Terminal Application T1A3

Jennifer Lai
14407@coderacademy.edu.au
https://github.com/jennlai95/T1A3_Terminal_application

Hotel Booking System

```
# About

This is a hotel booking system application that allows clients/users to view available rooms and make a booking.
It also lets users check for previous bookings
```

Menu
The first featue is the main menu that displays all the available features and functions of the program. This allows them to select their choices or to exit the program

Below is the discord discussion for this application.

View and enter current user
This inputs the current user so it can be saved

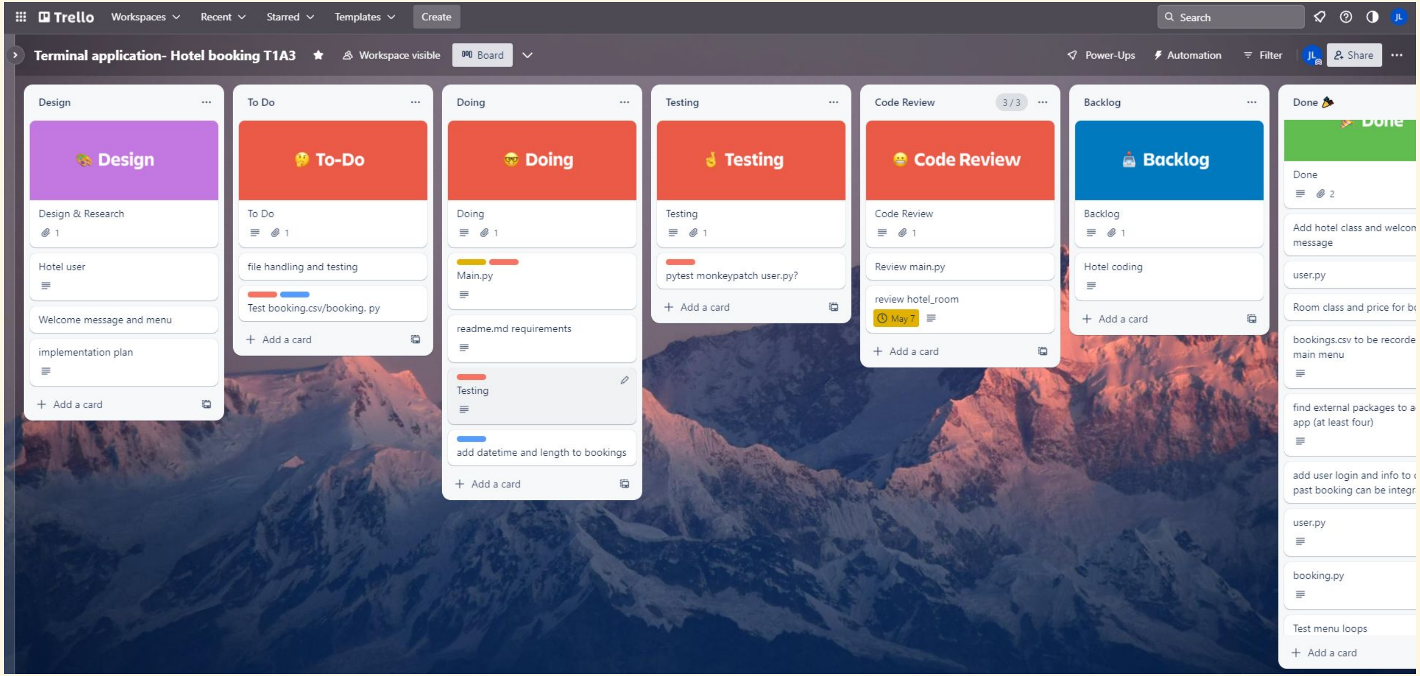
View available rooms and process
This allows users to view the current available rooms and prices, it will also prompt the users if they want to make a booking or if they want to continue browsing

Booking
This allows the users to make a booking and choose the length of their stay

Booking records
This provides a list of the previously made bookings and the user who made the booking.

Exit

Implementation plan



Terminal application- Hotel booking T1A3

Design
- Design
- Design & Research
- Hotel user
- Welcome message and menu
- implementation plan
- Add a card

To Do
- To-Do
- To Do
- file handling and testing
- Test booking.csv/booking. py
- Add a card

Doing
- Doing
- Doing
- Main.py
- readme.md requirements
- Testing
- add datetime and length to bookings
- Add a card

Testing
- Testing
- Testing
- pytest monkeypatch user.py?
- Add a card

Code Review 3 / 3
- Code Review
- Code Review
- Review main.py
- review hotel_room May 7
- Add a card

Backlog
- Backlog
- Backlog
- Hotel coding
- Add a card

Done
- Done
- Done
- Add hotel class and welcon message
- user.py
- Room class and price for bo
- bookings.csv to be recorde main menu
- find external packages to a app (at least four)
- add user login and info to c past booking can be integr
- user.py
- booking.py
- Test menu loops
- Add a card

