

Add Expense

Expense Tracker

Personal Expense Tracker

ExpenseID	Description	Category	Amount	Date
1	Bought an apple	Others	15.00	1/15/2025
2	Bought a car	Transportation	50000.00	1/15/2025
3	Went to a cine...	Entertainment	170.00	1/6/2025
5	Electricity	Utilities	17000.00	1/15/2025
6	Toys	Others	3900.00	1/15/2025
11	aaaa	Food	13.00	1/16/2025
*				

Add Expense

Update Expense

Delete Expense

Search Expense

View Summary

Add / Update Expense

Add / Update Expense

Description:

Category:

Amount:

Date:

Save

Expense Tracker

Personal Expense Tracker

ExpenseID	Description	Category	Amount	Date
1	Bought an apple	Others	15.00	1/15/2025
2	Bought a car	Transportation	50000.00	1/15/2025
3	Went to a cine...	Entertainment	170.00	1/6/2025
5	Electricity	Utilities	17000.00	1/15/2025
6	Toys	Others	3900.00	1/15/2025
11	aaaa	Food	13.00	1/16/2025
13	Banana	Food	14.00	1/13/2025
*				

Add Expense

Update Expense

Delete Expense

Search Expense

View Summary

Update Expense

Expense Tracker

Personal Expense Tracker

	ExpenseID	Description	Category	Amount	Date
	1	Bought an apple	Others	15.00	1/15/2025
	2	Bought a car	Transportation	50000.00	1/15/2025
	3	Went to a cine...	Entertainment	170.00	1/6/2025
	5	Electricity	Utilities	17000.00	1/15/2025
	6	Toys	Others	3900.00	1/15/2025
▶	11	aaaa	Food	13.00	1/16/2025
	13	Banana	Food	14.00	1/13/2025
*					

Add Expense

Update Expense

Delete Expense

Search Expense

View Summary

Add / Update Expense

Add / Update Expense

Description: Pear

Category: Food

Amount: 23.00

Date: Tuesday, January 14, 2025

Save

Expense Tracker

Personal Expense Tracker

	ExpenseID	Description	Category	Amount	Date
	1	Bought an apple	Others	15.00	1/15/2025
	2	Bought a car	Transportation	50000.00	1/15/2025
	3	Went to a cine...	Entertainment	170.00	1/6/2025
	5	Electricity	Utilities	17000.00	1/15/2025
	6	Toys	Others	3900.00	1/15/2025
▶	11	Pear	Food	23.00	1/14/2025
	13	Banana	Food	14.00	1/13/2025
*					

