

Python Development

ATM Simulator

Student Name: Senay KULA

Student Number: 699375422 Instructor: Ami Dave

Date: 17July 2024

## **Table of Contents**

1. Introduction
2. Project Specifications
3. System Design
   * Diagrams
   * Flowcharts
   * Pseudocode
4. Implementation
5. User Manual
6. Testing and Test Data
7. Testing Screens
8. Conclusion

## **1. Introduction**

This project report details the development of an ATM simulator. The simulator allows users to perform basic banking functions such as checking balances, withdrawing money, and depositing money. Additionally, it includes an admin mode for managing user accounts.

## **2. Project Specifications**

* **Programming Language:** Python
* **External Libraries:** matplotlib for plotting user balances
* **Data Storage:** JSON file to store user information

## **3. System Design**

### **Diagrams**

#### **Class Diagram**

### **Flowcharts**

#### **User Authentication Flowchart**

+-------------------+ | Start | +-------------------+ | v +-------------------+ | Enter Username | +-------------------+ | v +-------------------+ | Enter PIN | +-------------------+ | v +----------------------------+ | Authenticate (username, PIN)| +----------------------------+ | v +----------------------------+ | Success? | +----------------------------+ | | Yes No | | v v +-----------------------+ +-----------------+ | Display Main Menu | | Display Error | +-----------------------+ +-----------------+ | | v v +-----------------------+ +-----------------+ | User or Admin? | | Restart | +-----------------------+ +-----------------+

### **Pseudocode**

#### **Main Program Flow**

BEGIN

LOAD users from users.json

WHILE True DO

DISPLAY login prompt

IF authenticate(username, pin) THEN

IF username == "SysAdmin" THEN

DISPLAY admin\_menu()

ELSE

DISPLAY user\_menu(username)

ELSE

DISPLAY error message

END

## **4. Implementation**

The ATM simulator is implemented in Python. The main functionalities include user authentication, balance checking, withdrawing and depositing money, and admin functionalities like adding and deleting users, viewing all balances, and plotting user balances.

Main Class: ATM

class ATM:

def \_\_init\_\_(self):

self.users\_file = 'users.json'

self.load\_users()

def load\_users(self):

...

def save\_users(self):

...

def authenticate(self, username, pin):

...

def main\_menu(self):

...

def user\_menu(self, username):

...

def plot\_balances(self):

...

def admin\_menu(self):

...

def check\_balance(self, username):

...

def withdraw(self, username):

...

def deposit(self, username):

...

def change\_pin(self, username):

...

def add\_user(self):

...

def delete\_user(self):

...

def view\_all\_balances(self):

...

## **5. User Manual**

### **Getting Started**

**Run the Program:**

python atm\_simulator.py

**Login:**

* Enter your username.
* Enter your 4-digit PIN.

### **User Menu**

**Check Balance:**

* + Select option 1 to view your current balance.

**Withdraw Money:**

* + Select option 2 and enter the amount to withdraw (must be a multiple of 10 and up to $1000).

**Deposit Money:**

* + Select option 3 and enter the amount to deposit.

**Change PIN:**

* + Select option 4 to change your 4-digit PIN.

**Exit:**

* + Select option 5 to log out.

### **Admin Menu**

**Add User:**

* + Select option 1 to add a new user with a username and 4-digit PIN.

**Delete User:**

* + Select option 2 to delete an existing user.

**View All Balances:**

* + Select option 3 to view the balances of all users.

**Plot User Balances:**

* + Select option 4 to display a bar chart of user balances.

**Exit:**

* + Select option 5 to log out.