https://trello.com/b/kLus5Ies/terminal-application-hotel-booking-t1a3

Main priorities

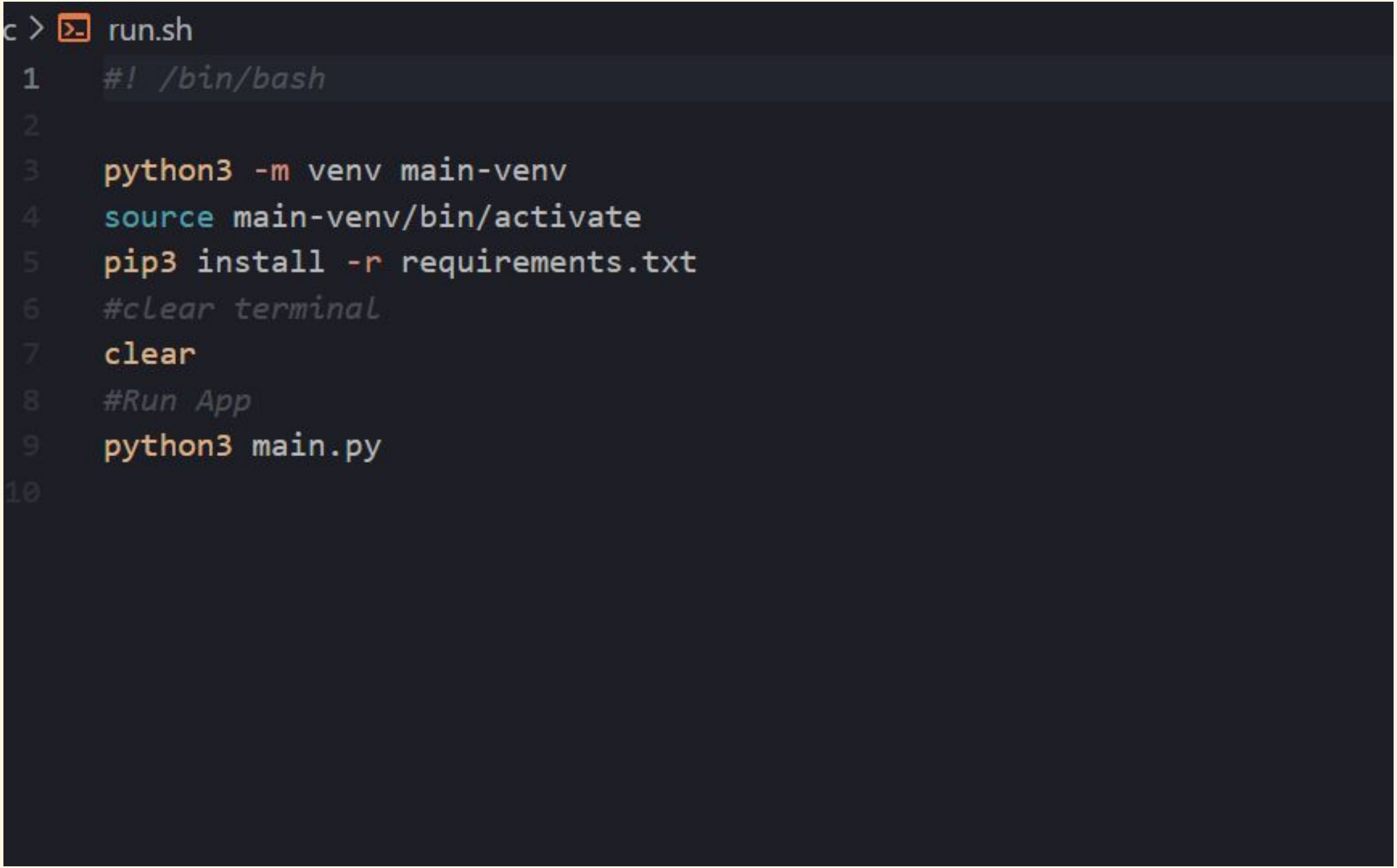
- Main Menu
- Making bookings
- User
- Previous bookings display
- Python testing

In each cases, making sure that there is sufficient error handling

Packages used

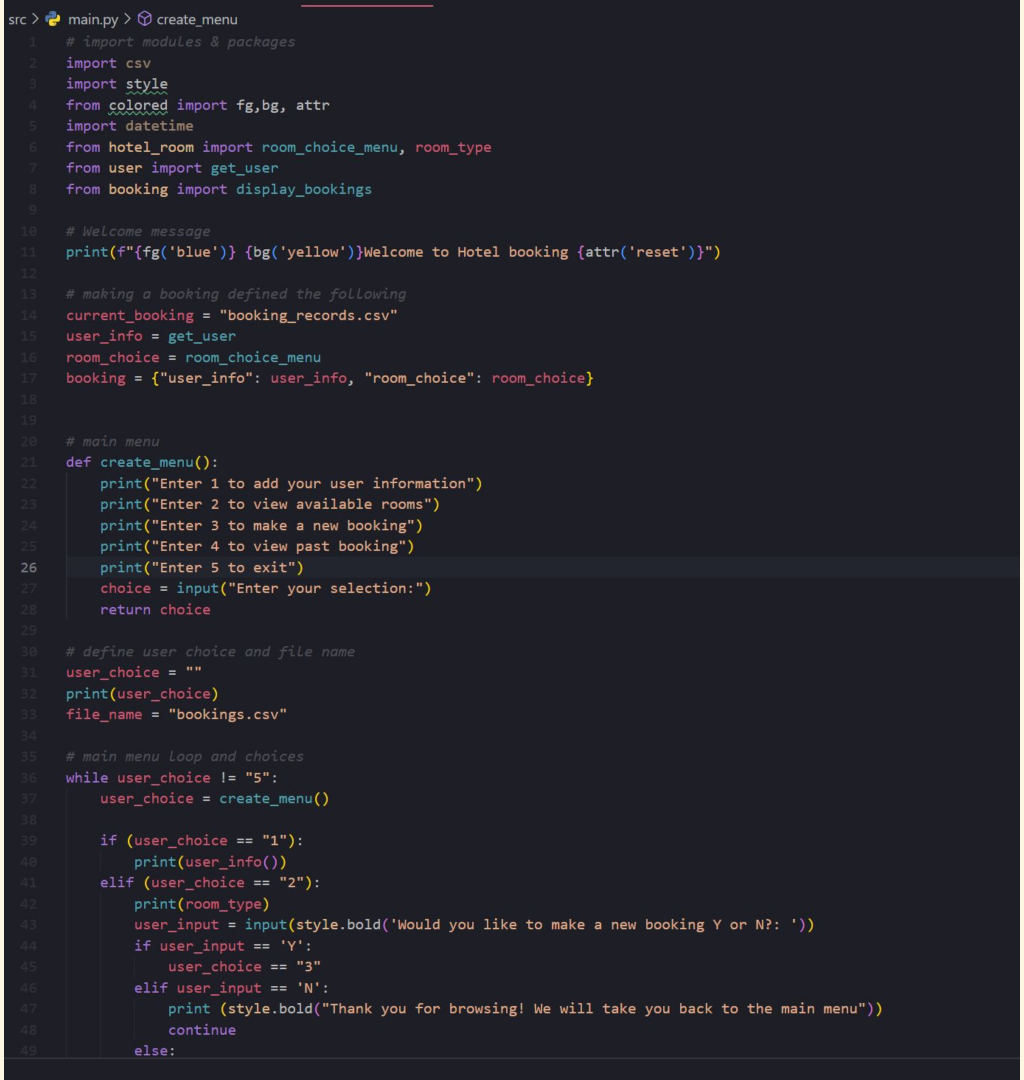
Internal packages used:

