Objectives of project

To create a program that produces the bill of the plants that are sold by the seller in a user friendly graphical interface.

Function description

- **buyrose** Initiate a dialog box to enter the quantity of rose that seller selects and add the rose into the bill with its respective quantity
- **buysunflower** Generate a dialog box that accepts the quantity of sunflower that seller selects and add the sunflower plant into the bill with its respective quantity
- **buyjasmine** Pops a selection box to enter the number of jasmine plant that seller selects and add the jasmine plant into the bill with its respective quantity
- **buyaloevera** Generate a dialog box that accepts the number of aloevera that seller selects and add the aloevera plant into the bill with its respective quantity.
- **buymoneyplant** Opens a selection box to take the amount of moneyplant that seller selects and add the moneyplant into the bill with its respective quantity
- **buyjade** Initiate a dialog box to enter the quantity of jade that seller selects and add the jade into the bill with its respective quantity
- **buyadenium** Generate a dialog box that accepts the quantity of adenium plant that seller selects and add the sunflower plant into the bill with its respective quantity
- **buycactus** Pops a selection box to enter the number of cactus plant that seller selects and add the cactus plant into the bill with its respective quantity
- **buypalm** Initiate a dialog box to enter the quantity of palm that seller selects and add the palm into the bill with its respective quantity
- **gotoflower** Open up another window that contains different varieties of plant in the same domain. It will open up the flower category
- **gotodecoration**This function open up another window that contains different varieties of plant in the same domain. It will open up the category of plant that are usually used for decoration
- **gotodesert** The plants that belong to the desert are selected by this function. It will pop up a window that contains desert plants.
- **billf** Generates the bill of the plants that are sold with the respective quantity, and the total amount corresponding to various plants. It also gives the total amount the buyer has to pay
- **pay** The function which keeps the track of payment. It double conforms if the bill is paid or not.
- **main** It is the runner of the code. It initializes the program.