## **6. Testing and Test Data**

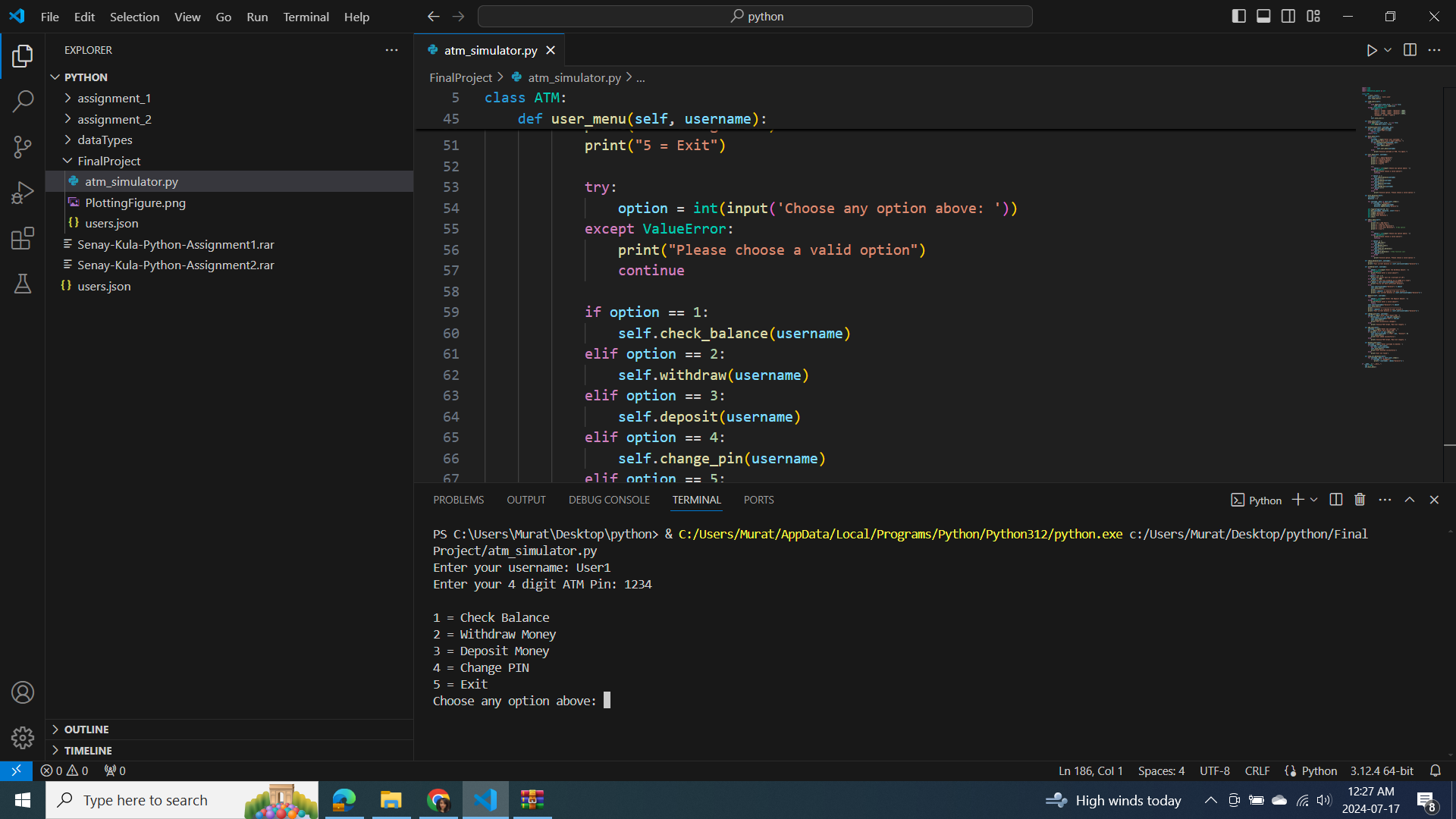
### **Test Cases**

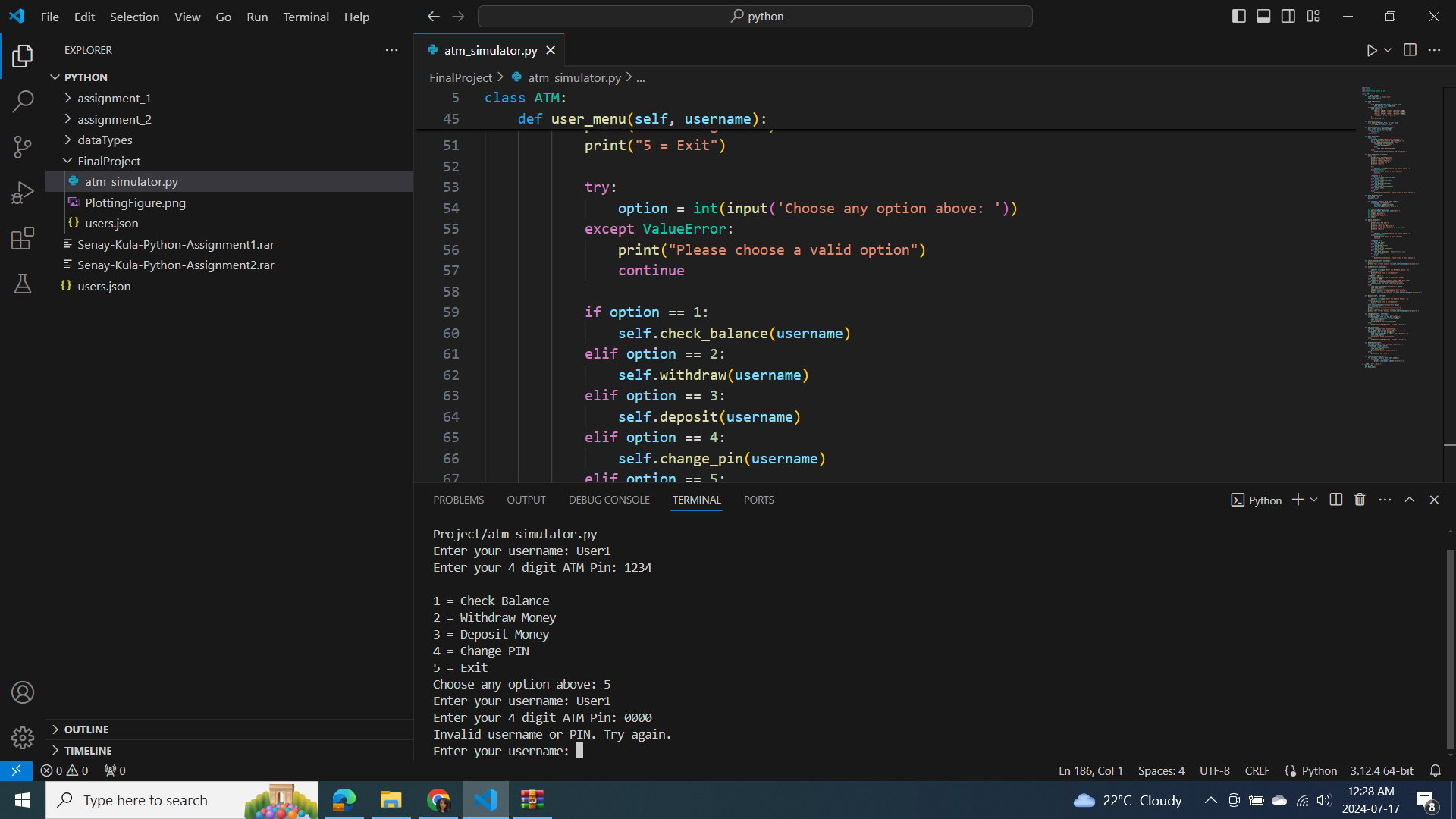
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| TC01 | Successful login | Username: User1, PIN: 1234 | Main menu displayed | Main menu displayed | Pass |
| TC02 | Incorrect PIN | Username: User1, PIN: 0000 | Error message | Error message | Pass |
| TC03 | Check balance | Option: 1 | Current balance displayed | Current balance displayed | Pass |
| TC04 | Withdraw amount (valid) | Option: 2, Amount: 100 | Balance updated, success message | Balance updated, success message | Pass |
| TC05 | Withdraw amount (invalid) | Option: 2, Amount: 1500 | Error message | Error message | Pass |
| TC06 | Deposit amount | Option: 3, Amount: 500 | Balance updated, success message | Balance updated, success message | Pass |
| TC07 | Change PIN | Option: 4, New PIN: 5678 | PIN updated, success message | PIN updated, success message | Pass |

