# Project Completion Report

**Name of the Group Members:** 1. Tanisha Mangaonkar: 16010422200

2. Samrudhi Bhat : 16010422240

3. Aastha Rai : 16010422091

**Internship Period:** 18/12/2023 to 31/12/2023

**Internship Organization:** K. J. Somaiya College of Engineering

**Department:** Information Technology

**Team Name:** Group 4

**Supervisor/Mentor:** Prof. Vaibhav Chunekar & Prof. Anagha Raich.

## I. Introduction:-

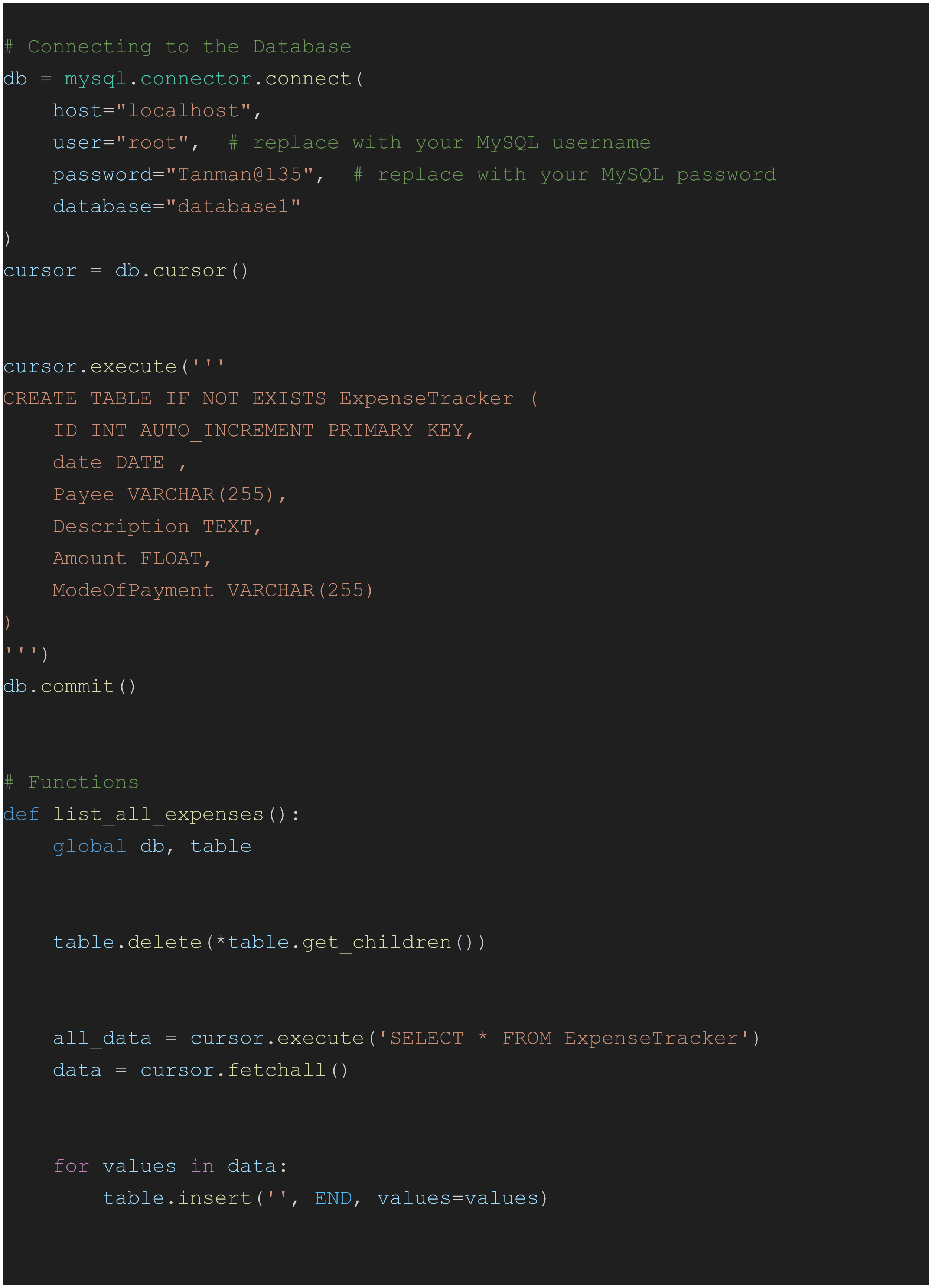
The project on "Data Analysis of Expenses" creates a simple expense tracker that allows users(students) to log their daily expenses, categorize spending, and view a summary of their transactions. This initiative aims to bridge the gap between raw financial data and actionable information, enabling stakeholders to make informed decisions and streamline their financial processes.

## II. Technologies Used:-

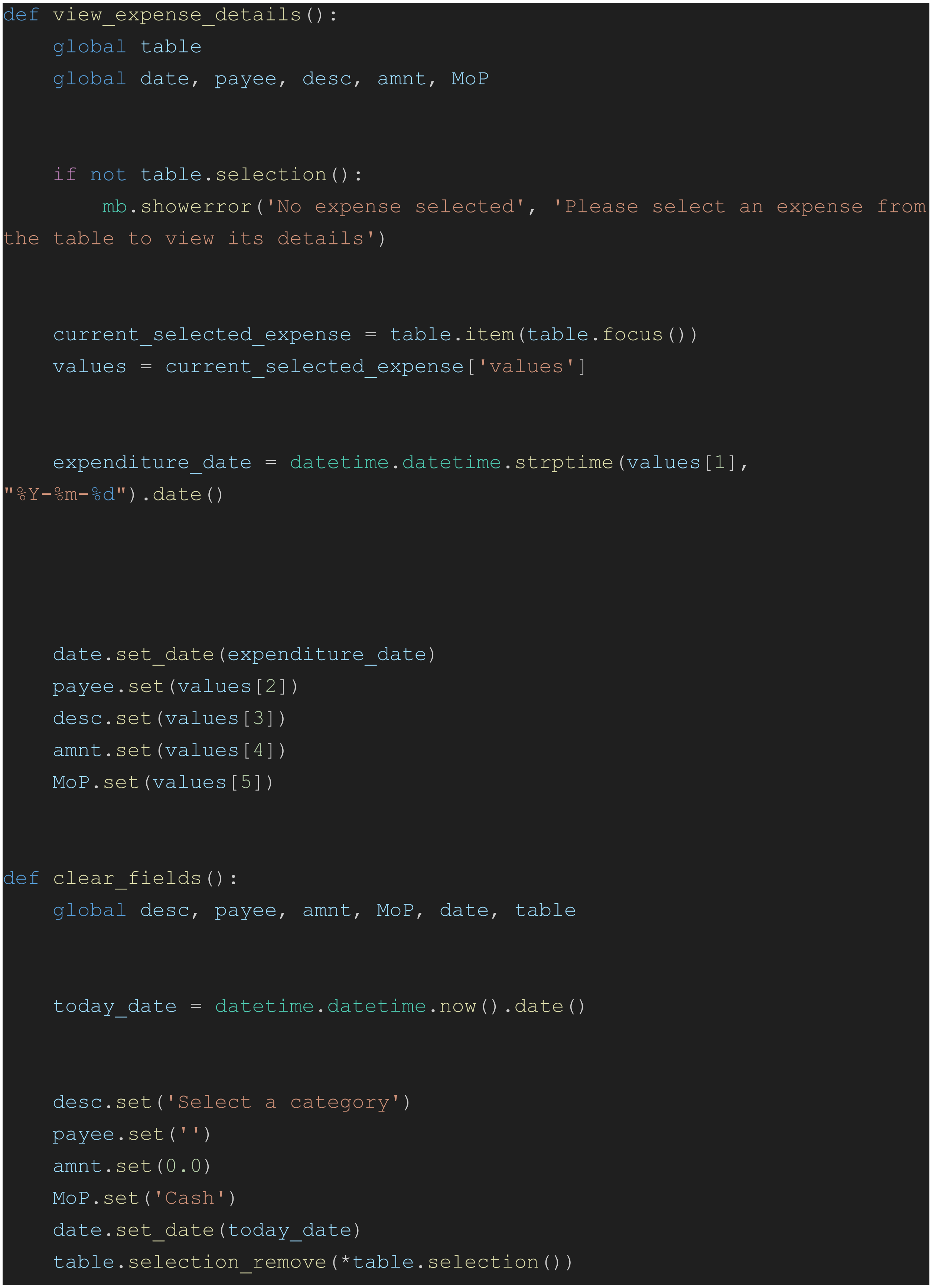
Python: The core programming language for developing the system.
Tkinter: A Python GUI toolkit utilized for creating an interactive user interface.
MySQL: A relational database management system employed for storing user data and expenses
Matplotlib: A Data visualization library employed to include data analysis and visualization to identify and track spending conduct of the user.