PROGRAM CODE

```
## import tkinter for GUI support
import tkinter

## importing simpledialog and message box exclusively for simpler use
from tkinter import simpledialog, messagebox

## module for profiling
import cProfile

# module for reading profiled data
import pstats
from pstats import SortKey

## dictionary to keep the quantity of each plant sold
quantity_dictionary = dict()

## dictionary that keep track of plants price
price_dictionary = {"rose":50, "sunflower":40, "jasmine":60, "aloevera":50,
                    "moneyplant":40, "jade":60, "adenium":100, "cactus":300,
                    "palm":200}


## method to sell the plants
def buyrose():
    rose_simple_dialog_box = simpledialog.askinteger("Rose", "Qty.")
    quantity_dictionary["rose"] = rose_simple_dialog_box

## method to sell the plants
def buysunflower():
    sunflower_simple_dialog_box = simpledialog.askinteger("Sunflower", "Qty.")
    quantity_dictionary["sunflower"] = sunflower_simple_dialog_box

## method to sell the plants
def buyjasmine():
    jasmine_simple_dialog_box = simpledialog.askinteger("Jasmine", "Qty.")
    quantity_dictionary["jasmine"] = jasmine_simple_dialog_box

## method to sell the plants
def buyaloevera():
    aloevera_simple_dialog_box = simpledialog.askinteger("Aloe Vera", "Qty.")
    quantity_dictionary["aloevera"] = aloevera_simple_dialog_box

## method to sell the plants
def buymoneyplant():
    moneyplant_simple_dialog_box = simpledialog.askinteger("Money Plant",
                                                           "Qty.")
    quantity_dictionary["moneyplant"] = moneyplant_simple_dialog_box

## method to sell the plants
def buyjade():
    jade_simple_dialog_box = simpledialog.askinteger("Jade", "Qty.")
    quantity_dictionary["jade"] = jade_simple_dialog_box
```

```
## method to sell the plants
def buyadenium():
    adenium_simple_dialog_box = simpledialog.askinteger("Adenium", "Qty.")
    quantity_dictionary["adenium"] = adenium_simple_dialog_box

## method to sell the plants
def buycactus():
    cactus_simple_dialog_box = simpledialog.askinteger("Cactus", "Qty.")
    quantity_dictionary["cactus"] = cactus_simple_dialog_box


# method to sell the plants
def buypalm():
    palm_simple_dialog_box = simpledialog.askinteger("Palm", "Qty.")
    quantity_dictionary["palm"] = palm_simple_dialog_box




# The given function will be assigned to the command of flower button
# Hence it will be called when button is clicked
def gotoflower():

    # Structure of the window that appears after entering the flower
    # section
    flower_window = tkinter.Toplevel()
    flower_window.title("Flowers")
    flower_window.config(bg = "white")
    flower_window.geometry("500x500+150+150")

    flowerlabel = tkinter.Label(flower_window, text = "Flowers",
                                font = ("algerian", 25),
                                image = fphoto, compound = "left", bg = "white",
                                height = 100, width = 500).grid(row = 0, column = 0,
                                                                columnspan = 2)


    ## Buttons denoting the plant name to be selected
    rose = tkinter.Button(flower_window, text = "Rose", image = fphoto,
                          height = 100,width = 300, font = ("algerian", 25),
                          compound = "left", bg = "white",
                          command = buyrose).grid(row = 1, column = 0)


    priceboxrose = tkinter.Label(flower_window, text = ("‘",
                                 price_dictionary["rose"]),
                                 font = ("indian rupee", 25),
                                 bg = "white").grid(row = 1, column = 1)

    ## Buttons denoting the plant name to be selected
    sunflower = tkinter.Button(flower_window, text = "sunflower",
                               image = fphoto,
                               height = 100, width = 300, font = ("algerian", 25),
                               compound = "left", bg = "white",
                               command = buysunflower).grid(row = 2, column = 0)
```

```
    priceboxsunflower = tkinter.Label(flower_window, text = ("‘",
                                      price_dictionary["sunflower"]),
                                      font = ("indian rupee", 25),
                                      bg = "white").grid(row = 2, column = 1)

    ## Buttons denoting the plant name to be selected
    jasmine = tkinter.Button(flower_window, text = "jasmine", image = fphoto,
                             height = 100, width = 300, font = ("algerian", 25),
                             compound = "left", bg = "white",
                             command = buyjasmine).grid(row = 3, column = 0)

    priceboxjasmine = tkinter.Label(flower_window, text = ("‘",
                                    price_dictionary["jasmine"]),
                                    font = ("indian rupee", 25),
                                    bg = "white").grid(row = 3, column = 1)

# The given function will be assigned to the command of decoration button
# Hence it will be called when button is clicked
def gotodecoration():
    # Structure of the window that appears after entering the flower
    # section
    decoration_window = tkinter.Toplevel()
    decoration_window.title("Dedoration Plants")
    decoration_window.config(bg = "white")
    decoration_window.geometry("500x500+150+150")

    decorationlabel = tkinter.Label(decoration_window,
                                    text = "Decoration Plants",
                                    font = ("algerian", 25), image = fphoto,
                                    compound = "left", bg = "white", height = 100,
                                    width = 500).grid(row = 0,
                                    column = 0, columnspan = 2)


    ## Buttons denoting the plant name to be selected
    aloevera = tkinter.Button(decoration_window, text = "Aloe vera",
                              image = fphoto, height = 100, width = 300,
                              font = ("algerian", 25), compound = "left",
                              bg = "white", command = buyaloevera).grid(row = 1,
                                                                        column = 0)

    priceboxaloevera = tkinter.Label(decoration_window, text = ("‘",
                                     price_dictionary["aloevera"]),
                                     font = ("indian rupee", 25),
                                     bg = "white").grid(row = 1, column = 1)

    ## Buttons denoting the plant name to be selected
    moneyplant = tkinter.Button(decoration_window, text = "Money Plant",
                                image = fphoto, height = 100, width = 300,
                                font = ("algerian", 25), compound = "left",
                                bg = "white",
                                command = buymoneyplant).grid(row = 2,
                                                              column = 0)

    priceboxmoneyplant = tkinter.Label(decoration_window, text = ("‘",
```

```
                                        price_dictionary["moneyplant"]),
                                        font = ("indian rupee", 25),
                                        bg = "white").grid(row = 2, column = 1)

    ## Buttons denoting the plant name to be selected
    jade = tkinter.Button(decoration_window, text = "Jade", image = fphoto,
                          height = 100, width = 300, font = ("algerian", 25),
                          compound = "left", bg = "white",
                          command = buyjade).grid(row = 3, column = 0)

    priceboxjade = tkinter.Label(decoration_window, text = ("`",
                                 price_dictionary["jade"]),
                                 font = ("indian rupee", 25),
                                 bg = "white").grid(row = 3, column = 1)


# The given function will be assigned to the command of desert button
# Hence it will be called when button is clicked
def gotodesert():
    # Structure of the window that appears after entering the flower
    # section
    decoration_window = tkinter.Toplevel()
    decoration_window.title("Desert plants")
    decoration_window.config(bg = "white")
    decoration_window.geometry("500x500+150+150")

    desertlabel = tkinter.Label(decoration_window, text = "Desert Plants",
                                font = ("algerian", 25), image = fphoto,
                                compound = "left", bg = "white", height = 100,
                                width = 500).grid(row = 0, column = 0,
                                                  columnspan = 2)


    ## Buttons denoting the plant name to be selected
    adenium = tkinter.Button(decoration_window, text = "Adenium",
                             image = fphoto,
                             height = 100, width = 300, font = ("algerian", 25),
                             compound = "left", bg = "white",
                             command = buyadenium).grid(row = 1, column = 0)

    priceboxadenium = tkinter.Label(decoration_window, text = ("`",
                                    price_dictionary["adenium"]),
                                    font = ("indian rupee", 25),
                                    bg = "white").grid(row = 1, column = 1)

    ## Buttons denoting the plant name to be selected
    cactus = tkinter.Button(decoration_window, text = "Cactus", image = fphoto,
                            height = 100, width = 300, font = ("algerian", 25),
                            compound = "left", bg = "white",
                            command = buycactus).grid(row = 2, column = 0)

    priceboxcactus = tkinter.Label(decoration_window, text = ("`",
                                   price_dictionary["cactus"]),
                                   font = ("indian rupee", 25),
                                   bg = "white").grid(row = 2, column = 1)
```

```
    ## Buttons denoting the plant name to be selected
    palm = tkinter.Button(decoration_window, text = "Palm", image = fphoto,
                          height = 100, width = 300, font = ("algerian", 25),
                          compound = "left", bg = "white",
                          command = buypalm).grid(row = 3, column = 0)

    priceboxpalm = tkinter.Label(decoration_window, text = ("`",
                                 price_dictionary["palm"]),
                                 font = ("indian rupee", 25),
                                 bg = "white").grid(row = 3, column = 1)


## The billf function will generate the bill
## and is assigned to the command argument of bill button
def billf():
    ## bill window
    bill_window = tkinter.Toplevel()
    bill_window.title("BILL")
    bill_window.config(bg = "white")
    bill_window.geometry("600x650+150+0")


    ## The structure of the bill
    billh = tkinter.Label(bill_window, text = "XYZ Nursery",
                          font = ("agency fb", 20, "bold"),
                          bg = "white").grid(row = 0, column = 1,
                                             columnspan = 5)

    address = tkinter.Label(bill_window, text = "MR 2, NEAR PQY PARK, AB ROAD",
                            font = ("agency fb", 15),
                            bg = "white").grid(row = 1, column = 1, columnspan = 5)

    mobno = tkinter.Label(bill_window, text = "Mob : 98765 _____",
                          font = ("agency fb", 15),
                          bg = "white").grid(row = 2, column = 1, columnspan = 5)

    email = tkinter.Label(bill_window, text = "E Mail : xyznursery@gmail.com",
                          font = ("agency fb", 15),
                          bg = "white").grid(row = 3, column = 1, columnspan = 5)

    cashsalesinvoice = tkinter.Label(bill_window, text = "CASH SALES INVOICE",
                                     font = ("agency fb", 15, "bold"),
                                     bg = "white").grid(row = 4, column = 1,
                                                        columnspan = 5)

    sno = tkinter.Label(bill_window, text = "{0:^5s}".format("S.No"),
                        font = ("indian rupee", 15), bg = "white", borderwidth = 2,
                        relief = "groove", width = 5).grid(row = 5, column = 0)

    description = tkinter.Label(bill_window, text = "{0:^40s}"
                                .format("Description of goods"),
                                font = ("indian rupee", 15), bg = "white",
                                borderwidth = 2, relief = "groove",
                                width = 23).grid(row = 5, column = 1)

    quantity = tkinter.Label(bill_window, text = "{0:^15s}".format("Quantity"),
```

```
                        font = ("indian rupee", 15), bg = "white",
                        borderwidth = 2, relief = "groove",
                        width = 8).grid(row = 5, column = 2)

    rate = tkinter.Label(bill_window, text = "{0:^10s}".format("Rate"),
                         font = ("indian rupee", 15), bg = "white",
                         borderwidth = 2,
                         relief = "groove", width = 8).grid(row = 5, column = 3)

    amount = tkinter.Label(bill_window, text = "{0:^10s}".format("Amount"),
                           font = ("indian rupee", 15), bg = "white",
                           borderwidth = 2,
                           relief = "groove", width = 8).grid(row = 5, column = 4)



    row_number = 1
    total_amount = 0
    total_quantity = 0


    ## Conditon to add the plants that are sold to the bill
    if "rose" in quantity_dictionary:

        snor = tkinter.Label(bill_window, text = "{0:^5d}".format(row_number),
                             font = ("indian rupee", 15), bg = "white",
                             borderwidth = 2, relief = "groove",
                             width = 5).grid(row = row_number+6, column = 0)

        descriptionr = tkinter.Label(bill_window, text = "{0:^40s}"
                                     .format("Rose"), font = ("indian rupee", 15),
                                     bg = "white", borderwidth = 2,
                                     relief = "groove",
                                     width = 23).grid(row = row_number+6,
                                     column = 1)

        quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                                  .format(quantity_dictionary["rose"]),
                                  font = ("indian rupee", 15), bg = "white",
                                  borderwidth = 2, relief = "groove",
                                  width = 8).grid(row = row_number+6, column = 2)

        rater = tkinter.Label(bill_window, text = "{0:^10d}"
                              .format(price_dictionary["rose"]),
                              font = ("indian rupee", 15), bg = "white",
                              borderwidth = 2, relief = "groove",
                              width = 8).grid(row = row_number+6, column = 3)

        amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                                .format(quantity_dictionary["rose"]
                                *price_dictionary["rose"]),
                                font = ("indian rupee", 15), bg = "white",
                                borderwidth = 2, relief = "groove",
                                width = 8).grid(row = row_number+6, column = 4)

        row_number += 1
```

```
    total_amount += quantity_dictionary["rose"]*price_dictionary["rose"]
    total_quantity += quantity_dictionary["rose"]


## Conditon to add the plants that are sold to the bill
if "sunflower" in quantity_dictionary:
    snor = tkinter.Label(bill_window, text = "{0:^5d}".format(row_number),
                         font = ("indian rupee", 15), bg = "white",
                         borderwidth = 2, relief = "groove",
                         width = 5).grid(row = row_number+6, column = 0)

    descriptionr = tkinter.Label(bill_window,
                                 text = "{0:^40s}".format("Sunflower"),
                                font = ("indian rupee", 15), bg = "white",
                                borderwidth = 2, relief = "groove",
                                width = 23).grid(row = row_number+6,
                                column = 1)

    quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                              .format(quantity_dictionary["sunflower"]),
                              font = ("indian rupee", 15), bg = "white",
                              borderwidth = 2, relief = "groove",
                              width = 8).grid(row = row_number+6, column = 2)

    rater = tkinter.Label(bill_window, text = "{0:^10d}"
                          .format(price_dictionary["sunflower"]),
                          font = ("indian rupee", 15), bg = "white",
                          borderwidth = 2, relief = "groove",
                          width = 8).grid(row = row_number+6, column = 3)

    amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                            .format(quantity_dictionary["sunflower"]
                            *price_dictionary["sunflower"]),
                            font = ("indian rupee", 15), bg = "white",
                            borderwidth = 2, relief = "groove",
                            width = 8).grid(row = row_number+6, column = 4)
    row_number += 1
    total_amount +=   quantity_dictionary["sunflower"] \
                    * price_dictionary["sunflower"]

    total_quantity += quantity_dictionary["sunflower"]



## Conditon to add the plants that are sold to the bill
if "jasmine" in quantity_dictionary:
    snor = tkinter.Label(bill_window, text = "{0:^5d}".format(row_number),
                         font = ("indian rupee", 15), bg = "white",
                         borderwidth = 2, relief = "groove",
                         width = 5).grid(row = row_number+6, column = 0)

    descriptionr = tkinter.Label(bill_window,
                                 text = "{0:^40s}".format("Jasmine"),
                                 font = ("indian rupee", 15), bg = "white",
                                 borderwidth = 2, relief = "groove",
                                 width = 23).grid(row = row_number+6,
```

```
                                    column = 1)

        quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                                  .format(quantity_dictionary["jasmine"]),
                                  font = ("indian rupee", 15), bg = "white",
                                  borderwidth = 2, relief = "groove",
                                  width = 8).grid(row = row_number+6, column = 2)

        rater = tkinter.Label(bill_window, text = "{0:^10d}"
                              .format(price_dictionary["jasmine"]),
                              font = ("indian rupee", 15), bg = "white",
                              borderwidth = 2, relief = "groove",
                              width = 8).grid(row = row_number+6, column = 3)

        amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                              .format(quantity_dictionary["jasmine"]
                              *price_dictionary["jasmine"]),
                              font = ("indian rupee", 15), bg = "white",
                              borderwidth = 2, relief = "groove",
                              width = 8).grid(row = row_number+6, column = 4)
        row_number += 1
        total_amount += quantity_dictionary["jasmine"] \
                        * price_dictionary["jasmine"]
        total_quantity += quantity_dictionary["jasmine"]


    ## Conditon to add the plants that are sold to the bill
    if "aloevera" in quantity_dictionary:
        snor = tkinter.Label(bill_window, text = "{0:^5d}"
                              .format(row_number), font = ("indian rupee", 15),
                              bg = "white", borderwidth = 2, relief = "groove",
                              width = 5).grid(row = row_number+6, column = 0)

        descriptionr = tkinter.Label(bill_window, text = "{0:^40s}"
                                     .format("Aloe Vera"),
                                     font = ("indian rupee", 15), bg = "white",
                                     borderwidth = 2, relief = "groove",
                                     width = 23).grid(row = row_number+6,
                                     column = 1)

        quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                                  .format(quantity_dictionary["aloevera"]),
                                  font = ("indian rupee", 15), bg = "white",
                                  borderwidth = 2, relief = "groove",
                                  width = 8).grid(row = row_number+6, column = 2)

        rater = tkinter.Label(bill_window, text = "{0:^10d}"
                              .format(price_dictionary["aloevera"]),
                              font = ("indian rupee", 15), bg = "white",
                              borderwidth = 2, relief = "groove",
                              width = 8).grid(row = row_number+6, column = 3)

        amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                                .format(quantity_dictionary["aloevera"]
                                *price_dictionary["aloevera"]),
                                font = ("indian rupee", 15), bg = "white",
```

```
                                borderwidth = 2, relief = "groove",
                                width = 8).grid(row = row_number+6, column = 4)
    row_number += 1
    total_amount +=   quantity_dictionary["aloevera"] \
                    * price_dictionary["aloevera"]
    total_quantity += quantity_dictionary["aloevera"]


## Conditon to add the plants that are sold to the bill
if "moneyplant" in quantity_dictionary:

    snor = tkinter.Label(bill_window, text = "{0:^5d}"
                         .format(row_number), font = ("indian rupee", 15),
                         bg = "white", borderwidth = 2, relief = "groove",
                         width = 5).grid(row = row_number+6, column = 0)

    descriptionr = tkinter.Label(bill_window, text = "{0:^40s}"
                                 .format("Money Plant"),
                                 font = ("indian rupee", 15), bg = "white",
                                 borderwidth = 2, relief = "groove",
                                 width = 23).grid(row = row_number+6,
                                 column = 1)

    quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                              .format(quantity_dictionary
                              ["moneyplant"]), font = ("indian rupee", 15),
                              bg = "white", borderwidth = 2,
                              relief = "groove",
                              width = 8).grid(row = row_number+6, column = 2)

    rater = tkinter.Label(bill_window, text = "{0:^10d}"
                          .format(price_dictionary["moneyplant"]),
                          font = ("indian rupee", 15), bg = "white",
                          borderwidth = 2, relief = "groove",
                          width = 8).grid(row = row_number+6, column = 3)

    amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                                .format(quantity_dictionary["moneyplant"]
                                *price_dictionary["moneyplant"]),
                                font = ("indian rupee", 15), bg = "white",
                                borderwidth = 2, relief = "groove",
                                width = 8).grid(row = row_number+6, column = 4)
    row_number += 1
    total_amount +=   quantity_dictionary["moneyplant"]\
                    * price_dictionary["moneyplant"]
    total_quantity += quantity_dictionary["moneyplant"]


## Conditon to add the plants that are sold to the bill
if "jade" in quantity_dictionary:
    snor = tkinter.Label(bill_window, text = "{0:^5d}"
                         .format(row_number), font = ("indian rupee", 15),
                         bg = "white", borderwidth = 2, relief = "groove",
                         width = 5).grid(row = row_number+6, column = 0)

    descriptionr = tkinter.Label(bill_window, text = "{0:^40s}"
```

```
                                    .format("Jade"), font = ("indian rupee",
                                    15), bg = "white", borderwidth = 2,
                                    relief = "groove",
                                    width = 23).grid(row = row_number+6,
                                    column = 1)

        quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                    .format(quantity_dictionary["jade"]),
                    font = ("indian rupee", 15), bg = "white", borderwidth = 2,
                    relief = "groove",
                    width = 8).grid(row = row_number+6, column = 2)

        rater = tkinter.Label(bill_window, text = "{0:^10d}"
                            .format(price_dictionary["jade"]),
                            font = ("indian rupee", 15), bg = "white",
                            borderwidth = 2, relief = "groove",
                            width = 8).grid(row = row_number+6, column = 3)

        amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                                .format(quantity_dictionary["jade"]
                                *price_dictionary["jade"]),
                                font = ("indian rupee", 15), bg = "white",
                                borderwidth = 2, relief = "groove",
                                width = 8).grid(row = row_number+6, column = 4)
        row_number +=   1
        total_amount +=   quantity_dictionary["jade"]   \
                        * price_dictionary["jade"]
        total_quantity +=   quantity_dictionary["jade"]


    ## Conditon to add the plants that are sold to the bill
    if "adenium" in quantity_dictionary:
        snor = tkinter.Label(bill_window, text = "{0:^5d}"
                            .format(row_number), font = ("indian rupee", 15),
                            bg = "white", borderwidth = 2, relief = "groove",
                            width = 5).grid(row = row_number+6, column = 0)

        descriptionr = tkinter.Label(bill_window, text = "{0:^40s}"
                                    .format("Adenium"),
                                    font = ("indian rupee", 15), bg = "white",
                                    borderwidth = 2, relief = "groove",
                                    width = 23).grid(row = row_number+6,
                                    column = 1)

        quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                                .format(quantity_dictionary["adenium"]),
                                font = ("indian rupee", 15), bg = "white",
                                borderwidth = 2, relief = "groove",
                                width = 8).grid(row = row_number+6, column = 2)

        rater = tkinter.Label(bill_window, text = "{0:^10d}"
                            .format(price_dictionary["adenium"]),
                            font = ("indian rupee", 15), bg = "white",
                            borderwidth = 2, relief = "groove",
                            width = 8).grid(row = row_number+6, column = 3)
```

