B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

SPC AAT Report on

Wending/Event Budget Planner Tracker

TITLE

Submitted in partial fulfillment of the requirements for AAT

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

NAME OF THE CANDIDATE

1. V Manohar USN: **1BM25CS038**

2. Vishnu R USN: **1BM25CS188**

Department of Computer Science and Engineering
B.M.S College of Engineering

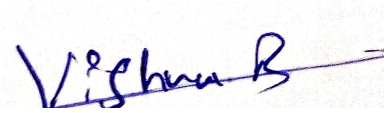
Bull Temple Road, Basavanagudi, Bangalore 560 019
2025-2026

B.M.S COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, **V Manohar** And **Vishnu R** students of 1st Semester, B.E, Department of Computer Science And Engineering, BMS College of Engineering, Bangalore, hereby declare that, this AAT Project entitled "Wending/Event Budget Planner Tracker" has been carried out in Department of CSE, BMS College of Engineering, Bangalore during the academic semester Sep 2025 – Jan 2026. We also declare that to the best of our knowledge and belief, the AAT Project report is not from part of any other report by any other students.

Student Name	**Student Signature**
1. V Manohar	**1.** 
2. Vishnu R	**2.** 

BMS COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the AAT Project titled "**Wending/Event Budget Planner Tracker**" has been carried out by V Manohar (1BM25CS038) And Vishnu R (1BM25CS188) during the academic year 2025-2026.

Signature of the Faculty in Charge

Table of Contents

1. INTRODUCTION

The **Wedding/Event Planner Budget Tracker** is a C-based application designed to help users plan, monitor, and manage their event expenses in an organized and user-friendly manner. This program allows users to create an event, define different budget categories such as venue, food, decoration, photography, and other expenses, and then record both estimated and actual costs for each category. By comparing estimated and actual expenditures, the program helps users analyze whether the event is under budget, over budget, or perfectly balanced.

This project makes use of fundamental C programming concepts such as **structures**, **arrays**, **loops**, **conditional statements**, **string handling**, and **file handling**. Structures are used to group related data such as category name, estimated cost, and actual cost, ensuring better data organization. Arrays of structures allow the program to manage multiple budget categories efficiently. File handling enables permanent storage of event data, making it possible to save and retrieve budget information whenever required.

The menu-driven interface of the program provides ease of use, even for users with minimal technical knowledge. Users can add, edit, delete, and search budget categories, as well as view a complete summary of total expenses. This project not only simulates a real-world event management scenario but also strengthens practical understanding of core programming concepts.

Overall, the Wedding/Event Planner Budget Tracker serves as an effective tool for learning structured programming in C while addressing a real-life problem of budget management in event planning.

2. ALGORITHM

Step-1: Start the program.

Step-2: Declare a structure named Budget to store category name, estimated cost, and actual cost.

Step-3: In the main function, declare an array of Budget structures to store multiple budget categories and initialize required variables.

Step-4: Read the event name from the user. Step-5: Read the

event date from the user.

Step-6: Ask the user to enter the number of budget categories for the event.

Step-7: Initialize total estimated cost and total actual cost to zero.

Step-8: Repeat the following steps for each category using a loop:

- Read category name
- Read estimated cost.
- Read actual cost.
- Add estimated cost to total estimated cost.
- Add actual cost to total actual cost.

Step-9: Display the event name and event date

Step-10: Display all budget categories along with their estimated and actual costs.

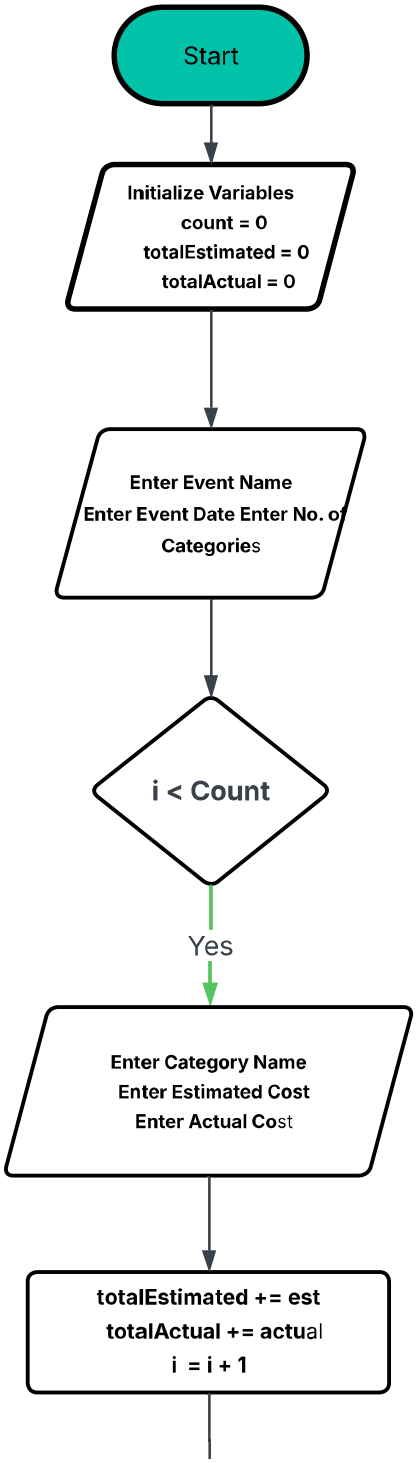
Step-11: Display the total estimated cost and total actual cost.

