

Expense Tracker Code

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

#define MAX_EXPENSES 100
#define MAX_CATEGORIES 10

typedef struct {
    char category[50];
    float amount;
    char date[11]; // format: YYYY-MM-DD
} Expense;

typedef struct {
    Expense expenses[MAX_EXPENSES];
    int expense_count;
    float budget;
} ExpenseTracker;

void addExpense(ExpenseTracker *tracker);
void viewExpenses(ExpenseTracker *tracker);
void monthlyReport(ExpenseTracker *tracker);
void annualReport(ExpenseTracker *tracker);
void setBudget(ExpenseTracker *tracker);
float get_total_expense(ExpenseTracker *tracker);
void displayMenu();

int main() {
    ExpenseTracker tracker = { .expense_count = 0, .budget = 0.0 };
    int choice;

    setBudget(&tracker);
    do {
        displayMenu();
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: addExpense(&tracker); break;
            case 2: viewExpenses(&tracker); break;
            case 3: monthlyReport(&tracker); break;
            case 4: annualReport(&tracker); break;
            case 5: setBudget(&tracker); break;
            case 0: printf("Exiting...\n"); break;
        }
    } while (choice != 0);
}
```

```

        default: printf("Invalid choice! Please try again.\n");
    }
} while (choice != 0);

return 0;
}

void displayMenu() {
    printf("\nExpense Tracker Menu:\n");
    printf("1. Add Expense\n");
    printf("2. View Expenses\n");
    printf("3. Monthly Report\n");
    printf("4. Annual Report\n");
    printf("5. Set Budget\n");
    printf("0. Exit\n");
}

void addExpense(ExpenseTracker *tracker) {

    Expense new_expense;
    printf("Enter category: ");
    scanf("%s", new_expense.category);
    printf("Enter amount: ");
    scanf("%f", &new_expense.amount);
    printf("Enter date (YYYY-MM-DD): ");
    scanf("%s", new_expense.date);

    tracker->expenses[tracker->expense_count++] = new_expense;
    printf("Expense added successfully!\n");

    float total_expense = get_total_expense(tracker);
    if (total_expense > tracker->budget) {
        printf("!!! Alert: Total expense %.2f exceeds the
budget%.2f!!!", total_expense, tracker->budget);
    }
}

void viewExpenses(ExpenseTracker *tracker) {
    printf("\nExpenses:\n");
    for (int i = 0; i < tracker->expense_count; i++) {
        printf("Category: %s, Amount: %.2f, Date: %s\n",
            tracker->expenses[i].category,
            tracker->expenses[i].amount,
            tracker->expenses[i].date);
    }
}

void monthlyReport(ExpenseTracker *tracker) {
    char ask_budget[2];
    float total = 0.0;
    char month[3];
    char year[5];

```

```

printf("Enter month (MM): ");
scanf("%s", month);
printf("Enter year (YYYY): ");
scanf("%s", year);

printf("\nMonthly Report for %s-%s:\n", year, month);
for (int i = 0; i < tracker->expense_count; i++) {
    if (strcmp(tracker->expenses[i].date + 5, month, 2) == 0 &&
        strcmp(tracker->expenses[i].date, year, 4) == 0) {
        printf("Category: %s, Amount: %.2f\n",
            tracker->expenses[i].category,
            tracker->expenses[i].amount);
        total += tracker->expenses[i].amount;
    }
}
printf("Total Expenses: %.2f\n", total);
}

void annualReport(ExpenseTracker *tracker) {
    char ask_budget[2];
    float total = 0.0;
    char year[5];
    printf("Enter year (YYYY): ");
    scanf("%s", year);

    printf("\nAnnual Report for %s:\n", year);
    for (int i = 0; i < tracker->expense_count; i++) {
        if (strcmp(tracker->expenses[i].date, year, 4) == 0) {
            printf("Category: %s, Amount: %.2f\n",
                tracker->expenses[i].category,
                tracker->expenses[i].amount);
            total += tracker->expenses[i].amount;
        }
    }
    printf("Total Expenses: %.2f\n", total);
}

void setBudget(ExpenseTracker *tracker) {
    printf("Enter budget: ");
    scanf("%f", &tracker->budget);
    printf("Budget set to %.2f\n", tracker->budget);
}

float get_total_expense(ExpenseTracker *tracker) {
    float total = 0.0;
    for (int i = 0; i < tracker->expense_count; i++) {
        total += tracker->expenses[i].amount;
    }
    return total;
}

```