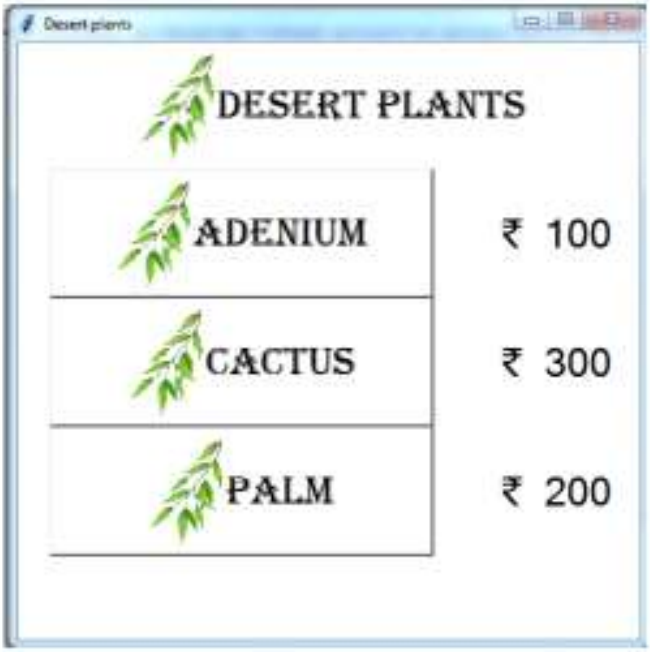
```
    amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                            .format(quantity_dictionary["adenium"]
                            *price_dictionary["adenium"]),
                            font = ("indian rupee", 15), bg = "white",
                            borderwidth = 2, relief = "groove",
                            width = 8).grid(row = row_number+6, column = 4)
    row_number += 1
    total_amount +=   quantity_dictionary["adenium"]\
                    * price_dictionary["adenium"]
    total_quantity += quantity_dictionary["adenium"]


## Conditon to add the plants that are sold to the bill
if "cactus" in quantity_dictionary:
    snor = tkinter.Label(bill_window, text = "{0:^5d}"
                         .format(row_number), font = ("indian rupee", 15),
                         bg = "white", borderwidth = 2, relief = "groove",
                         width = 5).grid(row = row_number+6, column = 0)

    descriptionr = tkinter.Label(bill_window, text = "{0:^40s}"
                                 .format("Cactus"),
                                 font = ("indian rupee", 15), bg = "white",
                                 borderwidth = 2, relief = "groove",
                                 width = 23).grid(row = row_number+6,
                                 column = 1)

    quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                                 .format(quantity_dictionary["cactus"]),
                                 font = ("indian rupee", 15), bg = "white",
                                 borderwidth = 2, relief = "groove",
                                 width = 8).grid(row = row_number+6,
                                 column = 2)

    rater = tkinter.Label(bill_window, text = "{0:^10d}"
                         .format(price_dictionary["cactus"]),
                         font = ("indian rupee", 15), bg = "white",
                         borderwidth = 2, relief = "groove",
                         width = 8).grid(row = row_number+6, column = 3)

    amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                            .format(quantity_dictionary["cactus"]
                            *price_dictionary["cactus"]),
                            font = ("indian rupee", 15), bg = "white",
                            borderwidth = 2, relief = "groove",
                            width = 8).grid(row = row_number+6, column = 4)
    row_number +=   1
    total_amount +=   quantity_dictionary["cactus"]\
                  * price_dictionary["cactus"]
    total_quantity += quantity_dictionary["cactus"]


## Conditon to add the plants that are sold to the bill
if "palm" in quantity_dictionary:
    snor = tkinter.Label(bill_window, text = "{0:^5d}"
                         .format(row_number),
                         font = ("indian rupee", 15), bg = "white",
```

