Project Title: **EXPENSE TRACKER**

GROUP MEMBERS:

- Syed Muhammad Areeb (CT-25026)
- Muhammad Mahad Amir (CT-25027)
- Faiq Ahmed (CT-25024)

SEMESTER:

- FALL 2025

COURSE CODE AND NAME:

- CT-175 (Programming Fundamentals)

Project Description:

The **Expense Tracker** is a C-based console application designed to help users efficiently manage their income and expenses. It supports **multi-user accounts**, allowing each user to securely store and track their financial data independently.

The system allows users to add income, record categorized expenses, view spending summaries, generate reports, and maintain CSV-based logs for persistent storage. The program ensures that users do not exceed their income, provides category-wise breakdowns, and highlights the top 3 spending items.

This project demonstrates core programming concepts including:

- ✓ Structures
- ✓ File handling (CSV + TXT storage)
- ✓ Conditional logic
- ✓ Functions for modular design
- ✓ Dynamic data loading and saving
- ✓ Basic data analysis

Overall, the Expense Tracker offers a simple but practical budgeting tool for financial management.

Project Features / Functionalities:

1. User Account System:

* Multiple users supported (up to 10).
* Secure sign-in and sign-up.
* Each user's expenses stored separately in their own CSV file.
* Option to **delete account,** removing all personal records.

2. Income Management:

* Users can add or update their total monthly income.
* Income is stored in **all_incomes.csv.**
* Program ensures expenses never exceed income.

3. Expense Management:

Add expenses with:

* Item name
* Category
* Amount
* Date (manual entry or automatic "Today")
* Delete the most recent expense.

4. Expense Summary & Analysis:

* Total income, total expense, and remaining balance.
* Top 3 highest-expense items.
* Category-wise breakdown with percentages.

5. File Handling:

* Users file: `users.txt`
* Income file: `all_incomes.csv`
* Per-user data file: `<username>.csv`
* Automatic loading and saving.

6. Structured Menu System:

* Add Income
* Add Expense
* Delete Last Expense
* View Totals
* Generate Summary
* Delete Account
* Logout

DATA STORAGE MECHANISM:

We used structures and arrays to store the data of users, income, and expenses for each user.

STRUCTURES:

- **STRUCT User:** Stores user credentials. It's members are:
  - username[MAX_USERNAME_LEN]: character array for username.

- password[MAX_PASSWORD_LEN]: character array for password.

ARRAYS / VARIABLES:

User Management:

* **User users[MAX_USERS]**: array of User structures to store all users (username and passwrod) in memory.
* **userCount**: integer to track the number of registered users.
* **currentUser[MAX_USERNAME_LEN]**: character array storing the username of the logged-in user.

Expense Management:

* **expenseItem[MAX_ENTRIES][50]:** stores the names of all expenses.
* **expenseCategory[MAX_ENTRIES][50]**: stores the category of each expense.
* **expenses[MAX_ENTRIES**]: array stores amount of each expense.
* e**xpenseDate[MAX_ENTRIES][11]**: stores the date of each expense in DD/MM/YYYY format.
* **expenseCount**: integer to track the number of expenses added.

Temporary Arrays:

* **sortedAmt[MAX_ENTRIES]:** used to sort expenses for generating top items.

* **sortedItem[MAX_ENTRIES][50]:** stores sorted expense items corresponding to sortedAmt.
* **cats[MAX_ENTRIES][50]:** stores unique categories for category breakdown.
* **catTotal[MAX_ENTRIES]:** array storing total expense per category

Conclusion:

The Expense Tracker project successfully demonstrates real-world use of C programming and file handling. It efficiently manages multi-user data, validates user inputs, prevents overspending, and produces analytical reports. This project strengthened our understanding of:
The system is practical, scalable, and easily extendable for future improvements.

Future Improvements:

* Add budgeting goals
* Keeping predefined categories for items (e.g. food, transport, clothes)
* Export reports as PDF
* Add charts/graphs
* Make GUI version using GTK or web interface
* Password encryption
* Sorting/filtering expense history