expense-tracker-2

August 19, 2024

Task: As a Python developer, you are tasked to create an expense tracker application that helps users manage and analyze their daily expenses by providing functionalities for expense management, user authentication, and reporting. Action: 1. Open the Python environment: a. Open the Python practice lab and start your work in a blank code file. 2. Create Classes and Methods: a. Define a class named Expense to represent an expense record b. Use the **init** method to initialize an expense object with the following attributes: i. expense_id: A unique identifier for the expense ii. date: The date of the expense iii. category: The category of the expense (e.g., food, transportation) iv. description: A brief description of the expense v. amount: The amount spent c. Define the **str** method to return the string representation of the expense object 3. Data Storage: a. Create an empty list named expenses to store expense records b. Define the following functions to manipulate the list: i. add_expense(expense): Adds a new expense object to the list ii. update_expense(expense_id, new_expense): Updates an existing expense object based on expense_id iii. delete_expense(expense_id): Deletes an expense object from the list based on expense_id iv. display_expenses(): Displays all expense objects in the list 4. User Authentication: a. Create a dictionary named users with predefined usernames and passwords b. Define the function authenticate_user(username, password). This function will: i. Check if the provided username exists in the user's dictionary ii. Verify if the provided password matches the password in the user dictionary iii. Print a success message if authentication is successful; otherwise, print a failure message iv. Return True if authentication is successful, otherwise returns False 5. Categorization and Summarization: a. Define the function categorize_expenses(). This function will: i. Create an empty dictionary named categories ii. Iterate over each expense in the expenses list iii. Add the expense amount to the corresponding category in the categories dictionary iv. Return the categories dictionary b. Define the function summarize_expenses(). This function will: i. Initialize a variable total to 0 ii. Iterate over each expense in the expenses list and add the expense amount to total iii. Return the total sum of expenses 6. Functions for Repetitive Tasks: a. Define the function calculate_total_expenses(): i. Use a generator expression to sum the amount of all expenses in the expenses list ii. Return the total sum of expenses b. Define the function generate_summary_report(): i. Call categorize_expenses() to get the categorized expenses ii. Print the total amount for each category iii. Print the total sum of all expenses by calling calculate_total_expenses() 7. Simple CLI for Interaction: a. Define the function cli () to provide a menu for user interaction: i. Print the menu options ii. Take user input to select an option iii. Write if-Elif conditions to execute the corresponding function based on user input: 1. Adds a new expense 2. Updates an existing expense 3. Deletes an expense 4. Displays all expenses 5. Generates a summary report 6. Exits the application b. Call the authenticate_user() function before showing the menu to ensure the user is authenticated 8. Run the program and verify the results: a. Run the program by executing the file in your Python environment b. Follow the prompts and inputs to simulate user interactions and admin functions

[ ]:
```
from collections import defaultdict


users={"user1":"user1pass","user2":"user2pass","user3":"user3pass"}

class Expense():
    def __init__(self,expense_id,date,category,description,amount):
        self.expense_id=expense_id
        self.date=date
        self.category=category
        self.description=description
        self.amount=amount

    def __str__(self):
        return f"{self.expense_id}\t{self.date}\t\t{self.category}\t\t\t{self.
 ↪description}\t\t{self.amount} "

expenses=[]


def add_expense():
    date=input("Enter the date (YYYY-MM-DD): ")
    category=input("Enter expense category : ")
    description=input("Enter description: ")
    amount=int(input("Enter the amount: "))
    if len(expenses)>0:
        expenses.append(Expense(1+int(expenses[-1].
 ↪expense_id),date,category,description,amount))
    else:
        expenses.append(Expense(1,date,category,description,amount))

def delete_expense(expense_id):
        for item in expenses:
                if item.expense_id == int(expense_id):
                        expenses.remove(item)
                        print("Expense Deleted!")
                        return
        print("Deletion failed!")

def update_expense(expense_id):
        for item in expenses:
                if item.expense_id == int(expense_id) :
                        print("Enter details to update: ")
                        date_up = input("Enter the date (YYYY-MM-DD): ")
                        if date_up:
                                item.date = date_up
                        catg=input("Enter Category :")
                        if catg:
                                item.category = catg
                        desc=input("Enter Description :")
                        if desc:
                                item.descripiton = desc
                        amt=input("Enter Amount :")
                        if amt:
                                item.amount = int(amt)
                        print("Expense Updated!")
                        return
        print('Could not be Updated!')

def display_expenses():
        if len(expenses)==0:
                print("Currently no expenses!")
                return
        print("Exp_ID\tDate\t\t\tCategory\t\t\tDescription\t\tAmount")
        for item in expenses:
                print(str(item))




def authenticate_user(username, password):
    if (username, password) in users.items():
        print("Authentication Successfull")
        return True
    else:
        print("Authentication Failed")
        return False

def categorize_expenses():
        cat_expense=defaultdict(list)
        for item in expenses:
                cat_expense[item.category].append(item)
        return cat_expense

def summarize_expenses(catg_expense):
        catg_total=defaultdict(int)
        print("List of Expenses: ")
        for catg,exp in catg_expense.items():
                catg_amount=0
                for expense in exp:
                        print(expense)
                        catg_amount+=expense.amount
                catg_total[catg]=catg_amount
        return catg_total
```

```
def calculate_total():
        total=0
        for item in expenses:
                total+=item.amount
        return total

def generate_summary():
        if len(expenses)==0:
                print("Expense List is Empty")
                return
        categorized_exp=categorize_expenses()
        expense_total=summarize_expenses(categorized_exp)
        print("CATEGORY WISE EXPENSE : \n")
        for item,exp in categorized_exp.items():
                print(item)
                print("------------")
                print("ID\tDate\t\t\tCategory\t\t\tDescription\t\tAmount")
                for expense in exp:
                        print(str(expense))
                print(f"Total expenditure for {item} is {expense_total[item]}")
        print(f"Expense Total : {calculate_total()}")


def cli():
    while True:
        print("MENU\n____")
        print(f"1.Add a new expense \n2.Update an existing expense \n3.Delete␣
 ↪an expense \n4.Display all expenses \n5.Generate a summary report \n6.Exit␣
 ↪the application")
        ch= int(input("Enter choice:"))
        if ch==1:
            add_expense()
        elif ch==2:
            up=int(input("Enter the expense id of the entry to be updated : "))
            update_expense(up)
        elif ch==3:
            de=int(input("Enter the expense of the entry to be deleted : "))
            delete_expense(de)
        elif ch==4:
            display_expenses()
        elif ch==5:
            generate_summary()
        elif ch==6:
            exit()
        else:
            print("Invalid Option. Select one from the menu!!!")
```

```
def Expense_Tracker():
        print("EXPENSE TRACKER")
        print("Login\n-----")
        username=input("Enter your username: ")
        password=input("Enter your password: ")
        if authenticate_user(username,password):
                print("Logged in Successfully\n")
                cli()
        else:
                print("Invalid Username or Password\n")
                Expense_Tracker()

Expense_Tracker()
```

