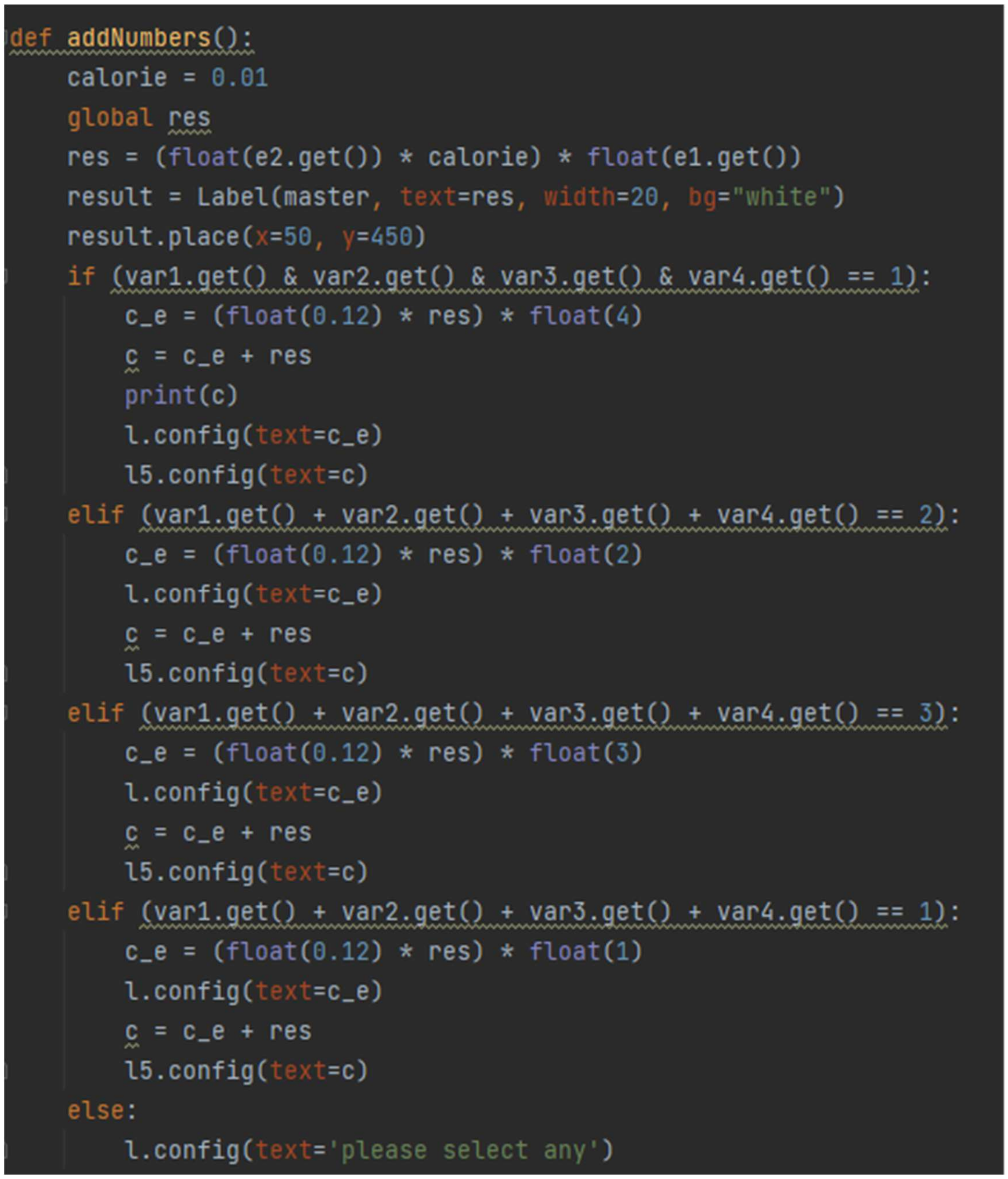


```
text_box = Text(
    master,
    height=12,
    width=50
)
text_box.place(x=300, y=150)
text_box.insert('end', message)
text_box.config(state='disabled')

var1 = IntVar()
var2 = IntVar()
var3 = IntVar()
var4 = IntVar()

c1 = Checkbutton(master, text='Electricity', variable=var1, onvalue=1, offvalue=0)
c1.place(x=70, y=160)
c2 = Checkbutton(master, text='Natural Gas', variable=var2, onvalue=1, offvalue=0)
c2.place(x=70, y=200)
c3 = Checkbutton(master, text='Fuel', variable=var3, onvalue=1, offvalue=0)
c3.place(x=70, y=240)
c4 = Checkbutton(master, text='Oil', variable=var4, onvalue=1, offvalue=0)
c4.place(x=70, y=280)
e1 = Entry(master)
e1.place(x=100, y=50)
e2 = Entry(master)
e2.place(x=100, y=100)

e3 = Entry(master)
e3.place(x=100,y=530)
i=e3.get()
```



```
b = Button(master, text="Calculate", command=addNumbers, bg="green")
b.place(x=80, y=330)



master.mainloop()
```

