Muhammad Huzaifa

Roll no:Sp23691

Bsse-05-afternoon

Software Construction and Development

Assignment :02

Expense Tracker Project Report

1. Implementation Overview

The Expense Tracker application is implemented using Object-Oriented Programming (OOP) in Java.
It includes the following features:
- Add Expense
- View Expenses
- Set Budget
- Generate Expense Report
- Exit Program

Structure:
- **Expense class:** Represents a single expense entry.
- **ExpenseTracker class:** Manages the collection of expenses and budget.
- **Main class:** Provides a menu-driven interface using a switch statement and Scanner input.

2. Manual Test Cases

Test Case	**Action**	**Expected Output**
Add an expense	Input description "Lunch", amount 10	"Expense added."
View expenses	Option 2	List of all expenses with total spent
Set budget	Option 3 → Enter 50	"Budget set to $50"
Generate report under budget	Expenses = 10, Budget = 50	"You are under budget by $40"

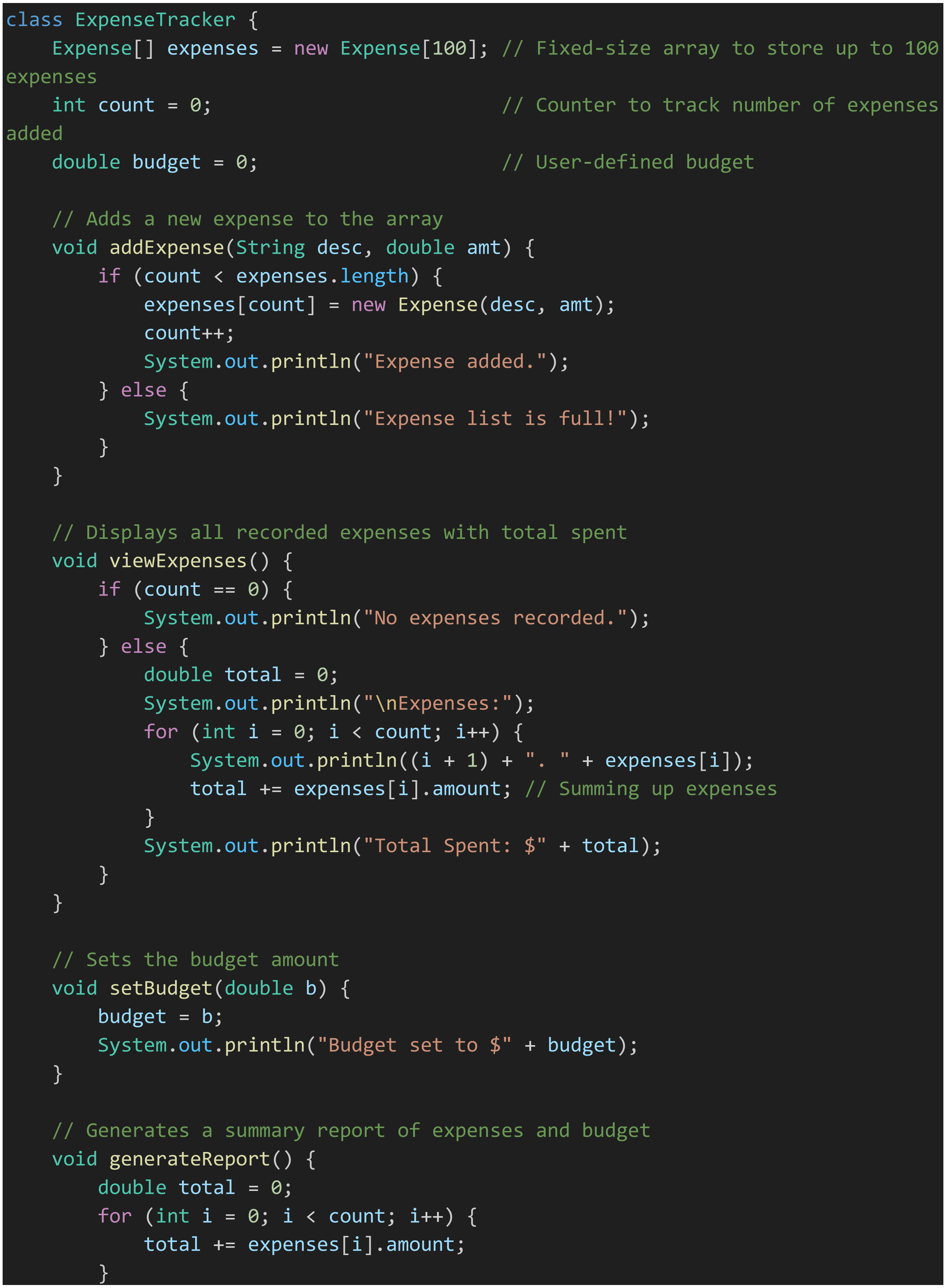
Generate report over budget	Add more to total > 50	"You are over budget by $X"
Exceed 100 expenses	Add 101st expense	"Expense list is full!"
View expenses without adding	Option 2 on fresh start	"No expenses recorded."

3. Bugs / Issues

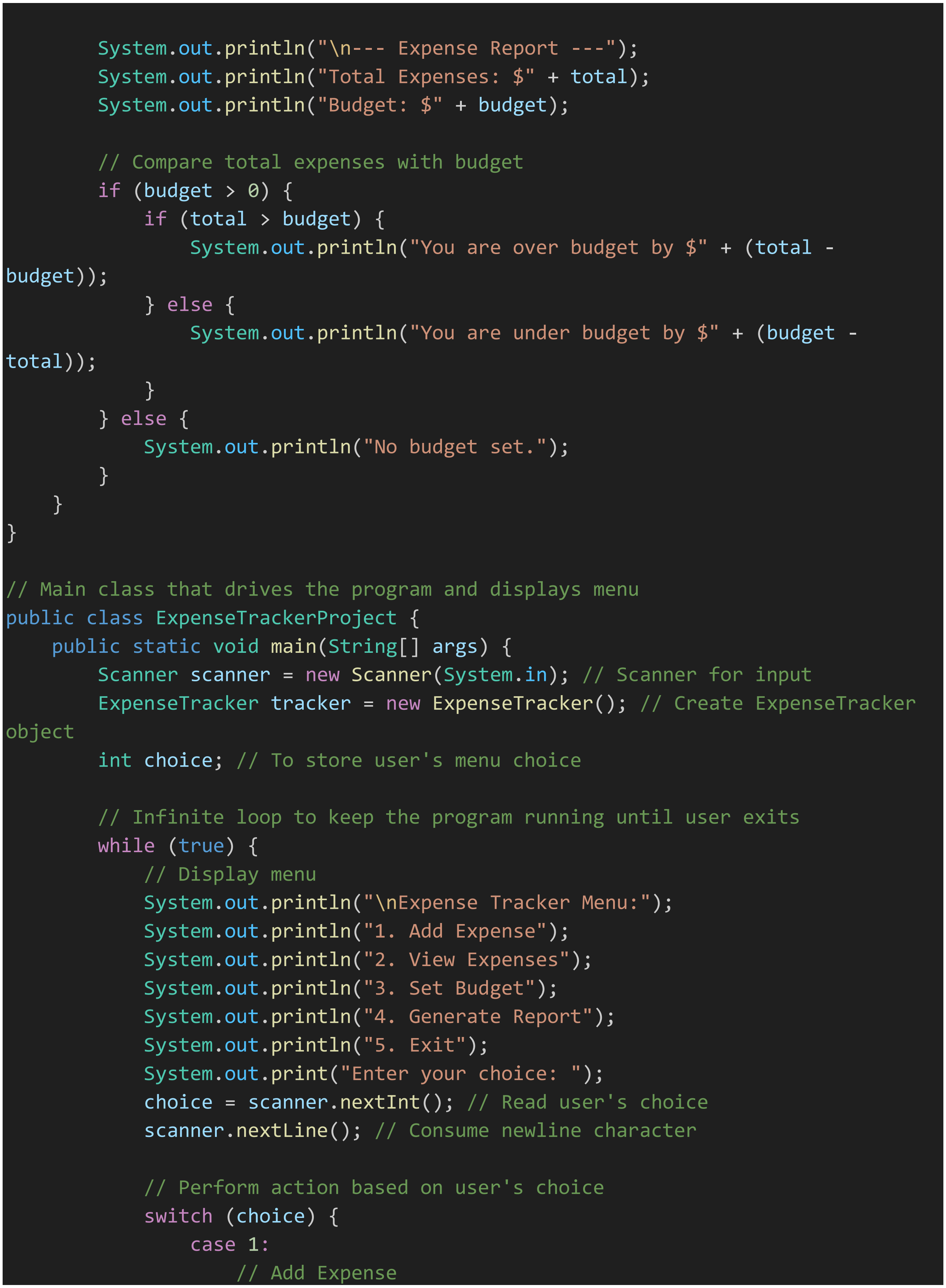
Issue	Description	Status
No delete feature	Cannot remove an expense once added	Future update
No input validation	Crash if invalid input (e.g., string instead of number)	Known issue
Data not saved to file	All data lost after program exits	Future update
Fixed-size array limit	Only supports up to 100 expenses	Design choice