## Code:-

```
import tkinter as tk
from tkinter import *
from PIL import Image, ImageTk
import datetime
from tkcalendar import DateEntry
import tkinter.ttk as ttk
from tkinter import messagebox as mb
import mysql.connector
import matplotlib.pyplot as plt
```

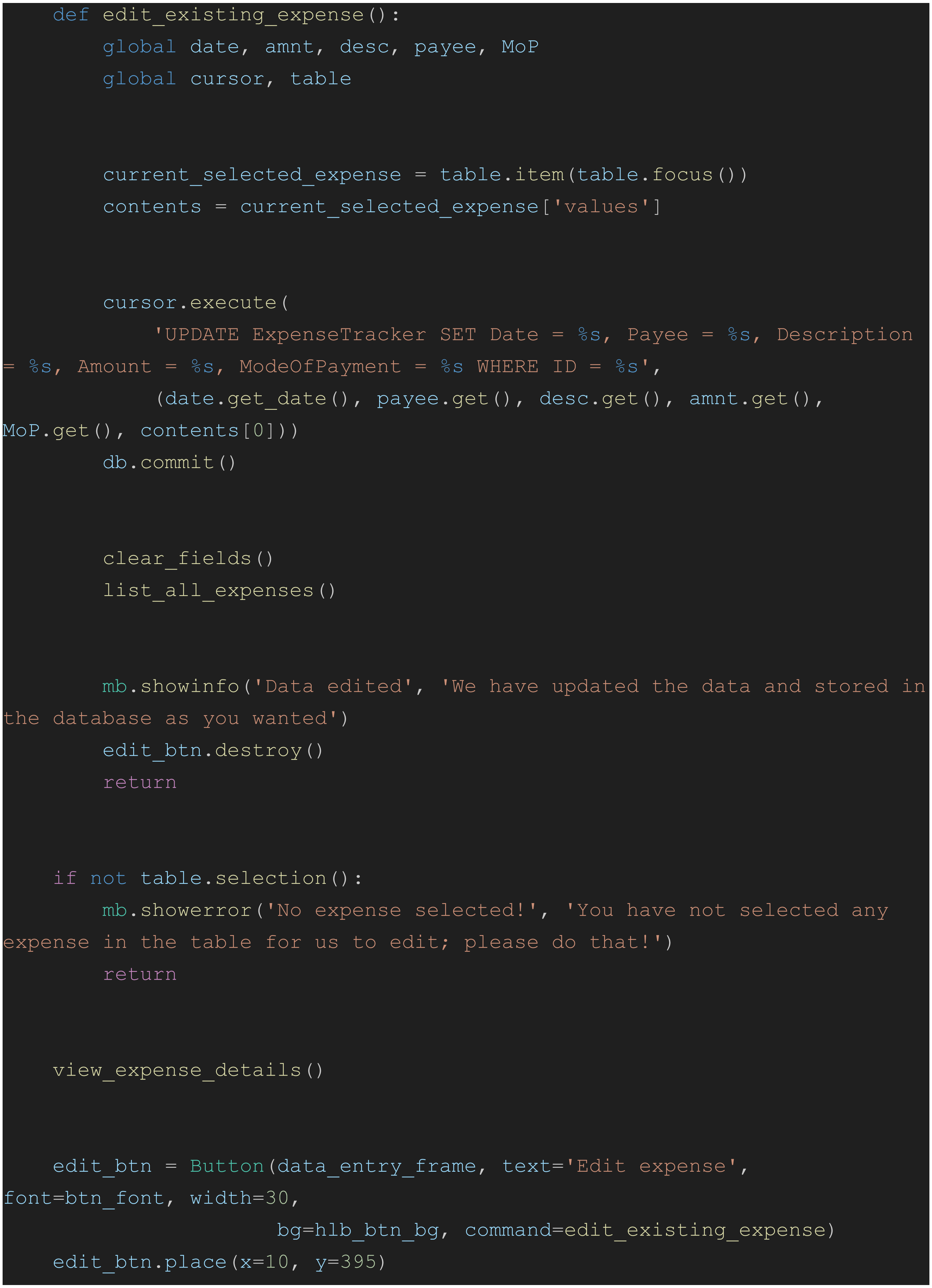


```
# Connecting to the Database
db = mysql.connector.connect(
    host="localhost",
    user="root",  # replace with your MySQL username
    password="Tanman@135",  # replace with your MySQL password
    database="database1"
)
cursor = db.cursor()


cursor.execute('''
CREATE TABLE IF NOT EXISTS ExpenseTracker (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    date DATE ,
    Payee VARCHAR(255),
    Description TEXT,
    Amount FLOAT,
    ModeOfPayment VARCHAR(255)
)
''')
db.commit()


# Functions
def list_all_expenses():
    global db, table


    table.delete(*table.get_children())


    all_data = cursor.execute('SELECT * FROM ExpenseTracker')
    data = cursor.fetchall()


    for values in data:
        table.insert('', END, values=values)
```



```
def view_expense_details():
    global table
    global date, payee, desc, amnt, MoP


    if not table.selection():
        mb.showerror('No expense selected', 'Please select an expense from
the table to view its details')


    current_selected_expense = table.item(table.focus())
    values = current_selected_expense['values']


    expenditure_date = datetime.datetime.strptime(values[1],
"%Y-%m-%d").date()




    date.set_date(expenditure_date)
    payee.set(values[2])
    desc.set(values[3])
    amnt.set(values[4])
    MoP.set(values[5])


def clear_fields():
    global desc, payee, amnt, MoP, date, table


    today_date = datetime.datetime.now().date()


    desc.set('Select a category')
    payee.set('')
    amnt.set(0.0)
    MoP.set('Cash')
    date.set_date(today_date)
    table.selection_remove(*table.selection())
```

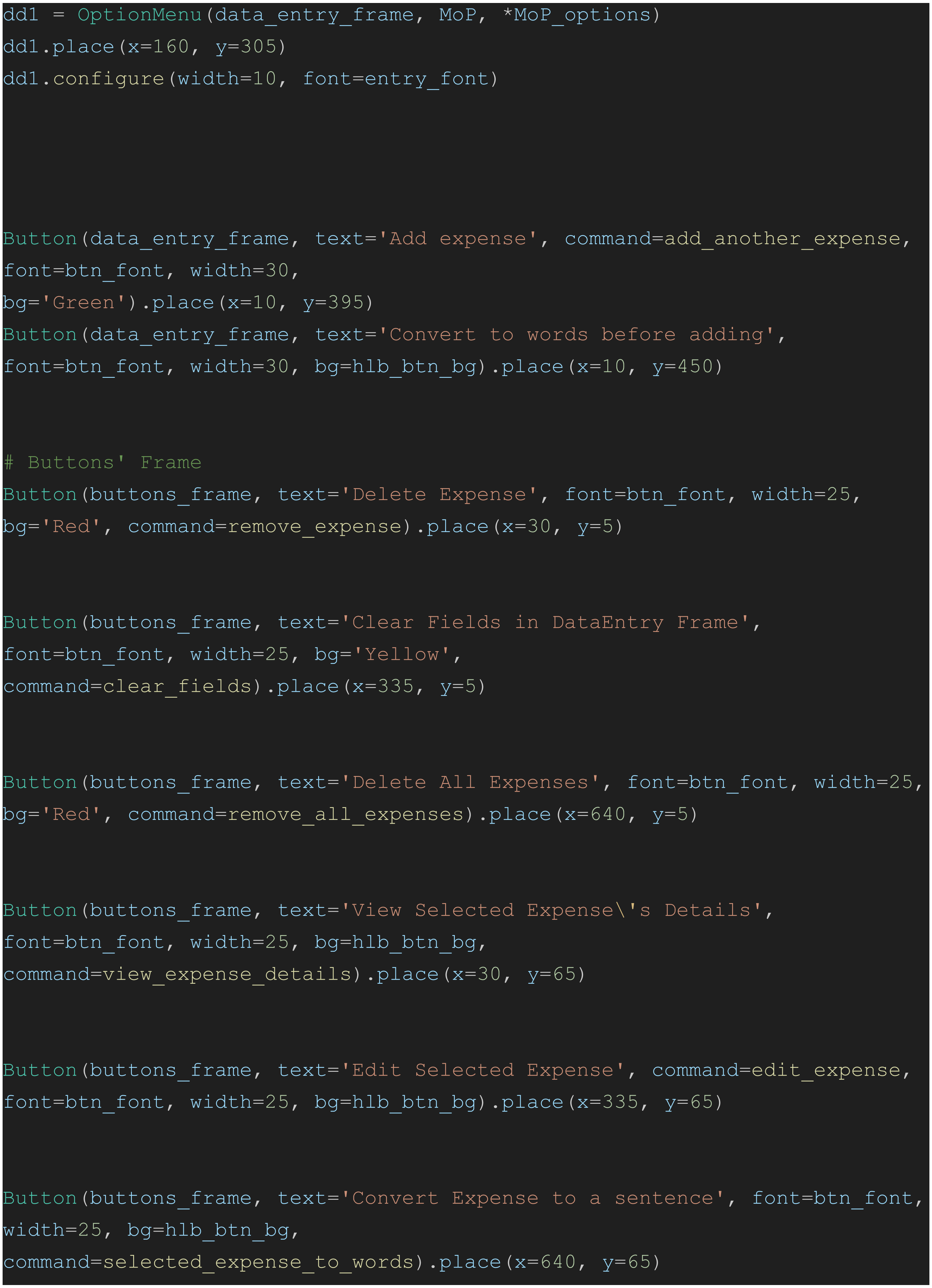
```
def remove_expense():
    if not table.selection():
        mb.showerror('No record selected!', 'Please select a record to
delete!')
        return


    current_selected_expense = table.item(table.focus())
    values_selected = current_selected_expense['values']


    surety = mb.askyesno('Are you sure?', f'Are you sure that you want to
delete the record of {values_selected[2]}?')


    if surety:
        cursor.execute('DELETE FROM ExpenseTracker WHERE ID=%s',
(values_selected[0],))
        db.commit()


        list_all_expenses()
        mb.showinfo('Record deleted successfully!', 'The record you wanted
to delete has been deleted successfully')


def remove_all_expenses():
    surety = mb.askyesno('Are you sure?', 'Are you sure that you want to
delete all the expense items from the database?',
                        icon='warning')


    if surety:
        table.delete(*table.get_children())


        cursor.execute('DROP TABLE ExpenseTracker')
        db.commit()
        clear_fields()
        list_all_expenses()
        mb.showinfo('All Expenses deleted', 'All the expenses were successfully deleted')
    else:
        mb.showinfo('Ok then', 'The task was aborted and no expense was deleted!')


def add_another_expense():
    global date, payee, desc, amnt, MoP
    global cursor


    if not date.get_date() or not payee.get() or not desc.get() or not amnt.get() or not MoP.get():
        mb.showerror('Fields empty!', "Please fill all the missing fields before pressing the add button!")
    else:
        cursor.execute(
            'INSERT INTO ExpenseTracker (Date, Payee, Description, Amount, ModeOfPayment) VALUES (%s, %s, %s, %s, %s)',
            (date.get_date(), payee.get(), desc.get(), amnt.get(), MoP.get())
        )
        db.commit()


        clear_fields()
        list_all_expenses()
        mb.showinfo('Expense added', 'The expense whose details you just entered has been added to the database')


def edit_expense():
    global table
```



```
    def edit_existing_expense():
        global date, amnt, desc, payee, MoP
        global cursor, table


        current_selected_expense = table.item(table.focus())
        contents = current_selected_expense['values']


        cursor.execute(
            'UPDATE ExpenseTracker SET Date = %s, Payee = %s, Description
= %s, Amount = %s, ModeOfPayment = %s WHERE ID = %s',
            (date.get_date(), payee.get(), desc.get(), amnt.get(),
MoP.get(), contents[0]))
        db.commit()


        clear_fields()
        list_all_expenses()


        mb.showinfo('Data edited', 'We have updated the data and stored in
the database as you wanted')
        edit_btn.destroy()
        return


    if not table.selection():
        mb.showerror('No expense selected!', 'You have not selected any
expense in the table for us to edit; please do that!')
        return


    view_expense_details()


    edit_btn = Button(data_entry_frame, text='Edit expense',
font=btn_font, width=30,
                      bg=hlb_btn_bg, command=edit_existing_expense)
    edit_btn.place(x=10, y=395)
```