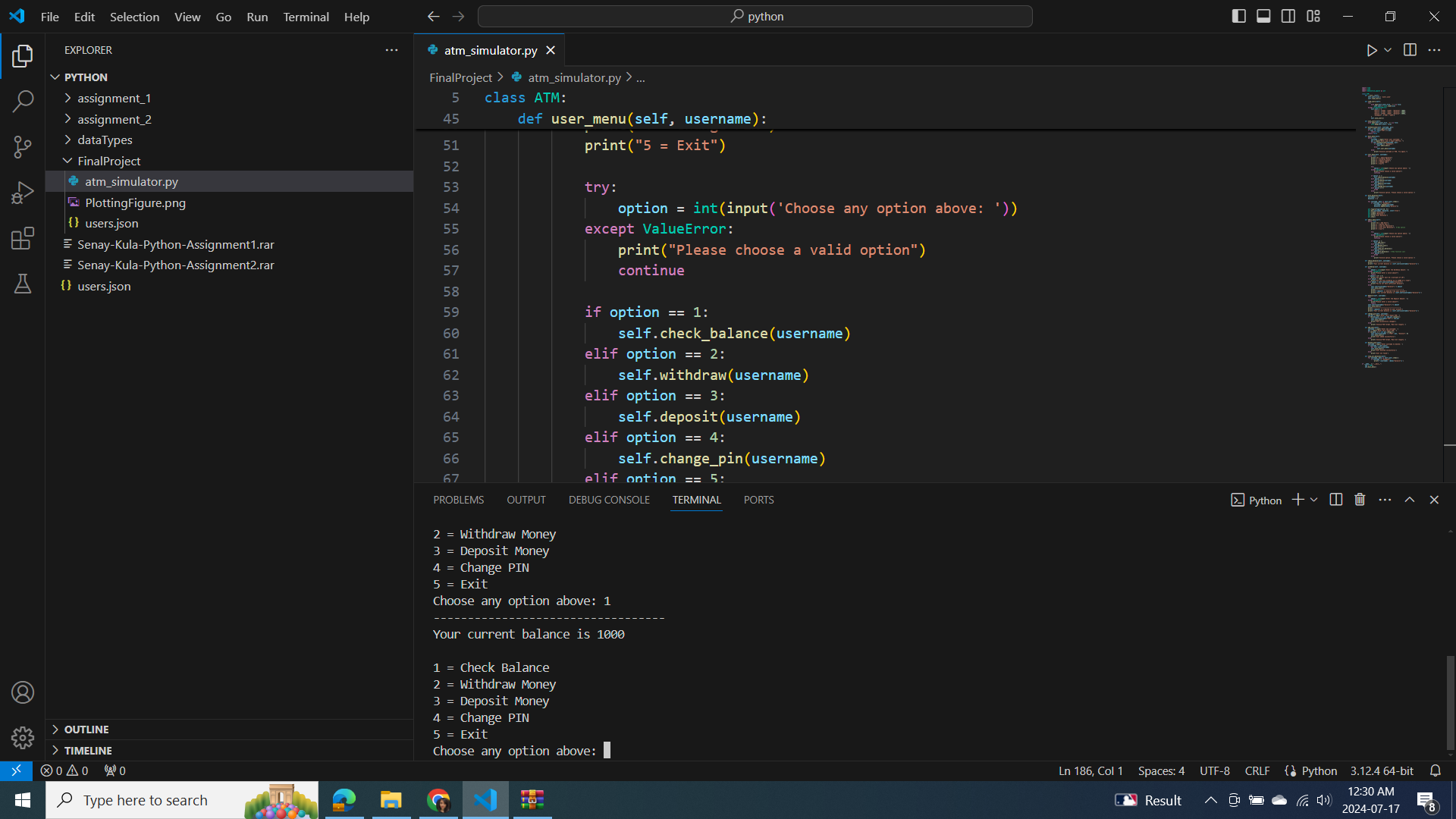
### **Test Data**

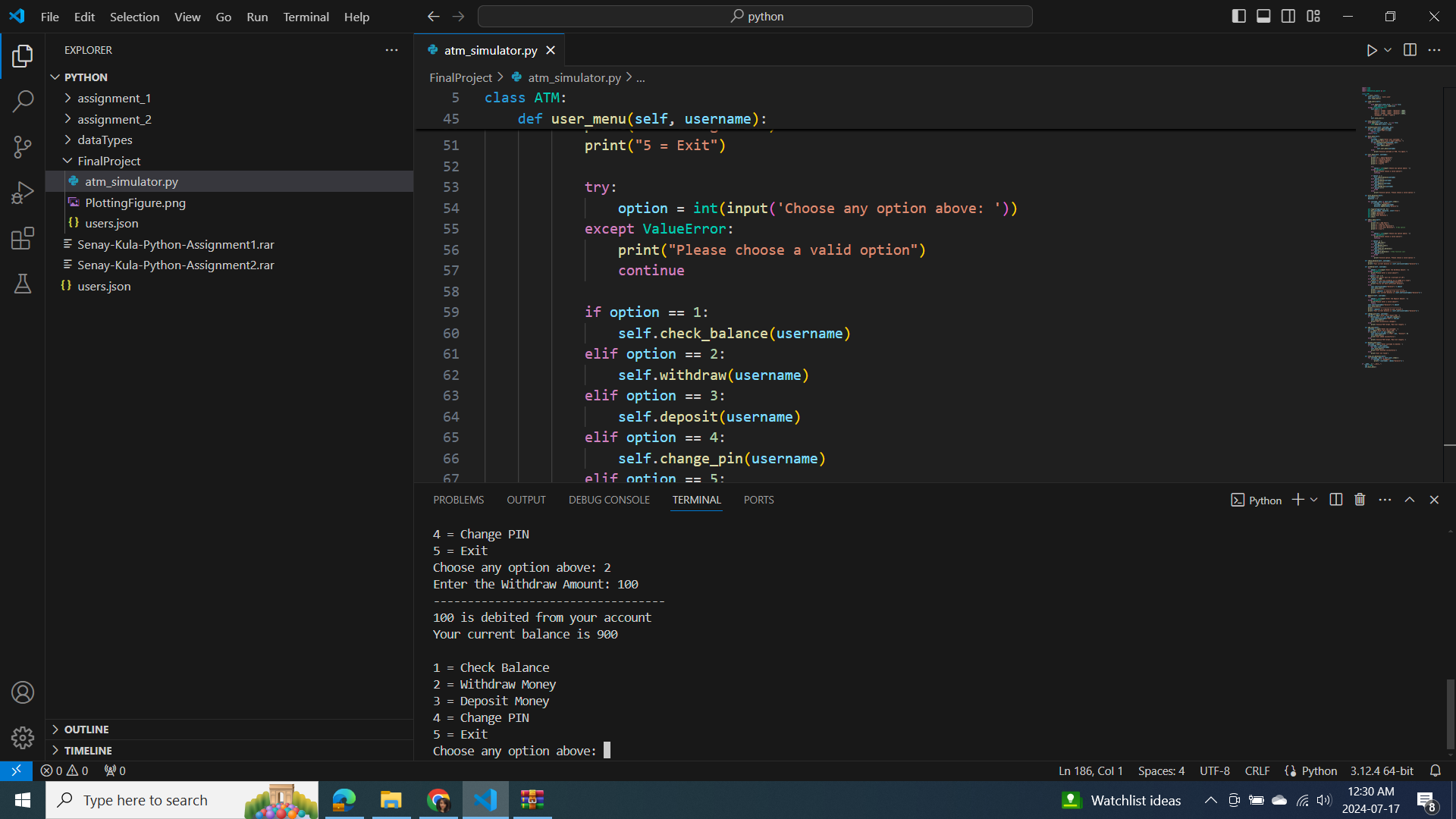
* **User1:** Username: User1, PIN: 1234, Balance: $1000
* **User2:** Username: User2, PIN: 2222, Balance: $2000
* **User3:** Username: User3, PIN: 3333, Balance: $3000
* **SysAdmin:** Username: SysAdmin, PIN: 1357

