

# **MIGAL IT SOLUTIONS**

## **JAVA DEVELOPMENT INTERNSHIP**

**NAME: KAVYA.P**

**DEPARTMENT: B.TECH IT**

**COMPANY: MIGAL IT SOLUTIONS**

### **1. BASIC LEVEL PROJECT**

**PROJECT TITLE:** DAILY HABIT TRACKER(CONSOLE APPLICATION)

#### **DESCRIPTION:**

The Daily Habit Tracker is a console-based Java application that helps user strack their daily habits. It allows users to enter habits, mark them as completed or pending, and calculates a weekly habit score.

#### **JAVA PROGRAM:**

```
import java.util.Scanner;  
  
public class SimpleHabitTracker {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        int completed = 0;  
  
        System.out.print("Enter total number of habits: ");  
  
        int total = sc.nextInt();  
  
        for (int i = 1; i <= total; i++) {
```

```

System.out.print("Is habit " + i + " completed? (1-Yes / 0-No):
");

    Int status = sc.nextInt();

    if (status == 1) {

        completed++;

    }

}

System.out.println("\nCompleted Habits: " + completed);

System.out.println("Pending Habits: " + (total - completed));

int score = (completed * 100) / total;

System.out.println("Weekly Habit Score: " + score + "%");

sc.close();

}

}

```

## **OUTPUT/SCREENSHOTS:**

SimpleHabitTracker.java

```

1~ import java.util.Scanner;
2
3~ public class SimpleHabitTracker {
4
5~     public static void main(String[] args) {
6
7~         Scanner sc = new Scanner(System.in);
8
9~         int completed = 0;
10
11~        System.out.print("Enter total number of habits: ");
12~        int total = sc.nextInt();
13
14~        for (int i = 1; i <= total; i++) {
15
16~            System.out.print("Is habit " + i + " completed? (1-Yes /
0-No): ");

```

Run

Output

```

Enter total number of habits: 4
Is habit 1 completed? (1-Yes / 0-No): 1
Is habit 2 completed? (1-Yes / 0-No): 0
Is habit 3 completed? (1-Yes / 0-No): 1
Is habit 4 completed? (1-Yes / 0-No): 0

Completed Habits: 2
Pending Habits: 2
Weekly Habit Score: 50%

== Code Execution Successful ==

```

## **2. MEDIUM LEVEL PROJECT**

**PROJECT TITLE:** Expense Splitter (Mini Split wise Console App)

**DESCRIPTION:**

The Expense Splitter is a console-based Java application that divides a total expense equally among people and shows how much each person has to pay.

**JAVA PROGRAM:**

```
import java.util.Scanner;  
  
public class ExpenseSplitter {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter number of people: ");  
  
        int n = sc.nextInt();  
  
        System.out.print("Enter total expense amount: ");  
  
        double total Amount = sc.nextDouble();  
  
        double split Amount = total Amount / n;  
  
        System.out.println("\n---- PAYMENT SUMMARY ----");
```

```

        for (int i = 1; i <= n; i++) {
            System.out.println("Person " + i + " should pay: ₹" +
splitAmount);
        }
        sc.close();
    }
}

```

## **OUTPUT/SCREENSHOT:**

The screenshot shows a Java code editor with the following code in the ExpenseSplitter.java file:

```

1~ import java.util.Scanner;
2
3~ public class ExpenseSplitter {
4
5~     public static void main(String[] args) {
6
7~         Scanner sc = new Scanner(System.in);
8
9~         System.out.print("Enter number of people: ");
10~        int n = sc.nextInt();
11
12~        System.out.print("Enter total expense amount: ");
13~        double totalAmount = sc.nextDouble();
14
15~        double splitAmount = totalAmount / n;
16
17~        System.out.println("\n---- PAYMENT SUMMARY ----");
18~        for (int i = 1; i <= n; i++) {
19~            System.out.println("Person " + i + " should pay: ₹" +
splitAmount);
20~        }
21
22~        sc.close();
23~    }
24~}

```

The output window shows the following interaction:

```

Enter number of people: 4
Enter total expense amount: 2000
---- PAYMENT SUMMARY ----
Person 1 should pay: ₹500.0
Person 2 should pay: ₹500.0
Person 3 should pay: ₹500.0
Person 4 should pay: ₹500.0
== Code Execution Successful ==

```

## **3. ADVANCED LEVEL PROJECT**

**PROJECT TITLE:** Smart Parking Lot System

**DESCRIPTION:**

Smart Parking Lot System is a console-based Java application that calculates parking charges and stores ticket details using file handling.

### **JAVA PROGRAM:**

```
import java.util.Scanner;  
  
import java.io.FileWriter;  
  
import java.io.IOException;  
  
public class SmartParkingLot {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter vehicle number: ");  
  
        String vehicleNo = sc.nextLine();  
  
        System.out.print("Enter parking hours: ");  
  
        int hours = sc.nextInt();  
  
        int charge = hours * 50;  
  
        System.out.println("\n--- PARKING TICKET ---");  
  
        System.out.println("Vehicle Number: " + vehicleNo);  
  
        System.out.println("Parking Hours: " + hours);
```

```
System.out.println("Total Charge: ₹" + charge);

// File handling

try {

    FileWriter fw = new FileWriter("parking_ticket.txt");

    fw.write("Vehicle Number: " + vehicleNo + "\n");

    fw.write("Parking Hours: " + hours + "\n");

    fw.write("Total Charge: ₹" + charge + "\n");

    fw.close();

} catch (IOException e) {

    System.out.println("File error");

}

sc.close();

}

}
```

### **OUTPUT/SCREENSHOT:**

The screenshot shows a Java code editor interface with the following details:

- Code Area:** The file `SmartParkingLot.java` is open, containing a Java program that calculates parking charges.
- Run Button:** A blue "Run" button is visible at the top of the editor.
- Output Area:** The output window displays the execution results of the code.
- Output Content:**

```
Enter vehicle number: TN38CH6530
Enter parking hours: 2

--- PARKING TICKET ---
Vehicle Number: TN38CH6530
Parking Hours: 2
Total Charge: ₹100
File error

== Code Execution Successful ==
```
- Toolbar:** Standard toolbar icons for copy, paste, share, and run are present.
- Clear Button:** A "Clear" button is located in the top right corner of the output area.