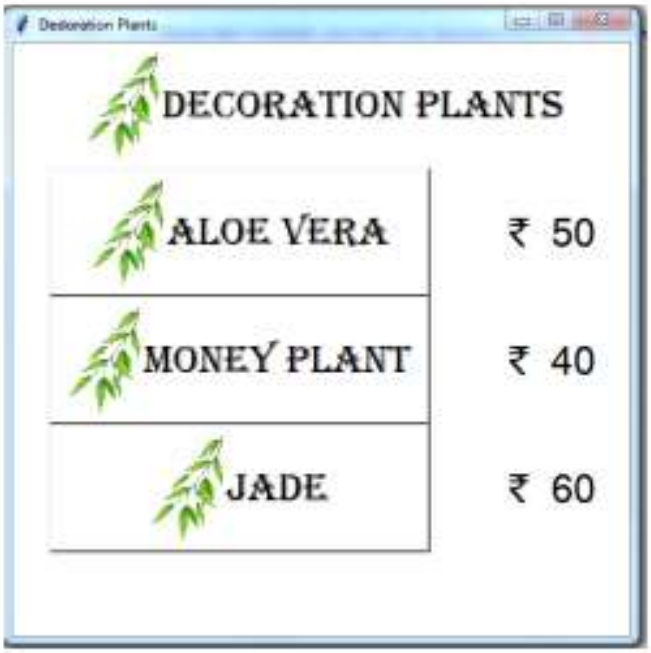
```
                        borderwidth = 2, relief = "groove",
                        width = 5).grid(row = row_number+6, column = 0)

    descriptionr = tkinter.Label(bill_window, text = "{0:^40s}"
                                .format("Palm"),
                                font = ("indian rupee", 15), bg = "white",
                                borderwidth = 2, relief = "groove",
                                width = 23).grid(row = row_number+6,
                                column = 1)

    quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                            .format(quantity_dictionary["palm"]),
                            font = ("indian rupee", 15), bg = "white",
                            borderwidth = 2, relief = "groove",
                            width = 8).grid(row = row_number+6, column = 2)

    rater = tkinter.Label(bill_window, text = "{0:^10d}"
                        .format(price_dictionary["palm"]),
                        font = ("indian rupee", 15), bg = "white",
                        borderwidth = 2, relief = "groove",
                        width = 8).grid(row = row_number+6, column = 3)

    amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                            .format(quantity_dictionary["palm"]
                            *price_dictionary["palm"]),
                            font = ("indian rupee", 15), bg = "white",
                            borderwidth = 2, relief = "groove",
                            width = 8).grid(row = row_number+6, column = 4)
    row_number +=   1
    total_amount +=   quantity_dictionary["palm"]\
                    * price_dictionary["palm"]
    total_quantity +=   quantity_dictionary["palm"]




## calculate the total of the bill
total = tkinter.Label(bill_window, text = "{0:^40s}"
                    .format("Total"), font = ("indian rupee", 15),
                    bg = "white", borderwidth = 2, relief = "groove",
                    width = 23).grid(row = row_number+6, column = 1)

quantityr = tkinter.Label(bill_window, text = "{0:^15d}"
                        .format(total_quantity),
                        font = ("indian rupee", 15), bg = "white",
                        borderwidth = 2, relief = "groove",
                        width = 8).grid(row = row_number+6, column = 2)

amountr = tkinter.Label(bill_window, text = "{0:^10d}"
                        .format(total_amount),
                        font = ("indian rupee", 15), bg = "white",
                        borderwidth = 2, relief = "groove",
                        width = 8).grid(row = row_number+6, column = 4)


def pay():
```

