

# expense-tracker-1

August 17, 2024

```
[5]: """
QUESTION
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.
"""

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"Expense id: {self.expense_id}, date: {self.date}, category:
↳ {self.category}, description: {self.description}, amount: {self.amount}"

expenses = []

def add_expense(expense):
    # Adds a new expense object to the list
    expenses.append(expense)

def update_expense(expense_id, new_expense):
    # Updates an existing expense object based on expense_id
    for i in expenses:
        if i.expense_id == expense_id:
            i.date = new_expense.date
            i.category = new_expense.category
            i.description = new_expense.description
            i.amount = new_expense.amount

def delete_expense(expense_id):
    # Deletes an expense object from the list based on expense_id
    for i in expenses:
```

```

        if(i.expense_id==expense_id:
            expenses.remove(i)

def display_expenses():
    #Displays all expense objects in the list
    for i in expenses:
        print(i)

def categorize_expenses():
    categories={}
    category=[]
    for i in expenses:
        if(i.category in category):
            categories[i.category]+=int(i.amount)
        else:
            category.append(i.category)
            categories[i.category]=int(i.amount)
    return categories

def summarize_expenses():
    total=0
    for i in expenses:
        total+=int(i.amount)
    return total

def cli():
    while(True):
        choice=int(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"""))
        if choice==1:
            expense_id=int(input("Enter the expense id:"))
            date=input("Enter the date")
            category=input("Enter the category")
            description=input("Enter the description")
            amount=input("Enter the amount")
            expense=Expense(expense_id,date,category,description,amount)
            add_expense(expense)
        elif choice==2:
            expense_id=int(input("Enter the expense id to update:"))
            date=input("Enter the new date: ")
            category=input("Enter the new category: ")
            description=input("Enter the new description: ")

```

```

        amount=input("Enter the amount: ")
        expense=Expense(expense_id,date,category,description,amount)
        update_expense(expense_id,expense)
    elif choice==3:
        expense_id=int(input("Enter the expense id to update:"))
        delete_expense(expense_id)
    elif choice==4:
        display_expenses()
    elif choice==5:
        print("Expense per category: ",categorize_expenses())
        print("Total expense: ",summarize_expenses())
    elif choice==6:
        print("exit")
        break
    else:
        print("invalid")

username=input("Enter the username: ")
password=input("Enter the password: ")

user_db={"username":"Melissa","password":"Chakki"}

def authenticate_user(username,password):
    if(username==user_db["username"] and password==user_db["password"]):
        print("Login successful as user")
        return True
    else:
        print("Login failed")
        return False

if(authenticate_user(username,password)):
    cli()
else:
    print("Incorrect username or password")

```

Enter the username: Melissa

Enter the password: Chakki

Login successful as user

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 4

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 1

Enter the expense id: 1  
Enter the date 17/08/2024  
Enter the category Food  
Enter the description Biryani  
Enter the amount 150

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 1

Enter the expense id: 2  
Enter the date 16/08/2024  
Enter the category Food  
Enter the description Snacks  
Enter the amount 40

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 4

Expense id: 1, date: 17/08/2024, category: Food, description: Biryani, amount: 150

Expense id: 2, date: 16/08/2024, category: Food, description: Snacks, amount: 40

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 2

Enter the expense id to update: 2  
Enter the new date: 16/08/2024  
Enter the new category: Snacks  
Enter the new description: Bread pocket  
Enter the amount: 20

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 4

Expense id: 1, date: 17/08/2024, category: Food, description: Biryani, amount: 150

Expense id: 2, date: 16/08/2024, category: Snacks, description: Bread pocket, amount: 20

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 5

Expense per category: {'Food': 150, 'Snacks': 20}

Total expense: 170

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 3
- Enter the expense id to update: 2

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 4

Expense id: 1, date: 17/08/2024, category: Food, description: Biryani, amount: 150

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 6

exit

[ ]: