# **In English:**

I'm gonna walk you through this Expense Manager code like we're just two pals chatting about a cool project. No stiff explanations here, just chill breakdowns! Let's dive in~

**BIG PICTURE FIRST!** This is a **C++ Expense Tracker app** that lets users:

* Create accounts (with passwords!)
* Start/load sessions based on their role (Student, Gamer, etc.)
* Track expenses by category
* View summaries and budgets

It stores everything in files like a little personal finance assistant. Now, piece by piece!

**STRUCTS & VARIABLES:**

struct Expense {

string date;

string category;

float amount;

};

Simple structure to hold each expense!

* Date
* Category (like "Snacks")
* Amount

vector<Expense> expenses;

float setBudget = 0;

string userType;

string sessionName;

vector<string> categories;

string loggedInUser = "";

These are global variables used to keep track of the current user's data and session.

defaultCategories: A cool map that suggests default expense categories based on user type (e.g., Gamer, Student). Super thoughtful!

**FILE & FOLDER HELPERS:**

* fileExists() - Checks if a file exists.
* folderExists() - Checks if a folder exists.
* createFolderIfNotExist() - Makes a folder if it ain't there.
* savePassword()/verifyPassword() - For account login!
* getUserPath() - Returns the path to current user's folder.

**LOGIN MENU:**

loginMenu() handles:

1. Login
2. Account creation
3. Refresh screen

It asks for username & password and sets loggedInUser. Uses the file system to store and retrieve data. Basic but solid!

**SESSION STUFF:**

**Saving/Loading:**

* saveSession() saves user role, budget, and categories into a \_session.txt
* loadSession() loads that session
* saveExpenses()/loadExpenses() handle expenses via a \_expenses.txt file

**listSessions() lists all saved sessions for the user.**

**MAIN FEATURES:**

**createNewSession()**

* Asks for session name
* Role type (e.g., Gamer)
* Budget
* Picks default categories or lets user add custom ones
* Clears previous expenses and saves it all

**loadExistingSession()**

* Lists available sessions and loads the one you pick

**CORE ACTIONS:**

**addExpense()**

* Asks date, category, amount
* Adds to vector
* Saves to file

**addNewCategory()**

* Adds a new category to the list (no dupes!)

**showExpenses()**

* Neatly prints all expenses with numbering
* You can edit or delete an expense too! 😎

**budgetSummary()**

* Shows how much you’ve spent vs. your budget
* Warns you if you’ve overspent 💸

**categorySummary()**

* Shows how much you’ve spent in each category (very handy)

**MAIN LOOP (in main()):**

After logging in:

1. You get options to create/load sessions
2. Once in a session, you can:
   * Add categories/expenses
   * View all expenses
   * View budget summary
   * View category-wise breakdown
   * Refresh or exit

It's like a personal finance dashboard! 🧾

**FINAL THOUGHTS:**

This project is super cool and actually really complete:

* Good use of file handling
* Keeps data per user (using folders and files)
* Supports sessions so different budgets/categories can be made
* Very beginner-friendly UI
* Easy to scale too!

If you're thinking of improving it in the future, you could:

* Add date validations
* Save expenses in JSON/CSV for compatibility
* Make a UI with a framework
* Encrypt passwords (currently plaintext)

# **In Roman Urdu:**

Heyyy~ buckle up, dost! Hum dono milke chill vibes ke saath is Expense Manager code ka full tour karte hain — bina kisi boring explanation ke! Just two pals talking tech 🤖 Let's dive in~

**BIG PICTURE!** Yeh ek **C++ Expense Tracker app** hai jo users ko allow karta hai:

* Account banane ke liye (password ke saath!)
* Role-based sessions start/load karne ke liye (Student, Gamer, etc.)
* Har expense ko category ke through track karne ke liye
* Budget aur summaries view karne ke liye

Saara data files mein save hota hai jaise ek chhota sa financial assistant ho — smart & loyal! 📅

**STRUCTS & VARIABLES:**

struct Expense {

string date;

string category;

float amount;

};

Yeh simple sa struct har kharcha ko track karta hai:

* Date
* Category (jaise "Snacks")
* Amount

vector<Expense> expenses;

float setBudget = 0;

string userType;

string sessionName;

vector<string> categories;

string loggedInUser = "";

Yeh global variables har session/user ki info yaad rakhte hain.

Aur sabse cool part? defaultCategories map har role ke liye relevant categories suggest karta hai! ✨

**FILE & FOLDER HELPERS:**

* fileExists() - Check karta hai file exist karti hai ya nahi
* folderExists() - Folder exist karta hai ya nahi, yeh dekhta hai
* createFolderIfNotExist() - Agar folder nahi hai toh bana deta hai!
* savePassword() / verifyPassword() - Account login ke liye
* getUserPath() - Logged in user ke folder ka path deta hai

**LOGIN MENU:**

loginMenu() handle karta hai:

1. Existing account login
2. Naya account creation
3. Screen refresh

Username aur password input hota hai, and loggedInUser set hota hai. Data files ke through store aur verify hota hai — simple, solid, secure (ish) ✨

**SESSION STUFF:**

**Save/Load Functions:**

* saveSession() user ka role, budget aur categories ko \_session.txt mein save karta hai
* loadSession() wohi session load karta hai
* saveExpenses() / loadExpenses() expenses ko \_expenses.txt file mein handle karte hain

**listSessions()**

Saare saved sessions ko dikhata hai — bohot convenient! 📃

**MAIN FEATURES:**

**createNewSession()**

* Session ka naam poochta hai
* Role select karne deta hai (jaise Gamer, Student)
* Budget input hota hai
* Default ya custom categories choose karne ka option
* Expenses clear karke sab save karta hai

**loadExistingSession()**

* Pehle se saved sessions dikhaata hai aur select karne deta hai

**CORE ACTIONS:**

**addExpense()**

* Date, category, aur amount input karta hai
* Vector mein add karta hai
* File mein save bhi karta hai

**addNewCategory()**

* Nayi category add karta hai (duplicates nahi chalte!)

**showExpenses()**

* Saare expenses numbered list ke form mein dikhata hai
* Edit ya delete bhi kar sakte ho — power user vibes 🧱

**budgetSummary()**

* Kitna budget hai vs. kitna spend kiya, yeh dikhata hai
* Overspending alert bhi deta hai! 😱

**categorySummary()**

* Har category mein kitna kharcha hua, yeh batata hai

**MAIN LOOP (in main()):**

Login ke baad:

1. Session create/load ka option aata hai
2. Phir aap:
   * Category/expense add kar sakte ho
   * Saare expenses dekh sakte ho
   * Budget summary check kar sakte ho
   * Category-wise spending breakdown dekh sakte ho
   * Refresh ya exit kar sakte ho

Personal finance dashboard ka maza! 🧰

**FINAL THOUGHTS:**

Project is:

* Strong file handling use karta hai
* User-based data store karta hai
* Session-based budgeting allow karta hai
* UI beginner-friendly hai
* Scale karna easy hai

**FUTURE UPGRADES IDEA:**

* Date format validation
* Expense data ko JSON/CSV mein save karna
* Visual UI using GUI framework (jaise Qt or SFML)
* Password encryption (abhi plaintext hai bro 😅)

Tayyarr ho next level ke liye? Yeh toh bas shuruaat hai! Let's gooo Abdullah! 🌟🚀