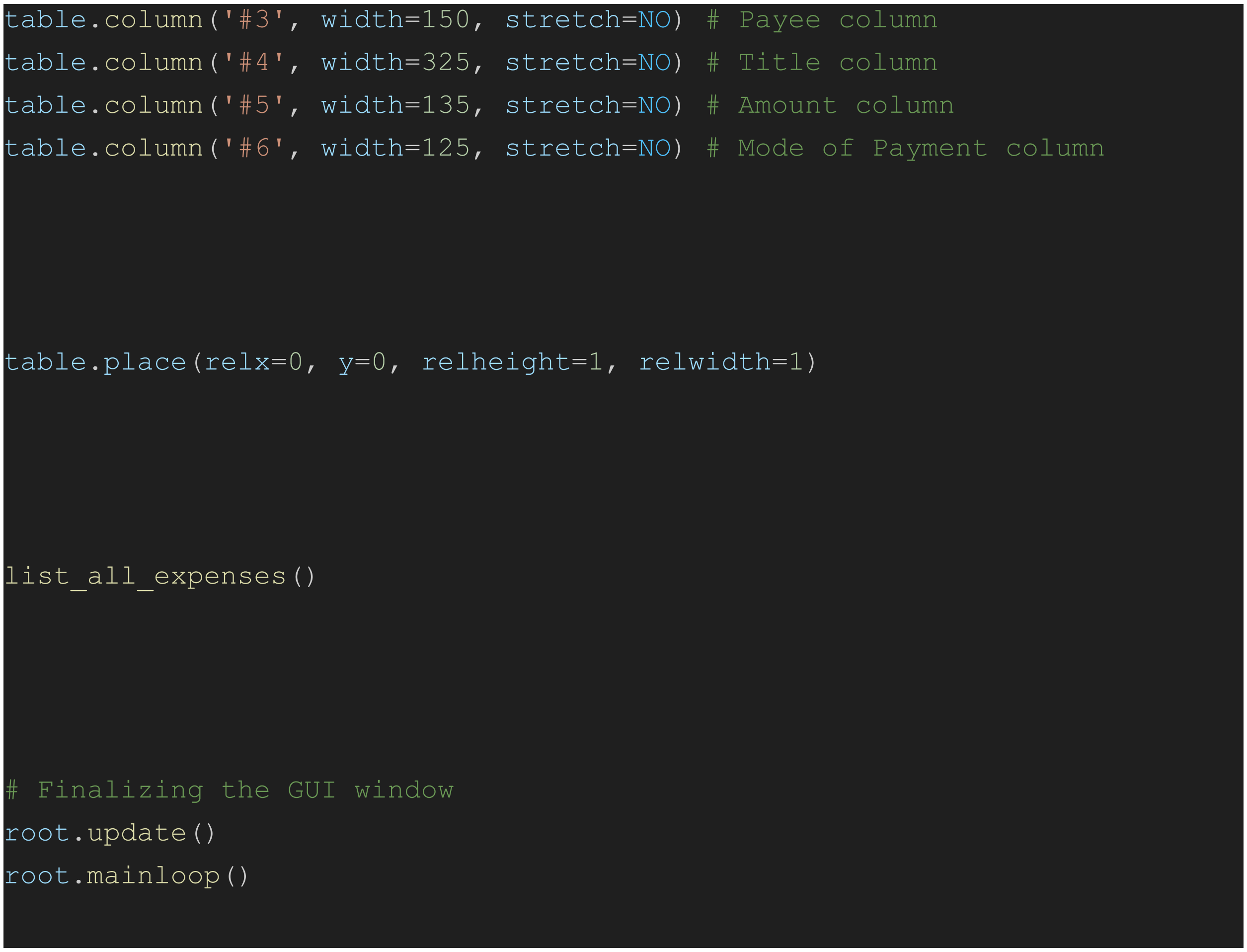
```
def selected_expense_to_words():
    global table


    if not table.selection():
        mb.showerror('No expense selected!', 'Please select an expense
from the table for us to read')
        return


    current_selected_expense = table.item(table.focus())
    values = current_selected_expense['values']


    message = f'Your expense can be read like: \n"You paid {values[4]} to
{values[2]} for {values[3]} on {values[1]} via {values[5]}"'


    mb.showinfo('Here\'s how to read your expense', message)

def expense_to_words_before_adding():
    global date, desc, amnt, payee, MoP


    if not date or not desc or not amnt or not payee or not MoP:
        mb.showerror('Incomplete data', 'The data is incomplete, meaning
fill all the fields first!')


    message = f'Your expense can be read like: \n"You paid {amnt.get()} to
{payee.get()} for {desc.get()} on {date.get_date()} via {MoP.get()}"'


    add_question = mb.askyesno('Read your record like: ',
f'{message}\n\nShould I add it to the database?')


    if add_question:
        add_another_expense()
    else:
        mb.showinfo('Ok', 'Please take your time to add this record')







def visualize_expenses_per_category():
    global cursor


    cursor.execute('SELECT Description, COUNT(*) FROM ExpenseTracker GROUP
BY Description')
    data = cursor.fetchall()


    categories = [row[0] for row in data]
    counts = [row[1] for row in data]


    plt.figure(figsize=(10, 6))
    plt.bar(categories, counts, color='skyblue')
    plt.xlabel('Expense Category')
    plt.ylabel('Number of Expenses')
    plt.title('Number of Expenses per Category')
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()


    plt.show()
    # Backgrounds and Fonts
dataentery_frame_bg = '#fab885'
buttons_frame_bg = '#f27457'
hlb_btn_bg = 'White'


lbl_font = ('Georgia', 13)
entry_font = 'Times 13 bold'
btn_font = ('Gill Sans MT', 13)


# Initializing the GUI window
root = Tk()
root.title('Expense Tracker')
root.geometry('1100x680+0+0')


Label(root, text='EXPENSE TRACKER', fg="White", highlightcolor='black',
bd=2, relief=RIDGE,
font=('Noto Sans CJK TC', 15, 'bold'), bg='#fab885').pack(side=TOP,
fill=X)


# StringVar and DoubleVar variables
desc = StringVar(value='Select a category')
amnt = DoubleVar()
payee = StringVar()
MoP = StringVar(value='Cash')


# Frames
data_entry_frame = Frame(root, bg=dataentery_frame_bg)
data_entry_frame.place(x=0, y=30, relheight=0.95, relwidth=0.25)


buttons_frame = Frame(root, bg=buttons_frame_bg)
buttons_frame.place(relx=0.25, rely=0.05, relwidth=0.75, relheight=0.21)
```

```
tree_frame = Frame(root)
tree_frame.place(relx=0.25, rely=0.26, relwidth=0.75, relheight=0.74)
# Data Entry Frame
Label(data_entry_frame, text='Date (M/DD/YY) :', font=lbl_font,
bg=dataentery_frame_bg).place(x=10, y=50)
date = DateEntry(data_entry_frame, date=datetime.datetime.now().date(),
font=entry_font)
date.place(x=160, y=50)


Label(data_entry_frame, text='Payee\t :', font=lbl_font,
bg=dataentery_frame_bg).place(x=10, y=230)
Entry(data_entry_frame, font=entry_font, width=31, text=payee).place(x=10,
y=260)


Label(data_entry_frame, text='Category :', font=lbl_font,
bg=dataentery_frame_bg).place(x=10, y=100)
categories = ['Select a catogory', 'Food', 'Stationery', 'Fees',
'Clothing', 'Others']
desc.set(categories[0])
cat_opt = OptionMenu(data_entry_frame, desc, *categories)
cat_opt.place(x=100, y=100)




Label(data_entry_frame, text='Amount :', font=lbl_font,
bg=dataentery_frame_bg).place(x=10, y=180)
Entry(data_entry_frame, font=entry_font, width=14, text=amnt).place(x=95,
y=180)


Label(data_entry_frame, text='Mode of Payment:', font=lbl_font,
bg=dataentery_frame_bg).place(x=10, y=310)
MoP_options = ['Cash', 'Cheque', 'Credit Card', 'Debit Card', 'Paytm',
'Google Pay', 'Razorpay']
MoP.set(MoP_options[0])  # Set default value
```