- .csv
- datetime
- Unittest
- Pytest
- Styles
- colored

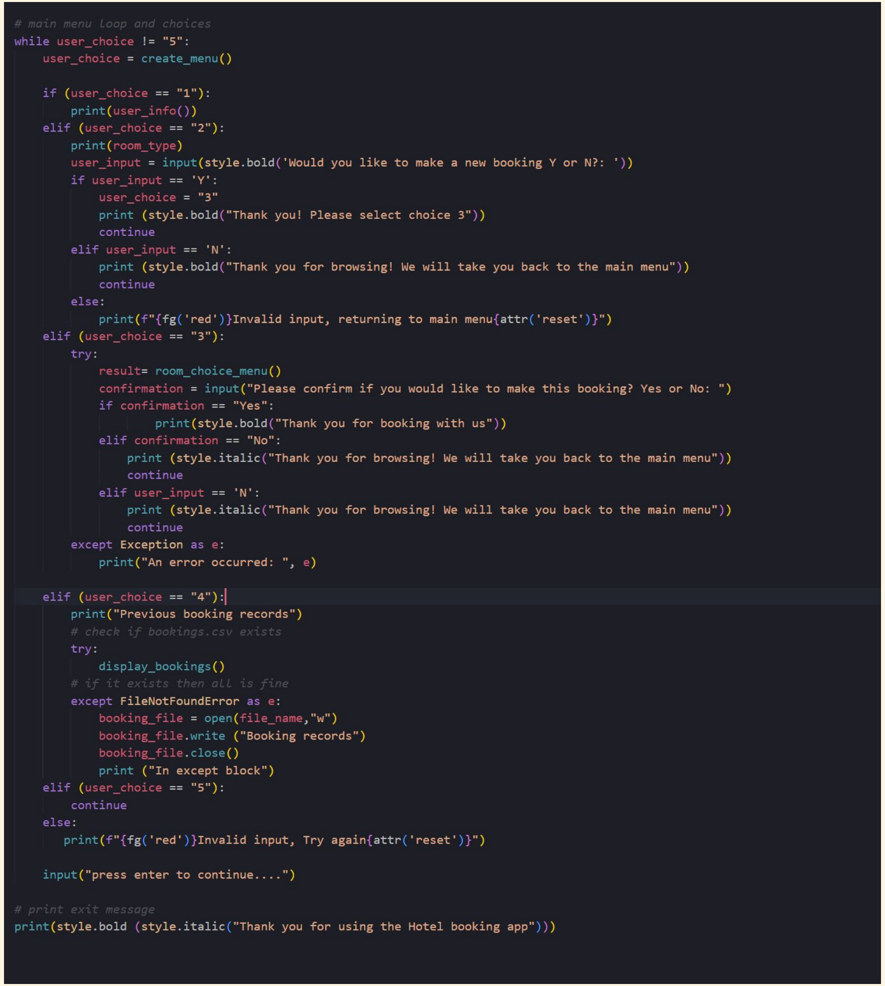


```bash
#! /bin/bash

python3 -m venv main-venv
source main-venv/bin/activate
pip3 install -r requirements.txt
#clear terminal
clear
#Run App
python3 main.py
```

