Project Result Report

Programming Fundamentals

University Name: FAST NUCES Karachi Campus

Department: Department of Computer Science

Course: Programming Fundamentals

Project Title: Budget Tracker with Data Privacy and Security

Submitted By: Muhammad Umer (25K-0983) & Khizar Khurseed (25K-0916)

Submitted To: Sir Sheeraz Iqbal

Semester: Fall 2025

Date: 22 Nov 2025

Abstract

The Budget Tracker with Data Privacy & Security is a command-line (CLI) based project developed as a part of the Programming Fundamentals course.

The system focuses on simple account creation, secure login, & user-specific budget record management.

Basic security features like password hashing, data encryption & separation of user data help in preserving user privacy.

1. Introduction

Managing budgets manually often leads to errors, missing records, & privacy concerns. In modern applications, users expect their financial data to be separated, private, and accessible only through secure login.

This project implements a simple yet functional Budget Tracking System using the C language. It allows users to:

- Register and login securely
- Add, view, edit, delete expenses or income
- Generate monthly reports or financial alerts

2. Objectives

- To develop a secure user account system using password hashing
- To maintain separate transaction files per user to ensure smooth CRUD Operations.
- To allow users to add, view, and track their transaction records with ensuring basic data privacy

3. System Design

Flow of the program:

Start
→ Home Menu → Login/ Register
→ If Login Success → Load User Dashboard
→ Dashboard Menu → Add/ View/ Edit/ Delete transactions
→ Dashboard Menu → Generate Monthly Report/ Alerts
→ Save data to file to avoid data loss when program stops
Exit

Algorithm

1. Start the program
2. Display home menu (login / signup)
3. If user registers:
    - Prompt user for username
    - Prompt user for password
    - Hash password using hashing algorithm
    - Save these credentials to file

4. If user logs in:
    - Validate credentials
    - If matches → load dashboard

5. Dashboard operations:
    - Add record → encrypt amount → save to file
    - View record → decrypt amount → display
    - Edit/delete → rewrite records in file
    - Monthly report → total sum/ expense by month
    - Alert → compare total expense with given limit

6. Continue until user exits

Input & Output

Input:

- Username
- Password

- Date
- Amount
- Transaction type (income/expense)

Output:

- Login success or failure
- Transaction recorded successfully
- Perform CRUD operations on transaction

4. Implementation

Language used: C

Compiler/IDE: GCC/ Visual Studio (VS) Code

Key Features

- ➢ Password hashing
- ➢ Basic data encryption
- ➢ CRUD operations on transactions (Add/View/Edit/Delete)
- ➢ Monthly report
- ➢ Expense alert
- ➢ CLI interface with clean menu navigation

Code Snippet

1. Password Hash Function

```c
unsigned long hashString(const char *str) {
    unsigned long hash = 5381;
    int c;
    while (*str != '\0') {
        c= *str;
        hash = ((hash << 5) + hash) + c;  // hash * 33 + 97
        str++;
    }
    return hash;
}
```

2. Structure for Budget Entry

```c
struct budgetEntry{
    char date[30]; // DD-MM-YYYY e.g: 27-11-2025
    float amount;
    char type[30];
};
```

3. Encryption algorithm

```
float encryptAmount(float amount) {
    return (amount *4) + 147.59;
}
```

Sample Output

For hashing algorithm

```
Enter a password: pass123

Hash: 399470012
```

```
Enter a password: pass12

Hash: 2615115593
```

5. Testing & Results

Test No	Input	Expected Output	Actual Output	Status
1	Login (correct credentials)	Login successful	Successful	Pass
2	Login (wrong password)	Invalid Username or Password	Invalid Username or Password	Pass
3	Add transaction (valid data)	Record saved	Record saved	Pass
5	View transaction history	Display user records	Displayed correctly	Pass

The system performed correctly across repeated tests. User files were created correctly, and credentials were validated without errors.

6. Conclusion, Limitations & References

Conclusion

This project demonstrates how fundamental concepts of programming can be used to create a secure CLI Budget Tracker.

The system ensures data privacy using separate user files, hashed passwords, and encrypted amounts, making it more secure than typical beginner projects.

Limitations

- Password hashing is simple and not cryptographically secure.
- Records encryption is reversible (not strong encryption)
- No sorting or advanced analytics

Future Enhancements

- Use stronger encryption/ hashing (using external libraries).
- Add backup/restore system
- Convert CLI into GUI or web application.
- Add charts/graphs for reports

References

- Money Tracker-Expense & Budget App (By Horoscope365)