T1A3 Assessment

Terminal Application – Jonathan Ow

Terminal App Idea

Hotel Reservation system with the following features:

- User account creation and storage of user details
- Reserve various types of rooms
- Check in and Check out date selection
- Outputs reservation details in terminal

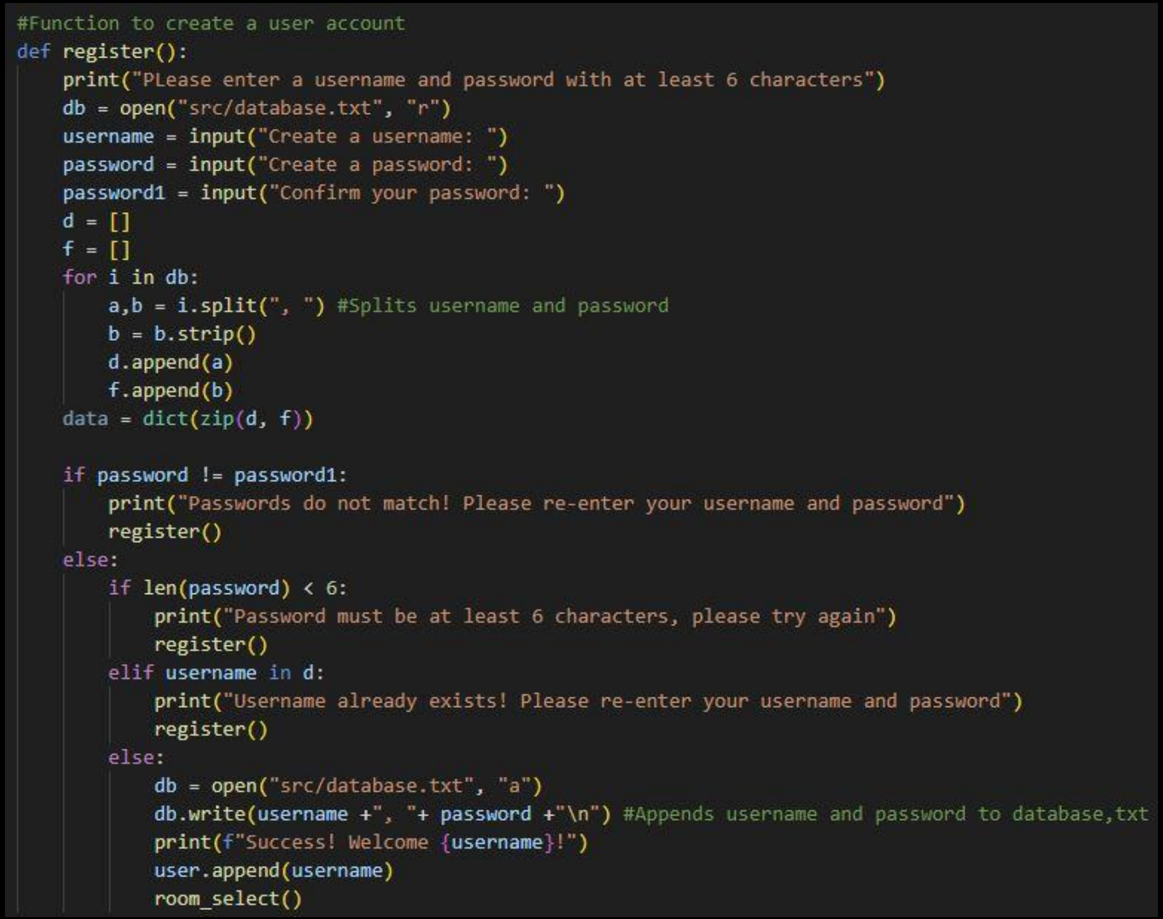
Program run through

Run through steps

1. User creates account or login to existing account
2. User is given list of available room types as well as their cost per night
3. User is given the option to view a calendar that will assist them in selecting dates
4. User is asked to input check in and checkout date
5. Room number and room access pin are generated
6. All reservation details outputted to terminal

Feature 1: Register Account

- User assign unique username and password
- Password must be 6 characters long
- Uses separate text file (database.txt) to store user details
- Separates username and password using split and appends into dictionary
- Username is appended to user universal variable