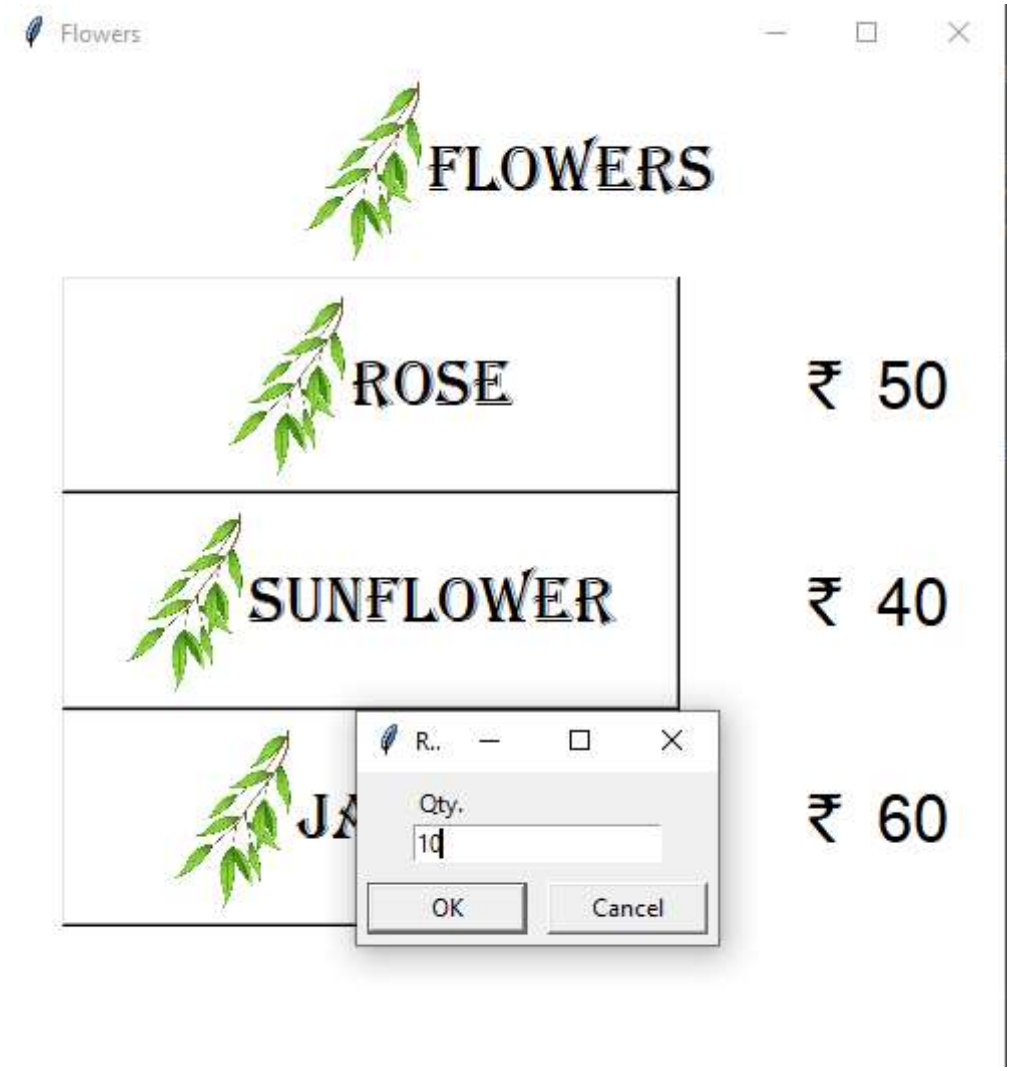
```
        simpledialog_box_ask_yes_no = messagebox.askyesno("", "Do you really \
                                                want to recieve transaction.")
        if simpledialog_box_ask_yes_no:
            messageboxpaydone = messagebox.showinfo("Thankyou", "Payment \
                                                    Successfull")

    pay_button = tkinter.Button(bill_window, text = "Recieve",
                                font = ("algerian", 15), bg = "white",
                                borderwidth = 3, relief = "raised",
                                command = pay).place(x = 220, y = 600)



## The runner of the code
def main():

    img = tkinter.Label(main_window, image = fphoto, compound = "left",
                        bg = "white", height = 100,
                        width = 500).grid(row = 0, column = 0)

    flower = tkinter.Button(main_window, text = "flowery plant",
                            font = ("algerian", 25), bg = "white", width = 24,
                            height = 2, command = gotoflower).grid(row = 1,
                            column = 0)

    decoration = tkinter.Button(main_window, text = "decoration plant",
                                font = ("algerian", 25), bg = "white",
                                 width = 24,
                                height = 2,
                                command = gotodecoration).grid(row = 2,
                                column = 0)

    desert = tkinter.Button(main_window, text = "desert plant",
                            font = ("algerian", 25), bg = "white", width = 24,
                            height = 2, command = gotodesert).grid(row = 3,
                            column = 0)

    space = tkinter.Label(main_window, bg = "white", width = 24,
                          height = 2).grid(row = 4, column = 0)

    bill = tkinter.Button(main_window, text = "Bill", font = ("algerian", 15,
                        "bold"), bg = "white", command = billf).grid(row = 5,
                        column = 0)
    main_window.mainloop()

## The block that will be run by default
if __name__ == "__main__":
    main_window = tkinter.Tk()
    main_window.title("Nursery")
    main_window.config(bg = "white")
    main_window.geometry("500x500+150+150")
    photo = tkinter.PhotoImage(file = "photo.png")
    fphoto = photo.subsample(3, 3)

    ## module function for profiling
```

