**PYTHON CASE STUDY**

**1. ATM Simulation System:**

**CODE:**

*class* ATM:

*def* \_\_init\_\_(*self*, *balance*=1000):

    self.balance = balance

*def* check\_balance(*self*):

    print(*f*"Your balance: ${self.balance}")

*def* deposit(*self*, *amount*):

    self.balance += amount

    print(*f*"Deposited: ${amount}")

*def* withdraw(*self*, *amount*):

    if amount > self.balance:

         print("Insufficient funds!")

    else:

        self.balance -= amount

        print(*f*"Withdrawn: ${amount}")

*def* main():

    atm = ATM()

    while True:

        print("\n1. Check Balance\n2. Deposit\n3. Withdraw\n4. Exit")

        choice = input("Enter choice: ")

        if choice == "1":

            atm.check\_balance()

        elif choice == "2":

            amt = *float*(input("Enter deposit amount: "))

            atm.deposit(amt)

        elif choice == "3":

             amt = *float*(input("Enter withdrawal amount: "))

             atm.withdraw(amt)

        elif choice == "4":

            print("Thank you for using the ATM!")

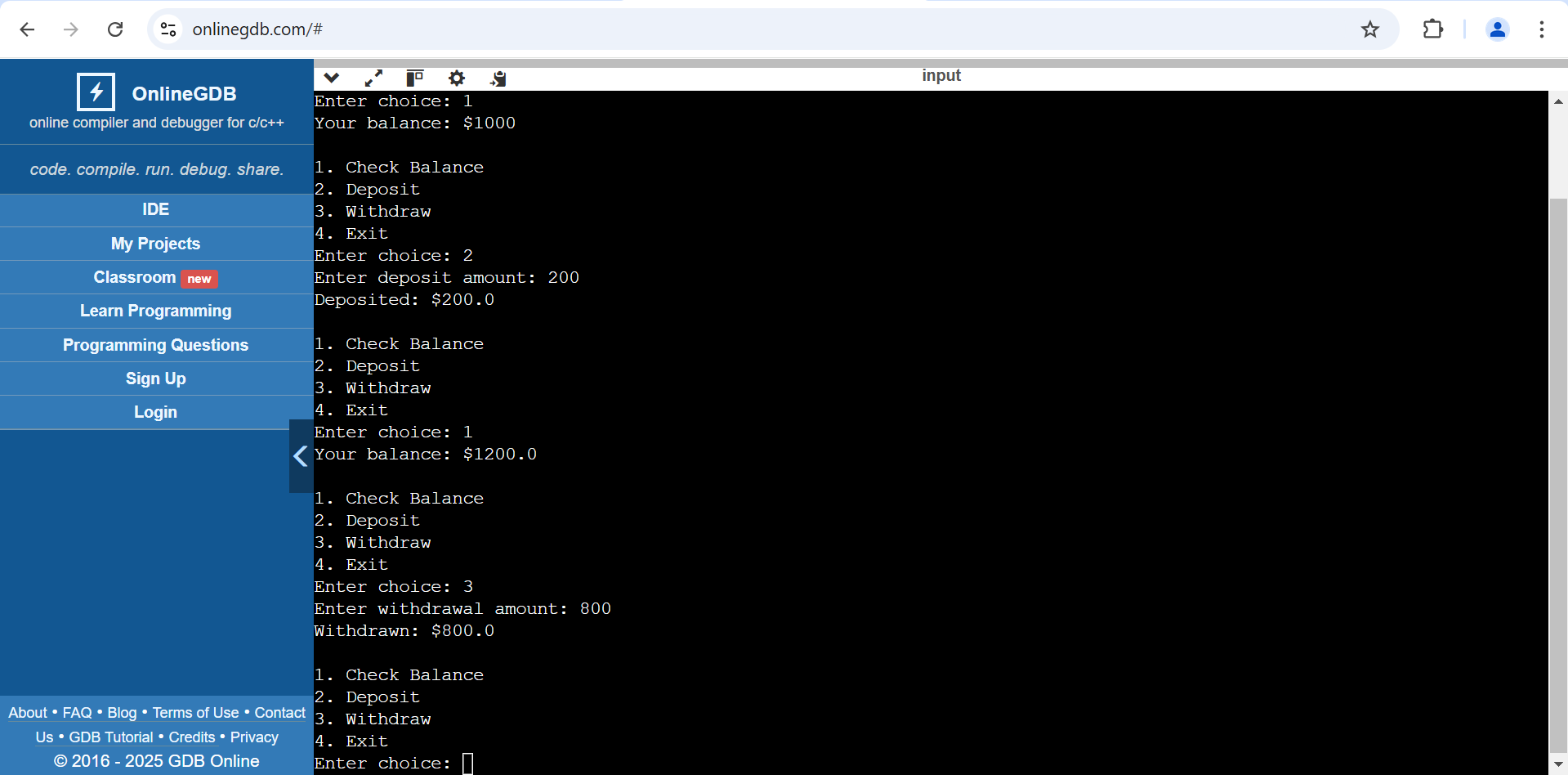
            break

        else:

            print("Invalid choice! Try again.")

