**A black and white logo

AI-generated content may be incorrect.CAROLINA UNIVERSITY**

**School of Computing, Applied Science, & Engineering**

**Computer Science Department**

**Name of Project: ExpenseTracker**

**Prepared By (Name): Mahesh Karki**

**Student Id: 1063667**

**Course:**[**CS 222 50 - C# Programming**](https://my.carolinau.edu/ICS/Academics/CS/CS___222/2024_SU-CS___222-50/)

**Submitted To (Instructor): Professor James Boston Belton**

**Date of Submission: 06/22/2025**

**GitHub Link: *https://github.com/karkim7/Week7-ExpenseTracker***

**Table of Contents:**

1. **Abstract**
2. **Acknowledgments**
3. **Introduction**
4. **Objectives**
5. **Limitations**
6. **ExpenseTracker**
7. **Summary**
8. **Appendix**
9. **References**
10. **Screenshot of Program Execution Successfully**

**1. Abstract**

This project presents a simple console-based **Expense Tracker** application developed in C#. The primary goal of the application is to help users manage and monitor their daily expenses by allowing them to input, view, and calculate their total spending. Users can add expenses with descriptions and amounts, view a list of all recorded expenses, and see the total amount spent. The program uses object-oriented programming concepts, such as classes and lists, to organize and store the data efficiently. It also includes user input validation and a clear menu-based interface. This project simulates a real-world personal finance tool and demonstrates practical skills in C# programming.

**2. Acknowledgments**

I would like to express my sincere gratitude to my instructor for providing guidance and support throughout this project. The opportunity to apply my C# programming knowledge in a practical, real-world scenario has greatly enhanced my understanding of core programming concepts such as classes, lists, loops, and user input handling.

I also appreciate the resources available in our course materials and online tutorials that helped me design and complete this Expense Tracker application successfully. Lastly, I would like to thank my peers and family for their encouragement during the development of this project.

**3. Introduction**

Managing personal expenses is an essential part of daily life. This C# Expense Tracker application is designed to provide a simple and effective way for users to record and monitor their spending. The program allows users to add expenses with descriptions, view a list of all recorded expenses, and calculate the total amount spent.

Built using object-oriented programming principles, the application uses a class to represent each expense and a list to store multiple entries. The program features a user-friendly console menu that makes it easy to navigate and use. This project not only demonstrates practical C# programming skills but also shows how technology can be used to solve real-world problems.

**4. Objectives**

The main objective of this project is to develop a simple and user-friendly console-based application using the C# programming language. The application is designed to help users record and manage their daily expenses by allowing them to add expenses with descriptions and amounts, view all recorded expenses, and calculate the total amount spent. Through this project, object-oriented programming concepts such as classes and objects are applied to represent and organize expense data effectively. Additionally, the project aims to demonstrate the use of data structures like lists for storing multiple records, and to strengthen programming skills in areas such as user input handling, loops, conditionals, and basic validation. Overall, this project serves as a practical implementation of C# fundamentals in solving a real-world problem.

**5. Limitations**

Although the Expense Tracker application serves its basic purpose, it has some limitations. It is a console-based program and does not offer a graphical user interface (GUI), which may limit ease of use for non-technical users. The application does not store data permanently, meaning all expenses are lost once the program is closed, as there is no database or file storage implemented. Additionally, the program lacks features like editing or deleting specific expenses, categorizing expenses, or generating detailed reports. These limitations present opportunities for future improvements and expansion of the project.

**6. ExpenseTracker**

**ExpenseTracker** is a simple desktop application developed in C# that helps users manage their daily expenses efficiently. The main goal of the application is to provide users with an easy way to record, categorize, and analyze their spending habits to improve financial awareness and budgeting.

Key features of ExpenseTracker include:

* **Add and Edit Expenses:** Users can input details such as amount, date, category (e.g., food, transport, bills), and description for each expense.
* **Expense Categories:** Predefined categories help organize expenses, and users can add custom categories.
* **View Expense History:** A clear list or table shows all recorded expenses with sorting and filtering options by date or category.
* **Summary and Reports:** Basic summary views display total expenses over a period and category-wise spending, helping users identify where most of their money goes.
* **User-friendly Interface:** The app uses Windows Forms for a simple graphical interface that is easy to navigate even for beginners.
* **Data Persistence:** Expenses are saved locally (e.g., in a file or database) so that data is retained between sessions.

ExpenseTracker serves as a foundational project to practice object-oriented programming concepts, user interface design, and basic file/database operations in C#.

**7. Conclusion**

The Expense Tracker project successfully demonstrates the use of C# programming to solve a real-world problem in a simple and practical way. It allows users to record expenses, view their spending history, and calculate the total amount spent through a user-friendly console interface. The project applies fundamental programming concepts such as object-oriented design, loops, lists, and input validation. Although the application has limitations, it provides a strong foundation for future enhancements like data storage, editing features, and graphical interfaces. Overall, this project has helped strengthen core C# skills and provided valuable experience in building functional console applications.

**8. Appendices**

Appendix A: Sample Input and Output

Sample Run:

=== Welcome to Simple Expense Tracker ===

Choose an option:

1. Add Expense

2. View Expenses

3. View Total

4. Exit

Your choice: 1

Enter description: Grocery

Enter amount: $25

Expense added!

Choose an option:

1. Add Expense

2. View Expenses

3. View Total

4. Exit

Your choice: 2

Your Expenses:

- Grocery: $25.00

Choose an option:

1. Add Expense

2. View Expenses

3. View Total

4. Exit

Your choice: 3

Total Spent: $25.00

**Appendix B: Technologies Used**

* Programming Language: C#
* IDE: Visual Studio / Visual Studio Code
* .NET Runtime: .NET 6.0 or later

**Appendix C: File List**

* ExpenseTracker.cs – Main program source code.
* ExpenseTrackerDoc.txt – Project documentation.
* README.md (optional) – GitHub project summary.

**9. References**

1. Microsoft Docs – [C# Programming Guide](https://learn.microsoft.com/en-us/dotnet/csharp/)
2. W3Schools – C# Tutorial
3. GeeksforGeeks – C# Basics
4. Stack Overflow – Community solutions for C# coding issues
5. .NET Official Site – <https://dotnet.microsoft.com>
6. C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development 4th edition by Mark J. Price.

**10. Screenshot of Program Execution Successfully**

**A screenshot of a computer

AI-generated content may be incorrect.**

**-The End-**