4. Code Implementation With Comments:

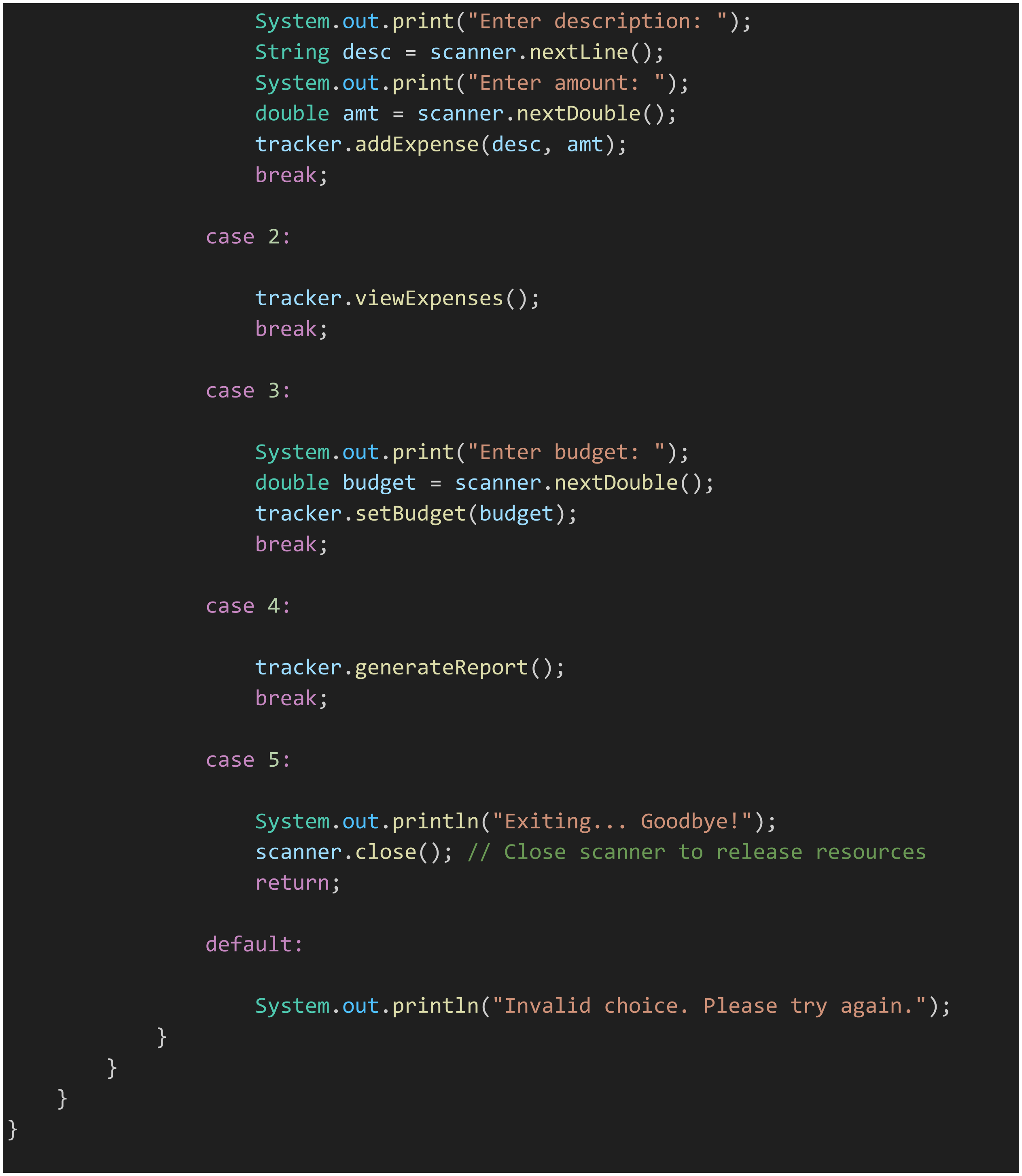
```
import java.util.Scanner;

// Class to hold a single expense
class Expense {
    String description; // Description of the expense
    double amount;      // Amount of the expense

    // Constructor to initialize expense details
    Expense(String description, double amount) {
        this.description = description;
        this.amount = amount;
    }

    // Override toString to return expense in readable format
    public String toString() {
        return description + " - $" + amount;
    }
}

// Class to manage all expenses and the budget
```



```
class ExpenseTracker {
    Expense[] expenses = new Expense[100]; // Fixed-size array to store up to 100
expenses
    int count = 0;                         // Counter to track number of expenses
added
    double budget = 0;                     // User-defined budget

    // Adds a new expense to the array
    void addExpense(String desc, double amt) {
        if (count < expenses.length) {
            expenses[count] = new Expense(desc, amt);
            count++;
            System.out.println("Expense added.");
        } else {
            System.out.println("Expense list is full!");
        }
    }

    // Displays all recorded expenses with total spent
    void viewExpenses() {
        if (count == 0) {
            System.out.println("No expenses recorded.");
        } else {
            double total = 0;
            System.out.println("\nExpenses:");
            for (int i = 0; i < count; i++) {
                System.out.println((i + 1) + ". " + expenses[i]);
                total += expenses[i].amount; // Summing up expenses
            }
            System.out.println("Total Spent: $" + total);
        }
    }

    // Sets the budget amount
    void setBudget(double b) {
        budget = b;
        System.out.println("Budget set to $" + budget);
    }

    // Generates a summary report of expenses and budget
    void generateReport() {
        double total = 0;
        for (int i = 0; i < count; i++) {
            total += expenses[i].amount;
        }
```