main()

**Output:**

****

**2. E-Comerce Order Management:**

**CODE:**

*class* Product:

*def* \_\_init\_\_(*self*, *name*, *price*):

*self*.name = *name*

*self*.price = *price*

*class* ShoppingCart:

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

*self*.cart = []

*def* add\_product(*self*, *product*):

*self*.cart.append(*product*)

        print(*f*"{*product*.name} added to cart!")

*def* view\_cart(*self*):

        if not *self*.cart:

            print("Cart is empty!")

        else:

            print("\nShopping Cart:")

            total = 0

            for p in *self*.cart:

                print(*f*"- {p.name}: ${p.price}")

                total += p.price

            print(*f*"Total: ${total}")

*def* checkout(*self*):

        if not *self*.cart:

            print("Cart is empty!")

        else:

*self*.view\_cart()

            print("Proceeding to checkout...")

*def* main():

    cart = ShoppingCart()

    products = {

        "1": Product("Laptop", 1000),

        "2": Product("Headphones", 150),

        "3": Product("Mouse", 50),

    }

    while True:

        print("\n1. Add Laptop ($1000)\n2. Add Headphones ($150)\n3. Add Mouse ($50)\n4. View Cart\n5. Checkout\n6. Exit")

        choice = input("Enter choice: ")

        if choice in products:

            cart.add\_product(products[choice])

        elif choice == "4":

            cart.view\_cart()

        elif choice == "5":

            cart.checkout()

            break

        elif choice == "6":

            print("Thank you for shopping!")