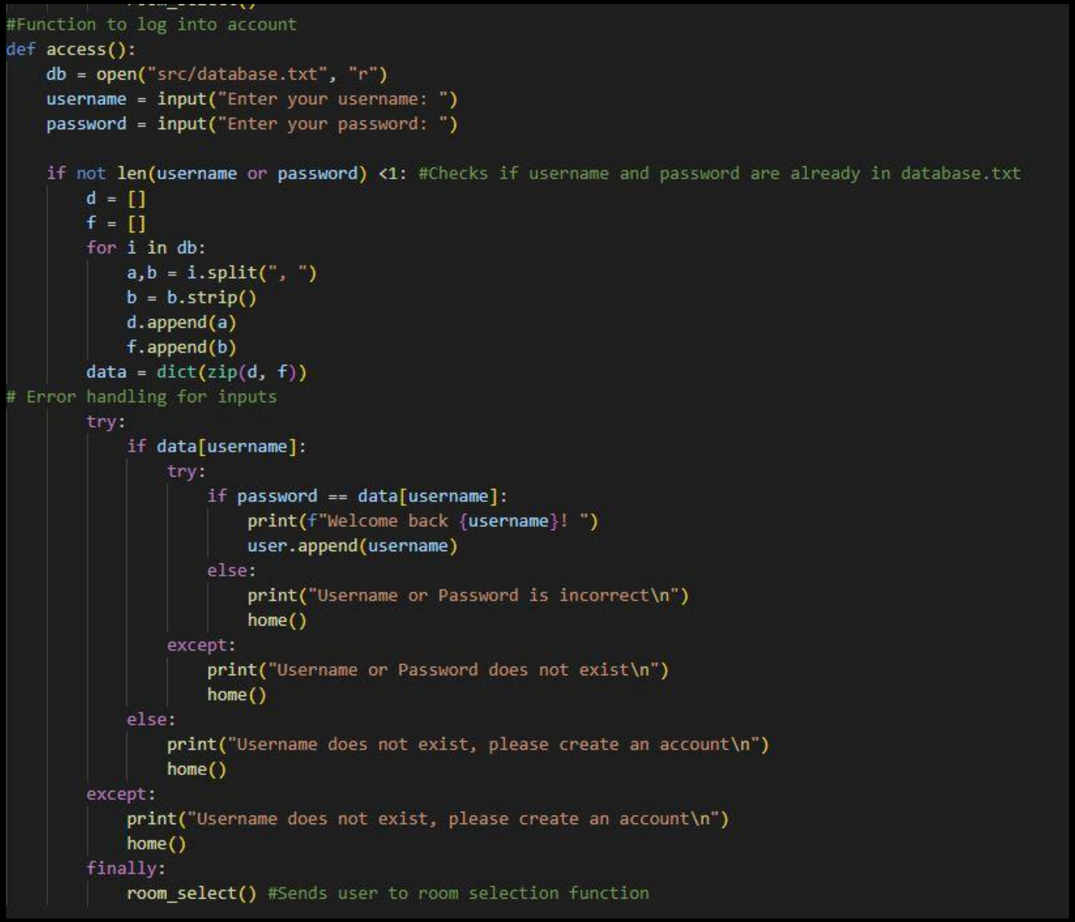


```
#Function to create a user account
def register():
    print("PLease enter a username and password with at least 6 characters")
    db = open("src/database.txt", "r")
    username = input("Create a username: ")
    password = input("Create a password: ")
    password1 = input("Confirm your password: ")
    d = []
    f = []
    for i in db:
        a,b = i.split(", ") #Splits username and password
        b = b.strip()
        d.append(a)
        f.append(b)
    data = dict(zip(d, f))

    if password != password1:
        print("Passwords do not match! Please re-enter your username and password")
        register()
    else:
        if len(password) < 6:
            print("Password must be at least 6 characters, please try again")
            register()
        elif username in d:
            print("Username already exists! Please re-enter your username and password")
            register()
        else:
            db = open("src/database.txt", "a")
            db.write(username +", "+ password +"\n") #Appends username and password to database,txt
            print(f"Success! Welcome {username}!")
            user.append(username)
            room_select()
```

Feature 2: Access Account

- Gets username and password inputs from user
- Checks if username is already in database and if corresponding password matches
- Error handling that handles all potential incorrect inputs
- Returns user to home function if conditions are not met