Main.py

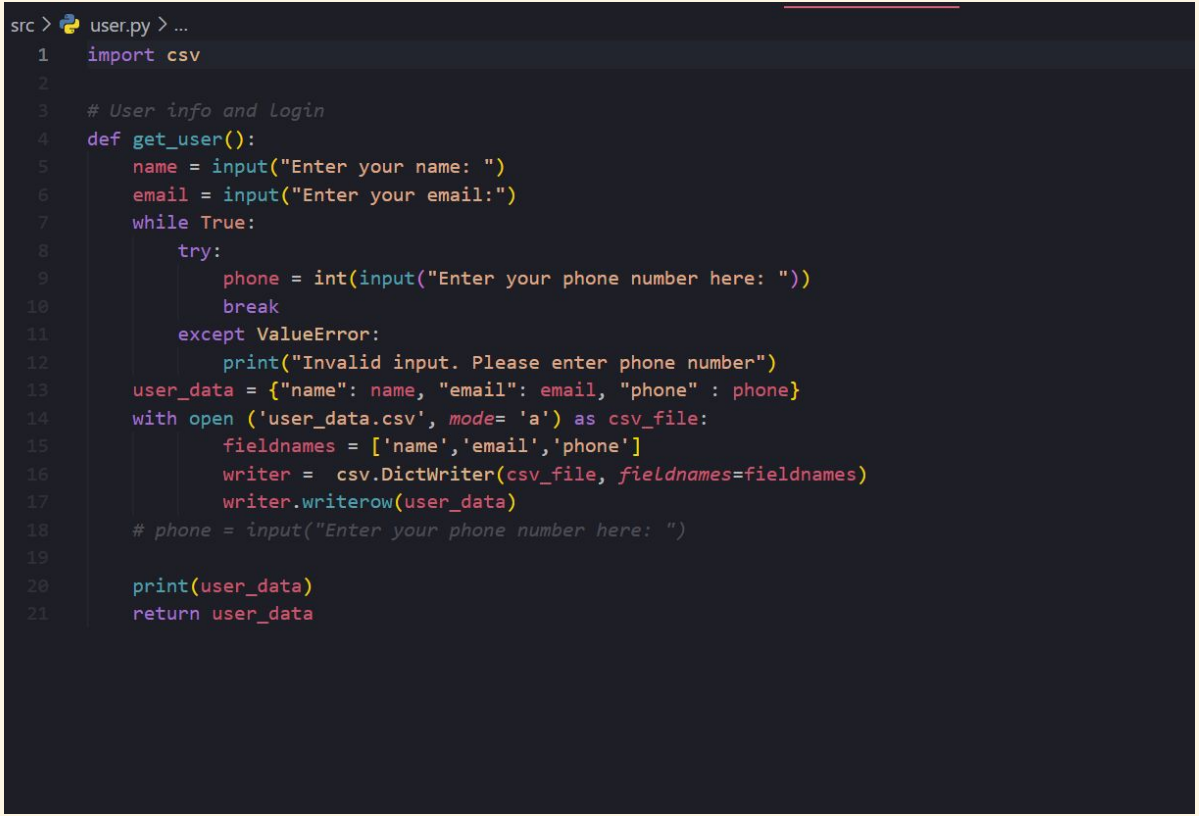


```python
# import modules & packages
import csv
import style
from colored import fg,bg, attr
import datetime
from hotel_room import room_choice_menu, room_type
from user import get_user
from booking import display_bookings

# Welcome message
print(f"{fg('blue')} {bg('yellow')}Welcome to Hotel booking {attr('reset')}")

# making a booking defined the following
current_booking = "booking_records.csv"
user_info = get_user
room_choice = room_choice_menu
booking = {"user_info": user_info, "room_choice": room_choice}


# main menu
def create_menu():
    print("Enter 1 to add your user information")
    print("Enter 2 to view available rooms")
    print("Enter 3 to make a new booking")
    print("Enter 4 to view past booking")
    print("Enter 5 to exit")
    choice = input("Enter your selection:")
    return choice

# define user choice and file name
user_choice = ""
print(user_choice)
file_name = "bookings.csv"

# main menu loop and choices
while user_choice != "5":
    user_choice = create_menu()

    if (user_choice == "1"):
        print(user_info())
    elif (user_choice == "2"):
        print(room_type)
        user_input = input(style.bold('Would you like to make a new booking Y or N?: '))
        if user_input == 'Y':
            user_choice == "3"
        elif user_input == 'N':
            print (style.bold("Thank you for browsing! We will take you back to the main menu"))
            continue
        else:
```