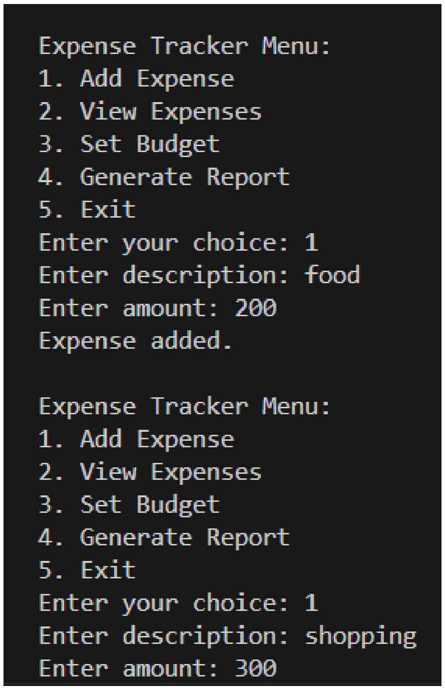


```
        System.out.println("\n--- Expense Report ---");
        System.out.println("Total Expenses: $" + total);
        System.out.println("Budget: $" + budget);

        // Compare total expenses with budget
        if (budget > 0) {
            if (total > budget) {
                System.out.println("You are over budget by $" + (total - 
budget));
            } else {
                System.out.println("You are under budget by $" + (budget - 
total));
            }
        } else {
            System.out.println("No budget set.");
        }
    }
}

// Main class that drives the program and displays menu
public class ExpenseTrackerProject {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Scanner for input
        ExpenseTracker tracker = new ExpenseTracker(); // Create ExpenseTracker 
object
        int choice; // To store user's menu choice

        // Infinite loop to keep the program running until user exits
        while (true) {
            // Display menu
            System.out.println("\nExpense Tracker Menu:");
            System.out.println("1. Add Expense");
            System.out.println("2. View Expenses");
            System.out.println("3. Set Budget");
            System.out.println("4. Generate Report");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt(); // Read user's choice
            scanner.nextLine(); // Consume newline character

            // Perform action based on user's choice
            switch (choice) {
                case 1:
                    // Add Expense
```