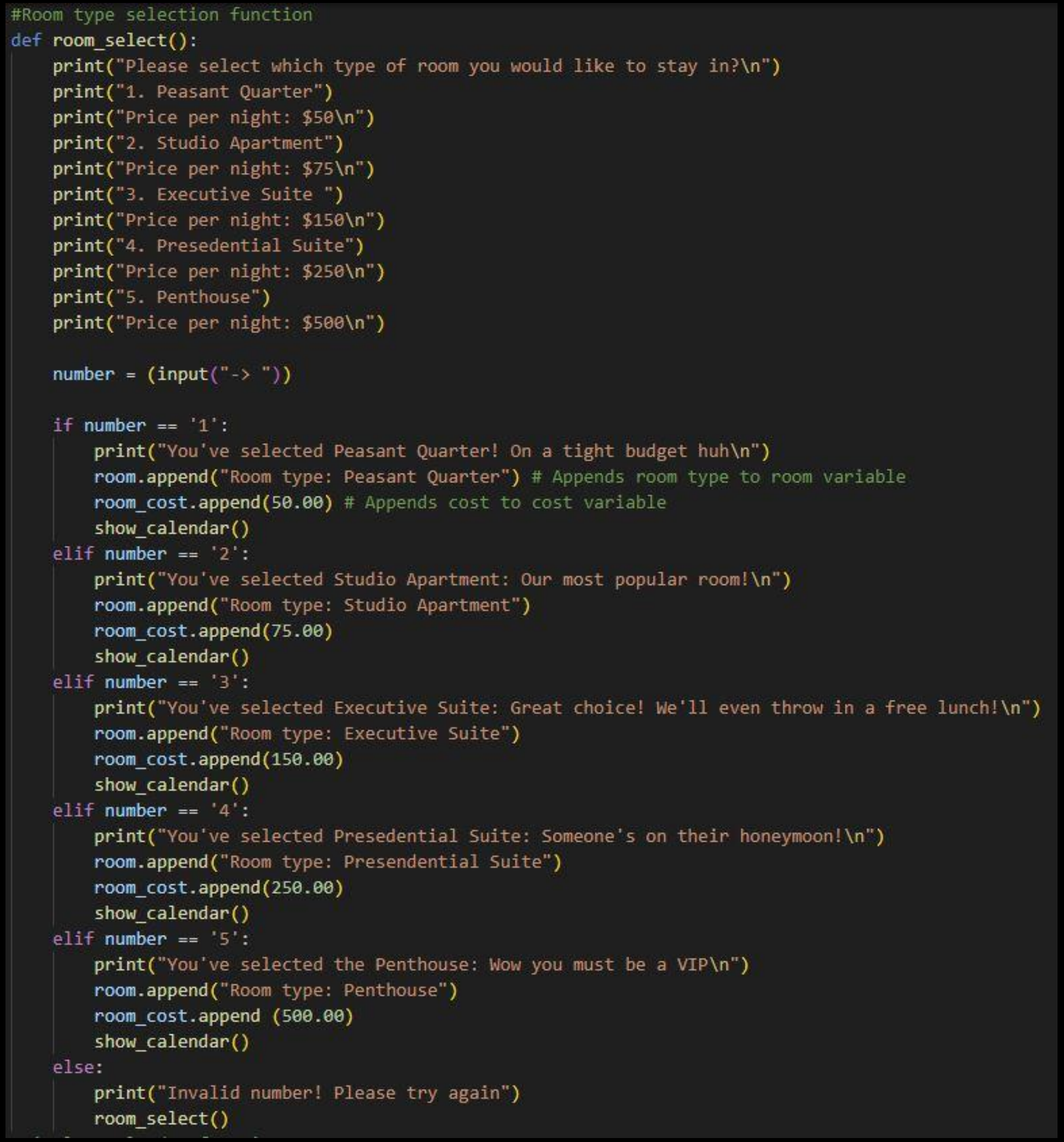
```
#Function to log into account
def access():
    db = open("src/database.txt", "r")
    username = input("Enter your username: ")
    password = input("Enter your password: ")

    if not len(username or password) <1: #Checks if username and password are already in database.txt
        d = []
        f = []
        for i in db:
            a,b = i.split(", ")
            b = b.strip()
            d.append(a)
            f.append(b)
        data = dict(zip(d, f))
# Error handling for inputs
        try:
            if data[username]:
                try:
                    if password == data[username]:
                        print(f"Welcome back {username}! ")
                        user.append(username)
                    else:
                        print("Username or Password is incorrect\n")
                        home()
                except:
                    print("Username or Password does not exist\n")
                    home()
            else:
                print("Username does not exist, please create an account\n")
                home()
        except:
            print("Username does not exist, please create an account\n")
            home()
        finally:
            room_select() #Sends user to room selection function
```

