**Expense Tracker Application**

def view\_expenses(expenses):

if not expenses:

print("No expenses recorded yet.")

else:

print("All expenses:")

total = 0

for i in range(len(expenses)):

desc, amount = expenses[i]

total += amount

print(f"{i + 1}. {desc}: ₹{amount}")

print(f"Total expenses: ₹{total}")

def calculate\_total\_expenses(expenses):

total = 0

for \_, amount in expenses:

total += amount

print(f"Total expenses: ₹{total}")

def main():

expenses = []

while True:

print("\nExpense Tracker Menu:")

print("1. Add Expense")

print("2. View All Expenses and Total")

print("3. Calculate Total Expenses")

print("4. Quit")

choice = input("Enter your choice: ")

if choice == "1":

description = input("Enter expense description: ")

amount = float(input("Enter expense amount in rupees: "))

expenses.append((description, amount))

print("Expense added successfully!")

elif choice == "2":

view\_expenses(expenses)

elif choice == "3":

calculate\_total\_expenses(expenses)

elif choice == "4":

print("Exiting Expense Tracker. Goodbye!")

break

else:

print("Invalid choice. Please choose a valid option.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Explanation of the above code**

def view\_expenses(expenses):

if not expenses:

print("No expenses recorded yet.")

else:

# Display all recorded expenses and their total

pass

def calculate\_total\_expenses(expenses):

total = 0

for \_, amount in expenses:

total += amount

print(f"Total expenses: ₹{total}")

* These are two functions defined to handle the visualization and calculation of expenses.
* **view\_expenses** takes a list of expenses as input and displays them along with their total. If there are no expenses, it prints a message saying so.
* **calculate\_total\_expenses** calculates the total expenses from the list of expenses passed to it.

def main():

expenses = []

while True:

# Display menu options

Pass

* This is the **main** function where the program starts its execution.
* It initializes an empty list **expenses** to store the expense data.
* It contains a **while** loop that runs indefinitely until the user chooses to exit.

print("\nExpense Tracker Menu:")

print("1. Add Expense")

print("2. View All Expenses and Total")

print("3. Calculate Total Expenses")

print("4. Quit")

choice = input("Enter your choice: ")

* This code snippet prints out the menu options for the user to choose from.
* It prompts the user to input their choice and stores it in the variable **choice**.

if choice == "1":

# Add a new expense

pass

elif choice == "2":

view\_expenses(expenses)

elif choice == "3":

calculate\_total\_expenses(expenses)

elif choice == "4":

print("Exiting Expense Tracker. Goodbye!")

break

else:

print("Invalid choice. Please choose a valid option.")

* This part of the code is responsible for handling the user's input based on their choice.
* If the user chooses to add a new expense, it will execute the corresponding code (which is not shown in the snippet).
* If the user chooses to view all expenses, it will call the **view\_expenses** function and pass the **expenses** list to it.
* If the user chooses to calculate the total expenses, it will call the **calculate\_total\_expenses** function and pass the **expenses** list to it.
* If the user chooses to quit, it will print a farewell message and break out of the loop, ending the program.

expenses.append((description, amount))

This line of code adds a new expense to the **expenses** list. The expense is represented as a tuple containing the description and the amount.

print(f"{i + 1}. {desc}: ₹{amount}")

print(f"Total expenses: ₹{total}")

These lines of code are responsible for displaying the expenses and the total expenses in Indian Rupees (₹).

**Assignment: Expense Tracker Application**

**Objective:** The objective of this assignment is to develop an Expense Tracker Application in Python. The application should allow users to add expenses, view all expenses, calculate the total expenses, and include additional functionalities to enhance the user experience.

**Instructions:**

* Develop a Python program that implements the Expense Tracker Application.
* Use the provided code snippets as a reference to structure your program.
* Add extra functionalities to the application to improve its usability and user experience.
* Document your code effectively with comments to explain the purpose of each function and major sections of code.
* Test your program thoroughly to ensure it works correctly and handles different scenarios gracefully.

**Basic Requirements:**

1. The application should allow users to add a new expense with a description and amount.
2. Users should be able to view all recorded expenses along with their total.
3. The application should calculate and display the total expenses.
4. Implement error handling to validate user input and handle unexpected scenarios gracefully.

**Additional Functionalities:**

1. **Delete Expense:** Allow users to delete a specific expense from the list of recorded expenses.
2. **Edit Expense:** Allow users to edit the description or amount of a recorded expense.
3. **Expense Categories:** Allow users to categorize expenses into different categories (e.g., Food, Transportation, Entertainment) and provide an option to view expenses by category.
4. **Export/Import Data:** Implement functionalities to export all recorded expenses to a file and import expenses from a file.
5. **Monthly Reports:** Provide an option to generate monthly expense reports, showing total expenses for each month.
6. **Currency Conversion:** Allow users to choose a currency for displaying expenses and implement currency conversion functionalities.
7. **Reminder Notifications:** Implement reminder notifications to remind users of upcoming expenses or due dates.
8. **Graphical User Interface (GUI):** Develop a graphical user interface for the application using a library such as Tkinter or PyQt.

**Note:** You are encouraged to explore additional functionalities beyond the ones listed above to enhance the functionality and usability of your Expense Tracker Application. Be creative and innovative in your approach!