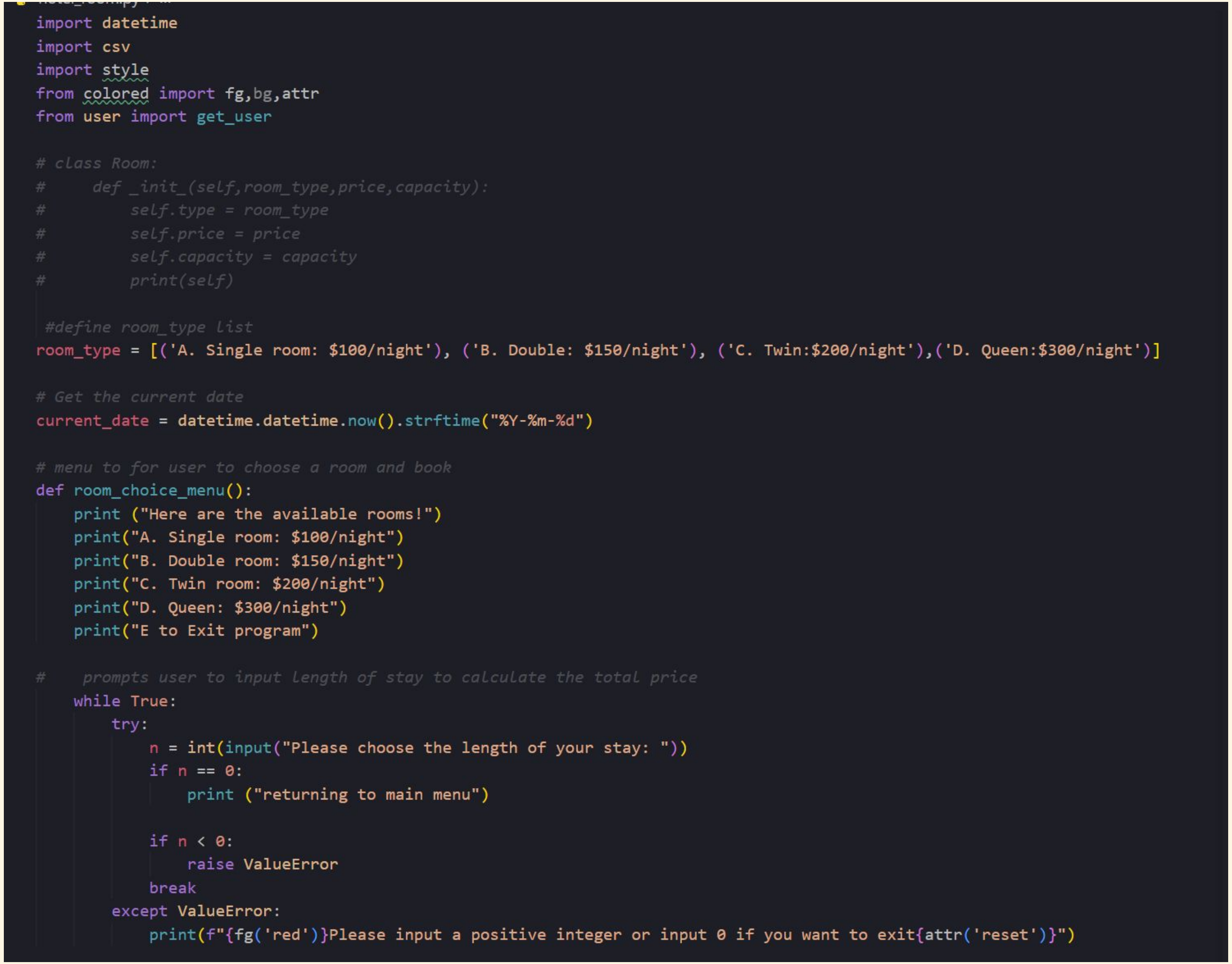
```python
# main menu loop and choices
while user_choice != "5":
    user_choice = create_menu()

    if (user_choice == "1"):
        print(user_info())
    elif (user_choice == "2"):
        print(room_type)
        user_input = input(style.bold('Would you like to make a new booking Y or N?: '))
        if user_input == 'Y':
            user_choice = "3"
            print (style.bold("Thank you! Please select choice 3"))
            continue
        elif user_input == 'N':
            print (style.bold("Thank you for browsing! We will take you back to the main menu"))
            continue
        else:
            print(f"{fg('red')}Invalid input, returning to main menu{attr('reset')}")
    elif (user_choice == "3"):
        try:
            result= room_choice_menu()
            confirmation = input("Please confirm if you would like to make this booking? Yes or No: ")
            if confirmation == "Yes":
                    print(style.bold("Thank you for booking with us"))
            elif confirmation == "No":
                print (style.italic("Thank you for browsing! We will take you back to the main menu"))
                continue
            elif user_input == 'N':
                print (style.italic("Thank you for browsing! We will take you back to the main menu"))
                continue
        except Exception as e:
            print("An error occurred: ", e)

    elif (user_choice == "4"):
        print("Previous booking records")
        # check if bookings.csv exists
        try:
            display_bookings()
        # if it exists then all is fine
        except FileNotFoundError as e:
            booking_file = open(file_name,"w")
            booking_file.write ("Booking records")
            booking_file.close()
            print ("In except block")
    elif (user_choice == "5"):
        continue
    else:
       print(f"{fg('red')}Invalid input, Try again{attr('reset')}")

    input("press enter to continue....")

# print exit message
print(style.bold (style.italic("Thank you for using the Hotel booking app")))
```

User.py to import get_user() function

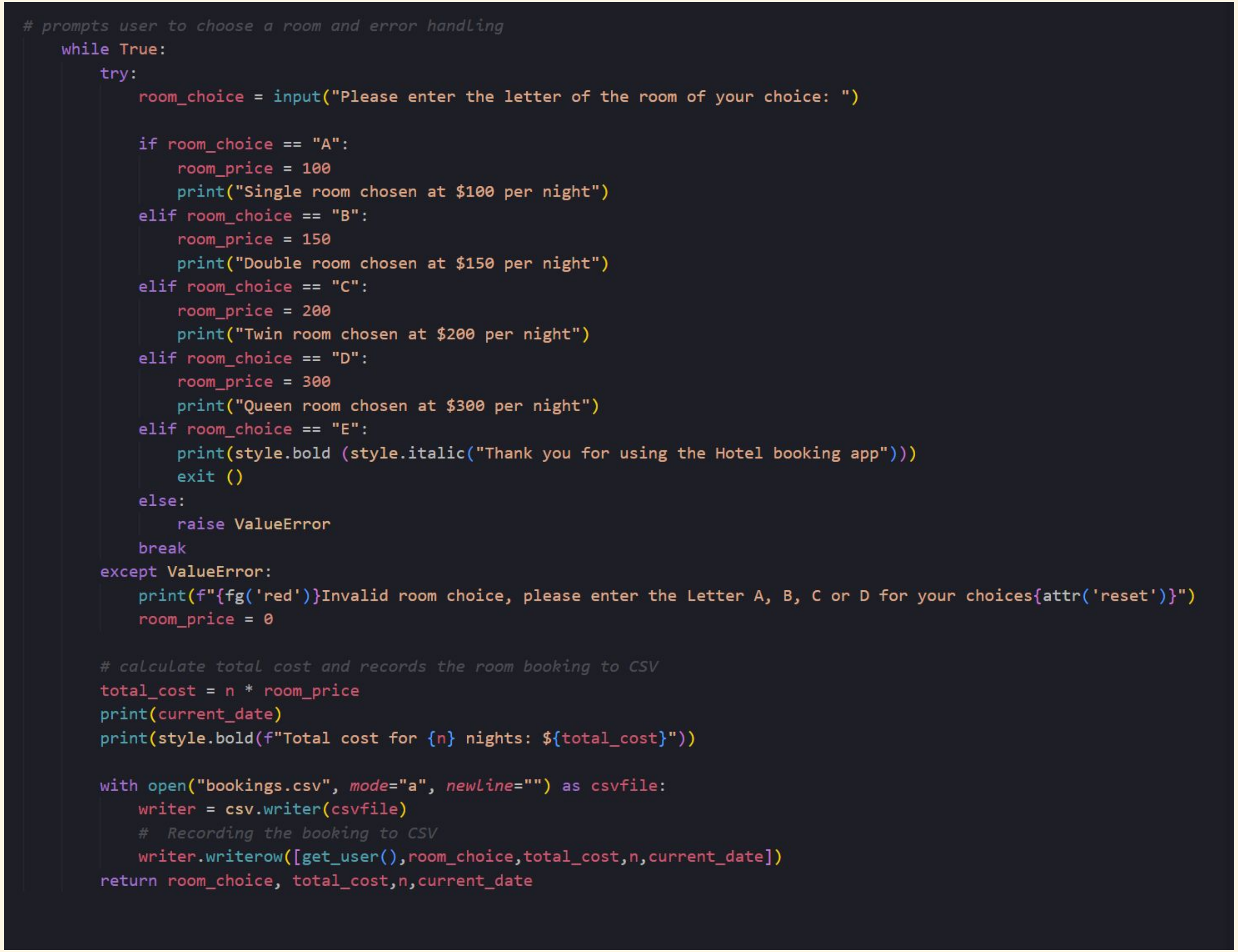


```python
import csv

# User info and login
def get_user():
    name = input("Enter your name: ")
    email = input("Enter your email:")
    while True:
        try:
            phone = int(input("Enter your phone number here: "))
            break
        except ValueError:
            print("Invalid input. Please enter phone number")
    user_data = {"name": name, "email": email, "phone" : phone}
    with open ('user_data.csv', mode= 'a') as csv_file:
            fieldnames = ['name','email','phone']
            writer =  csv.DictWriter(csv_file, fieldnames=fieldnames)
            writer.writerow(user_data)
    # phone = input("Enter your phone number here: ")

    print(user_data)
    return user_data
```