Feature 3: Room selection

- User is given a list of room types to select via inputting a corresponding number
- The types of rooms include

    Peasant Quarters
    Studio Apartment
    Executive Suite
    Presidential Suite
    Penthouse
- Room type and cost are appended to their universal variables room and room_cost



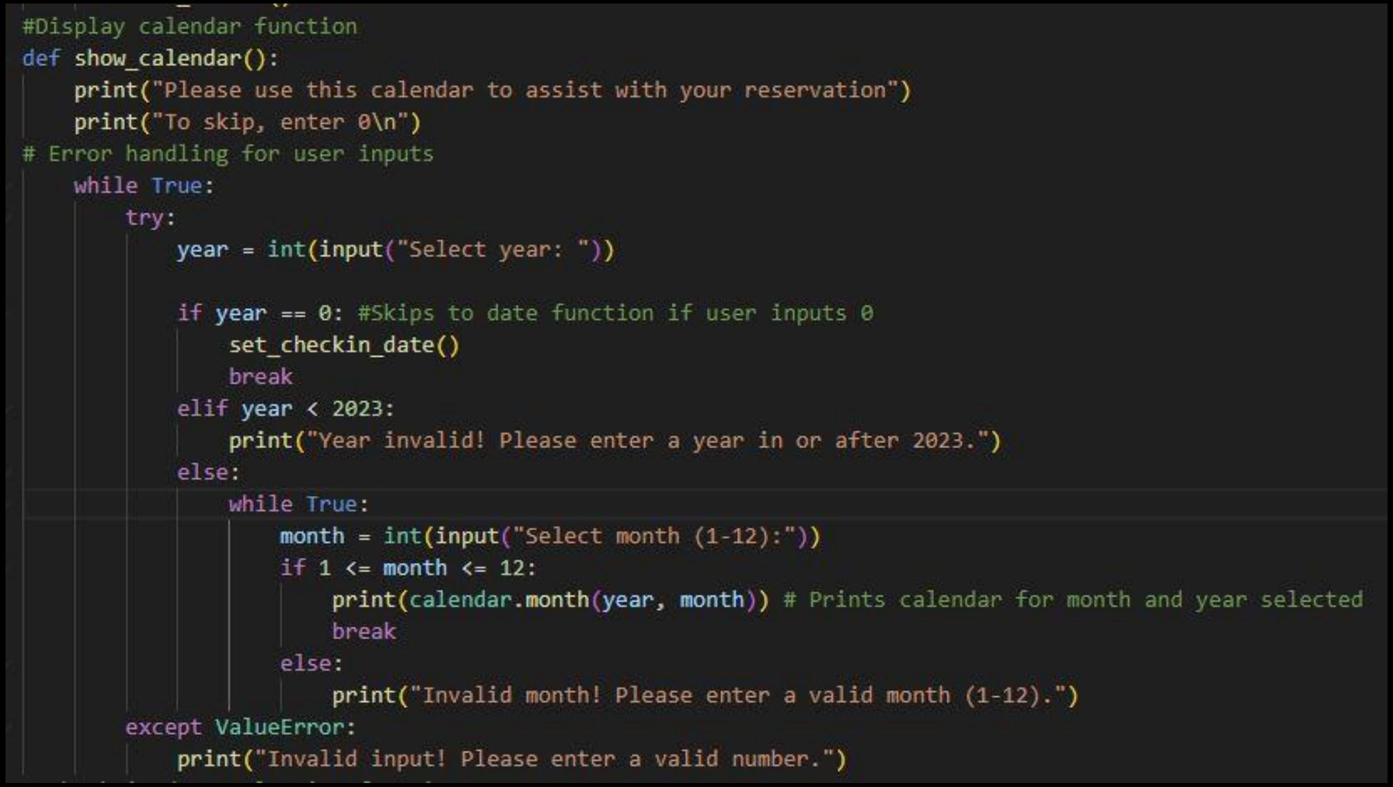
```
#Room type selection function
def room_select():
    print("Please select which type of room you would like to stay in?\n")
    print("1. Peasant Quarter")
    print("Price per night: $50\n")
    print("2. Studio Apartment")
    print("Price per night: $75\n")
    print("3. Executive Suite ")
    print("Price per night: $150\n")
    print("4. Presedential Suite")
    print("Price per night: $250\n")
    print("5. Penthouse")
    print("Price per night: $500\n")

    number = (input("-> "))

    if number == '1':
        print("You've selected Peasant Quarter! On a tight budget huh\n")
        room.append("Room type: Peasant Quarter") # Appends room type to room variable
        room_cost.append(50.00) # Appends cost to cost variable
        show_calendar()
    elif number == '2':
        print("You've selected Studio Apartment: Our most popular room!\n")
        room.append("Room type: Studio Apartment")
        room_cost.append(75.00)
        show_calendar()
    elif number == '3':
        print("You've selected Executive Suite: Great choice! We'll even throw in a free lunch!\n")
        room.append("Room type: Executive Suite")
        room_cost.append(150.00)
        show_calendar()
    elif number == '4':
        print("You've selected Presedential Suite: Someone's on their honeymoon!\n")
        room.append("Room type: Presendential Suite")
        room_cost.append(250.00)
        show_calendar()
    elif number == '5':
        print("You've selected the Penthouse: Wow you must be a VIP\n")
        room.append("Room type: Penthouse")
        room_cost.append (500.00)
        show_calendar()
    else:
        print("Invalid number! Please try again")
        room_select()
```