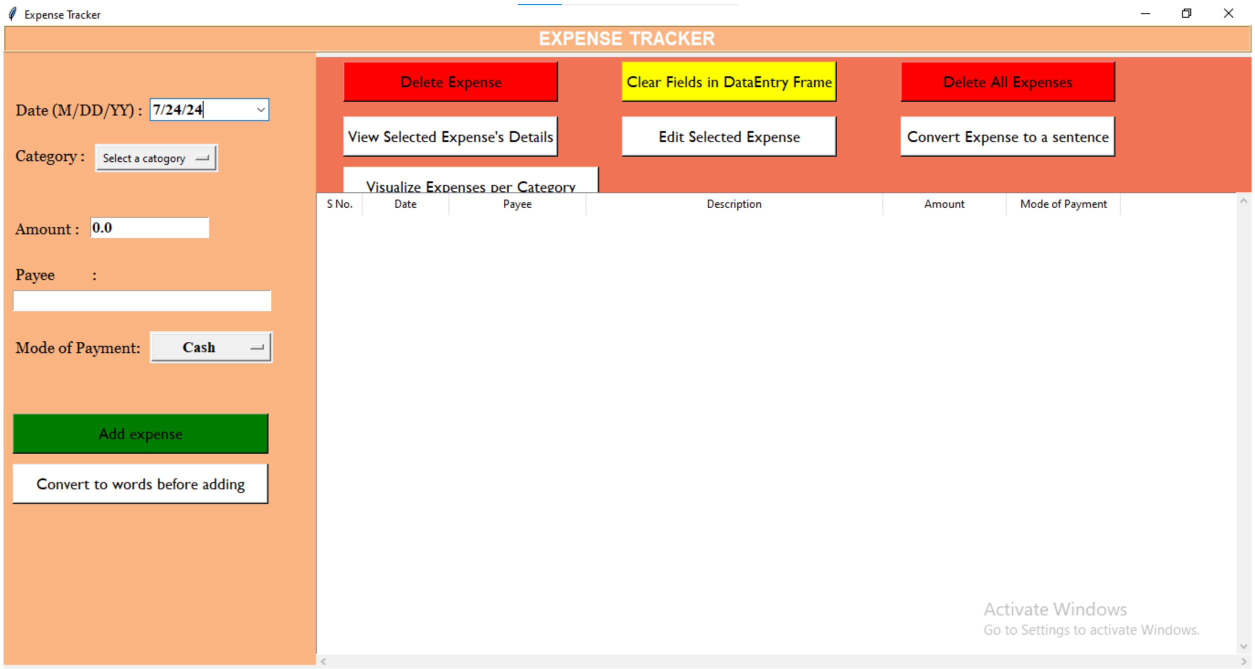


```
dd1 = OptionMenu(data_entry_frame, MoP, *MoP_options)
dd1.place(x=160, y=305)
dd1.configure(width=10, font=entry_font)




Button(data_entry_frame, text='Add expense', command=add_another_expense,
font=btn_font, width=30,
bg='Green').place(x=10, y=395)
Button(data_entry_frame, text='Convert to words before adding',
font=btn_font, width=30, bg=hlb_btn_bg).place(x=10, y=450)


# Buttons' Frame
Button(buttons_frame, text='Delete Expense', font=btn_font, width=25,
bg='Red', command=remove_expense).place(x=30, y=5)


Button(buttons_frame, text='Clear Fields in DataEntry Frame',
font=btn_font, width=25, bg='Yellow',
command=clear_fields).place(x=335, y=5)


Button(buttons_frame, text='Delete All Expenses', font=btn_font, width=25,
bg='Red', command=remove_all_expenses).place(x=640, y=5)


Button(buttons_frame, text='View Selected Expense\'s Details',
font=btn_font, width=25, bg=hlb_btn_bg,
command=view_expense_details).place(x=30, y=65)


Button(buttons_frame, text='Edit Selected Expense', command=edit_expense,
font=btn_font, width=25, bg=hlb_btn_bg).place(x=335, y=65)


Button(buttons_frame, text='Convert Expense to a sentence', font=btn_font,
width=25, bg=hlb_btn_bg,
command=selected_expense_to_words).place(x=640, y=65)
```

```
# Add a button for visualization
Button(buttons_frame, text='Visualize Expenses per Category',
font=btn_font, width=30, bg=hlb_btn_bg,
       command=visualize_expenses_per_category).place(x=30, y=120)


# Treeview Frame
table = ttk.Treeview(tree_frame, selectmode=BROWSE,
                columns=('ID', 'Date', 'Payee', 'Description', 'Amount',
'Mode of Payment'))


X_Scroller = Scrollbar(table, orient=HORIZONTAL, command=table)
Y_Scroller = Scrollbar(table, orient=VERTICAL, command=table.yview)
X_Scroller.pack(side=BOTTOM, fill=X)
Y_Scroller.pack(side=RIGHT, fill=Y)




table.config(yscrollcommand=Y_Scroller.set, xscrollcommand=X_Scroller.set)




table.heading('ID', text='S No.', anchor=CENTER)
table.heading('Date', text='Date', anchor=CENTER)
table.heading('Payee', text='Payee', anchor=CENTER)
table.heading('Description', text='Description', anchor=CENTER)
table.heading('Amount', text='Amount', anchor=CENTER)
table.heading('Mode of Payment', text='Mode of Payment', anchor=CENTER)




table.column('#0', width=0, stretch=NO)
table.column('#1', width=50, stretch=NO)
table.column('#2', width=95, stretch=NO)  # Date column
```



```
table.column('#3', width=150, stretch=NO) # Payee column
table.column('#4', width=325, stretch=NO) # Title column
table.column('#5', width=135, stretch=NO) # Amount column
table.column('#6', width=125, stretch=NO) # Mode of Payment column



table.place(relx=0, y=0, relheight=1, relwidth=1)



list_all_expenses()



# Finalizing the GUI window
root.update()
root.mainloop()
```



Expense Tracker

EXPENSE TRACKER

Date (M/DD/YY) : 7/24/24

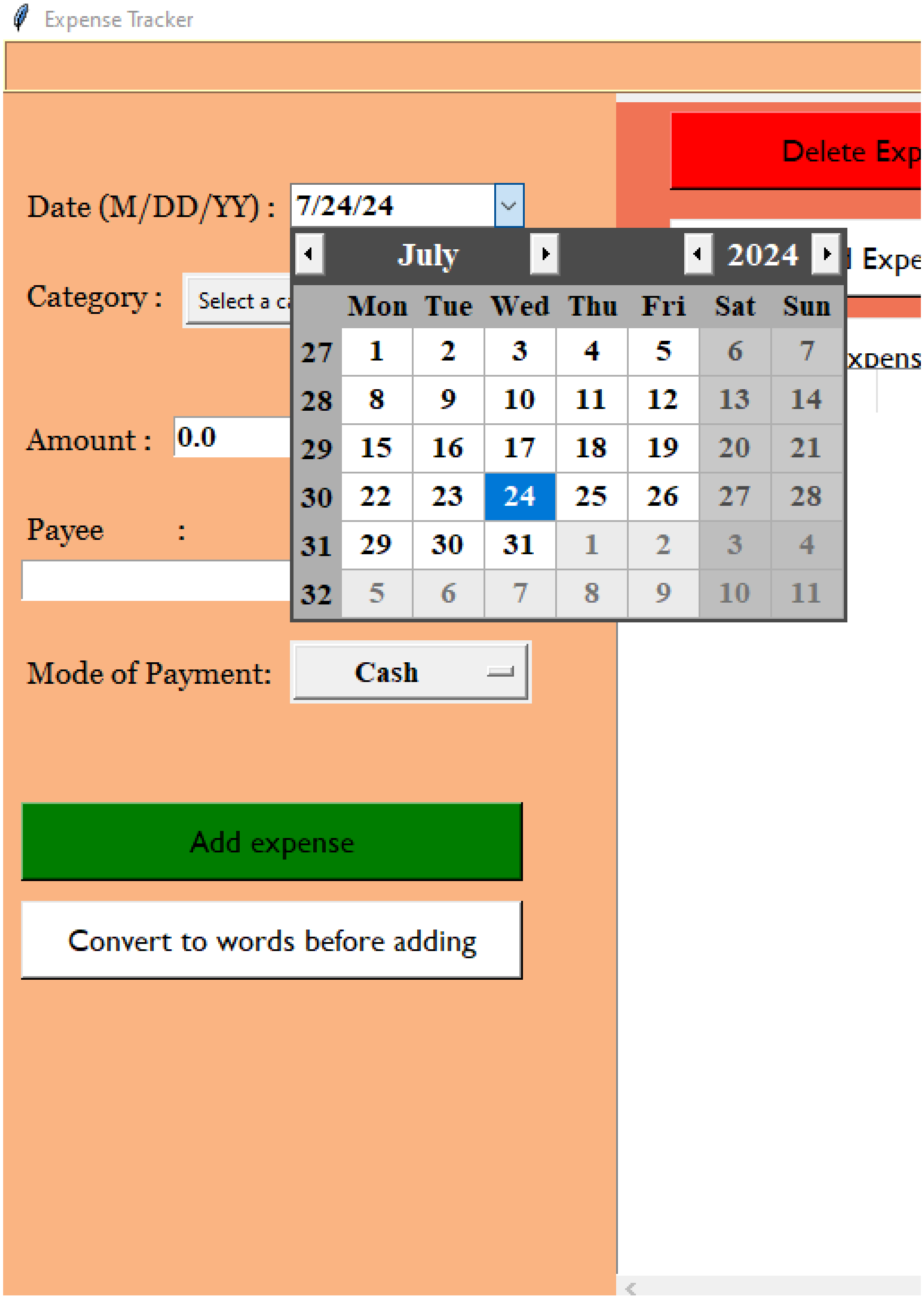
Category : Select a catogory

Amount : 0.0

Payee :

Mode of Payment: Cash

Add expense
Convert to words before adding
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
S No.
Date
Payee
Description
Amount
Mode of Payment
Activate Windows
Go to Settings to activate Windows.