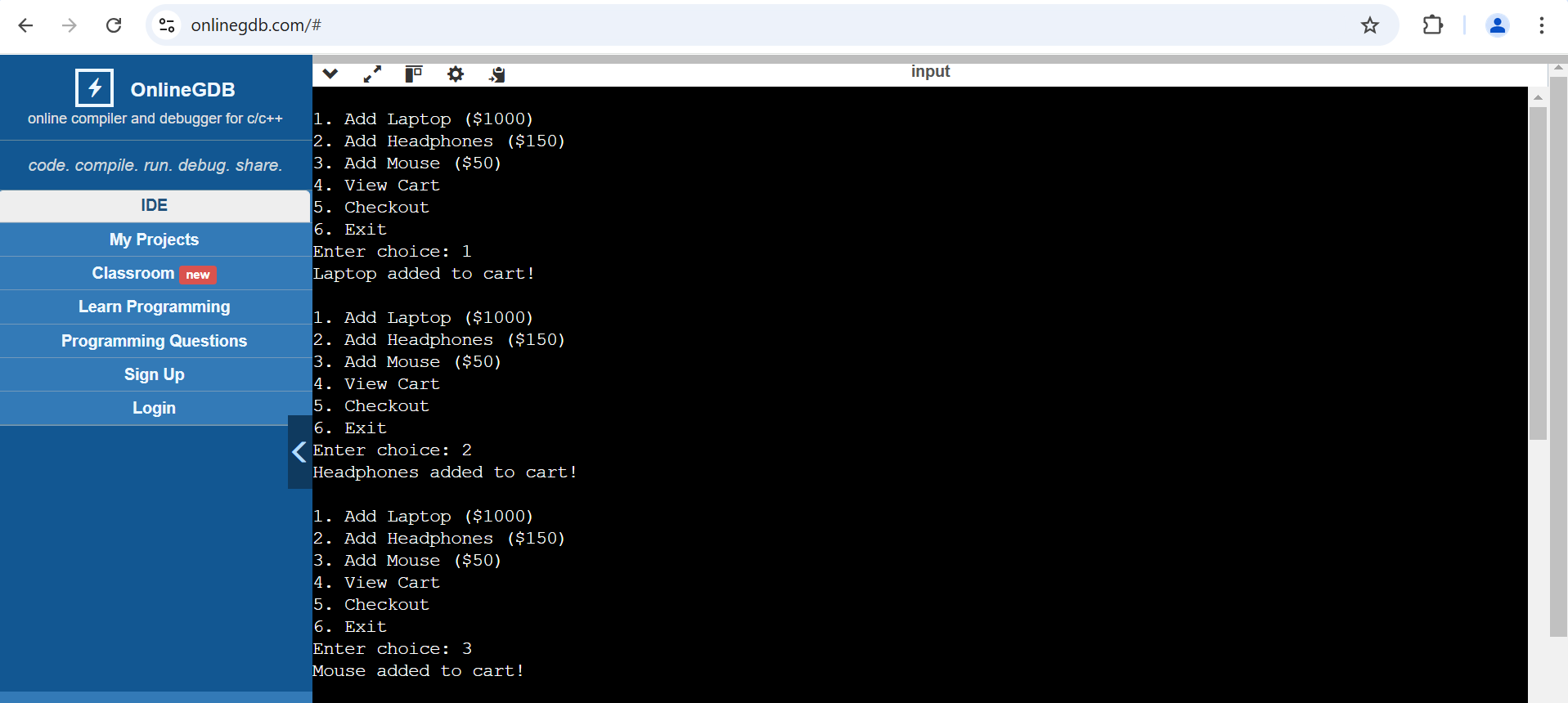
            break

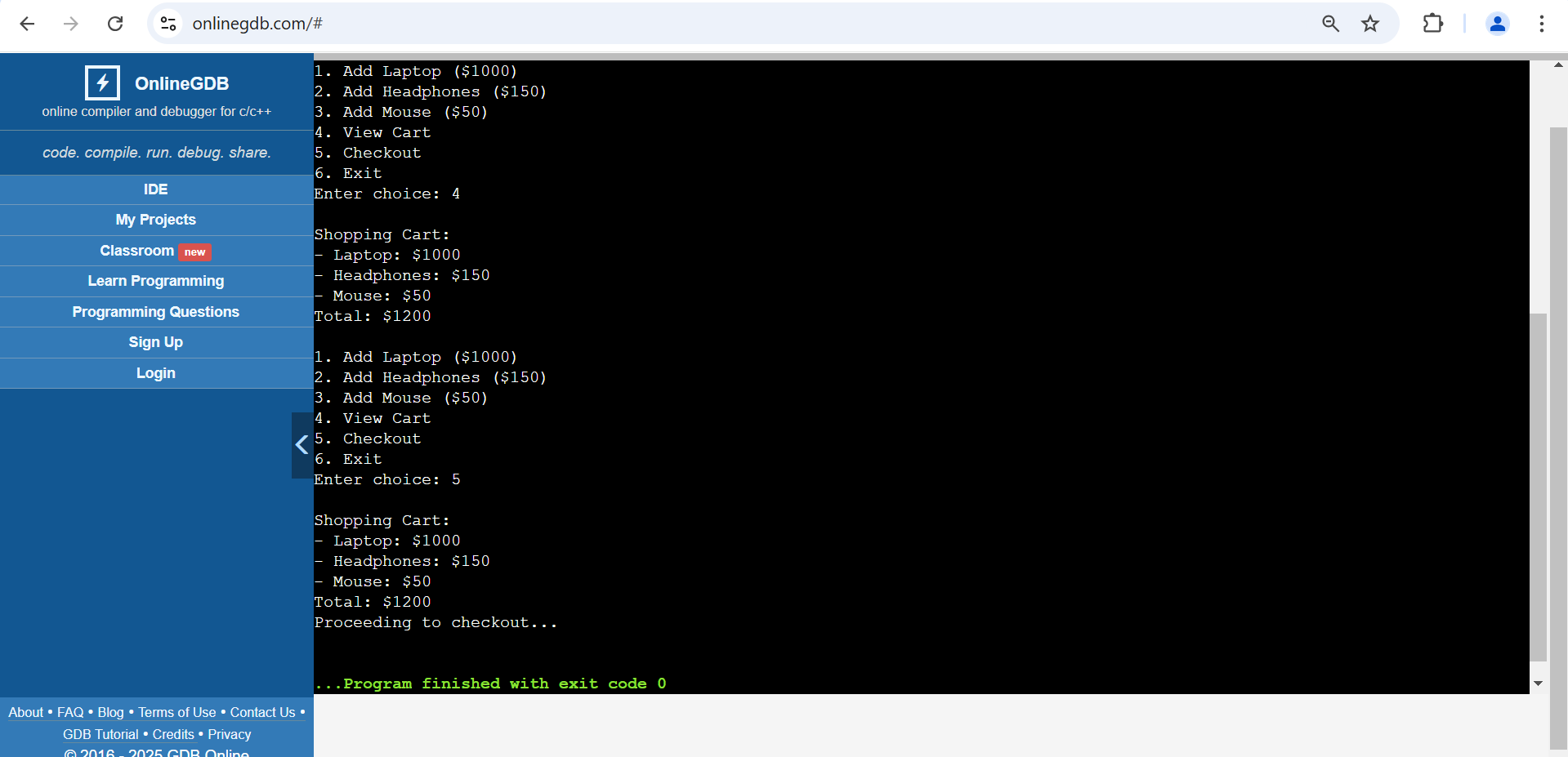
        else:

