

Smart Expense Tracker

A JavaFX desktop application to manage personal finances

Ben Gregory John
Jerome Philip John
Md Aquib Raza

Project Overview

- **Objective:** Develop a JavaFX desktop application to manage personal finances, including income, expenses, and budgets.
- **Scope:** User authentication, transaction tracking, budget setting, real-time financial summaries, and alerts.
- **Technologies:** Java 21, JavaFX, MySQL, Maven, JDBC.

Welcome, BenGJ

Add Income

Add Expense

Add Budget

View Profile

Logout

Total Income: ₹12750.00

Total Expense: ₹2450.00

Current Balance: ₹10300.00

Recent Activity

Expense: ₹1000.00 (Transport) on 2025-07-14

Expense: ₹1100.00 (Shopping) on 2025-07-14

Income: ₹2750.00 (Freelance) on 2025-07-14

Budget exceeded for Transport (WEEKLY): ₹1100.00/₹1000.00

OOPS Concepts used in Project

Encapsulation

Private fields in classes with public getters/setters ensure data protection and access.

Inheritance

BaseDAO provide database methods inherited by User, Income, Expense, Budget classes.

Polymorphism

Interfaces allow polymorphic behavior in services enabling flexible transaction handling.

Abstraction

Abstracted logic in service and dao packages separates rules from UI and data access.

Smart Expense Tracker

BenGJ

●●●●●●●|

Login

Register

Login and Authentication

- Secure Login: Validates username/password using PasswordUtil.
- Password Hashing: Uses SHA-256 for secure storage in MySQL.
- Password Validator: Ensures strong passwords.
- Session Management: SessionManager singleton tracks logged user.

Income Management

- Form Input: Users enter amount, source, date, and description via AddIncomeController.
- Validation: Ensures positive amount and valid Income using enums (SALARAY, FREELANCE, GIFTS, OTHERS)
- Database Storage: IncomeDAO uses JDBC prepared statements to store income in MySQL.
- Real-Time Update: TableView refreshes to display new expense records instantly.
- Logging: CustomLogger to record income actions (e.g., “Added income: ₹10000, SALARY”)

Manage your Income

Amount:

Source:

Description:

ID	Amount	Source	Date	Description	Actions	
32	10000.0	Salary	2025-07-14	Salary	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
33	2500.0	Gifts	2025-07-14	Gift	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

[Back to Dashboard](#)

Income added successfully

Expense Management

- Form Input: Users enter amount, category, date, and description via AddExpenseController.
- Validation: Ensures positive amount and valid Category using enums (FOOD, TRANSPORT, ENTERTAINMENT)
- Database Storage: ExpenseDAO uses JDBC prepared statements to store expense in MySQL.
- Real-Time Update: TableView refreshes to display new expense records instantly.
- Logging: CustomLogger to record expense actions (e.g., “Added expense: ₹5000, FOOD”)

Manage Expenses

Amount:

Category:

Description:

ID	Amount	Category	Date	Description	Actions	
22	100.0	Transport	2025-07-14	Auto	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
23	250.0	Food	2025-07-14	Dinner	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
24	1100.0	Shopping	2025-07-14	Amazon	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
25	1000.0	Transport	2025-07-14		<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

Manage Budgets

Amount:

Category:

Period (e.g., Monthly):

ID	Amount	Category	Period	Actions	
30	1000.0	Transport	WEEKLY	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

[Back to Dashboard](#)

Budget added successfully

Budget Management

- Budget CRUD: Set, update, delete budgets via `AddBudgetController`.
- BudgetPeriod: WEEKLY, MONTHLY enums for flexible budgeting.
- Real-Time Alerts: `BudgetService` checks for overspending (e.g., “Budget exceeded for Food: ₹8000/₹5000”).
- Multithreading: Uses `Task` for non-blocking alert checks.
- UI Feedback: Displays alerts on dashboard via `errorLabel`.

Data Flow Diagram

