CS214-3267

Project report

(Monthly Budget Tracker)

Presented by :

Abdullah waleed alshabi	431109334
Fahad mansour alkaridis	431109379
Ziad Refaei Almutairi	431109416
Hamad Saleh Almadi	431107688
Moath saleh alburaidi	421109543

Introduction:

This tool can be used as an all-inclusive system in which people successfully organize their budgets. It smoothens the procedure of checking all the income sources such part part-time work or allowances and simultaneously helps them record all the expenditures. This savings tool will offer well-defined suggestions which in turn enable the users to take control of their spending habits. Also, the expense Tracker acts as a guide that leads users to the financial independence of individuals.

Required Knowledge:

- **Understanding the Requirements:** With this tool, you should inspect all requirements mentioned in the project description, including income input, expense addition and modification, and budget summary display.
- **Data Structures:** The deployment of the Expense Tracker Tool entails having information about data structures. In this project, we implement the linked list data structure to create an inefficiently efficient expense data management. insertion and deletion of elements makes them a good choice for keeping tabs on expenses.
- **Header Files:** The project uses a header file that contains classes definitions and function prototypes. This enables us to create well-organized code, which separates the interface from implementation and increases code readability and maintainability.
- **Classes:** The "Monthly Budget" class in the Expense Tracker Tool deals with tracking expenses for one month. It helps with tasks like getting, adding, editing, and showing expenses. Knowing about classes and object-oriented programming is key to making a program that works well on its own and can be reused.

Project goals :

1. Create an easy-to-use and clear user interface.
2. Print a summary of expenses in the form of simplified categories.
3. Knowing what the user spends most of his money on.
4. Execute the program in C++ language without errors.

Inputs:

• **Income:** User enters monthly income.

• **Expense Addition:** The user selects the category of the expense and the amount.

• **Expense Modification:** The user specifies an expense category and enters a new expense amount.

• **Menu Selection:** The User selects actions for adding expenses, changing income, visualizing the summary, or canceling the operation.

Outputs:

• **Expense Added Successfully:** Proof of claim submission.

• **Expense Modified Successfully:** The role of consent in the modification of expenses.

• **Expense Category Not Found:** An error will occur if the mentioned category doesn’t exist.

• **Budget Summary:** Showing amount spent, amount left, and budget remaining.

• **Low Budget Warning:** Informing the passenger when their funds are running low.

• **Exit Message:** Emigration message upon course completion.

Conclusion:

Through this product, people will have a tool that is simple to use to control their financial duties. Having functions such as income input, expenditure tracking, and budget summary, a facility is created by such tool whereby financial headway and expenditure propriety are promoted.

Using data structures and a good OOP paradigm, we've designed a modularity and performance-improved instrument.

The code in C++ language

```
#include <iostream>

#include <iomanip>

using namespace std;

class MonthlyBudget {

private:

    struct node {

        double amount;

        string category;

        node* next;

    }*head;

    double income, total, remaining;

public:

    MonthlyBudget() {

        head = NULL;

        income = 0;

        int c;

        cout<<"Enter your encome:"<<endl;

        cin>>income;

        do{

            cout<<"Enter 1 for add expense"<<endl<<"2 for modify your encome"<<endl<<"3 for display"<<endl<<"-1 for
exit:"<<endl;



            cin>>c;

            if(c==1){

                this->AddE();



            }

            else if(c==2){
```

```
            cout<<"Enter your encome:"<<endl;
            int p;
            cin>>p;
            income=0;
            income+=p;




        }
        else if(c==3){
            this->display();
        }
        else if(c==-1){
            cout<<"You end the program!"<<endl;
        }
        else{
            cout<<"You enter wrong number"<<endl;}
    }while(c!=-1);
}
void AddE() {
    int num, choose;
    string str;
    // التعديل او الاضافة بين يختار
    cout << "1 For Modify" << endl;
    cout << "2 For Add" << endl;
    cin >> choose;
    if (choose == 1) {
        node* temp = head;
        bool m = false;
        // سعره يغير و النوع يدخل السعر في يكون التغيير
        cout << "Enter expense category to Modify: ";
```

```
        cin >> str;
        cout << "Enter expense amount to Modify: $";
        cin >> num;
        if (temp == NULL) {
            cout << "there is no Expense to Modify" << endl;
            return;
        }
        else {
            while (temp != NULL) {
                if (temp->category == str) {
                    // القديم السعر ينقص النوع لقى اذا
                    total -= temp->amount;
                    // الجديد السعر يضيف
                    total += num;
                    remaining = income - total;
                    temp->amount = num;
                    m = true;
                    cout << "Expense modified successfully!" << endl;
                    break;
                }
                temp = temp->next;
            }
            if (!m)
                cout << "Expense category not found." << endl;
        }
    }
    // عادية اضافة
    if (choose == 2) {
        cout << "Enter expense category to Add: ";
        cin >> str;
```

```
        cout << "Enter expense amount to Add: $";
        cin >> num;
        node* temp, * r;
        if (head == NULL) {
            temp = new node;
            temp->amount = num;
            temp->category = str;
            temp->next = NULL;
            head = temp;
            total += num;
            remaining = income - total;
            cout << "Expense added successfully!" << endl;
            return;
        }
        else {
            temp = head;
            while (temp->next != NULL)
                temp = temp->next;
        }
        remaining = income - total;
        if (num>remaining) {
            cout << "cant add Category: " << str << endl;
            cout << "Amount: $" << num << endl;
            cout << "There is no enough budget" << endl;
            return;
        }
        r = new node;
        r->amount = num;
        r->category = str;
        r->next = NULL;
```

```
            temp->next = r;
            total += num;
            remaining = income - total;
            cout << "Expense added successfully!" << endl;
        }
        if (remaining <= income / 5)
            cout << "Low budget " << remaining << endl;
    }
    void display() {
        cout << "Monthly Budget Summary" << endl;
        cout << "Income: $" << fixed << setprecision(2) << income << endl;
        cout << "Expenses:" << endl;
        cout << setw(15) << left << "Category" << setw(10) << right << "Amount" << endl;

        node* temp = head;
        total = 0;
        while (temp != nullptr) {
            cout << setw(15) << left << temp->category << setw(10) << right << temp->amount << endl;
            total += temp->amount;
            temp = temp->next;
        }

        cout << "Total Expenses: $" << fixed << setprecision(2) << total << endl;
        remaining = income - total;
        cout << "Remaining Budget: $" << fixed << setprecision(2) << remaining << endl;
        //thess void display function^^^^
    }
};
int main() {
    MonthlyBudget m = MonthlyBudget();
}
```