Add Expense

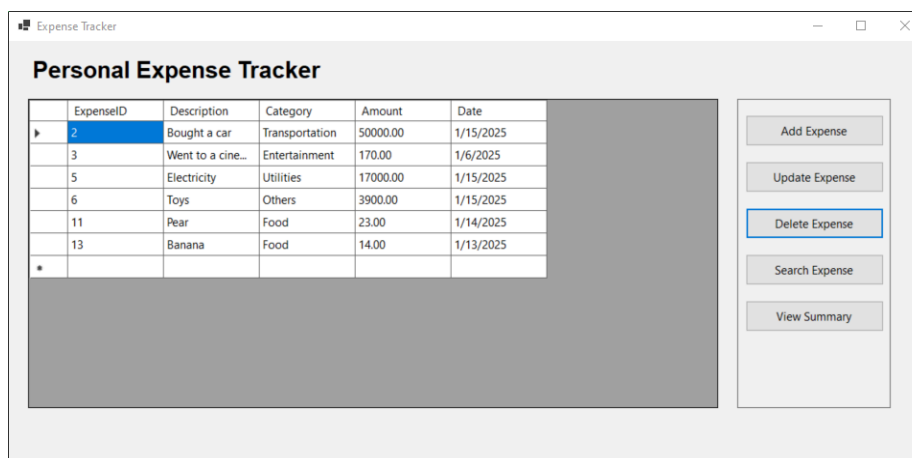
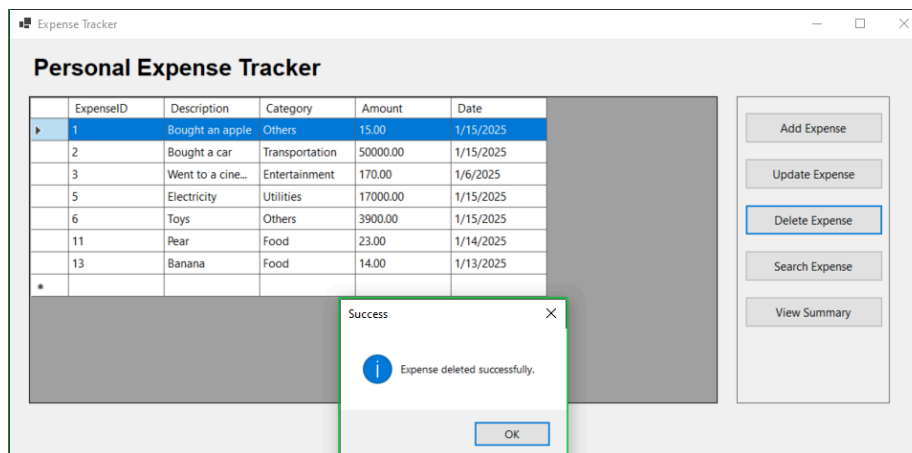
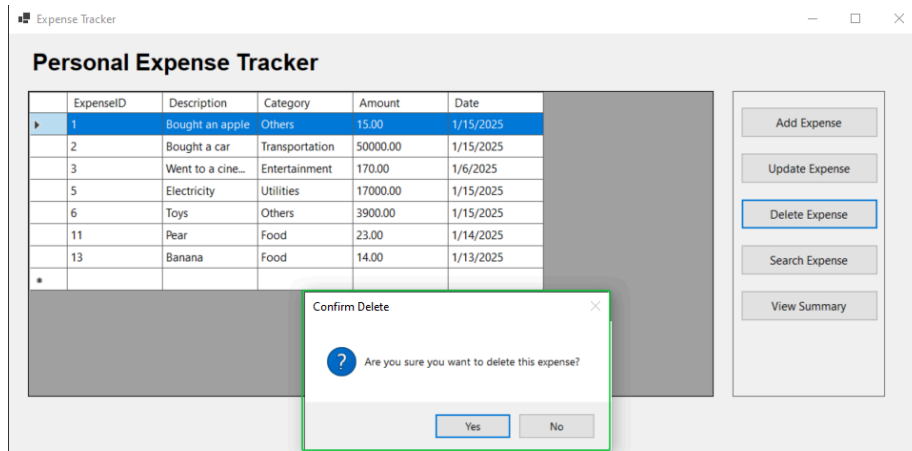
Update Expense