Hotel_room.py

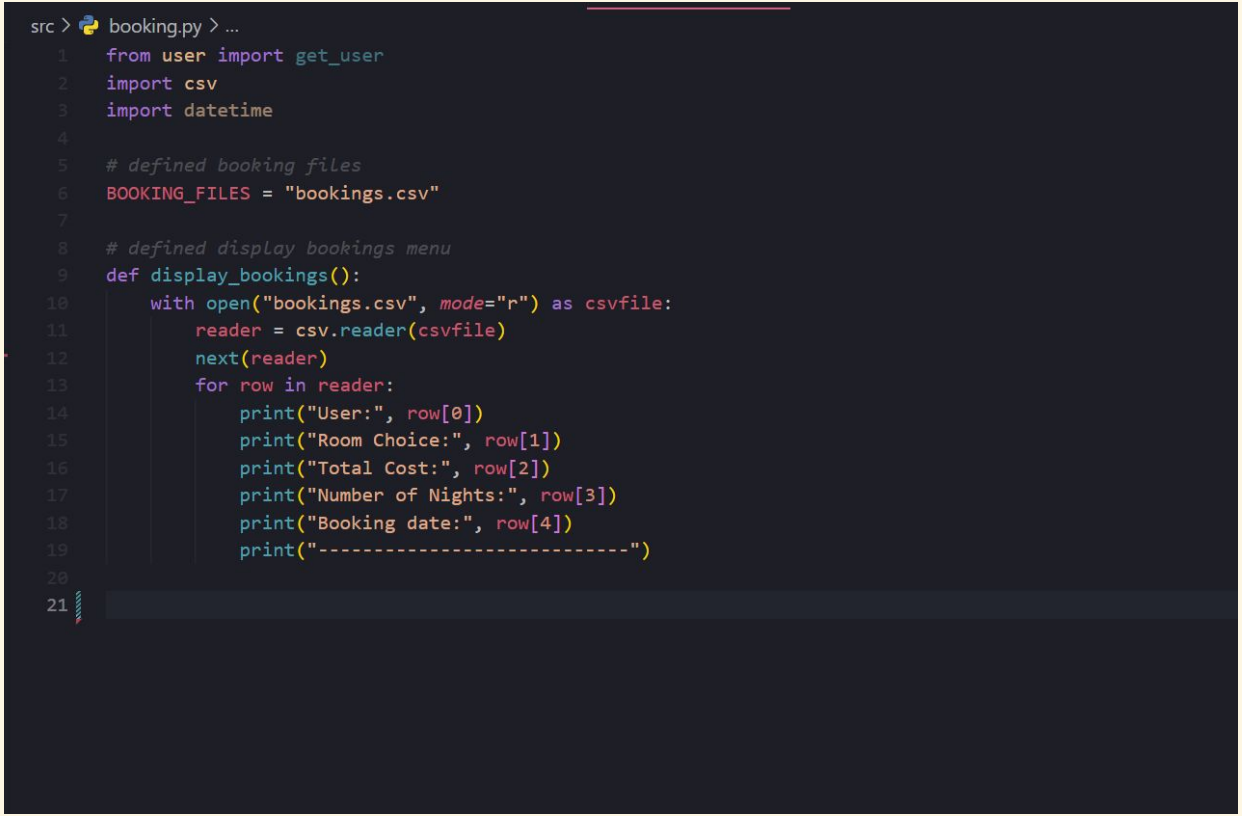


```python
import datetime
import csv
import style
from colored import fg,bg,attr
from user import get_user

# class Room:
#     def _init_(self,room_type,price,capacity):
#         self.type = room_type
#         self.price = price
#         self.capacity = capacity
#         print(self)

 #define room_type list
room_type = [('A. Single room: $100/night'), ('B. Double: $150/night'), ('C. Twin:$200/night'),('D. Queen:$300/night')]

# Get the current date
current_date = datetime.datetime.now().strftime("%Y-%m-%d")

# menu to for user to choose a room and book
def room_choice_menu():
    print ("Here are the available rooms!")
    print("A. Single room: $100/night")
    print("B. Double room: $150/night")
    print("C. Twin room: $200/night")
    print("D. Queen: $300/night")
    print("E to Exit program")

#    prompts user to input length of stay to calculate the total price
    while True:
        try:
            n = int(input("Please choose the length of your stay: "))
            if n == 0:
                print ("returning to main menu")

            if n < 0:
                raise ValueError
            break
        except ValueError:
            print(f"{fg('red')}Please input a positive integer or input 0 if you want to exit{attr('reset')}")
```



```python
# prompts user to choose a room and error handling
    while True:
        try:
            room_choice = input("Please enter the letter of the room of your choice: ")

            if room_choice == "A":
                room_price = 100
                print("Single room chosen at $100 per night")
            elif room_choice == "B":
                room_price = 150
                print("Double room chosen at $150 per night")
            elif room_choice == "C":
                room_price = 200
                print("Twin room chosen at $200 per night")
            elif room_choice == "D":
                room_price = 300
                print("Queen room chosen at $300 per night")
            elif room_choice == "E":
                print(style.bold (style.italic("Thank you for using the Hotel booking app")))
                exit ()
            else:
                raise ValueError
            break
        except ValueError:
            print(f"{fg('red')}Invalid room choice, please enter the Letter A, B, C or D for your choices{attr('reset')}")
            room_price = 0

        # calculate total cost and records the room booking to CSV
        total_cost = n * room_price
        print(current_date)
        print(style.bold(f"Total cost for {n} nights: ${total_cost}"))

        with open("bookings.csv", mode="a", newline="") as csvfile:
            writer = csv.writer(csvfile)
            #  Recording the booking to CSV
            writer.writerow([get_user(),room_choice,total_cost,n,current_date])
        return room_choice, total_cost,n,current_date
```

