PERSONAL BUDGET TRACKER

A CAPSTONE PROJECT REPORT

(Object Oriented Programming with C++ using encapsulation-DSA0199)

Submitted to

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE & ENGINEERING

By

Naresh K (192211178),

Selvarathinam M (192211183)

Course Faculty

Mrs. Jayanthi S

SAVEETHA SCHOOL OF ENGINEERING,

SIMATS, CHENNAI - 602105

MARCH-2024

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUE OF MEDICAL AND TECHNICAL SCIENCES,CHENNAI - 602105

BONAFIDE CERTIFICATE

Certified that this project report **"PERSONAL BUDGET TRACKER"** is the Bonafide work of **"M. SELVARATHINAM, K. NARESH"** who carried out the project work under my supervision.

Submitted to

Mrs. Jayanthi S

(Course Faculty)
Department of Information Security,
Saveetha School of Engineering,
SIMATS

SIGNATURE of Course Faculty

ACKNOWLEDGEMENT

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our Honorable Chancellor **Dr. N. M. Veeraiyan**, Saveetha Institute of Medical and Technical Sciences, for his blessings and for being a source of inspiration. I sincerely thank our Director of Academics **Dr. Deepak Nallaswamy,** SIMATS, for his visionary thoughts and support. I am indebted to extend my gratitude to our Director **Dr. Ramya Deepak,** Saveetha School of Engineering, for facilitating us all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. B. Ramesh,** Principal, Saveetha School of Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my Course faculty **Mrs. Jayanthi S** for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinators, Review Panel External and Internal Members and the entire faculty of the Department of Design, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

STUDENT NAME's

TABLE OF CONTENTS

CHAPTER NO	TITLE
1	Introduction
2	Project Description and Goals:
3	Technical Specifications:
4	Design Approach and Details
5	Schedule, Tasks, and Milestones:
6	Project Demonstration:
7	Cost Analysis:
8	Result:
9	Discussion:
10	Conclusion:

Introduction:

The Personal Budget Tracker Application represents a significant step forward in financial management, offering a comprehensive solution for tracking and managing personal finances. Built upon the foundation of C++ programming, this application leverages the power of object-oriented principles to streamline processes and enhance efficiency. At its core, it includes essential features such as a login page, user dashboard, and customizable financial tracking tools, providing secure access and personalized experiences for different users.

Moreover, the application facilitates seamless budget management, allowing users to efficiently input income, categorize expenses, and track their financial status in real-time. With the view expense functionality, users can monitor their spending habits and adjust budgets accordingly, ensuring responsible financial planning. Additionally, the report generation feature simplifies financial analysis by accurately calculating totals and generating summaries for better decision-making. By integrating these functionalities within a cohesive and user-friendly interface, the Personal Budget Tracker Application empowers individuals to optimize their financial planning, control spending, and achieve their savings goals. As financial awareness becomes increasingly crucial, this application emerges as an indispensable tool for anyone seeking to maintain healthy financial habits and achieve financial stability.

Project Description and Goals:

The Personal Budget Tracker using C++ is a comprehensive software solution designed to help users manage their finances efficiently. This application includes features such as a secure login system, a user-friendly dashboard for tracking income and expenses, customizable budgeting options, and real-time financial summaries. Users can categorize their spending, monitor their progress toward savings goals, and generate detailed financial reports.

The primary goal of this project is to develop a robust and user-friendly system that simplifies personal finance management, enhances financial awareness, and promotes better financial decision-making. By providing users with an easy way to track their income and expenses, the application empowers them to maintain control over their finances, minimize unnecessary spending, and work toward long-term financial stability.

Functionality:

The Personal Budget Tracker will provide users with essential financial management tools, including user authentication, income and expense tracking, and financial summaries. Key functionalities include:

User Authentication : A secure login system ensures that only authorized users can access their financial data.

Income and Expense Management : Users can input, categorize, and track both income and expenses, ensuring accurate budget management.

Transaction Display : All recorded transactions are displayed in a clear and organized format for easy review.

Balance Calculation: The system calculates the current balance by factoring in all recorded income and expenses.

User-Friendly Interface: The interface will be designed to be intuitive, featuring clear navigation and user-friendly controls. Users will easily access features like adding transactions, viewing financial summaries, and checking their current balance.

Accurate Calculation:
Precise algorithms are implemented to ensure accurate calculations of income, expenses, and overall balance, allowing users to rely on the tool for their financial planning.

Error Handling:
Robust error handling mechanisms will be incorporated to detect and manage exceptions gracefully, such as incorrect inputs or invalid choices, with informative error messages provided to assist users.

Platform Compatibility:
The application will be developed in C++ and can be executed on various platforms where C++ is supported. Although primarily a desktop application, the structure allows for future expansion to other platforms if needed.

Documentation and Support:
Comprehensive documentation, including user manuals and troubleshooting guides, will be provided. The program will also feature inline help and prompts to guide users through its functions.