7.5 Calculating an individual's carbon emission

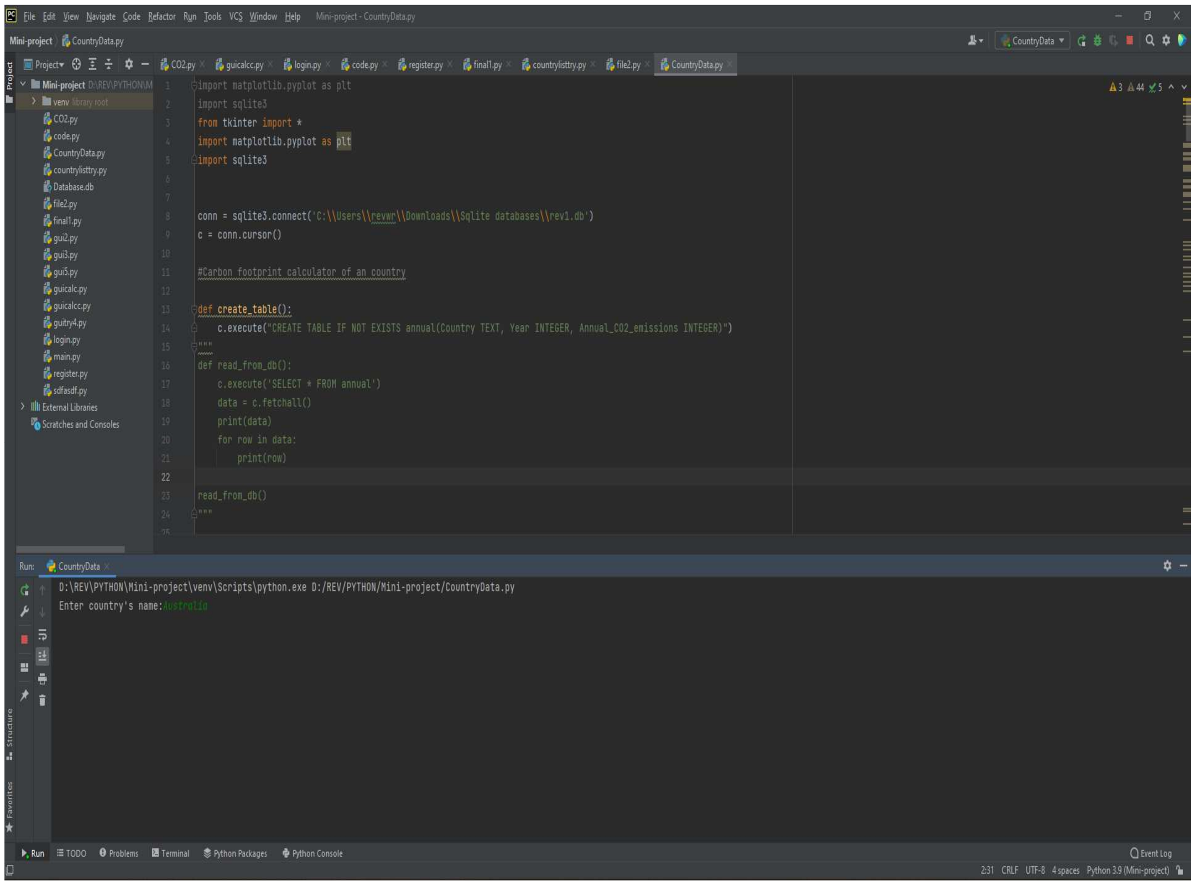


```
def addNumbers():
    calorie = 0.01
    global res
    res = (float(e2.get()) * calorie) * float(e1.get())
    result = Label(master, text=res, width=20, bg="white")
    result.place(x=50, y=450)
    if (var1.get() & var2.get() & var3.get() & var4.get() == 1):
        c_e = (float(0.12) * res) * float(4)
        c = c_e + res
        print(c)
        l.config(text=c_e)
        l5.config(text=c)
    elif (var1.get() + var2.get() + var3.get() + var4.get() == 2):
        c_e = (float(0.12) * res) * float(2)
        l.config(text=c_e)
        c = c_e + res
        l5.config(text=c)
    elif (var1.get() + var2.get() + var3.get() + var4.get() == 3):
        c_e = (float(0.12) * res) * float(3)
        l.config(text=c_e)
        c = c_e + res
        l5.config(text=c)
    elif (var1.get() + var2.get() + var3.get() + var4.get() == 1):
        c_e = (float(0.12) * res) * float(1)
        l.config(text=c_e)
        c = c_e + res
        l5.config(text=c)
    else:
        l.config(text='please select any')
```

