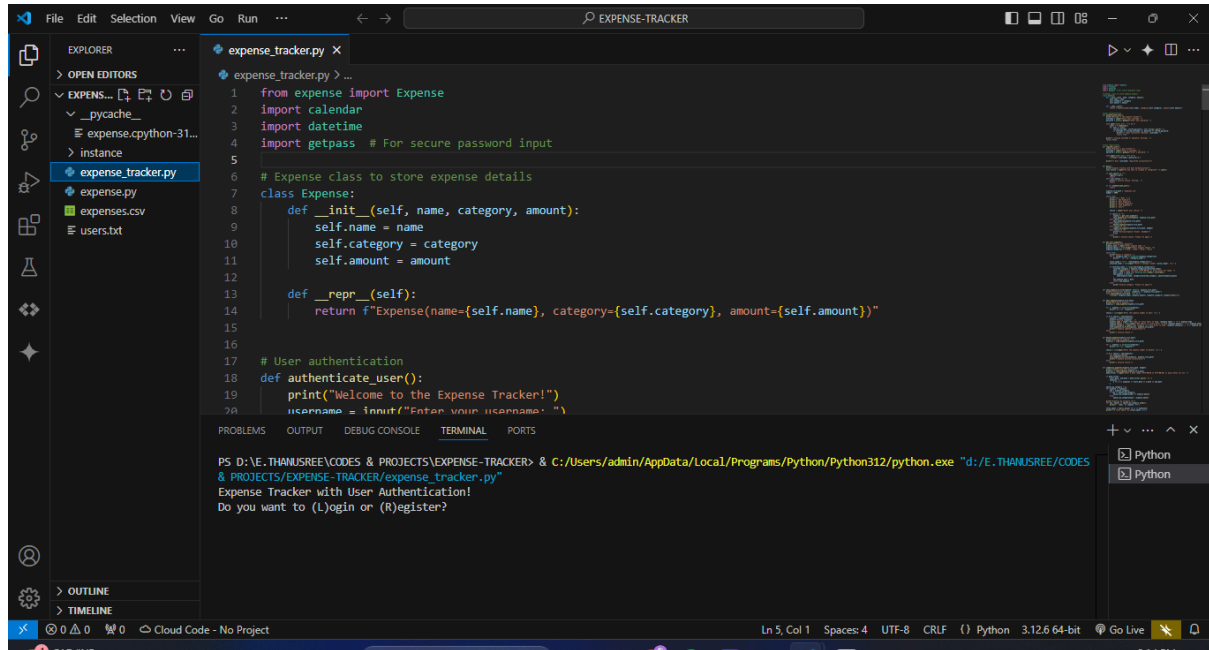


PennyWise: EXPENSE_TRACKER

1.CSV Version



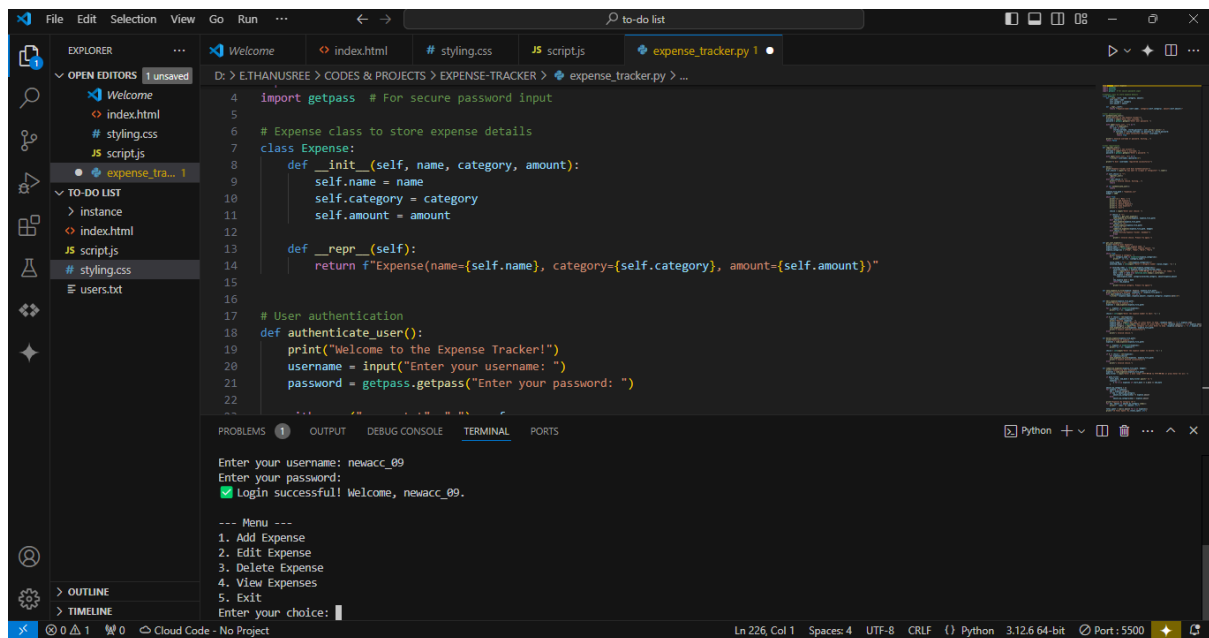
The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the project structure. The main editor window shows the `expense_tracker.py` file with the following code:

```
1 from expense import Expense
2 import calendar
3 import datetime
4 import getpass # For secure password input
5
6 # Expense class to store expense details
7 class Expense:
8     def __init__(self, name, category, amount):
9         self.name = name
10        self.category = category
11        self.amount = amount
12
13    def __repr__(self):
14        return f"Expense(name={self.name}, category={self.category}, amount={self.amount})"
15
16 # User authentication
17 def authenticate_user():
18     print("Welcome to the Expense Tracker!")
19     username = input("Enter your username: ")
20     password = getpass.getpass("Enter your password: ")
21     # ... (rest of the code is partially visible and blurred)
```