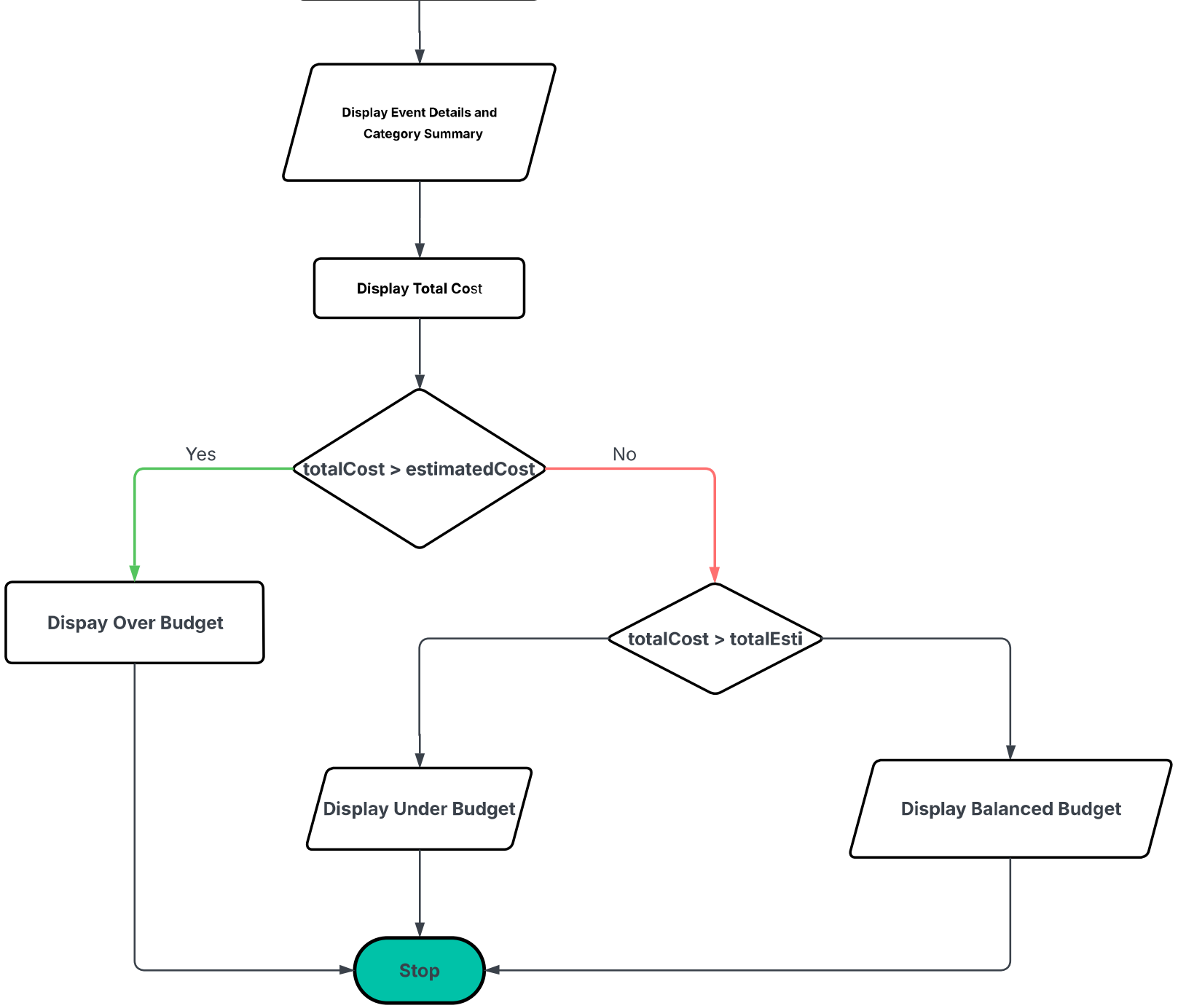
Step-12: Compare total estimated cost and total actual cost and display whether the event is over budget, under budget, or balanced.