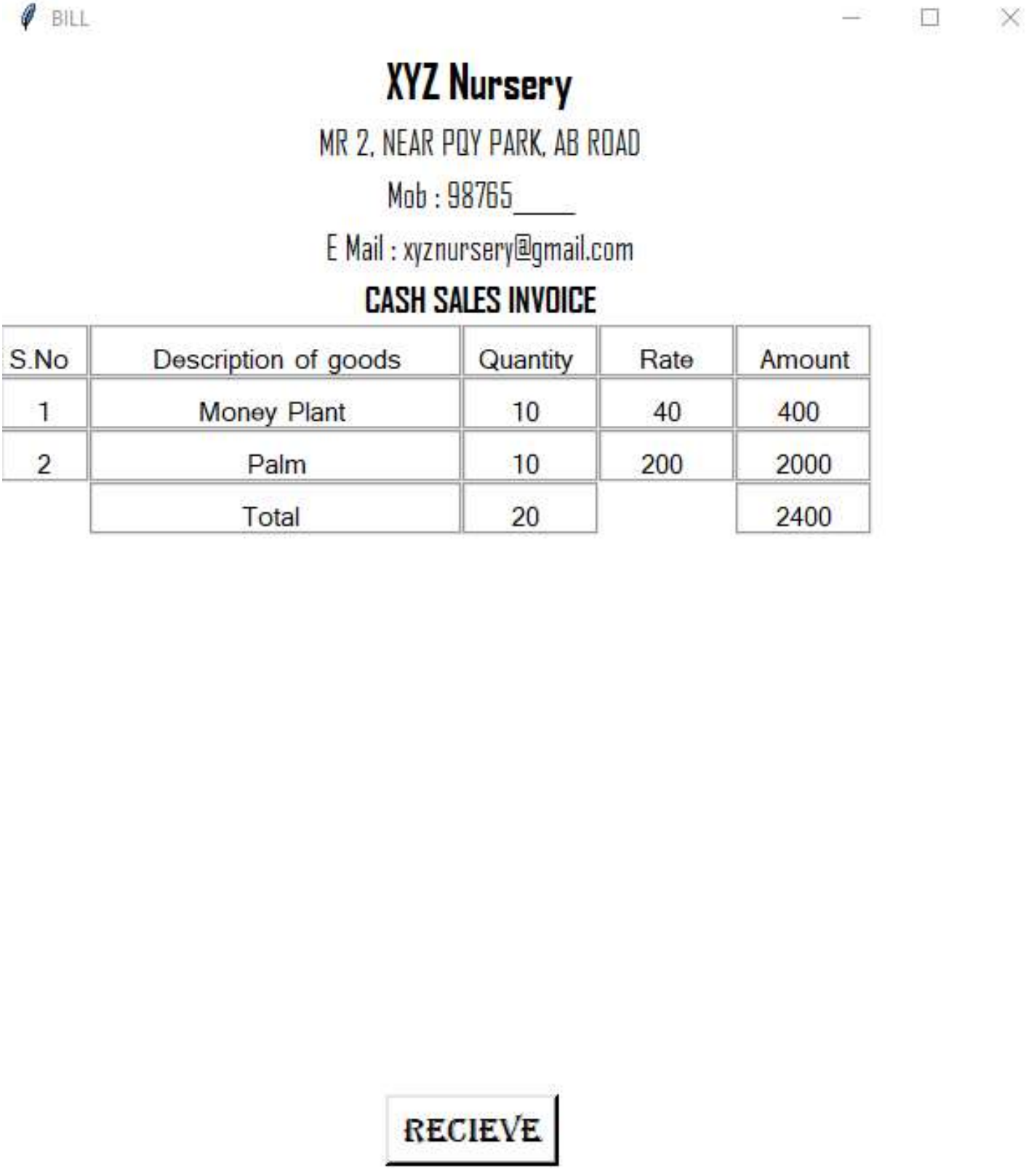
```
cProfile.run("main()", "output.dat")



## creating the output file that contains the profiled data
with open("output_time.txt", "w") as f:
    p = pstats.Stats("output.dat", stream = f)
    p.sort_stats("time").print_stats()
```

PROGRAM OUTPUT






Flowers
FLOWERS
ROSE
₹ 50
SUNFLOWER
₹ 40
₹ 60
R..
Qty.
10
OK
Cancel


BILL
XYZ Nursery
MR 2, NEAR PQY PARK, AB ROAD
Mob : 98765____
E Mail : xyznursery@gmail.com
CASH SALES INVOICE
S.No
Description of goods
Quantity
Rate
Amount
1
Money Plant
10
40
400
2
Palm
10
200
2000
Total
20
2400
RECIEVE