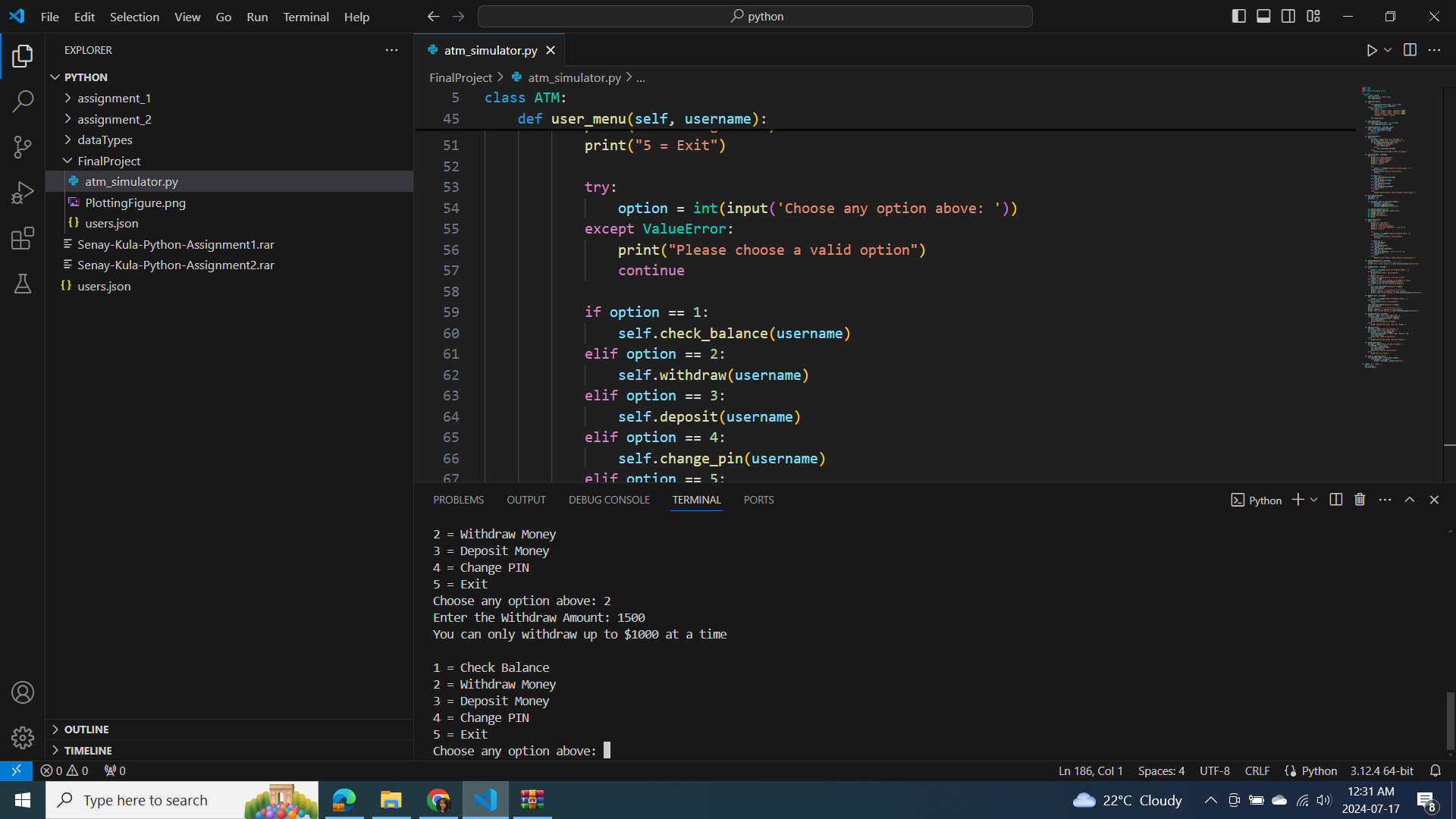
**7. Test SCREENS:**

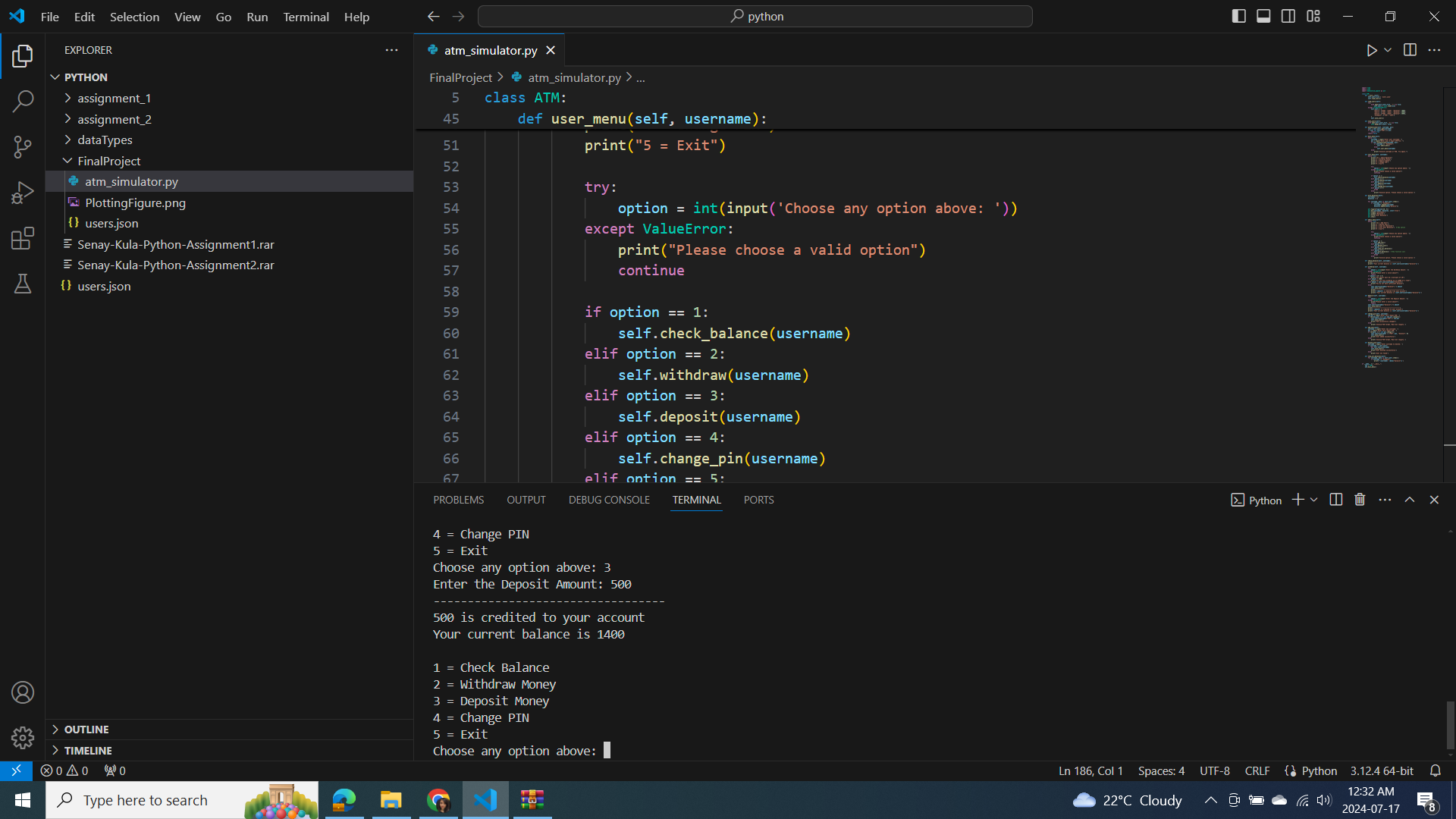


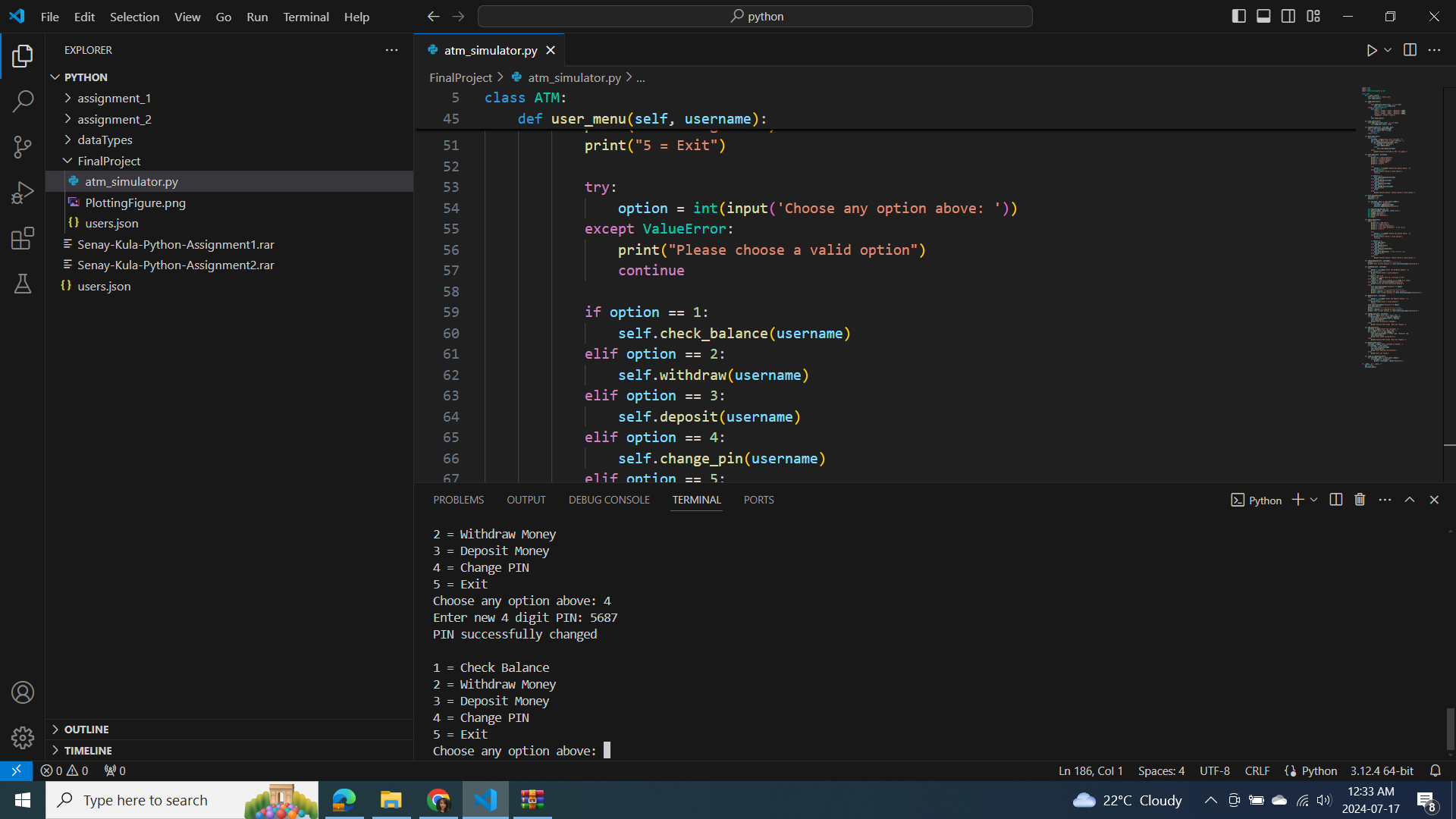












## **8. Conclusion**

The ATM simulator provides a robust simulation of basic ATM functionalities. It supports user authentication, balance management, and an admin interface for user management. This project demonstrates the effective use of Python for developing a user-interactive application with data persistence using JSON files.