Feature 4: Calendar

- Utilises the calendar module to display a calendar of the user's choice
- Receives a year and month input
- User given the option to skip the function entirely and go straight to check in date function

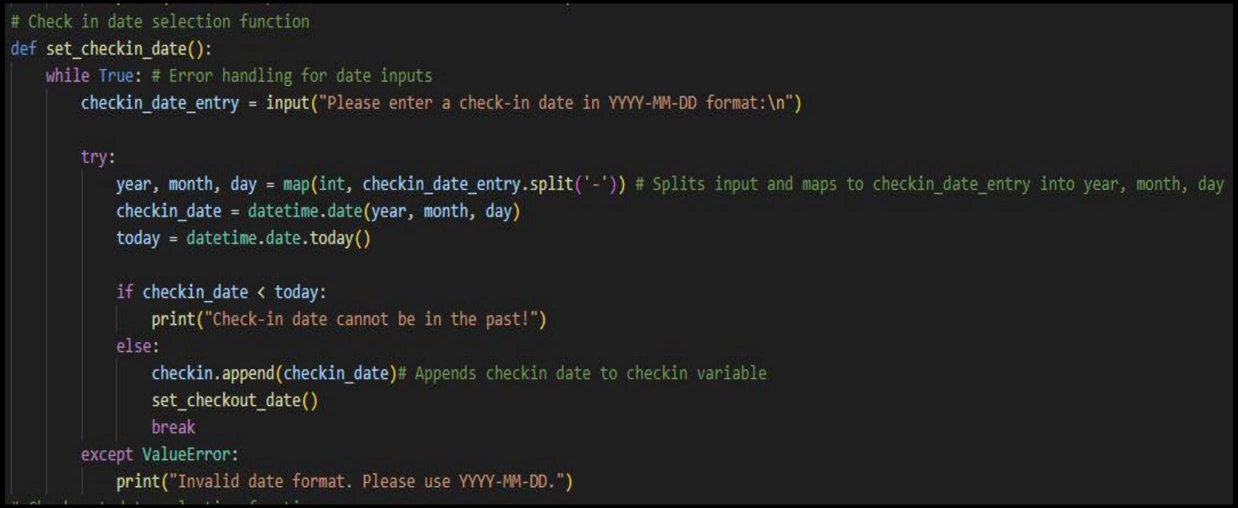


```
#Display calendar function
def show_calendar():
    print("Please use this calendar to assist with your reservation")
    print("To skip, enter 0\n")
# Error handling for user inputs
    while True:
        try:
            year = int(input("Select year: "))

            if year == 0: #Skips to date function if user inputs 0
                set_checkin_date()
                break
            elif year < 2023:
                print("Year invalid! Please enter a year in or after 2023.")
            else:
                while True:
                    month = int(input("Select month (1-12):"))
                    if 1 <= month <= 12:
                        print(calendar.month(year, month)) # Prints calendar for month and year selected
                        break
                    else:
                        print("Invalid month! Please enter a valid month (1-12).")
        except ValueError:
            print("Invalid input! Please enter a valid number.")
```

