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

// একটি খরচের ডেটা সংরক্ষণের জন্য ক্লাস
class Expense {
public:
    string date; // তারিখ (DD/MM/YYYY)
    string category; // খরচের ধরন (e.g., Food, Travel)
    double amount; // খরচের পরিমাণ

    Expense(string d, string c, double a) : date(d), category(c), amount(a) {}
};

// Expense Tracker ক্লাস
class ExpenseTracker {
private:
    vector<Expense> expenses;

public:
    // নতুন খরচ যোগ করা
    void addExpense(string date, string category, double amount) {
        expenses.push_back(Expense(date, category, amount));
        cout << "Expense added successfully!" << endl;
    }

    // সব খরচ দেখা
    void viewExpenses() {
        if (expenses.empty()) {
            cout << "No expenses recorded yet!" << endl;
            return;
        }

        cout << "\nAll Expenses:\n";
        for (const auto& expense : expenses) {
            cout << "Date: " << expense.date << ", Category: " << expense.category
                 << ", Amount: $" << expense.amount << endl;
        }
    }

    // নির্দিষ্ট দিনের খরচ দেখা
    void viewExpensesByDate(string date) {
        bool found = false;
        for (const auto& expense : expenses) {
            if (expense.date == date) {
                cout << "Date: " << expense.date << ", Category: " << expense.category
                     << ", Amount: $" << expense.amount << endl;
                found = true;
            }
        }
        if (!found) {
            cout << "No expenses found for the date: " << date << endl;
        }
    }

    // ফাইলে খরচ সংরক্ষণ করা
    void saveToFile() {
        ofstream outFile("expenses.txt");
        for (const auto& expense : expenses) {
            outFile << expense.date << "," << expense.category << "," << expense.amount << endl;
        }
        outFile.close();
        cout << "Expenses saved to file successfully!" << endl;
    }

    // ফাইল থেকে খরচ লোড করা
    void loadFromFile() {
        ifstream inFile("expenses.txt");
        if (!inFile) {
            cout << "No previous expenses found." << endl;
            return;
        }
        expenses.clear();
        string date, category;
        double amount;
        while (inFile >> date >> category >> amount) {
            expenses.push_back(Expense(date, category, amount));
        }
        inFile.close();
        cout << "Expenses loaded from file successfully!" << endl;
    }

    // মাসিক রিপোর্ট
    void generateMonthlyReport() {
        if (expenses.empty()) {
            cout << "No expenses recorded yet!" << endl;
            return;
        }

        double total = 0;
        for (const auto& expense : expenses) {
            total += expense.amount;
        }
        cout << "\nMonthly Report:\n";
        cout << "Total Expenses: $" << total << endl;
    }
};

int main() {
    ExpenseTracker tracker;
    int choice;
    string date, category;
    double amount;

    do {
        cout << "\nPersonal Expense Tracker:\n";
        cout << "1. Add Expense\n";
        cout << "2. View All Expenses\n";
        cout << "3. View Expenses by Date\n";
        cout << "4. Save Expenses to File\n";
        cout << "5. Load Expenses from File\n";
        cout << "6. Generate Monthly Report\n";
        cout << "7. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter date (DD/MM/YYYY): ";
                cin >> date;
                cout << "Enter category: ";
                cin >> category;
                cout << "Enter amount: ";
                cin >> amount;
                tracker.addExpense(date, category, amount);
                break;
            case 2:
                tracker.viewExpenses();
                break;
            case 3:
                cout << "Enter date (DD/MM/YYYY): ";
                cin >> date;
                tracker.viewExpensesByDate(date);
                break;
            case 4:
                tracker.saveToFile();
                break;
            case 5:
                tracker.loadFromFile();
                break;
            case 6:
                tracker.generateMonthlyReport();
                break;
            case 7:
                cout << "Exiting... Have a great day!" << endl;
                break;
            default:
```

```
            cout << "Invalid choice. Try again!" << endl;
        }
    } while (choice != 7);

    return 0;
}
```