CHAPTER 8

RESULTS

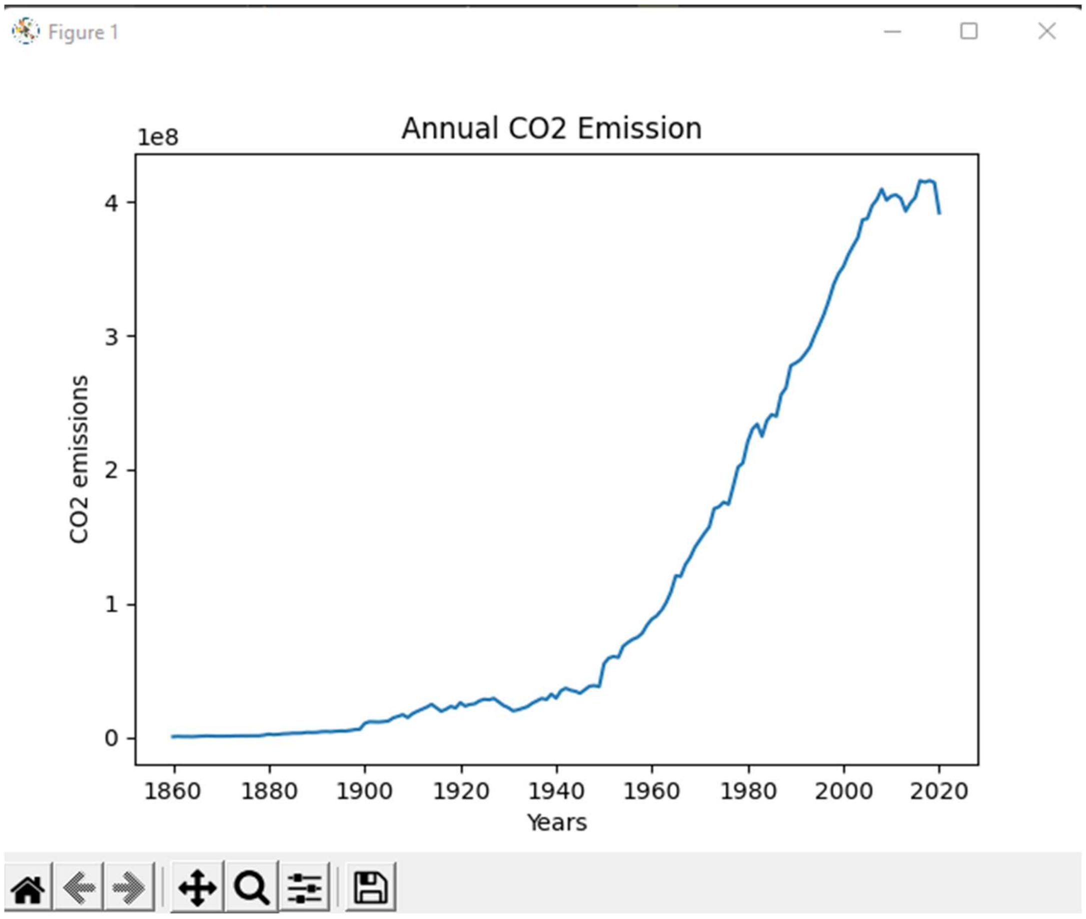
8.1 Input country name



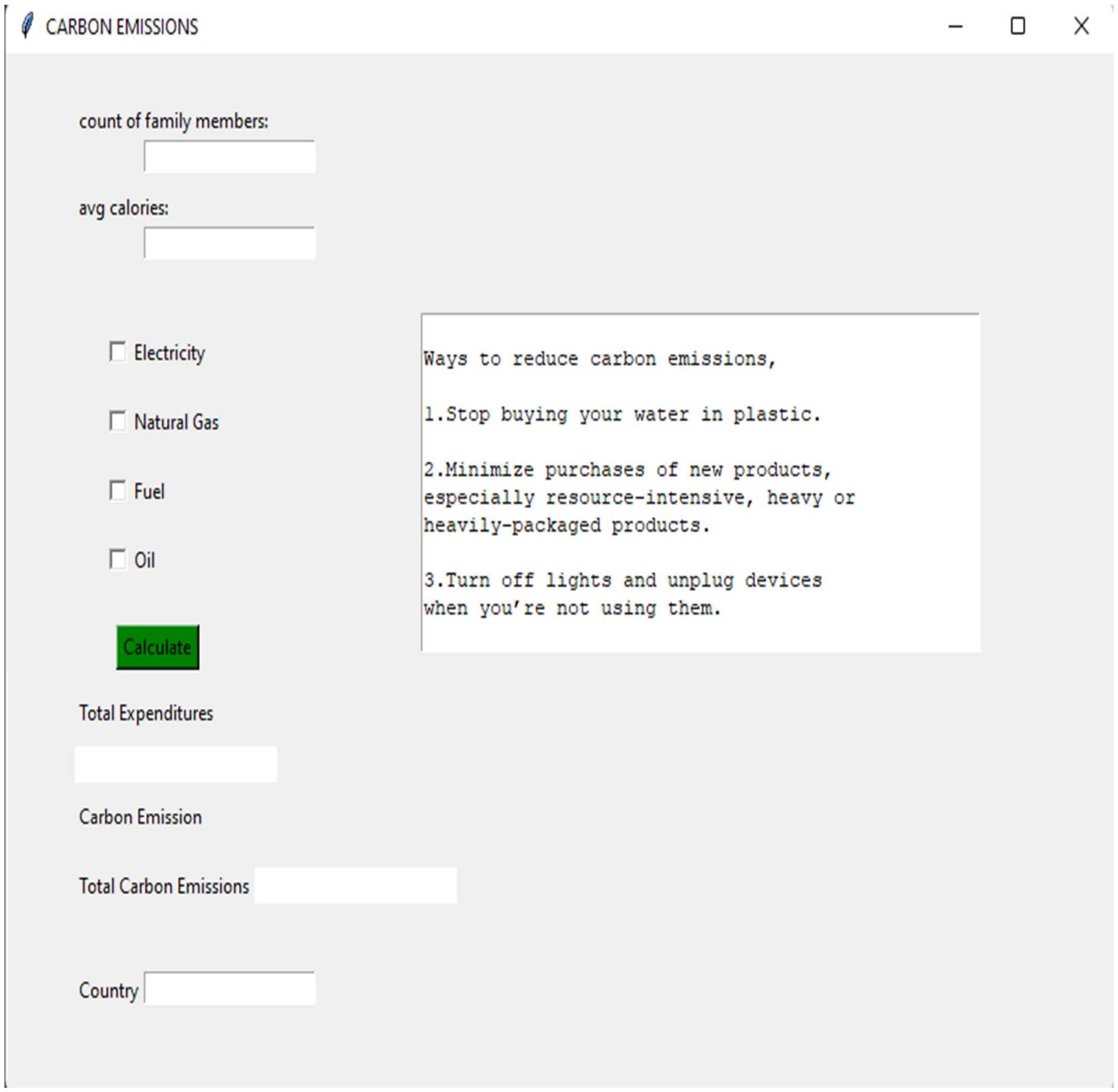
```python
import matplotlib.pyplot as plt
import sqlite3
from tkinter import *
import matplotlib.pyplot as plt
import sqlite3


conn = sqlite3.connect('C:\\Users\\revwr\\Downloads\\Sqlite databases\\rev1.db')
c = conn.cursor()

#Carbon footprint calculator of an country

def create_table():
    c.execute("CREATE TABLE IF NOT EXISTS annual(Country TEXT, Year INTEGER, Annual_CO2_emissions INTEGER)")
"""
def read_from_db():
    c.execute('SELECT * FROM annual')
    data = c.fetchall()
    print(data)
    for row in data:
        print(row)

read_from_db()
"""
```

```
D:\REV\PYTHON\Mini-project\venv\Scripts\python.exe D:/REV/PYTHON/Mini-project/CountryData.py
Enter country's name:Australia
```