Date (M/DD/YY) : 7/24/24

Category : Select a ca

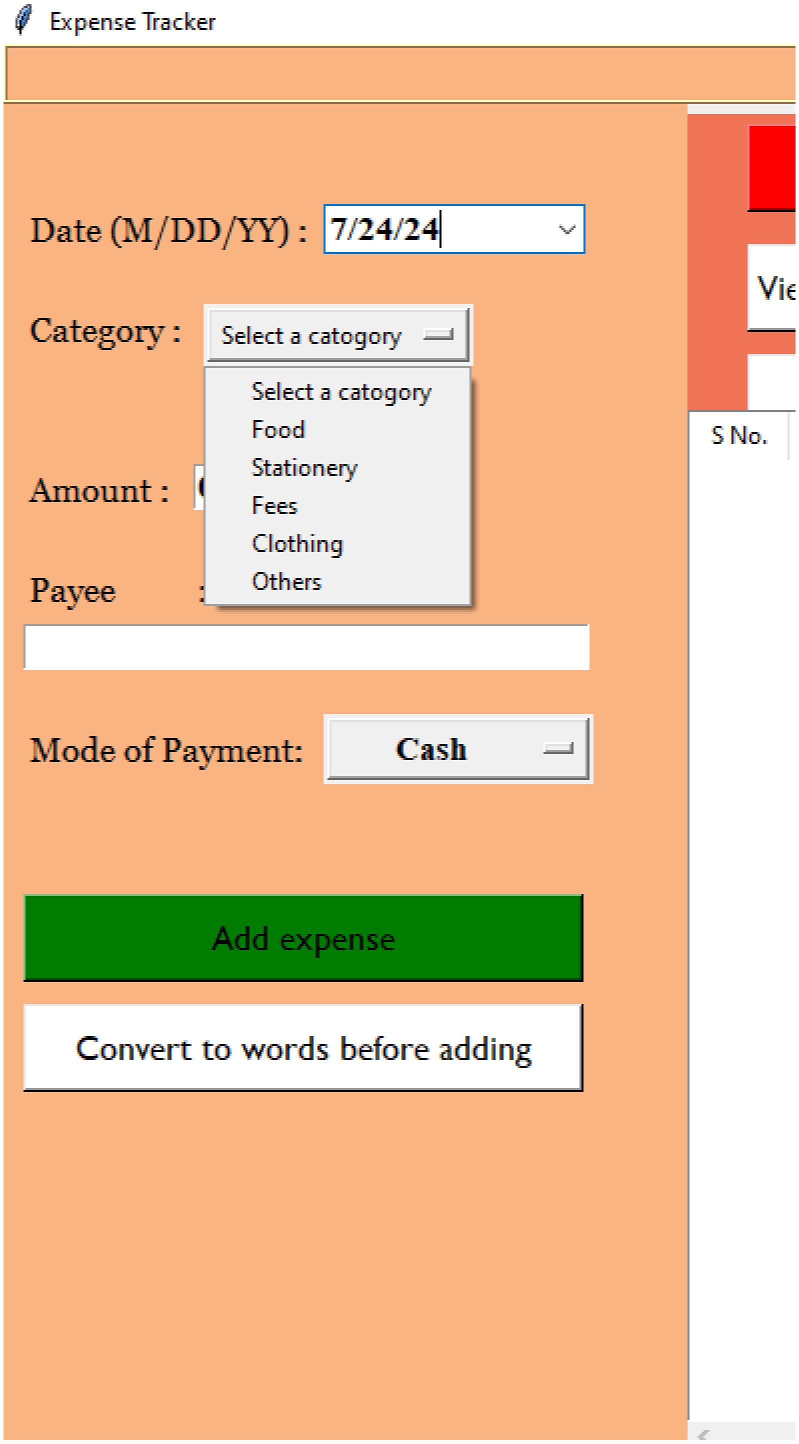
Amount : 0.0

Payee :

Mode of Payment: Cash

July 2024

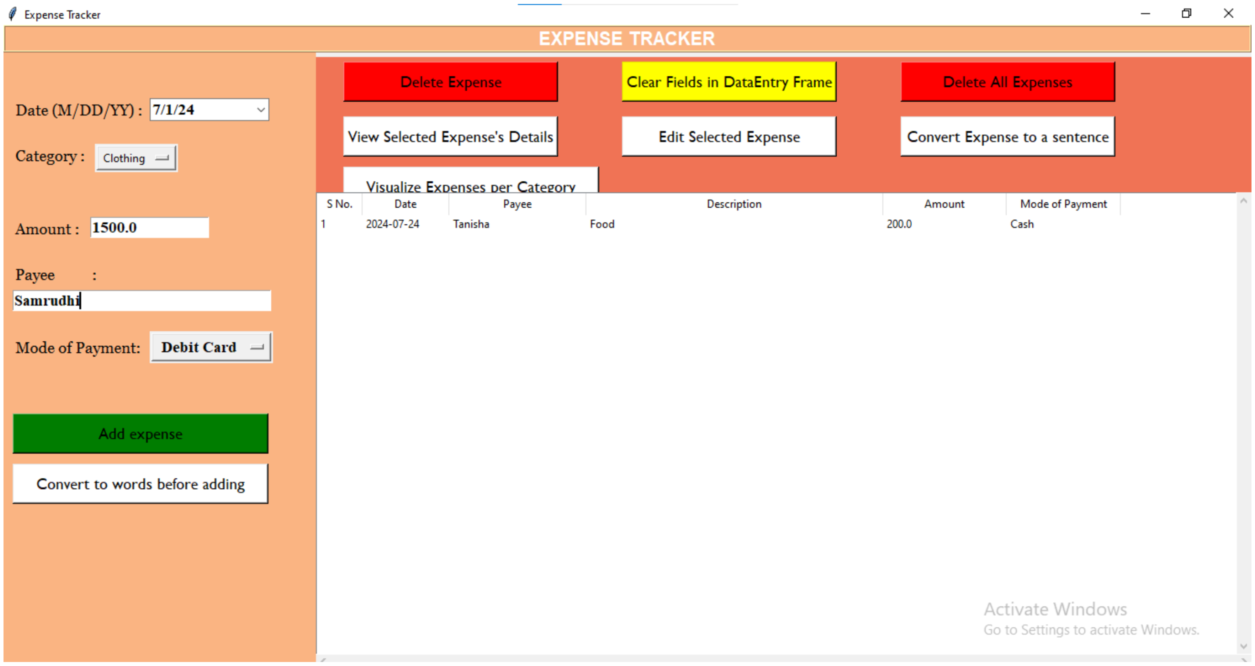
| | Mon | Tue | Wed | Thu | Fri | Sat | Sun |
|---|---|---|---|---|---|---|---|
| 27 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 28 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 29 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 30 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 31 | 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| 32 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |



Date (M/DD/YY) : 7/24/24

Category : Select a catogory

Mode of Payment: Cash


Expense Tracker
EXPENSE TRACKER
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
Add expense
Convert to words before adding
Activate Windows
Go to Settings to activate Windows.

Date (M/DD/YY) : 7/1/24

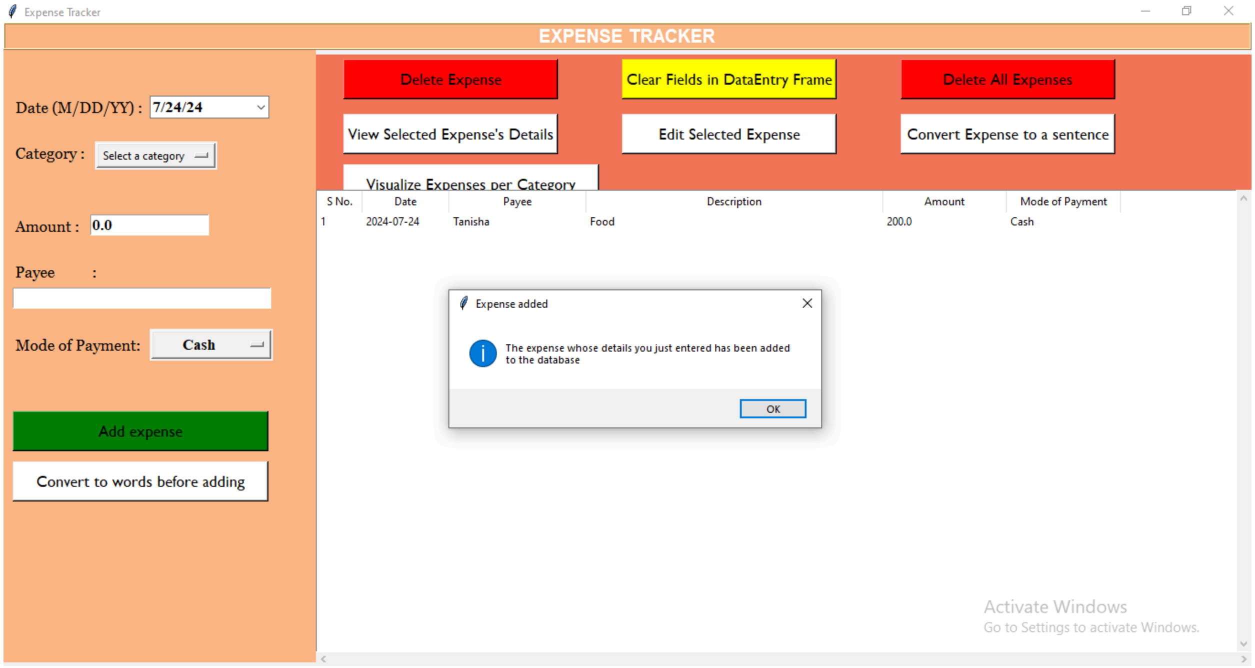
Category : Clothing

Amount : 1500.0

Payee : Samrudhi

Mode of Payment: Debit Card

| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|
| 1 | 2024-07-24 | Tanisha | Food | 200.0 | Cash |


Expense Tracker
EXPENSE TRACKER
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
Add expense
Convert to words before adding
OK
Activate Windows
Go to Settings to activate Windows.