PROFILLING DATA

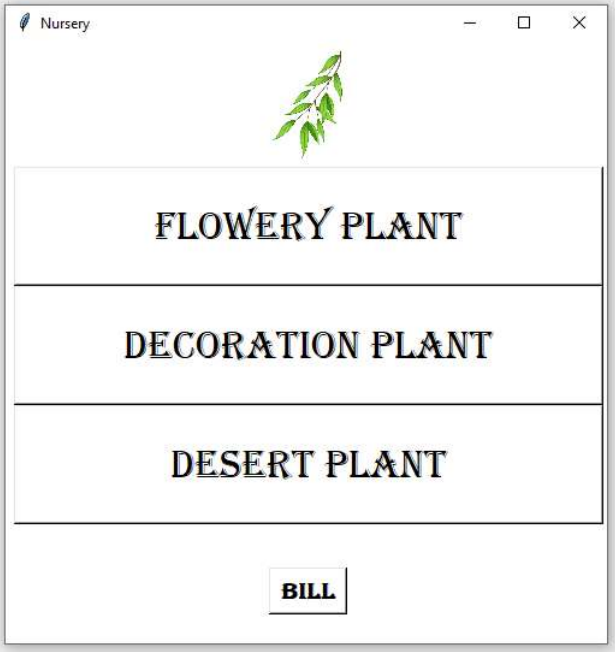
```
Sun Nov 13 18:25:31 2022    output.dat

         392 function calls in 0.991 seconds

   Ordered by: internal time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.964    0.964    0.984    0.984 {method 'mainloop' of '_tkinter.tkapp' objects}
       19    0.027    0.001    0.027    0.001 {method 'call' of '_tkinter.tkapp' objects}
       12    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:1497(_opt
        6    0.000    0.000    0.002    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2606(dest
        1    0.000    0.000    0.991    0.991 i:\##VISHESH##\pp python project\bill_generator.py:684(main)
        1    0.000    0.000    0.020    0.020 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2337(dest
        1    0.000    0.000    0.991    0.991 {built-in method builtins.exec}
        6    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2559(_set
        6    0.000    0.000    0.007    0.001 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2589(__in
       18    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:101(_cnfm
        7    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:657(destr
        4    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:1542(_reg
        7    0.000    0.000    0.000    0.000 {method 'deletecommand' of '_tkinter.tkapp' objects}
      117    0.000    0.000    0.000    0.000 {built-in method builtins.isinstance}
        5    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:61(_strin
        6    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2507(grid
       10    0.000    0.000    0.000    0.000 {method 'search' of 're.Pattern' objects}
        1    0.000    0.000    0.020    0.020 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:1916(__ca
        4    0.000    0.000    0.002    0.001 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2660(__in
       24    0.000    0.000    0.000    0.000 {method 'update' of 'dict' objects}
        6    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2598(<lis
        4    0.000    0.000    0.000    0.000 {method 'createcommand' of '_tkinter.tkapp' objects}
       12    0.000    0.000    0.000    0.000 {built-in method _tkinter._flatten}
        2    0.000    0.000    0.005    0.002 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:3159(__in
       42    0.000    0.000    0.000    0.000 {built-in method builtins.callable}
        1    0.000    0.000    0.984    0.984 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:1456(main
        7    0.000    0.000    0.000    0.000 {method 'values' of 'dict' objects}
        4    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:1910(__in
       18    0.000    0.000    0.000    0.000 {method 'items' of 'dict' objects}
        1    0.000    0.000    0.991    0.991 i:\##VISHESH##\pp python project\bill_generator.py:684(main)
        1    0.000    0.000    0.020    0.020 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2337(dest
        1    0.000    0.000    0.991    0.991 {built-in method builtins.exec}
        6    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2559(_set
        6    0.000    0.000    0.007    0.001 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2589(__in
       18    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:101(_cnfm
        7    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:657(destr
        4    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:1542(_reg
        7    0.000    0.000    0.000    0.000 {method 'deletecommand' of '_tkinter.tkapp' objects}
      117    0.000    0.000    0.000    0.000 {built-in method builtins.isinstance}
        5    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:61(_strin
        6    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2507(grid
       10    0.000    0.000    0.000    0.000 {method 'search' of 're.Pattern' objects}
        1    0.000    0.000    0.020    0.020 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:1916(__ca
        4    0.000    0.000    0.002    0.001 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2660(__in
       24    0.000    0.000    0.000    0.000 {method 'update' of 'dict' objects}
        6    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:2598(<lis
        4    0.000    0.000    0.000    0.000 {method 'createcommand' of '_tkinter.tkapp' objects}
       12    0.000    0.000    0.000    0.000 {built-in method _tkinter._flatten}
        2    0.000    0.000    0.005    0.002 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:3159(__in
       42    0.000    0.000    0.000    0.000 {built-in method builtins.callable}
        1    0.000    0.000    0.984    0.984 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:1456(main
        7    0.000    0.000    0.000    0.000 {method 'values' of 'dict' objects}
        4    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:1910(__in
       18    0.000    0.000    0.000    0.000 {method 'items' of 'dict' objects}
        1    0.000    0.000    0.991    0.991 <string>:1(<module>)
        6    0.000    0.000    0.000    0.000 {method 'get' of 'dict' objects}
        4    0.000    0.000    0.000    0.000 {method 'join' of 'str' objects}
       13    0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
        6    0.000    0.000    0.000    0.000 {method 'lower' of 'str' objects}
        4    0.000    0.000    0.000    0.000 {built-in method builtins.id}
        4    0.000    0.000    0.000    0.000 {built-in method builtins.repr}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
        1    0.000    0.000    0.000    0.000 C:\Users\Paridhi Educational\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py:4041(__st
```

DEBBUGING STEPS

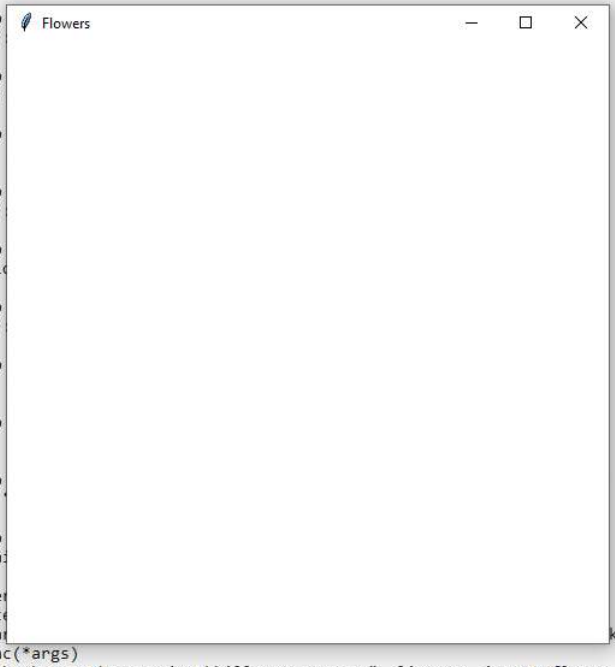
C:\Windows\system32\cmd.exe - python -m pdb billGenerator.py

```
I:\##VISHESH##\pp python project>python -m pdb billGenerator.py
> i:\##vishesh##\pp python project\billgenerator.py(2)<module>()
-> import tkinter
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(5)<module>()
-> from tkinter import simpledialog, messagebox
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(8)<module>()
-> import cProfile
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(11)<module>()
-> import pstats
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(12)<module>()
-> from pstats import SortKey
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(15)<module>()
-> quantity_dictionary = dict()
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(18)<module>()
-> price_dictionary = {"rose":50, "sunflower":40, "jasmine":60, "aloevera":50,
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(19)<module>()
-> "moneyplant":40, "jade":60, "adenium":100, "cactus":300,
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(20)<module>()
-> "palm":200}
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(18)<module>()
-> price_dictionary = {"rose":50, "sunflower":40, "jasmine":60, "aloevera":50,
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(24)<module>()
-> def buyrose():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(29)<module>()
-> def buysunflower():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(34)<module>()
-> def buyjasmine():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(39)<module>()
-> def buyaloevera():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(44)<module>()
```

```
C:\Windows\system32\cmd.exe - python -m pdb billGenerator.py
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(39)<module>()
-> def buyaloevera():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(44)<module>()
-> def buymoneyplant():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(50)<module>()
-> def buyjade():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(55)<module>()
-> def buyadenium():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(60)<module>()
-> def buycactus():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(66)<module>()
-> def buypalm():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(76)<module>()
-> def gotoflower():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(129)<module>()
-> def gotodecoration():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(184)<module>()
-> def gotodesert():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(236)<module>()
-> def billf():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(684)<module>()
-> def main():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(723)<module>()
-> if __name__ ==  "__main__":
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(728)<module>()
-> cProfile.run("main()", "output.dat")
(Pdb) n
```



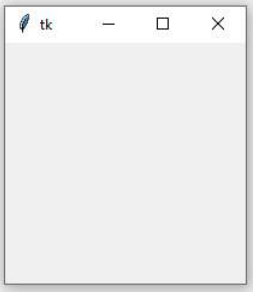

```
C:\Windows\system32\cmd.exe - python -m pdb billGenerator.py
> i:\##vishesh##\pp python project\billgenerator.py(44)<module>()
-> def buymoneyplant():
(Pdb) n
> i:\##vishesh##\pp python project\billgenerator.py(50)<module>()
-> def buyjade():
(Pdb) n
> i:\##vishesh##\pp
-> def buyadenium()
(Pdb) n
> i:\##vishesh##\pp
-> def buycactus():
(Pdb) n
> i:\##vishesh##\pp
-> def buypalm():
(Pdb) n
> i:\##vishesh##\pp
-> def gotoflower()
(Pdb) n
> i:\##vishesh##\pp
-> def gotodecorati
(Pdb) n
> i:\##vishesh##\pp
-> def gotodesert()
(Pdb) n
> i:\##vishesh##\pp
-> def billf():
(Pdb) n
> i:\##vishesh##\pp
-> def main():
(Pdb) n
> i:\##vishesh##\pp
-> if __name__ ==
(Pdb) n
> i:\##vishesh##\pp
-> cProfile.run("ma
(Pdb) n
Exception in Tkinte
Traceback (most rec
  File "C:\Users\Pa
    return self.func(*args)
  File "i:\##vishesh##\pp python project\billgenerator.py", line 87, in gotoflower
    image = fphoto, compound = "left", bg = "white",
NameError: name 'fphoto' is not defined
```



