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

#define FILE_NAME "expenses.txt"

struct Expense {
   char date[15];
   char category[50];
   float amount;
};

void clearInputBuffer() {
   while (getchar() != '\n');
}

void addExpense() {
   FILE *file = fopen(FILE_NAME, "a");
   if (file == NULL) {
      printf("Error opening file!\n");
      return;
   }

   struct Expense e;

   printf("Enter Date (DD-MM-YYYY): ");
   scanf("%s", e.date);
   clearInputBuffer();

   printf("Enter Category: ");
   fgets(e.category, sizeof(e.category), stdin);
   e.category[strcspn(e.category, "\n")] = 0;

   do {
      printf("Enter Amount: ");
      scanf("%f", &e.amount);
```

```
        if (e.amount <= 0)
            printf("Amount must be positive!\n");
    } while (e.amount <= 0);

    fprintf(file, "%s|%s|%.2f\n", e.date, e.category, e.amount);
    fclose(file);

    printf("Expense added successfully!\n");
}

void viewExpenses() {
    FILE *file = fopen(FILE_NAME, "r");
    if (file == NULL) {
        printf("No records found!\n");
        return;
    }

    struct Expense e;
    printf("\n%-15s %-20s %-10s\n", "Date", "Category", "Amount");
    printf("--------------------------------------------------\n");

    while (fscanf(file, "%[^|]|%[^|]|%f\n", e.date, e.category, &e.amount) != EOF) {
        printf("%-15s %-20s %.2f\n", e.date, e.category, e.amount);
    }

    fclose(file);
}

void totalExpense() {
    FILE *file = fopen(FILE_NAME, "r");
    if (file == NULL) {
        printf("No records found!\n");
        return;
    }

    struct Expense e;
    float total = 0;

    while (fscanf(file, "%[^|]|%[^|]|%f\n", e.date, e.category, &e.amount) != EOF) {
        total += e.amount;
    }

    printf("Total Expense: %.2f\n", total);
    fclose(file);
}

void searchByCategory() {
    FILE *file = fopen(FILE_NAME, "r");
    if (file == NULL) {
        printf("No records found!\n");
        return;
    }

    char search[50];
    clearInputBuffer();
    printf("Enter Category to Search: ");
    fgets(search, sizeof(search), stdin);
    search[strcspn(search, "\n")] = 0;

    struct Expense e;
    int found = 0;

    while (fscanf(file, "%[^|]|%[^|]|%f\n", e.date, e.category, &e.amount) != EOF) {
        if (strcmp(e.category, search) == 0) {
            printf("%-15s %-20s %.2f\n", e.date, e.category, e.amount);
            found = 1;
        }
    }

    if (!found)
        printf("No matching records found!\n");
    fclose(file);
}

void deleteExpense() {
    FILE *file = fopen(FILE_NAME, "r");
    FILE *temp = fopen("temp.txt", "w");

    if (file == NULL || temp == NULL) {
        printf("Error opening file!\n");
        return;
    }

    char deleteDate[15];
    printf("Enter Date of Expense to Delete (DD-MM-YYYY): ");
    scanf("%s", deleteDate);

    struct Expense e;
    int deleted = 0;

    while (fscanf(file, "%[^|]|%[^|]|%f\n", e.date, e.category, &e.amount) != EOF) {
        if (strcmp(e.date, deleteDate) != 0) {
            fprintf(temp, "%s|%s|%.2f\n", e.date, e.category, e.amount);
        } else {
            deleted = 1;
        }
    }

    fclose(file);
    fclose(temp);

    remove(FILE_NAME);
    rename("temp.txt", FILE_NAME);

    if (deleted)
        printf("Expense deleted successfully!\n");
    else
```

```
        printf("No matching record found!\n");
}

void monthlySummary() {
    FILE *file = fopen(FILE_NAME, "r");
    if (file == NULL) {
        printf("No records found!\n");
        return;
    }

    char month[8];
    printf("Enter Month and Year (MM-YYYY): ");
    scanf("%s", month);

    struct Expense e;
    float total = 0;

    while (fscanf(file, "%[^|]|%[^|]|%f\n", e.date, e.category, &e.amount) != EOF) {
        if (strstr(e.date, month) != NULL) {
            total += e.amount;
        }
    }

    printf("Total Expense for %s: %.2f\n", month, total);
    fclose(file);
}

int main() {
    int choice;

    while (1) {
        printf("\n===== Advanced Expense Tracker =====\n");
        printf("1. Add Expense\n");
        printf("2. View Expenses\n");
        printf("3. Total Expense\n");
        printf("4. Search by Category\n");
        printf("5. Delete by Date\n");
        printf("6. Monthly Summary\n");
        printf("7. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: addExpense(); break;
            case 2: viewExpenses(); break;
            case 3: totalExpense(); break;
            case 4: searchByCategory(); break;
            case 5: deleteExpense(); break;
            case 6: monthlySummary(); break;
            case 7: exit(0);
            default: printf("Invalid choice!\n");
        }
    }

    return 0;
}
```