Feature 5: Date selection (Check-in and Checkout)

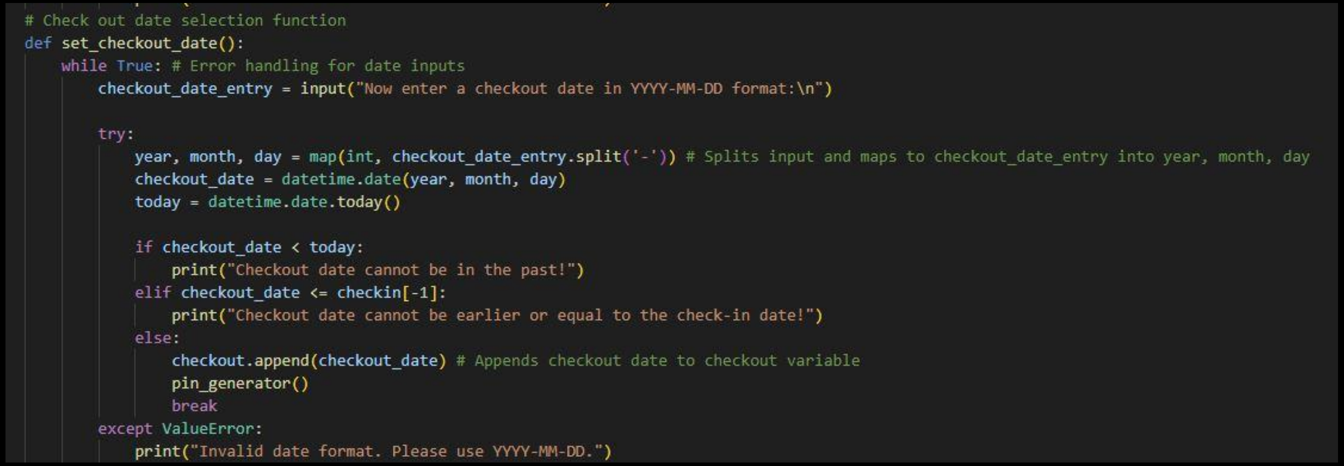
- Utilises the datetime module to set check in and check out dates
- Splits user inputs in year, month and day
- Appends dates to universal variable
- Separate functions for check in and check out dates



```python
# Check in date selection function
def set_checkin_date():
    while True: # Error handling for date inputs
        checkin_date_entry = input("Please enter a check-in date in YYYY-MM-DD format:\n")

        try:
            year, month, day = map(int, checkin_date_entry.split('-')) # Splits input and maps to checkin_date_entry into year, month, day
            checkin_date = datetime.date(year, month, day)
            today = datetime.date.today()

            if checkin_date < today:
                print("Check-in date cannot be in the past!")
            else:
                checkin.append(checkin_date)# Appends checkin date to checkin variable
                set_checkout_date()
                break
        except ValueError:
            print("Invalid date format. Please use YYYY-MM-DD.")
```