[ ]:
```
"""
EXPENSE TRACKER
Login
-----
Enter your username:  user1
Enter your password:  user1pass
Authentication Successfull
Logged in Successfully

MENU

----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 1
Enter the date (YYYY-MM-DD):  2024-12-30
Enter expense category :  Food
Enter description:  Lunch
Enter the amount:  200
MENU

----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 1
Enter the date (YYYY-MM-DD):  2023-11-24
Enter expense category :  Travel
Enter description:  Work
Enter the amount:  150
MENU

----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 1
Enter the date (YYYY-MM-DD):  2022-05-27
Enter expense category :  Food
Enter description:  Party
Enter the amount:  3500
MENU

----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 1
Enter the date (YYYY-MM-DD):  2024-05-08
Enter expense category :  Lifestyle
Enter description:  Club
Enter the amount:  10000
MENU

----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 4
Exp_ID        Date                          Category                        Description
1        2024-12-30                Food                           Lunch                 200␣
  ↪
2        2023-11-24                Travel                           Work                 150␣
  ↪
3        2022-05-27                Food                           Party                 3500␣
  ↪
4        2024-05-08                Lifestyle                           Club                 10000␣
  ↪
```

```
"""
```

```
[ ]: """
MENU
----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 2
Enter the expense id of the entry to be updated :  3
Enter details to update:
Enter the date (YYYY-MM-DD):  2023-08-09
Enter Category : Food
Enter Description : Party
Enter Amount : 4000
Expense Updated!
MENU
----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 2
Enter the expense id of the entry to be updated :  4
Enter details to update:
Enter the date (YYYY-MM-DD):  4
Enter Category : Food
Enter Description : Lunch
Enter Amount : 340
Expense Updated!
MENU
----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 3
Enter the expense of the entry to be deleted :  4
Expense Deleted!
"""
```

```
[ ]: """
MENU
----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 4
Exp_ID        Date                        Category                          Description
1         2024-12-30                Food                        Lunch                     200␣
 ↪
2         2023-11-24                Travel                        Work                     150␣
 ↪
3         2023-08-09                Food                        Party                     4000␣
 ↪
MENU
----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 5
List of Expenses:
1         2024-12-30                Food                        Lunch                     200␣
 ↪
3         2023-08-09                Food                        Party                     4000␣
 ↪
2         2023-11-24                Travel                        Work                     150␣
 ↪
CATEGORY WISE EXPENSE :

Food
-----------
ID        Date                        Category                        Description
1         2024-12-30                Food                        Lunch                     200␣
 ↪
3         2023-08-09                Food                        Party                     4000␣
 ↪
Total expenditure for Food is 4200
Travel
-----------
ID        Date                        Category                        Description
```

```
2        2023-11-24                Travel                       Work                150␣
  ↪
Total expenditure for Travel is 150
Expense Total : 4350
MENU
----
1.Add a new expense
2.Update an existing expense
3.Delete an expense
4.Display all expenses
5.Generate a summary report
6.Exit the application
Enter choice: 6
"""
```