**Project: Smart Expense Tracker**

This C program allows users to add expenses, view them, and calculate the total expenses in a simple way.

**Code for the project:**

#include <stdio.h>

#define MAX 100

struct Expense {

char category[30];

float amount;

} expenses[MAX];

int count = 0;

void addExpense() {

printf("\nEnter Category: ");

scanf(" %[^\n]", expenses[count].category);

printf("Enter Amount: ");

scanf("%f", &expenses[count].amount);

count++;

}

void viewExpenses() {

printf("\n--- Expenses ---\n");

for (int i = 0; i < count; i++) {

printf("%d. %s - $%.2f\n", i + 1, expenses[i].category, expenses[i].amount);

}

}

void totalExpense() {

float total = 0;

for (int i = 0; i < count; i++) total += expenses[i].amount;

printf("\nTotal Expenses: $%.2f\n", total);

}

// Main function with menu

int main() {

int choice;

while (1) {

printf("\n1. Add Expense\n2. View Expenses\n3. Total Expenses\n4. Exit\nChoice: ");

scanf("%d", &choice);

if (choice == 1) addExpense();

else if (choice == 2) viewExpenses();

else if (choice == 3) totalExpense();

else if (choice == 4) break;

else printf("\nInvalid choice!\n");

}

    return 0;

}

Code Breakdown & Explanation:

**1.Header file**

*#include <stdio.h>*

\*Includes the Standard Input/Output Library to use functions like printf() and scanf().

**2. Define Maximum Expenses**

*#define MAX 100*

\*Defines a constant MAX for the maximum number of expenses the program can store.

**3. Structure to Store Expenses**

*struct Expense {*

*char category[30];*

*float amount;*

*} expenses[MAX];*

\*Defines a structure (struct Expense) to store:

category[30] → A string to store the category (e.g., "Food", "Travel").

amount → A float variable to store the expense amount.

Declares an array of structures (expenses[MAX]) to store multiple expenses.

**4. Global Variable to Track Expense Count**

*int count = 0;*

\*Count keeps track of the total number of expenses added.

**5. Function to Add an Expense**

*void addExpense() {*

*printf("\nEnter Category: ");*

*scanf(" %[^\n]", expenses[count].category); // Reads category (including spaces)*

*printf("Enter Amount: ");*

*scanf("%f", &expenses[count].amount); // Reads the amount*

*count++; // Increases expense count*

*}*

\*Asks the user to enter a category and amount.

\*Stores the data in the expenses array.

\*Increments count to track the total expenses added.

**6. Function to View All Expenses**

*void viewExpenses() {*

*printf("\n--- Expenses ---\n");*

*for (int i = 0; i < count; i++) {*

*printf("%d. %s - $%.2f\n", i + 1, expenses[i].category, expenses[i].amount);*

*}*

*}*

\*Loops through the expenses array and prints each expense with:

\*Serial number (i + 1)

\*Category name

\*Amount spent

\*If no expenses exist (count == 0), it simply prints nothing.

**7. Function to Calculate Total Expenses**

*void totalExpense() {*

*float total = 0;*

*for (int i = 0; i < count; i++) total += expenses[i].amount;*

*printf("\nTotal Expenses: $%.2f\n", total);*

*}*

\*Loops through all expenses and adds up the amount field.

\*Prints the total expense value.

**8. Main Function (User Menu & Program Execution)**

*int main() {*

*int choice;*

*while (1) {*

*printf("\n1. Add Expense\n2. View Expenses\n3. Total Expenses\n4. Exit\nChoice: ");*

*scanf("%d", &choice);*

*if (choice == 1) addExpense();*

*else if (choice == 2) viewExpenses();*

*else if (choice == 3) totalExpense();*

*else if (choice == 4) break;*

*else printf("\nInvalid choice!\n");*

*}*

*return 0; }*

\*Displays a menu with 4 options:

1. Add Expense

2. View Expenses

3. Calculate Total Expenses

4. Exit

\*takes user input (scanf("%d", &choice)) and calls the corresponding function.

\*If the user enters 4, the program exits (break).

\*If the user enters an invalid choice, it shows an error message.

**How It Works (Example Run)**

1. Add Expense

2. View Expenses

3. Total Expenses

4. Exit

Choice: 1

Enter Category: Food

Enter Amount: 20.5

1. Add Expense

2. View Expenses

3. Total Expenses

4. Exit

Choice: 1

Enter Category: Travel

Enter Amount: 15

1. Add Expense

2. View Expenses

3. Total Expenses

4. Exit

Choice: 2

Expenses-

1. Food - $20.50

2. Travel - $15.00

1. Add Expense

2. View Expenses

3. Total Expenses

4. Exit

Choice: 3

Total Expenses: $35.50