8.2 Displaying carbon emission data for given country

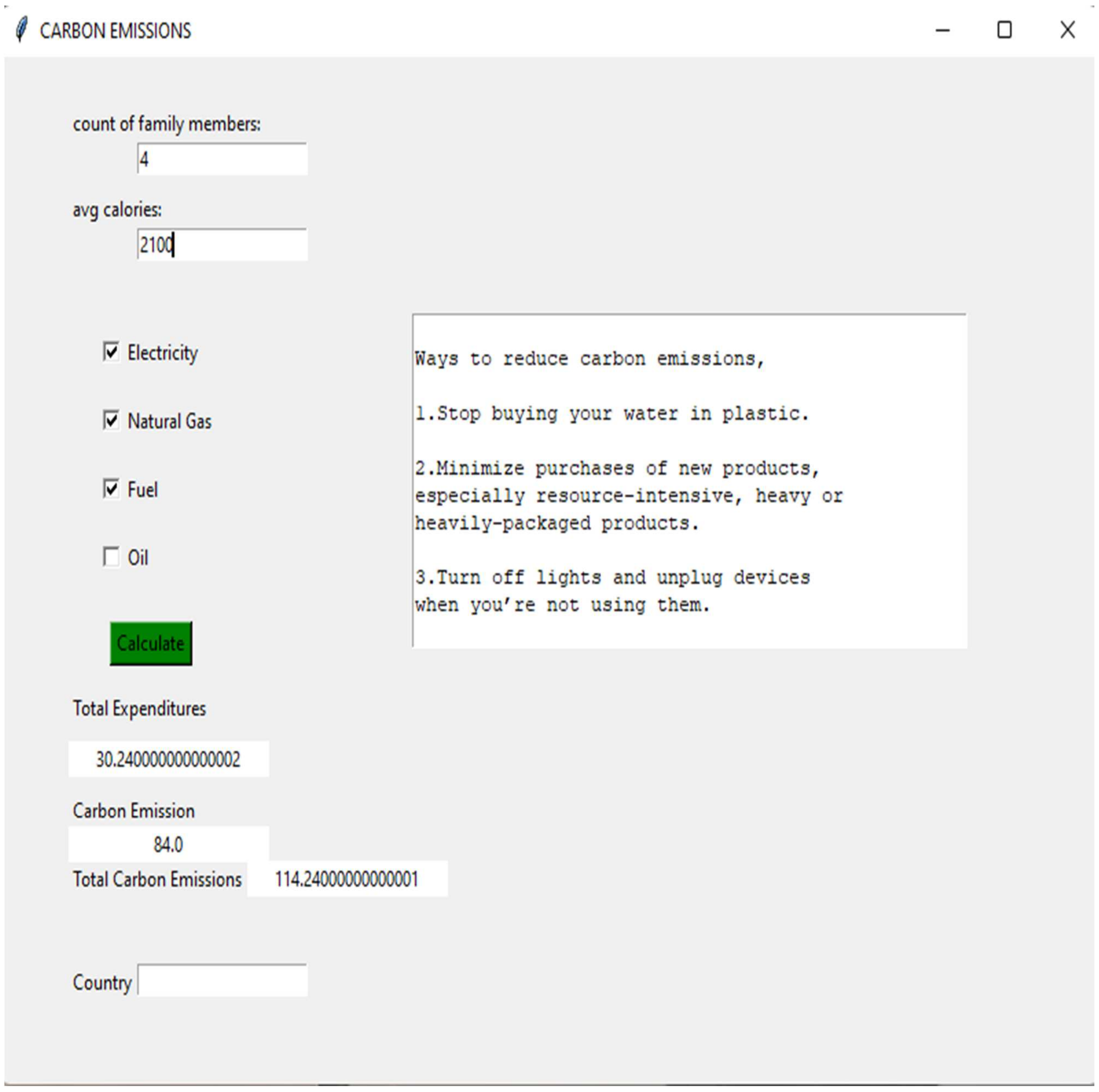


8.3 GUI



```
Ways to reduce carbon emissions,

1.Stop buying your water in plastic.

2.Minimize purchases of new products,
especially resource-intensive, heavy or
heavily-packaged products.

3.Turn off lights and unplug devices
when you’re not using them.
```

8.4 Input data into GUI for carbon emission calculation



count of family members: 4

avg calories: 2100

- [✓] Electricity
- [✓] Natural Gas
- [✓] Fuel
- [ ] Oil

```
Ways to reduce carbon emissions,

1.Stop buying your water in plastic.

2.Minimize purchases of new products,
especially resource-intensive, heavy or
heavily-packaged products.

3.Turn off lights and unplug devices
when you're not using them.
```

Total Expenditures: 30.240000000000002

Carbon Emission: 84.0

Total Carbon Emissions: 114.24000000000001

Country

8.5 Ways to reduce carbon emissions

Ways to reduce carbon emissions,

1.Stop buying your water in plastic.

2.Minimize purchases of new products, especially resource-intensive, heavy or heavily-packaged products.

3.Turn off lights and unplug devices when you're not using them.

4.Buy locally sourced, organic, plant-based, unprocessed foods from local farmers, farmers markets, green restaurants and health food stores. Minimize food waste by planning out meals ahead of time and freezing as much as possible.

5.Reduce water use (buy low flow shower and faucet heads, water efficient toilets/washing machines/dishwashers, check for leaks, buy native drought-tolerant plants, etc.).

6.Recycle as much as possible, even when travelling, and buy products with recycle-able/minimal packaging. Search online for ways to recycle hard-to-recycle items in your local community.

7.Consider installing solar panels on your home.

8.Start a Home Garden.

9.Raise awareness.

10.Use alternative transportation (bus, train, carpool, or bike) to get to work one day per week.

CHAPTER 9

CONCLUSION

Project’s main aim is to spread awareness about carbon emissions, letting every citizen know the carbon levels and how to overcome the crisis by taking the right steps to reduce carbon footprints is the project’s main idea. The project will give details about carbon emissions of the user during his daily life and to overcome or reduce the emissions. It will also provide insight on the carbon emissions in different countries all across the world. Hence, the user can become a responsible citizen by reducing his carbon emissions in his daily life.

REFERENCES

https://www.geeksforgeeks.org

https://www.w3schools.com

https://stackoverflow.com

https://www.tutorialspoint.com

https://www.google.com

https://docs.python.org

https://www.programiz.com