            print("Invalid choice!")

main()

**OUTPUT:**

****

****

**3. Student Grade Management System:**

**CODE:**

*class* GradeSystem:

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

*self*.grades = {}

*def* add\_grade(*self*, *name*, *grade*):

*self*.grades[*name*] = *grade*

        print(*f*"Added: {*name*} - {*grade*}")

*def* view\_grades(*self*):

         if not *self*.grades:

            print("No grades available!")

         else:

             print("\nStudent Grades:")

             for name, grade in *self*.grades.items():

                print(*f*"{name}: {grade}")

*def* calculate\_average(*self*):

        if not *self*.grades:

            print("No grades available!")

        else:

            avg = sum(*self*.grades.values()) / len(*self*.grades)

            print(*f*"Class Average: {avg*:.2f*}")

*def* main():

     system = GradeSystem()

     while True:

         print("\n1. Add Grade\n2. View Grades\n3. Calculate Average\n4.Exit")

         choice = input("Enter choice: ")

         if choice == "1":

             name = input("Enter student name: ")

             grade = float(input("Enter grade: "))

             system.add\_grade(name, grade)

         elif choice == "2":

            system.view\_grades()

         elif choice == "3":

            system.calculate\_average()

         elif choice == "4":

             print("Exiting Grade System.")

             break

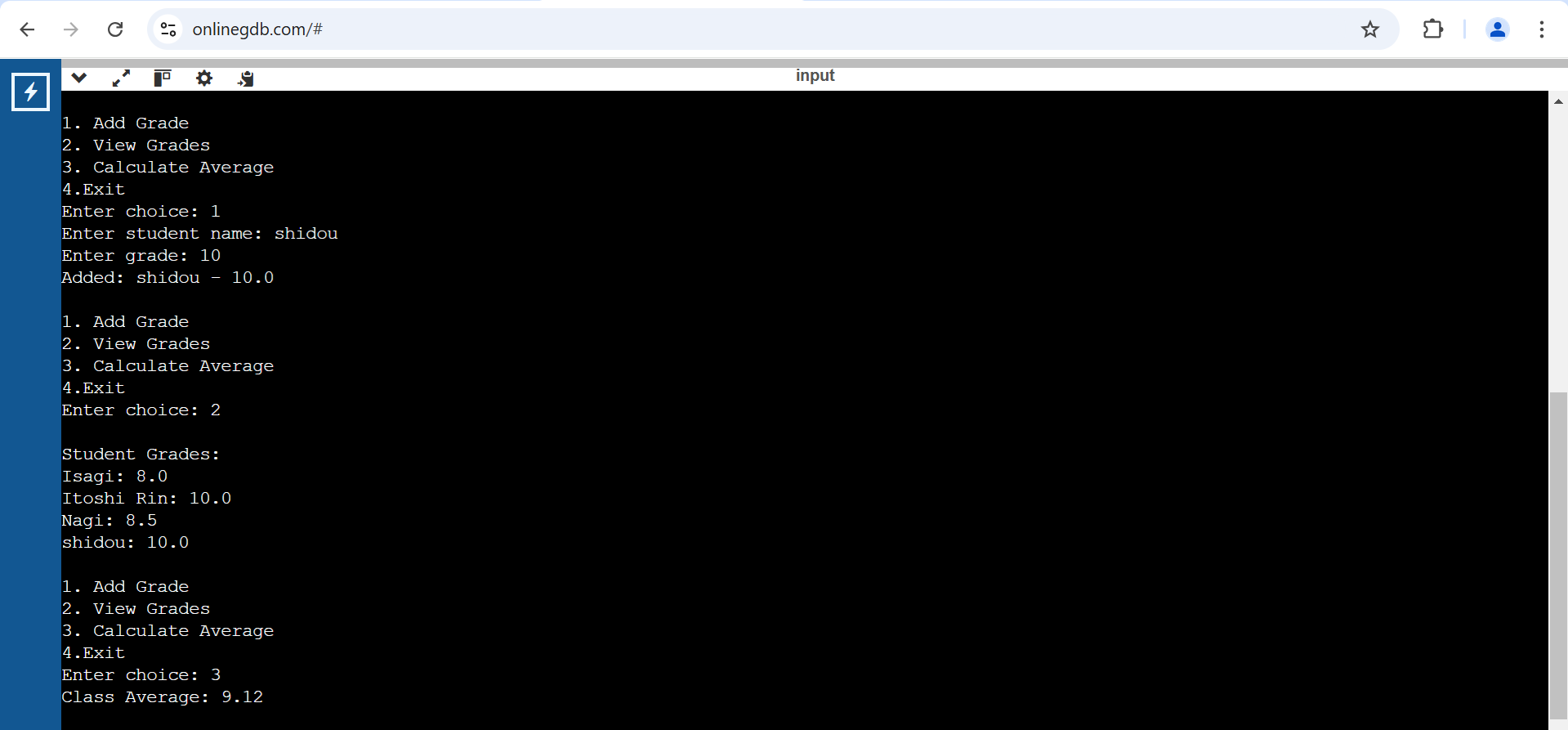
         else:

            print("Invalid choice!")

main()

**OUTPUT:**

****

****

**4. Hospital Patient Management System:**

**CODE:**

*class* Hospital:

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

*self*.patients = {}

*def* add\_patient(*self*, *id*, *name*, *age*, *disease*):

*self*.patients[*id*] = {"Name": *name*, "Age": *age*, "Disease": *disease*}

        print(*f*"Patient {*name*} added!")

*def* view\_patients(*self*):

        if not *self*.patients:

            print("No patients registered!")

        else:

            print("\nPatient Records:")

            for id, details in *self*.patients.items():

                print(*f*"ID: {id} - {details}")

*def* remove\_patient(*self*, *id*):

        if *id* in *self*.patients:

            del *self*.patients[*id*]

            print("Patient removed!")

        else:

            print("Patient not found!")

*def* main():

    hospital = Hospital()

    while True:

        print("\n1. Add Patient\n2. View Patients\n3. Remove Patient\n4. Exit")

        choice = input("Enter choice: ")

        if choice == "1":

            id = input("Enter Patient ID: ")

            name = input("Enter Name: ")

            age = input("Enter Age: ")

            disease = input("Enter Disease: ")

            hospital.add\_patient(id, name, age, disease)

        elif choice == "2":

            hospital.view\_patients()

        elif choice == "3":

            id = input("Enter Patient ID to remove: ")

            hospital.remove\_patient(id)

        elif choice == "4":

            print("Exiting Hospital System.")

            break

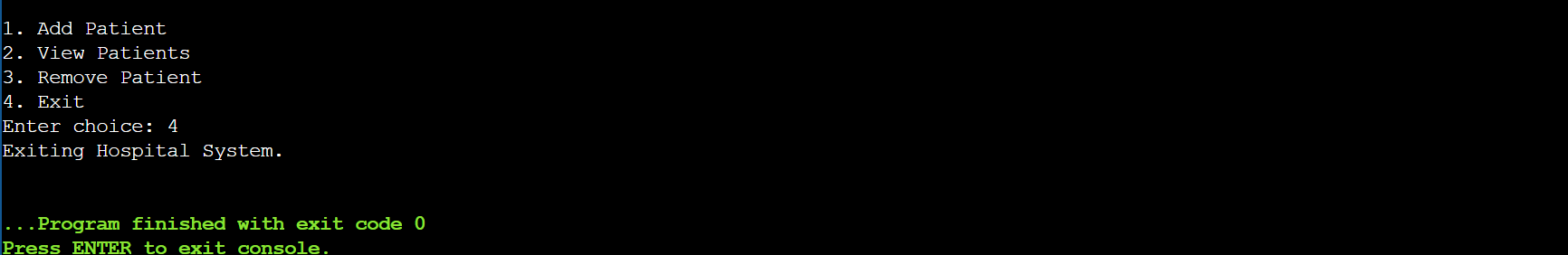
        else:

            print("Invalid choice!")

main()

**OUTPUT:**

****

****