Date (M/DD/YY) : 7/24/24

Category : Select a category

Amount : 0.0

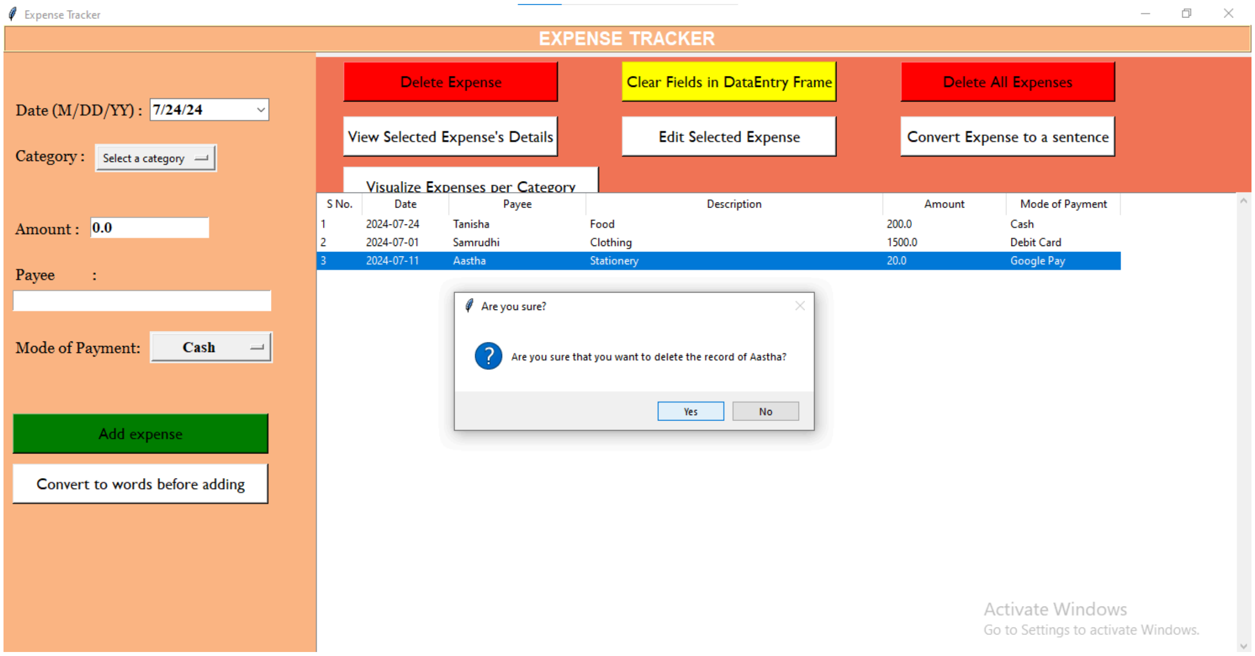
Payee :

Mode of Payment: Cash

| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|
| 1 | 2024-07-24 | Tanisha | Food | 200.0 | Cash |

Expense added

The expense whose details you just entered has been added to the database


Expense Tracker
EXPENSE TRACKER
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
Add expense
Convert to words before adding
Yes
No
Activate Windows
Go to Settings to activate Windows.

Date (M/DD/YY) : 7/24/24

Category : Select a category

Amount : 0.0

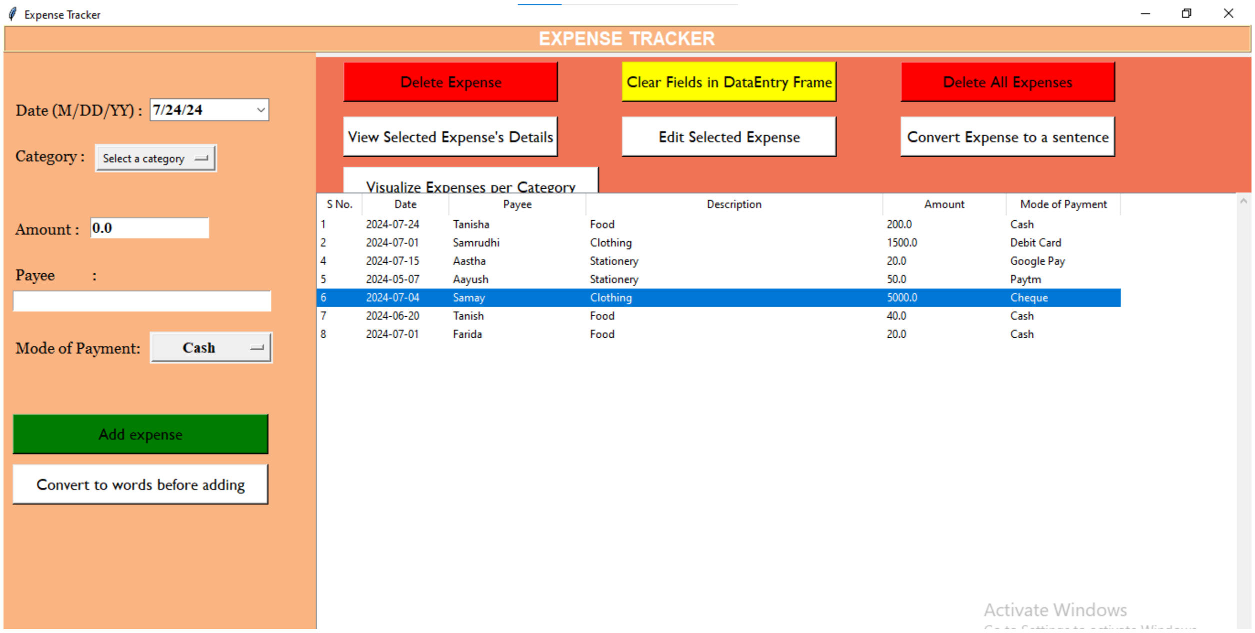
Payee :

Mode of Payment: Cash

| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|
| 1 | 2024-07-24 | Tanisha | Food | 200.0 | Cash |
| 2 | 2024-07-01 | Samrudhi | Clothing | 1500.0 | Debit Card |
| 3 | 2024-07-11 | Aastha | Stationery | 20.0 | Google Pay |

Are you sure?

Are you sure that you want to delete the record of Aastha?


Expense Tracker
EXPENSE TRACKER
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
Add expense
Convert to words before adding
Activate Windows

Date (M/DD/YY) : 7/24/24

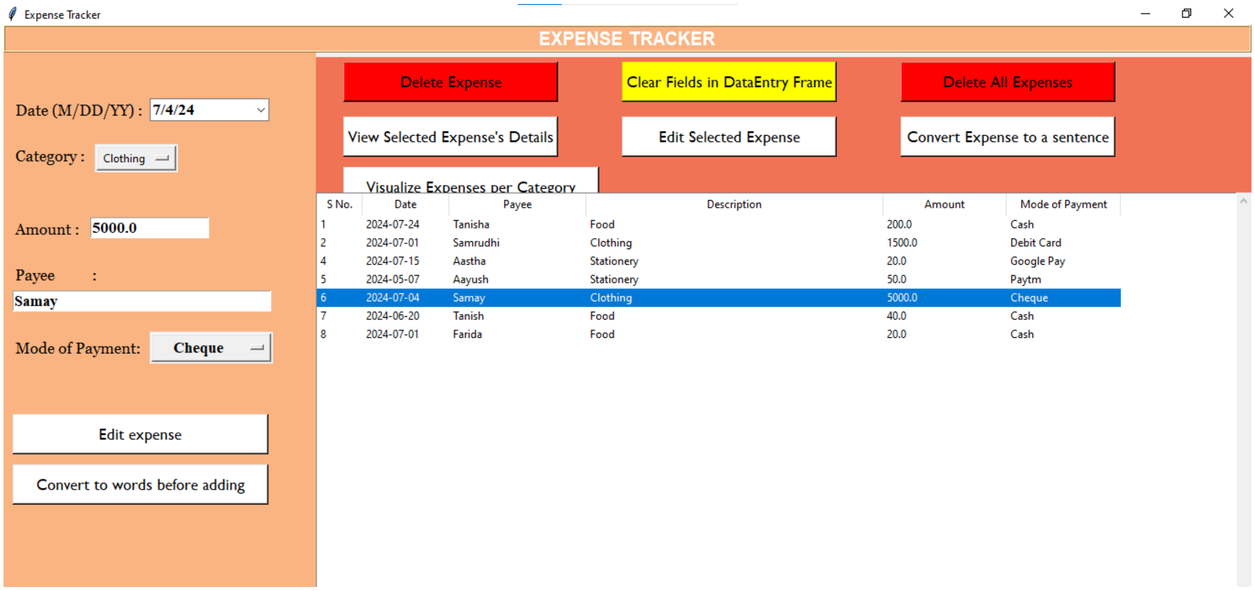
Category : Select a category

Amount : 0.0

Payee :

Mode of Payment: Cash

| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|
| 1 | 2024-07-24 | Tanisha | Food | 200.0 | Cash |
| 2 | 2024-07-01 | Samrudhi | Clothing | 1500.0 | Debit Card |
| 4 | 2024-07-15 | Aastha | Stationery | 20.0 | Google Pay |
| 5 | 2024-05-07 | Aayush | Stationery | 50.0 | Paytm |
| 6 | 2024-07-04 | Samay | Clothing | 5000.0 | Cheque |
| 7 | 2024-06-20 | Tanish | Food | 40.0 | Cash |
| 8 | 2024-07-01 | Farida | Food | 20.0 | Cash |


Expense Tracker
EXPENSE TRACKER
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
Edit expense
Convert to words before adding