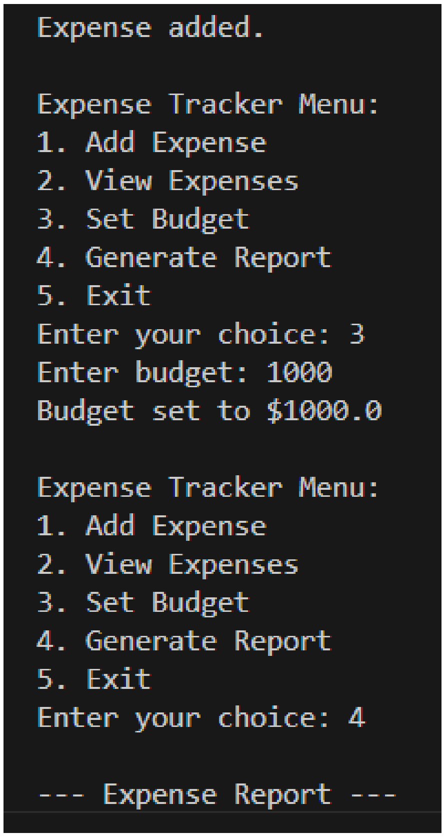


```
                    System.out.print("Enter description: ");
                    String desc = scanner.nextLine();
                    System.out.print("Enter amount: ");
                    double amt = scanner.nextDouble();
                    tracker.addExpense(desc, amt);
                    break;

                case 2:

                    tracker.viewExpenses();
                    break;

                case 3:

                    System.out.print("Enter budget: ");
                    double budget = scanner.nextDouble();
                    tracker.setBudget(budget);
                    break;

                case 4:

                    tracker.generateReport();
                    break;

                case 5:

                    System.out.println("Exiting... Goodbye!");
                    scanner.close(); // Close scanner to release resources
                    return;

                default:

                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }
}
```

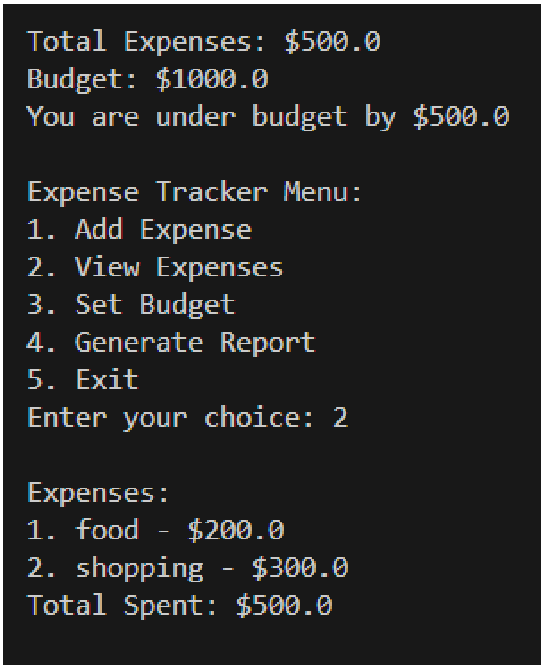
5. OutPut :



```
Expense Tracker Menu:
1. Add Expense
2. View Expenses
3. Set Budget
4. Generate Report
5. Exit
Enter your choice: 1
Enter description: food
Enter amount: 200
Expense added.

Expense Tracker Menu:
1. Add Expense
2. View Expenses
3. Set Budget
4. Generate Report
5. Exit
Enter your choice: 1
Enter description: shopping
Enter amount: 300
```



```
Expense added.

Expense Tracker Menu:
1. Add Expense
2. View Expenses
3. Set Budget
4. Generate Report
5. Exit
Enter your choice: 3
Enter budget: 1000
Budget set to $1000.0

Expense Tracker Menu:
1. Add Expense
2. View Expenses
3. Set Budget
4. Generate Report
5. Exit
Enter your choice: 4

--- Expense Report ---
```



```
Total Expenses: $500.0
Budget: $1000.0
You are under budget by $500.0

Expense Tracker Menu:
1. Add Expense
2. View Expenses
3. Set Budget
4. Generate Report
5. Exit
Enter your choice: 2

Expenses:
1. food - $200.0
2. shopping - $300.0
Total Spent: $500.0
```

Git Hub

Link:

https://github.com/huzaifachaudhry20s/Expense-Management-System.git