Database Assignment

CA Value: **5%**

Completion & Upload Date: **Sunday 16th February 9pm**

You are required to develop an application in Java that helps you keep track of your income and spending. This should be connected to a database with a table to store each expense, and a table to store income. Your solution must make use of DAOs and DTOs as shown in class.

An expense record in the database should contain, at minimum, the following:

* expenseID (integer, auto-increment)
* title (String)
* category (String)
* amount (double)
* dateIncurred (date)

E.g.: 1, weekly shop, groceries, €47.50, 7th January 2025

2, gym membership, fitness, €30, 9th January 2025

An income record should contain, at minimum, the following:

* incomeID (integer, auto-increment)
* title (String)
* amount (double)
* dateEarned (date)

E.g. 1, babysitting, €60, 5th January 2025

2, Bar work, €100, 7th January 2025

Your java program should allow you to do the following:

1. List all expenses incurred and calculate the total spend
2. Add a new expense
3. Delete an expense (by id)
4. List all income earned and calculate total income
5. Add a new income
6. Delete an income (by id)
7. List all income and expenses for a particular month and display the total income, expenditure, and how much money they should have left over.

You **must** use GitHub to show incremental progress on your assignment. Either add your lecturer as a collaborator or make your project public and share the link with them. On completion each student should be able to explain the code, design philosophy, and patterns used. Everyone **must** demo their project in person to the lecturer.

In your project read-me you should include links to any references or third-party tutorials you followed while creating your solution. Any sources (e.g. stack-overflow) you used when writing your code should be included in comments in the classes where they are used.

## Submission Checklist

Make sure to submit all the following to Moodle or your submission will not be marked:

|  |  |
| --- | --- |
| **GitHub Link**  Regular commits, relevant commit messages.  Must be either public, or have invited your lecturer as a collaborator |  |
| **Coversheet**  Signed by student – assignments without a coversheet will not be marked. |  |
| **Zipped project**  Include everything needed to run the project, including any SQL required to create and seed your database. |  |

## Academic Integrity

The assignment must be entirely the work of each student – in your own words. Students are not permitted to share any pseudocode or source code from their solution with any other student in the class. Students may not distribute the source code of their solution to any other student in any format (i.e., electronic, verbal, or hardcopy transmission). Any suspected plagiarism will be investigated, pursued, and reported to the Plagiarism Committee.

Generative artificial intelligence (AI) tools cannot be used in this assessment task. In this assessment, **you must not use** generative artificial intelligence (AI) (ChatGPT, ChatSonic, Bing Chat, Lex, DALL-E 2, or other tools) to generate any materials or content in relation to the assessment task.

The DkIT Academic Integrity Policy and Procedures, <https://www.dkit.ie/about-dkit/policies-and-guidelines/academic-policies.html>) states the following:

“Using generative artificial intelligence tools (e.g. ChatGPT) in an assessment unless explicitly permitted to do so and with proper acknowledgement, is a form of plagiarism”.

## Late Submissions

The institute’s standard policy on marking of late submissions will be applied:

* Submissions received up to 1 week late will have a 20% grade reduction applied.
* Submissions received up to 2 weeks late will have the above penalty applied and will be capped at 40%.
* Submissions received more than 2 weeks late cannot be graded.