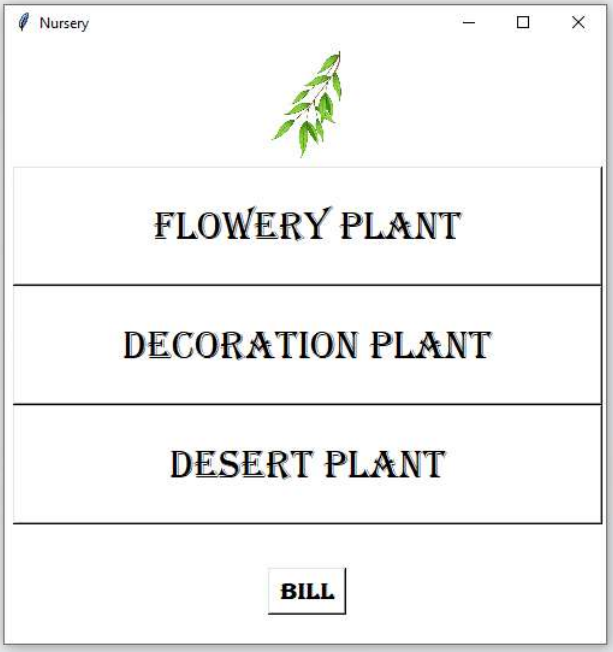
C:\Windows\system32\cmd.exe - python -m pdb bill_generator.py

```
I:\##VISHESH##\pp python project>python -m pdb bill_generator.py
> i:\##vishesh##\pp python project\bill_generator.py(2)<module>()
-> import tkinter
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(5)<module>()
-> from tkinter import simpledialog, messagebox
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(8)<module>()
-> import cProfile
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(11)<module>()
-> import pstats
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(12)<module>()
-> from pstats import SortKey
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(15)<module>()
-> quantity_dictionary = dict()
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(18)<module>()
-> price_dictionary = {"rose":50, "sunflower":40, "jasmine":60, "aloevera":50,
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(19)<module>()
-> "moneyplant":40, "jade":60, "adenium":100, "cactus":300,
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(20)<module>()
-> "palm":200}
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(18)<module>()
-> price_dictionary = {"rose":50, "sunflower":40, "jasmine":60, "aloevera":50,
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(24)<module>()
-> def buyrose():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(29)<module>()
-> def buysunflower():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(34)<module>()
-> def buyjasmine():
(Pdb)
> i:\##vishesh##\pp python project\bill_generator.py(39)<module>()
-> def buyaloevera():
(Pdb) n
```

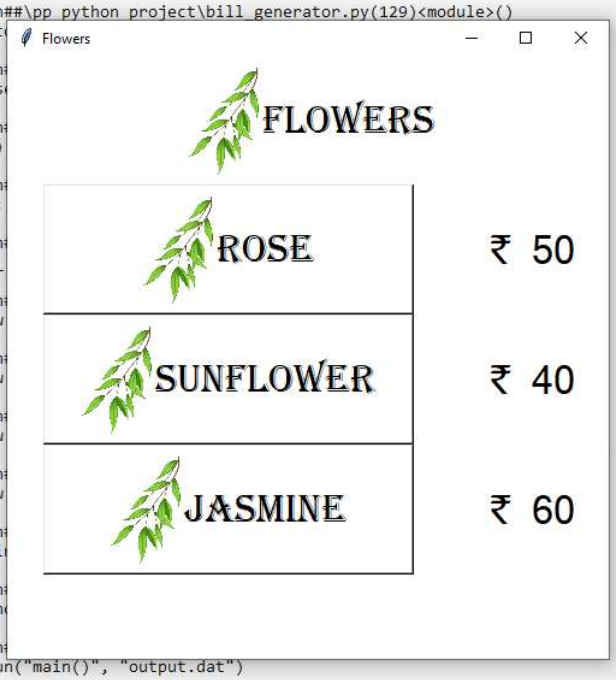
```
C:\Windows\system32\cmd.exe - python -m pdb bill_generator.py
-> def buyaloevera():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(44)<module>()
-> def buymoneyplant():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(50)<module>()
-> def buyjade():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(55)<module>()
-> def buyadenium():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(60)<module>()
-> def buycactus():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(66)<module>()
-> def buypalm():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(76)<module>()
-> def gotoflower():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(129)<module>()
-> def gotodecoration():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(184)<module>()
-> def gotodesert():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(236)<module>()
-> def billf():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(684)<module>()
-> def main():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(716)<module>()
-> if __name__ ==  "__main__":
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(717)<module>()
-> main_window = tkinter.Tk()
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(718)<module>()
-> main_window.title("Nursery")
(Pdb)
```




```
C:\Windows\system32\cmd.exe - python -m pdb bill_generator.py
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(66)<module>()
-> def buypalm():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(76)<module>()
-> def gotoflower():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(129)<module>()
-> def gotodecoration():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(184)<module>()
-> def gotodesert():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(236)<module>()
-> def billf():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(684)<module>()
-> def main():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(716)<module>()
-> if __name__ ==  "__main__":
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(717)<module>()
-> main_window = tkinter.Tk()
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(718)<module>()
-> main_window.title("Nursery")
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(719)<module>()
-> main_window.config(bg = "white")
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(720)<module>()
-> main_window.geometry("500x500+150+150")
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(721)<module>()
-> photo = tkinter.PhotoImage(file = "photo.png")
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(722)<module>()
-> fphoto = photo.subsample(3, 3)
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(726)<module>()
-> cProfile.run("main()", "output.dat")
(Pdb) n
```



```
C:\Windows\system32\cmd.exe - python  -m pdb bill_generator.py
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(66)<module>()
-> def buypalm():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(76)<module>()
-> def gotoflower():
(Pdb) n
> i:\##vishesh##\pp python project\bill_generator.py(129)<module>()
-> def gotodec
(Pdb) n
> i:\##vishesh
-> def gotodes
(Pdb) n
> i:\##vishesh
-> def billf()
(Pdb) n
> i:\##vishesh
-> def main():
(Pdb) n
> i:\##vishesh
-> if __name__
(Pdb) n
> i:\##vishesh
-> main_window
(Pdb) n
> i:\##vishesh
-> main_window
(Pdb) n
> i:\##vishesh
-> main_window
(Pdb) n
> i:\##vishesh
-> main_window
(Pdb) n
> i:\##vishesh
-> photo = tki
(Pdb) n
> i:\##vishesh
-> fphoto = ph
(Pdb) n
> i:\##vishesh
-> cProfile.run("main()", "output.dat")
(Pdb) n
```



MISCELLANEOUS DATA

Starting Date : 5 November, 2022
End Date : 13 November, 2022
Total time required : 9 Days
Total line of code : 733
No of functions : 15
Language used : Python
Profiller used : cProfile
Debugger used : pdb
Program Title : Bill Generator