Date (M/DD/YY) : 7/4/24

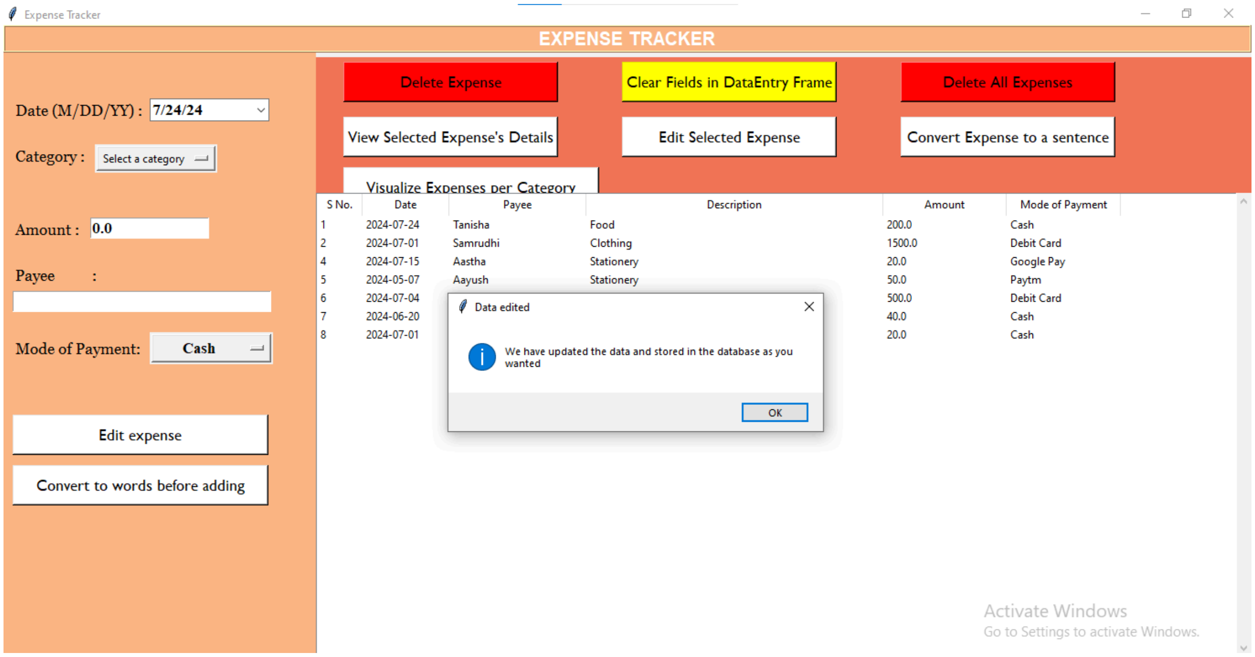
Category : Clothing

Amount : 5000.0

Payee : Samay

Mode of Payment: Cheque

| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|
| 1 | 2024-07-24 | Tanisha | Food | 200.0 | Cash |
| 2 | 2024-07-01 | Samrudhi | Clothing | 1500.0 | Debit Card |
| 4 | 2024-07-15 | Aastha | Stationery | 20.0 | Google Pay |
| 5 | 2024-05-07 | Aayush | Stationery | 50.0 | Paytm |
| 6 | 2024-07-04 | Samay | Clothing | 5000.0 | Cheque |
| 7 | 2024-06-20 | Tanish | Food | 40.0 | Cash |
| 8 | 2024-07-01 | Farida | Food | 20.0 | Cash |


Expense Tracker
EXPENSE TRACKER
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
Edit expense
Convert to words before adding
OK
Activate Windows
Go to Settings to activate Windows.

Date (M/DD/YY) : 7/24/24

Category : Select a category

Amount : 0.0

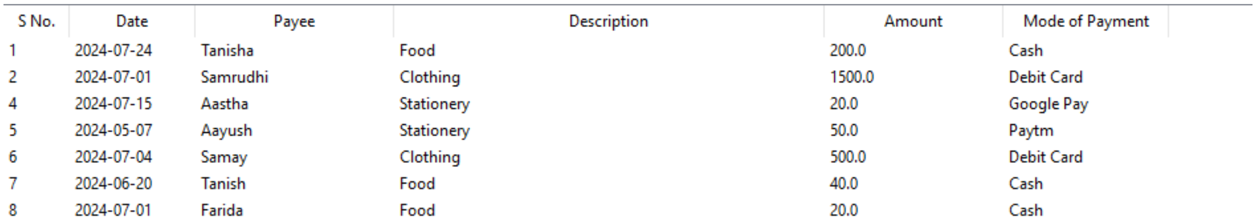
Payee :

Mode of Payment: Cash

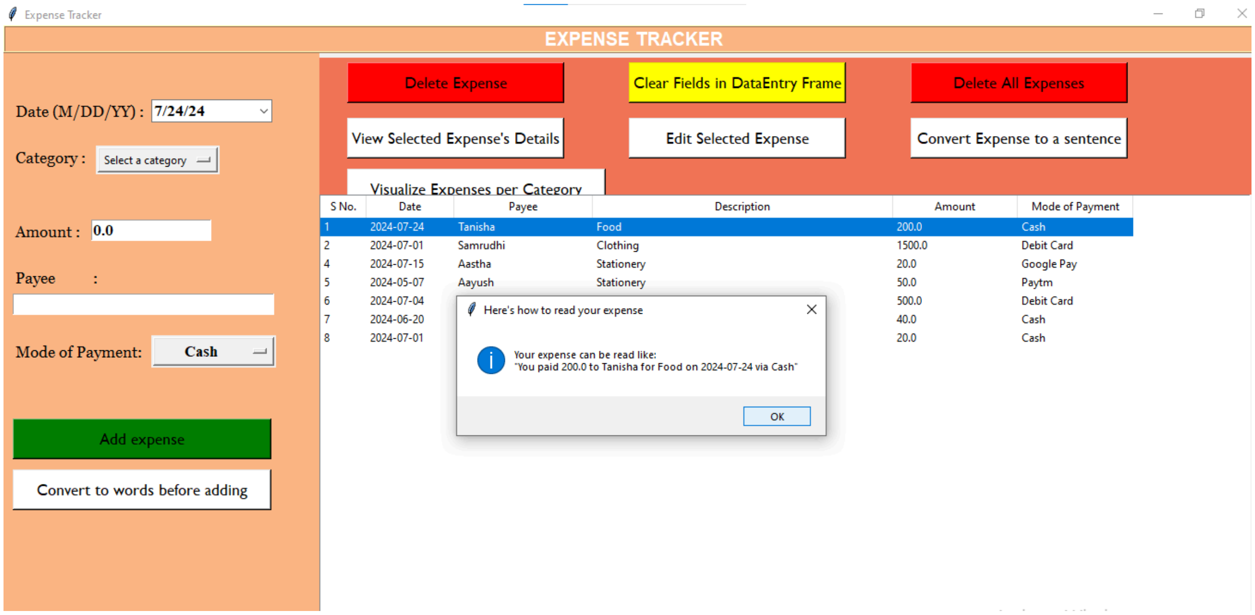
| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|
| 1 | 2024-07-24 | Tanisha | Food | 200.0 | Cash |
| 2 | 2024-07-01 | Samrudhi | Clothing | 1500.0 | Debit Card |
| 4 | 2024-07-15 | Aastha | Stationery | 20.0 | Google Pay |
| 5 | 2024-05-07 | Aayush | Stationery | 50.0 | Paytm |
| 6 | 2024-07-04 | | | 500.0 | Debit Card |
| 7 | 2024-06-20 | | | 40.0 | Cash |
| 8 | 2024-07-01 | | | 20.0 | Cash |

Data edited

We have updated the data and stored in the database as you wanted



| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|
| 1 | 2024-07-24 | Tanisha | Food | 200.0 | Cash |
| 2 | 2024-07-01 | Samrudhi | Clothing | 1500.0 | Debit Card |
| 4 | 2024-07-15 | Aastha | Stationery | 20.0 | Google Pay |
| 5 | 2024-05-07 | Aayush | Stationery | 50.0 | Paytm |
| 6 | 2024-07-04 | Samay | Clothing | 500.0 | Debit Card |
| 7 | 2024-06-20 | Tanish | Food | 40.0 | Cash |
| 8 | 2024-07-01 | Farida | Food | 20.0 | Cash |


Expense Tracker
EXPENSE TRACKER
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
Add expense
Convert to words before adding
OK

Date (M/DD/YY) : 7/24/24

Category : Select a category

Amount : 0.0

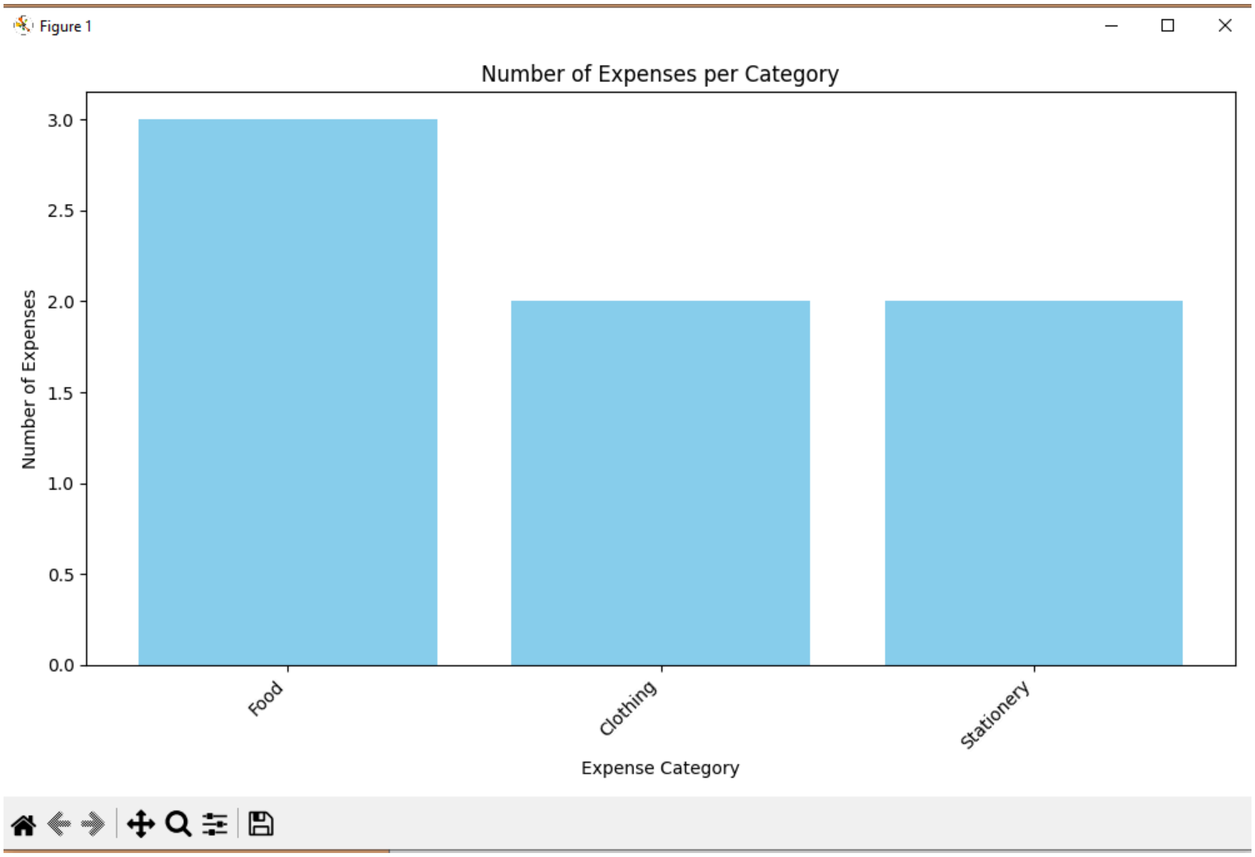
Payee :