Testing and Validation:
Thorough testing of all system components will be conducted to ensure functionality, performance, and security. Each feature will be validated against predefined test cases to ensure the system behaves as expected under various conditions.

Technical Specifications:
The application is built using C++ and employs object-oriented principles. Data structures such as vectors will be used to manage transactions, while classes will be implemented for modular design and maintainability.

Design Approach and Details:
A modular and scalable design approach is adopted to facilitate future enhancements and modifications. Object-oriented design principles ensure that code reusability and maintainability are maintained, with classes such as `Transaction` and `BudgetTracker` managing the core functionalities.

This design ensures that users can efficiently manage their personal finances while maintaining a straightforward and easy-to-navigate system.

Schedule, Tasks, and Milestones:

Planning Phase:
Define Project Scope, Objectives, and Requirement: Establish the goals of the Personal Budget Tracker, focusing on features like income/expense tracking, balance calculation, and transaction display.
Identify Stakeholders and User Personas: Understand the needs of potential users, such as individuals looking to manage personal finances, and tailor the system to meet their requirements.
Create a Project Plan with Timelines and Resource Allocation: Develop a detailed timeline, outlining key tasks and assigning resources to each phase of the project.

Design Phase:
Design System Architecture: Establish the overall structure of the application, focusing on key components such as the `Transaction` and `BudgetTracker` classes.
Develop Wireframes and Mockups for User Interfaces: Create simple mockups to visualize how users will interact with the system's interface, focusing on user-friendliness.
Define Class Structures and Relationships: Outline the class hierarchy and how data flows between components, ensuring modularity and maintainability.

Project Demonstration:
A demonstration will be organized to showcase the Personal Budget Tracker's features, such as adding income and expenses, calculating balance, and displaying transaction history. Feedback will be gathered from potential users to refine the system and ensure it meets user expectations.

Cost Analysis:
Estimate project costs, including development time, tools, and potential third-party libraries if needed. Compare these costs with the benefits the tracker provides in terms of improved financial awareness and savings for users.

Result:
The Personal Budget Tracker serves as a user-friendly and comprehensive solution for managing personal finances. The application successfully integrates features such as income and expense tracking, balance calculation, and transaction history display, streamlining the process of financial management. By adhering to C++ best practices, the system ensures accuracy and reliability, allowing users to manage their finances with confidence.

Discussion:
The development process involved careful planning, thorough testing, and iterative refinement. Regular feedback from users was crucial in shaping the system's functionalities and ensuring that it met real-world financial management needs. The system's scalability allows for future updates, ensuring it can accommodate new features as user needs evolve.

By leveraging a solid object-oriented design, the Personal Budget Tracker empowers individuals to manage their finances, promoting better budgeting habits and fostering long-term financial stability. The system's flexibility allows for further enhancements, such as the addition of advanced reporting tools or support for multiple users.

Summary:
The **Personal Budget Tracker** revolutionizes personal financial management by providing an intuitive platform for tracking income, expenses, and overall balance. Its user-friendly design and accurate functionality make financial management simple and accessible for users of all backgrounds.

CODE:

```
#include <iostream>

#include <vector>

#include <string>

#include <iomanip>


class Transaction {
```

```
private:
    std::string description;
    double amount;
    std::string type; // "Income" or "Expense"

public:
    Transaction(const std::string& desc, double amt, const std::string& type)
        : description(desc), amount(amt), type(type) {}

    double getAmount() const { return amount; }
    std::string getType() const { return type; }
    std::string getDescription() const { return description; }

    void display() const {
        std::cout << std::left << std::setw(20) << description
                  << std::right << std::setw(10) << amount
                  << std::setw(10) << type << "\n";
    }
};

class BudgetTracker {
private:
    std::vector<Transaction> transactions;

public:
```