Step-13: Display a message indicating successful execution of the program.

Step-14: Stop the program.

3. Flowchart


Start
Initialize Variables
count = 0
totalEstimated = 0
totalActual = 0
Enter Event Name
Enter Event Date Enter No. of
Categories
i < Count
Yes
Enter Category Name
Enter Estimated Cost
Enter Actual Cost
totalEstimated += est
totalActual += actual
i = i + 1


Display Event Details and Category Summary
Display Total Cost
totalCost > estimatedCost
Yes
No
Dispay Over Budget
totalCost > totalEsti
Display Under Budget
Display Balanced Budget
Stop

4. SOURCE CODE

```
#include <stdio.h>
#include <string.h>

struct Budget {
    char category[50];
    float estimated;
    float actual;
};

int main() {

    struct Budget b[10];
    int count = 0, i;

    char eventName[100];
    char eventDate[40];
    float totalEstimated = 0, totalActual = 0;

    /* Create Event */
    printf("Enter Event Name: ");
    scanf(" %[^\n]", eventName);
```

```
printf("Enter Event Date: ");
 scanf(" %[^\n]", eventDate);

 /* Add Categories */
 printf("\nEnter number of categories: ");
 scanf("%d", &count);

 for (i = 0; i < count; i++) {
    printf("\nEnter Category Name: ");
    scanf(" %[^\n]", b[i].category);

    printf("Enter Estimated Cost: ");
    scanf("%f", &b[i].estimated);

    printf("Enter Actual Cost: ");
    scanf("%f", &b[i].actual);

    totalEstimated += b[i].estimated;
    totalActual += b[i].actual;
 }
 /* Display Output */
 printf("\n=====================================\n");
 printf("WEDDING / EVENT BUDGET SUMMARY\n");
 printf("=====================================\n");
```

```
printf("Event Name : %s\n", eventName);
printf("Event Date : %s\n\n", eventDate);

for (i = 0; i < count; i++) {
    printf("%d. %s | Estimated: %.2f | Actual: %.2f\n",
           i + 1,
           b[i].category,
           b[i].estimated,
           b[i].actual);
}
printf("\nTotal Estimated Cost: %.2f\n", totalEstimated);
printf("Total Actual Cost    : %.2f\n", totalActual);

if (totalActual > totalEstimated)
    printf("Status: OVER Budget by %.2f\n",

        totalActual - totalEstimated);
else if (totalEstimated > totalActual)
    printf("Status: UNDER Budget by %.2f\n",
           totalEstimated - totalActual);
else
    printf("Status: Budget Balanced\n");
printf("\nProgram Executed Successfully\n");
return 0; }
```

5. RESULTS

```
Enter Event Name: Wedding
Enter Event Date: 06-03-2026

Enter number of categories: 3

Enter Category Name: Venue
Enter Estimated Cost: 200000
Enter Actual Cost: 220000

Enter Category Name: Food & Catering
Enter Estimated Cost: 300000
Enter Actual Cost: 290000

Enter Category Name: Photography
Enter Estimated Cost: 80000
Enter Actual Cost: 75000

==================================
WEDDING / EVENT BUDGET SUMMARY
==================================
Event Name : Wedding
Event Date : 06-03-2026

1. Venue | Estimated: 200000.00 | Actual: 220000.00
2. Food & Catering | Estimated: 300000.00 | Actual: 290000.00
3. Photography | Estimated: 80000.00 | Actual: 75000.00

Total Estimated Cost: 580000.00
Total Actual Cost    : 585000.00
Status: OVER Budget by 5000.00

Program Executed Successfully
```

6. REFERENCES

1. **E. Balagurusamy**, *Programming in ANSI C*, McGraw Hill Education – referred for understanding basic C programming concepts, structures, loops, and input/output operations.

2. **Brian W. Kernighan** and **Dennis M. Ritchie**, *The C Programming Language*, Pearson Publications – referred for fundamental syntax and structured programming concepts.

3. **GeeksforGeeks**, C Programming Tutorials – used for understanding practical implementation of arrays, structures, and program logic.

4. **TutorialsPoint,** C Programming – referred for examples related to control statements, loops, and formatted output.

5. Class Notes and **Lecture Materials** provided by faculty – used for understanding **algorithm writing, flowchart design**, and project documentation format.

6. **OpenAI – ChatGPT** – used for concept clarification, algorithm explanation, flowchart guidance, and assistance in understanding and improving the C program logic.