Display previous bookings in choice == 4

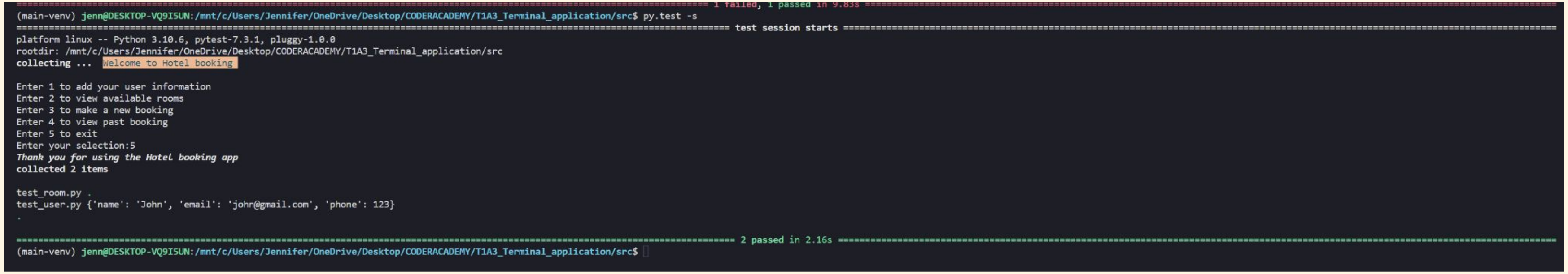


```python
from user import get_user
import csv
import datetime

# defined booking files
BOOKING_FILES = "bookings.csv"

# defined display bookings menu
def display_bookings():
    with open("bookings.csv", mode="r") as csvfile:
        reader = csv.reader(csvfile)
        next(reader)
        for row in reader:
            print("User:", row[0])
            print("Room Choice:", row[1])
            print("Total Cost:", row[2])
            print("Number of Nights:", row[3])
            print("Booking date:", row[4])
            print("----------------------------")