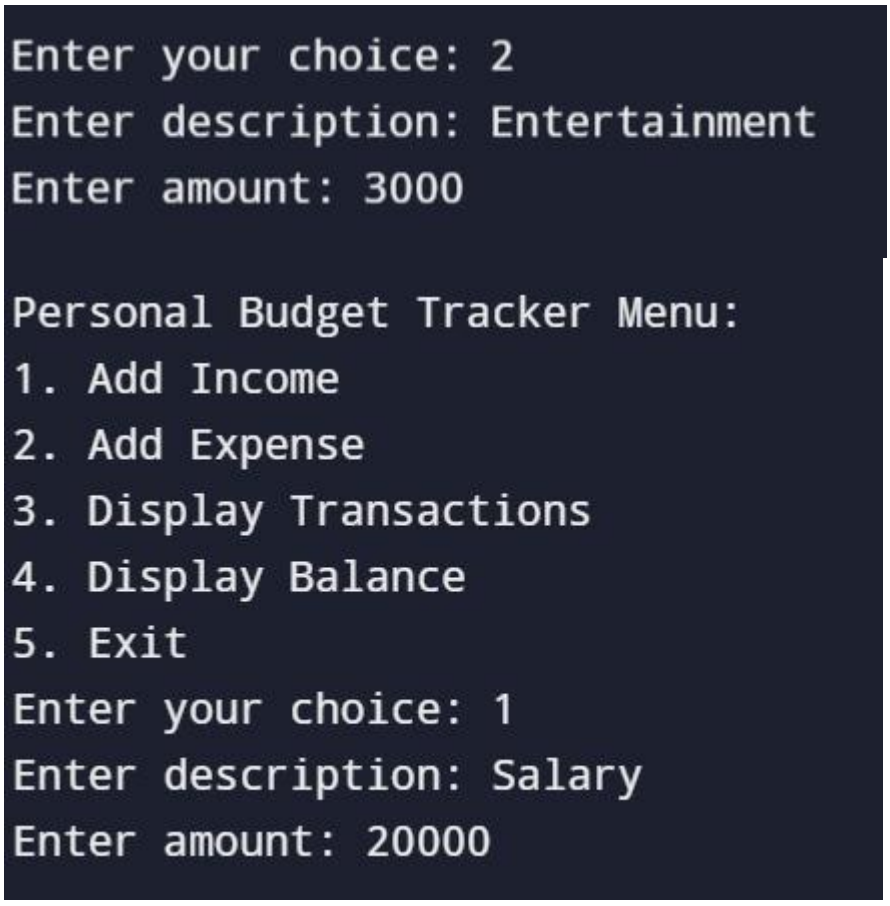
```
void addTransaction(const Transaction& transaction) {
    transactions.push_back(transaction);
}

void displayTransactions() const {
    std::cout << std::left << std::setw(20) << "Description"
              << std::right << std::setw(10) << "Amount"
              << std::setw(10) << "Type" << "\n";
    std::cout << "--------------------" << "----------" << "----------" << "\n";
    for (const auto& transaction : transactions) {
        transaction.display();
    }
}

double calculateBalance() const {
    double balance = 0.0;
    for (const auto& transaction : transactions) {
        if (transaction.getType() == "Income") {
            balance += transaction.getAmount();
        } else if (transaction.getType() == "Expense") {
            balance -= transaction.getAmount();
        }
    }
    return balance;
}
```

```
};

int main() {
    BudgetTracker budgetTracker;
    int choice;
    std::string description, type;
    double amount;

    while (true) {
        std::cout << "\nPersonal Budget Tracker Menu:\n";
        std::cout << "1. Add Income\n";
        std::cout << "2. Add Expense\n";
        std::cout << "3. Display Transactions\n";
        std::cout << "4. Display Balance\n";
        std::cout << "5. Exit\n";
        std::cout << "Enter your choice: ";
        std::cin >> choice;
        std::cin.ignore(); // Ignore newline character left in the input buffer

        switch (choice) {
            case 1:
                std::cout << "Enter description: ";
                std::getline(std::cin, description);
                std::cout << "Enter amount: ";
                std::cin >> amount;
```

```
            std::cin.ignore(); // Ignore newline character left in the input buffer
            budgetTracker.addTransaction(Transaction(description, amount, "Income"));
            break;

        case 2:
            std::cout << "Enter description: ";
            std::getline(std::cin, description);
            std::cout << "Enter amount: ";
            std::cin >> amount;
            std::cin.ignore(); // Ignore newline character left in the input buffer
            budgetTracker.addTransaction(Transaction(description, amount,
"Expense"));
            break;

        case 3:
            budgetTracker.displayTransactions();
            break;

        case 4:
            std::cout << "Current balance: " << budgetTracker.calculateBalance() <<
"\n";
            break;

        case 5:
            std::cout << "Exiting...\n";
```

```
            return 0;

        default:
            std::cout << "Invalid choice. Please try again.\n";
    }
}

return 0;
```

OUTPUT:



```
Enter your choice: 2
Enter description: Rent
Enter amount: 8000
```

```
Enter your choice: 4
Current balance: 3000
```

```
Enter your choice: 2
Enter description: Petrol
Enter amount: 3000
```

```
Enter your choice: 3
Description                 Amount      Type
-----------------------------------------
Salary                       20000    Income
Petrol                        3000   Expense
Entertainment                 3000   Expense
Rent                          8000   Expense
Food                          3000   Expense
```

```
Enter your choice: 4
Current balance: 3000
```

Conclusion:

In conclusion, the Personal Budget Tracker emerges as an essential tool for individuals looking to maintain control over their finances. By leveraging C++ to automate and streamline budgeting tasks, users can improve financial planning, reduce overspending, and achieve their savings goals. As financial literacy becomes increasingly important, adopting tools like the Personal Budget Tracker is crucial for fostering financial health and stability.