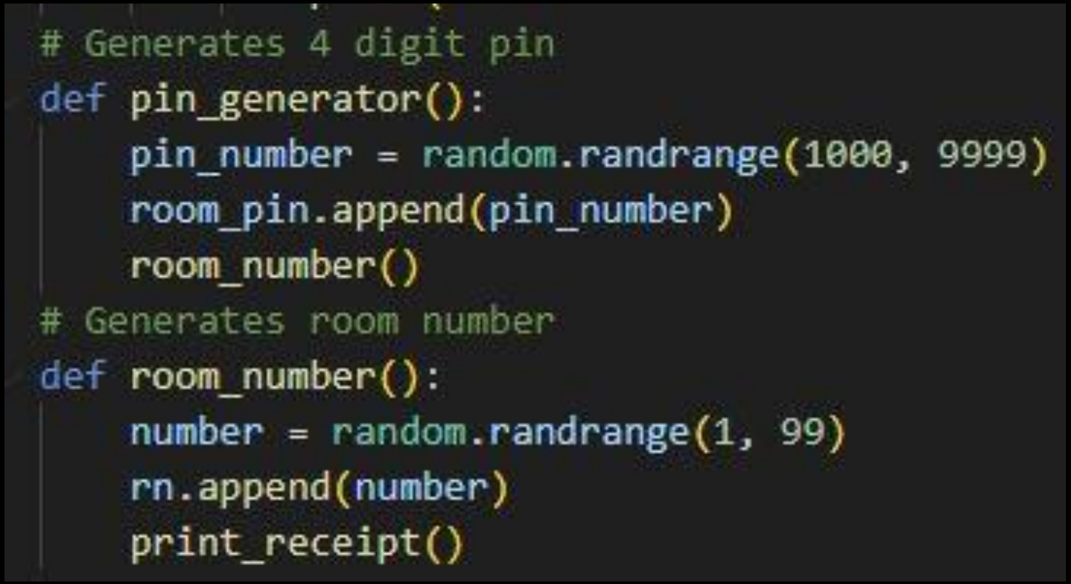
```python
# Check out date selection function
def set_checkout_date():
    while True: # Error handling for date inputs
        checkout_date_entry = input("Now enter a checkout date in YYYY-MM-DD format:\n")

        try:
            year, month, day = map(int, checkout_date_entry.split('-')) # Splits input and maps to checkout_date_entry into year, month, day
            checkout_date = datetime.date(year, month, day)
            today = datetime.date.today()

            if checkout_date < today:
                print("Checkout date cannot be in the past!")
            elif checkout_date <= checkin[-1]:
                print("Checkout date cannot be earlier or equal to the check-in date!")
            else:
                checkout.append(checkout_date) # Appends checkout date to checkout variable
                pin_generator()
                break
        except ValueError:
            print("Invalid date format. Please use YYYY-MM-DD.")
```

Feature 6: Pin and Room no. Randomiser

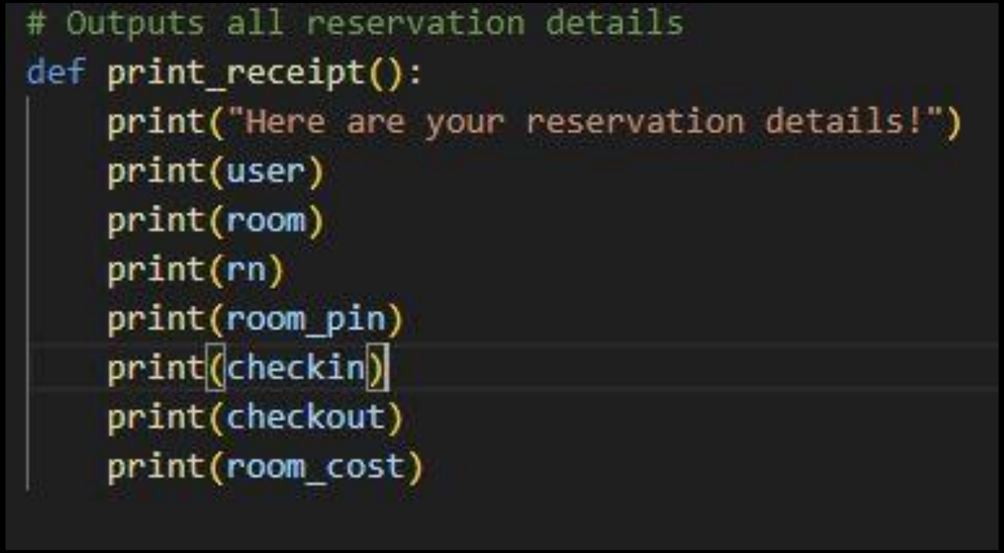
- Generates random pin number and hotel room number
- Uses the random python module to generate numbers
- Room pin is 4 digits
- Room number is 2 digits



```
# Generates 4 digit pin
def pin_generator():
    pin_number = random.randrange(1000, 9999)
    room_pin.append(pin_number)
    room_number()
# Generates room number
def room_number():
    number = random.randrange(1, 99)
    rn.append(number)
    print_receipt()
```

Final Output

- Receipt function outputs the reservation details that have been appended to the following variables



```python
# Outputs all reservation details
def print_receipt():
    print("Here are your reservation details!")
    print(user)
    print(room)
    print(rn)
    print(room_pin)
    print(checkin)
    print(checkout)
    print(room_cost)
```

Reflections

- Code could have more features which simulate a hotel e.g., being able to book multiple rooms, room service costs,
- Scrapped features such as being able to email user details of reservations (smtplib)

Issues

- Using multiple modules