C++ code for Expense Tracker which store the data of the user using file handling concept.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <iomanip>
using namespace std;

class Expense {
private:
    string date;
    string category;
    double amount;

public:
    // Constructor
    Expense(string d, string c, double a) : date(d), category(c), amount(a) {}

    // Accessors
    string getDate() const { return date; }
    string getCategory() const { return category; }
    double getAmount() const { return amount; }

    // Display Expense
    void displayExpense() const {
        cout << "Date: " << date << ", Category: " << category << ", Amount: $"
<< fixed << setprecision(2) << amount << endl;
    }
};

class ExpenseTracker {
private:
    vector<Expense> expenses;

    void saveToFile() {
        ofstream file("expenses.txt");
        if (file.is_open()) {
            for (const auto& expense : expenses) {
                file << expense.getDate() << "," << expense.getCategory() <<
"," << expense.getAmount() << "\n";
            }
            file.close();
        }
    }

    void loadFromFile() {
        ifstream file("expenses.txt");
        if (file.is_open()) {
            string date, category;
            double amount;
            while (file >> date >> category >> amount) {
                expenses.emplace_back(date, category, amount);
            }
            file.close();
        }
    }

public:
    // Constructor
    ExpenseTracker() { loadFromFile(); }

    // Add an expense
    void addExpense() {
        string date, category;
        double amount;

        cout << "Enter date (DD-MM-YYYY): ";
        cin >> date;
        cout << "Enter category (Food, Transport, Utilities, etc.): ";
        cin >> category;
        cout << "Enter amount: ";
        cin >> amount;

        expenses.emplace_back(date, category, amount);
        saveToFile();

        cout << "Expense added successfully!\n";
    }

    // View all expenses
    void viewExpenses() const {
        cout << "\n--- All Expenses ---\n";
        for (const auto& expense : expenses) {
            expense.displayExpense();
        }
    }
    // Generate report by category
    void generateReport() const {
        double total = 0.0;
        cout << "\n--- Expense Report ---\n";
        for (const auto& expense : expenses) {
            total += expense.getAmount();
        }
        cout << "Total expenses: $" << fixed << setprecision(2) << total <<
"\n";
    }
};

int main() {
    ExpenseTracker tracker;
    int choice;

    while (true) {
        cout << "\nExpense Tracker Menu:\n";
        cout << "1. Add Expense\n";
        cout << "2. View Expenses\n";
        cout << "3. Generate Report\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                tracker.addExpense();
                break;
            case 2:
                tracker.viewExpenses();
                break;
            case 3:
                tracker.generateReport();
                break;
            case 4:
                cout << "Exiting the program...\n";
                return 0;
            default:
                cout << "Invalid choice. Please try again.\n";
        }
    }

    return 0;
}
```

Output:

```
Expense Tracker Menu:
1. Add Expense
2. View Expenses
3. Generate Report
4. Exit
Enter your choice:
```