The terminal window at the bottom shows the command prompt output:

```
PS D:\E.THANUSREE\CODES & PROJECTS\EXPENSE-TRACKER> & C:\Users\admin\AppData\Local\Programs\Python\Python312/python.exe "d:\E.THANUSREE\CODES & PROJECTS\EXPENSE-TRACKER\expense_tracker.py"
Expense Tracker with User Authentication!
Do you want to (L)ogin or (R)egister?
```

- Allows user to register and then login using their credentials. These credentials will be stored in the .csv file(extension)



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the project structure. The main editor window shows the `expense_tracker.py` file with the following code:

```
4 import getpass # For secure password input
5
6 # Expense class to store expense details
7 class Expense:
8     def __init__(self, name, category, amount):
9         self.name = name
10        self.category = category
11        self.amount = amount
12
13    def __repr__(self):
14        return f"Expense(name={self.name}, category={self.category}, amount={self.amount})"
15
16 # User authentication
17 def authenticate_user():
18     print("Welcome to the Expense Tracker!")
19     username = input("Enter your username: ")
20     password = getpass.getpass("Enter your password: ")
21     # ... (rest of the code is partially visible and blurred)
```

The terminal window at the bottom shows the command prompt output with user interaction:

```
Enter your username: newacc_09
Enter your password:
✔ Login successful! Welcome, newacc_09.

--- Menu ---
1. Add Expense
2. Edit Expense
3. Delete Expense
4. View Expenses
5. Exit
Enter your choice: |
```

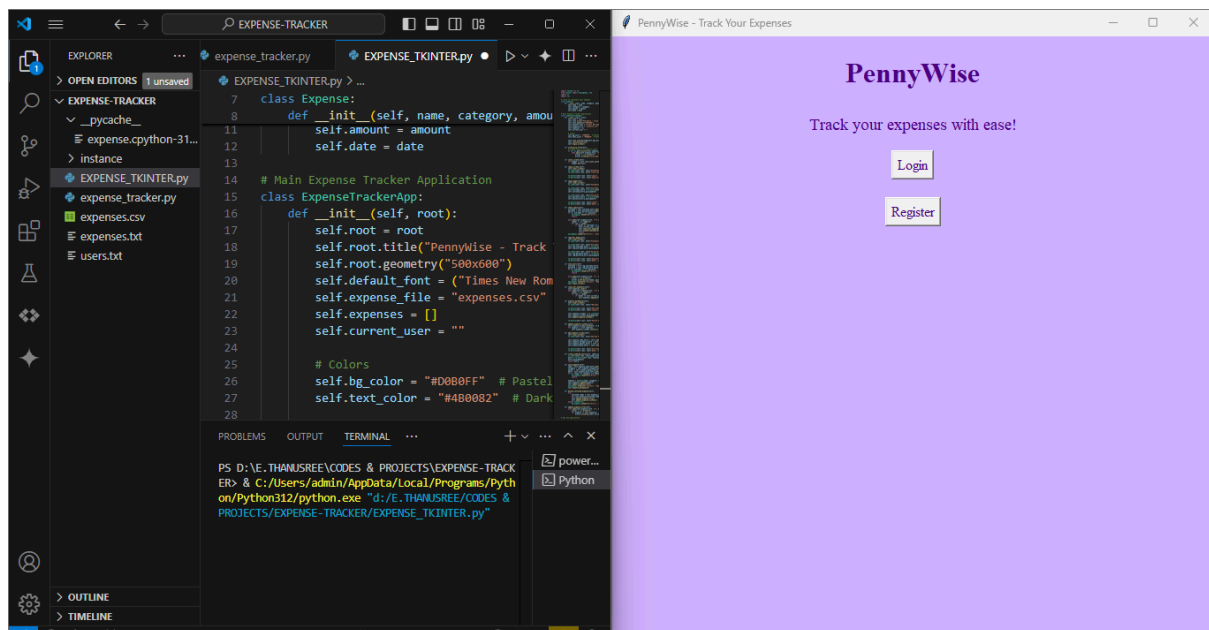
- After logging in, the user is then prompted to a menu based service, allowing them to select and enter the choice to perform the respective function.

```
4 import getpass # For secure password input
5
6 # Expense class to store expense details
7 class Expense:
8     def __init__(self, name, category, amount):
9         self.name = name
10        self.category = category
11        self.amount = amount
12
13    def __repr__(self):
14        return f"Expense(name={self.name}, category={self.category}, amount={self.amount})"
15
16 # User authentication
17 def authenticate_user():
18     print("Welcome to the Expense Tracker!")
19     username = input("Enter your username: ")
20     password = getpass.getpass("Enter your password: ")
21
22 ...
```

Enter expense name: Personal
Enter expense amount: 350
Select a category:
1. Food
2. Home
3. Work
4. Fun
Enter a category number [1 - 4]: 1
Enter the date (YYYY-MM-DD) on press Enter:
Saving User Expense: Expense(name=Personal, category=Food, amount=350.0, date=2024-11-17) to expenses.csv

- Followed by the categories which makes it easier to provide the monthly summary of the expenses.
- This data will be saved in a .csv file extension.

2.TKINTER Version



- Using tkinter as GUI, the same program has been created in order to track your expenses in a much easier manner, where the data gets saved in a .txt file extension.