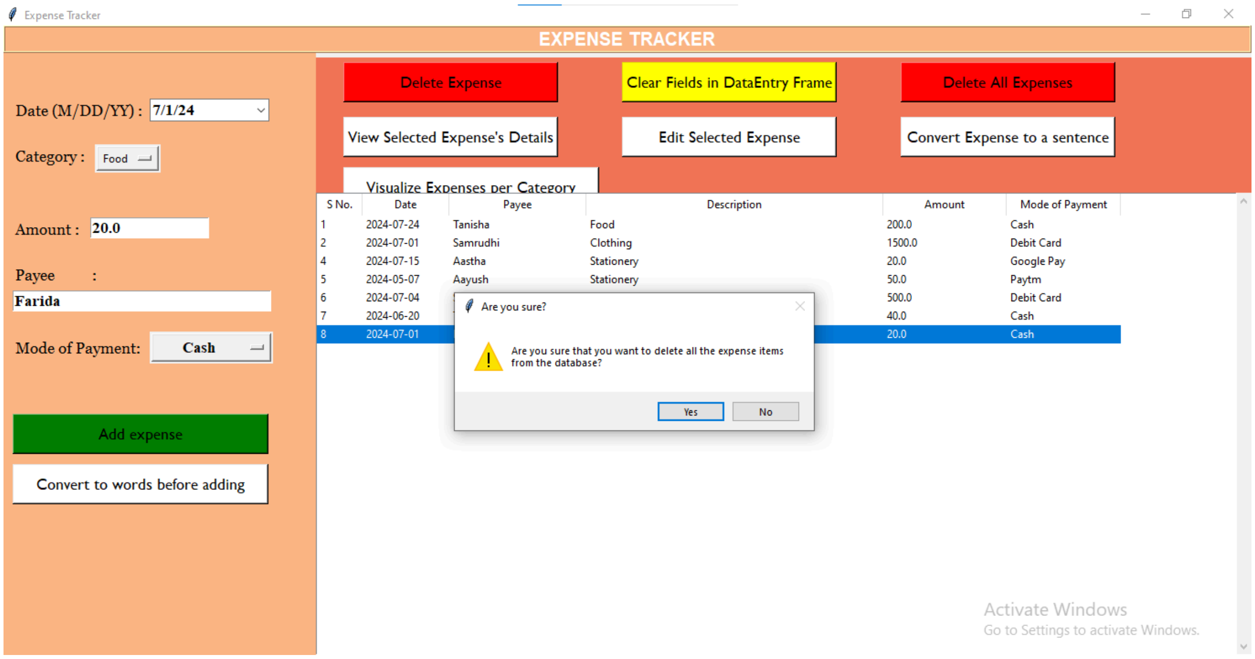
Mode of Payment: Cash

| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|
| 1 | 2024-07-24 | Tanisha | Food | 200.0 | Cash |
| 2 | 2024-07-01 | Samrudhi | Clothing | 1500.0 | Debit Card |
| 4 | 2024-07-15 | Aastha | Stationery | 20.0 | Google Pay |
| 5 | 2024-05-07 | Aayush | Stationery | 50.0 | Paytm |
| 6 | 2024-07-04 | | | 500.0 | Debit Card |
| 7 | 2024-06-20 | | | 40.0 | Cash |
| 8 | 2024-07-01 | | | 20.0 | Cash |

Here's how to read your expense

Your expense can be read like:
"You paid 200.0 to Tanisha for Food on 2024-07-24 via Cash"




Expense Tracker
EXPENSE TRACKER
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
Add expense
Convert to words before adding
Yes
No
Activate Windows
Go to Settings to activate Windows.

Date (M/DD/YY) : 7/1/24

Category : Food

Amount : 20.0

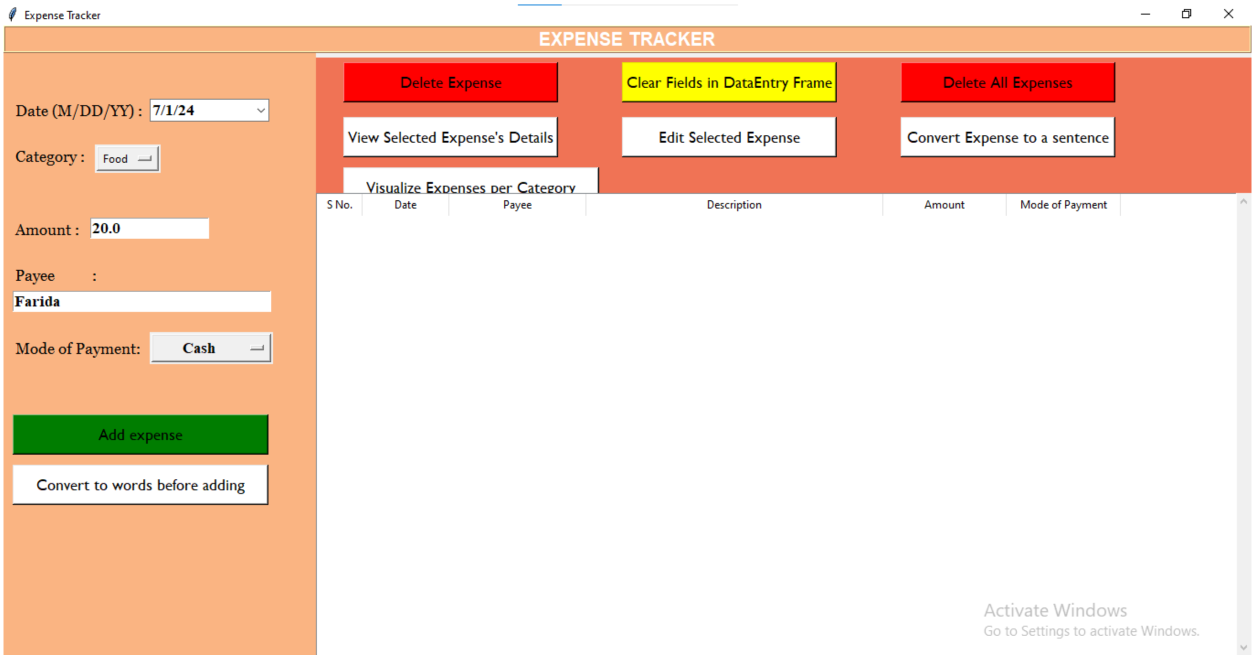
Payee : Farida

Mode of Payment: Cash

| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|
| 1 | 2024-07-24 | Tanisha | Food | 200.0 | Cash |
| 2 | 2024-07-01 | Samrudhi | Clothing | 1500.0 | Debit Card |
| 4 | 2024-07-15 | Aastha | Stationery | 20.0 | Google Pay |
| 5 | 2024-05-07 | Aayush | Stationery | 50.0 | Paytm |
| 6 | 2024-07-04 | | | 500.0 | Debit Card |
| 7 | 2024-06-20 | | | 40.0 | Cash |
| 8 | 2024-07-01 | | | 20.0 | Cash |

Are you sure?

Are you sure that you want to delete all the expense items from the database?


Expense Tracker
EXPENSE TRACKER
Delete Expense
Clear Fields in DataEntry Frame
Delete All Expenses
View Selected Expense's Details
Edit Selected Expense
Convert Expense to a sentence
Visualize Expenses per Category
Add expense
Convert to words before adding
Activate Windows
Go to Settings to activate Windows.

Date (M/DD/YY) : 7/1/24

Category : Food

Amount : 20.0

Payee : Farida

Mode of Payment: Cash

| S No. | Date | Payee | Description | Amount | Mode of Payment |
|---|---|---|---|---|---|

## III. Internship Experience

During the internship, participants engaged in practical exercises, real-world projects, and collaborative problem-solving sessions. The hands-on approach allowed for a deeper understanding of the concepts covered in the syllabus. The integration of MySQL with Python and the application of data handling techniques in real projects enhanced the overall learning experience.

## IV. Conclusion

The internship provided a comprehensive learning experience in the field of data management, visualization, and integration with databases using Python. Participants gained practical skills that are valuable in the context of real-world data science and engineering projects. The exposure to collaborative project work and problem-solving in a team setting further enriched the internship experience.

## V. Acknowledgments

We extend our gratitude to [Supervisor/Mentor Name] for their guidance and support throughout the internship period. Special thanks to [Company/Organization Name] for providing this valuable learning opportunity.

**Date:** [Date of Submission]

**Intern Signature:** [Your Signature]

**Supervisor/Mentor Signature:** [Supervisor/Mentor Signature]