```

Testings

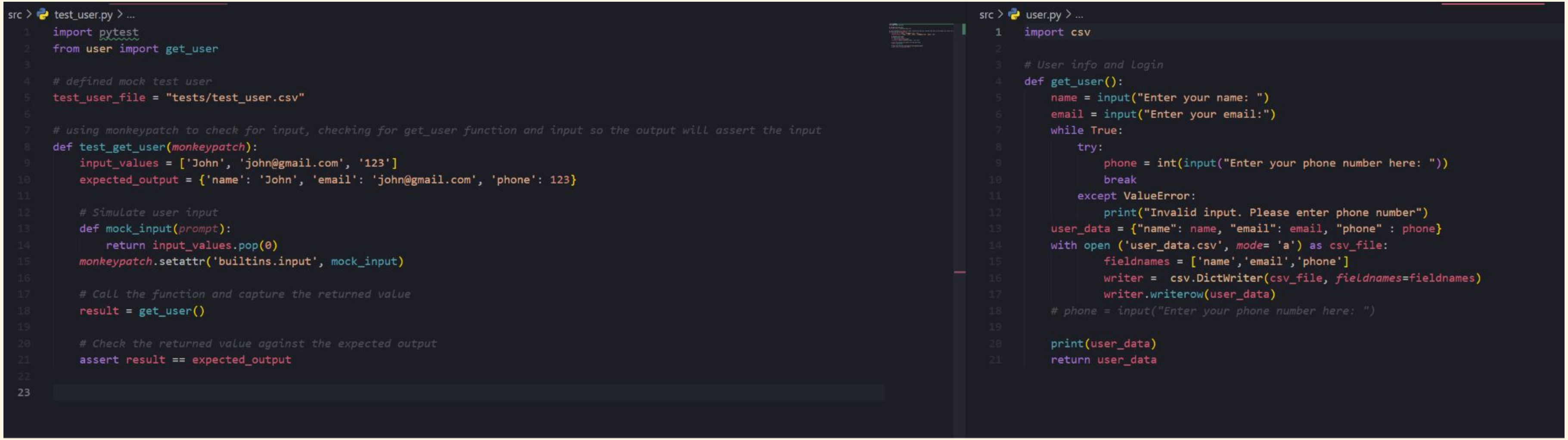
Did pytest and unittest

Pytest



```
================================================================ 1 failed, 1 passed in 9.83s ================================================================
(main-venv) jenn@DESKTOP-VQ9I5UN:/mnt/c/Users/Jennifer/OneDrive/Desktop/CODERACADEMY/T1A3_Terminal_application/src$ py.test -s
==================================================================== test session starts ====================================================================
platform linux -- Python 3.10.6, pytest-7.3.1, pluggy-1.0.0
rootdir: /mnt/c/Users/Jennifer/OneDrive/Desktop/CODERACADEMY/T1A3_Terminal_application/src
collecting ...  Welcome to Hotel booking 

Enter 1 to add your user information
Enter 2 to view available rooms
Enter 3 to make a new booking
Enter 4 to view past booking
Enter 5 to exit
Enter your selection:5
Thank you for using the Hotel booking app
collected 2 items

test_room.py .
test_user.py {'name': 'John', 'email': 'john@gmail.com', 'phone': 123}
.

==================================================================== 2 passed in 2.16s ====================================================================
(main-venv) jenn@DESKTOP-VQ9I5UN:/mnt/c/Users/Jennifer/OneDrive/Desktop/CODERACADEMY/T1A3_Terminal_application/src$ 
```

Test user.py



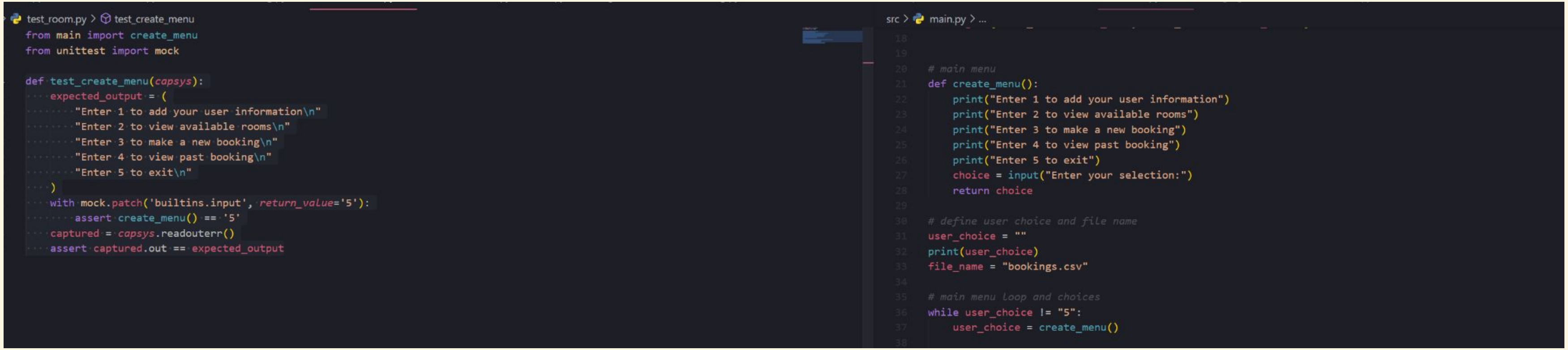
```python
import pytest
from user import get_user

# defined mock test user
test_user_file = "tests/test_user.csv"

# using monkeypatch to check for input, checking for get_user function and input so the output will assert the input
def test_get_user(monkeypatch):
    input_values = ['John', 'john@gmail.com', '123']
    expected_output = {'name': 'John', 'email': 'john@gmail.com', 'phone': 123}

    # Simulate user input
    def mock_input(prompt):
        return input_values.pop(0)
    monkeypatch.setattr('builtins.input', mock_input)

    # Call the function and capture the returned value
    result = get_user()

    # Check the returned value against the expected output
    assert result == expected_output

```

```python
import csv

# User info and login
def get_user():
    name = input("Enter your name: ")
    email = input("Enter your email:")
    while True:
        try:
            phone = int(input("Enter your phone number here: "))
            break
        except ValueError:
            print("Invalid input. Please enter phone number")
    user_data = {"name": name, "email": email, "phone" : phone}
    with open ('user_data.csv', mode= 'a') as csv_file:
            fieldnames = ['name','email','phone']
            writer =  csv.DictWriter(csv_file, fieldnames=fieldnames)
            writer.writerow(user_data)
    # phone = input("Enter your phone number here: ")

    print(user_data)
    return user_data
```

Tested by putting mock input values with monkeypatch and asserted that the result will be same as expected output

Expected output follows the format of the user.csv

Testing main menu



```python
from main import create_menu
from unittest import mock

def test_create_menu(capsys):
    expected_output = (
        "Enter 1 to add your user information\n"
        "Enter 2 to view available rooms\n"
        "Enter 3 to make a new booking\n"
        "Enter 4 to view past booking\n"
        "Enter 5 to exit\n"
    )
    with mock.patch('builtins.input', return_value='5'):
        assert create_menu() == '5'
    captured = capsys.readouterr()
    assert captured.out == expected_output
```

```python


# main menu
def create_menu():
    print("Enter 1 to add your user information")
    print("Enter 2 to view available rooms")
    print("Enter 3 to make a new booking")
    print("Enter 4 to view past booking")
    print("Enter 5 to exit")
    choice = input("Enter your selection:")
    return choice

# define user choice and file name
user_choice = ""
print(user_choice)
file_name = "bookings.csv"

# main menu loop and choices
while user_choice != "5":
    user_choice = create_menu()

```

Tested main menu so that it will exit when we input 5