Delete Expense

Search Expense

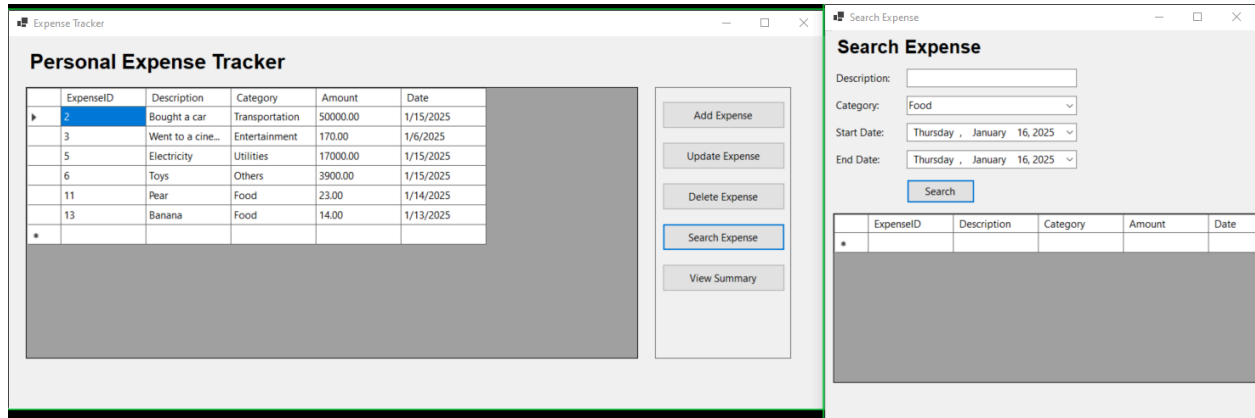
View Summary

Delete Expense

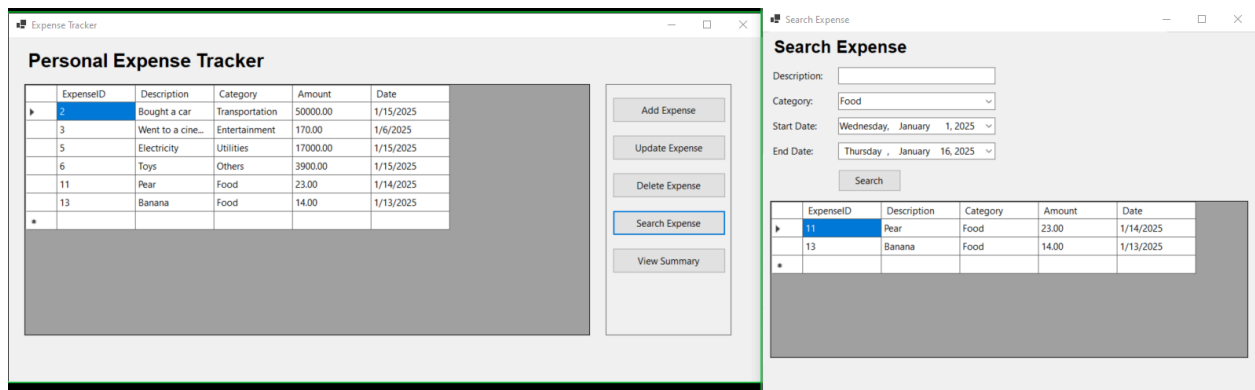


Search Expense

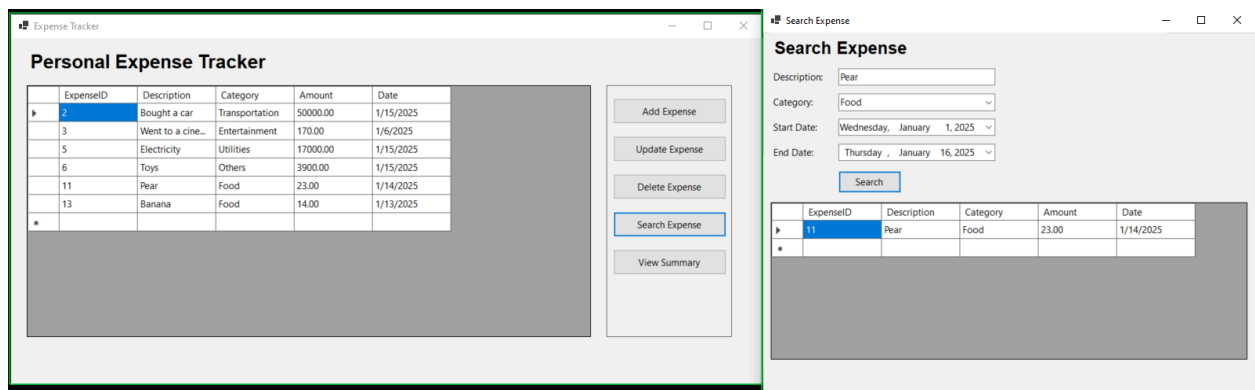
Nothing found



Searching by category



Searching by description



View Summary

Expense Summary

—

□

×

Expense Summary

	Category	TotalAmount
▶	Entertainment	170.00
	Food	37.00
	Others	3900.00
	Transportation	50000.00
	Utilities	17000.00
•		

Total Expenses: \$71,107.00

Personal Expense Tracker Application Documentation

Table of Contents

1. Overview
2. Architecture
3. Namespaces and Classes
 - 3.1 `PersonalExpenseTracker` Namespace
 - 3.1.1 `AddUpdateForm`
 - 3.1.2 `MainForm`
 - 3.1.3 `SearchForm`
 - 3.1.4 `SummaryForm`
 - 3.1.5 `Categories`
 - 3.1.6 `DatabaseHelper`
 - 3.1.7 `Expense`
 - 3.1.8 `Logger`
 - 3.1.9 `Program`
4. Database Setup
5. User Guide
6. Troubleshooting

1. Overview

The Personal Expense Tracker is a Windows Forms application designed to help users keep track of their expenses. It allows users to add, update, delete, search, and view summaries of their expenses. The application uses a local database to store expense data and provides a user-friendly interface for interacting with the data.

2. Architecture

The application follows a Model-View-Controller (MVC) inspired architecture:

- **Model**: Represents the data and business logic. This includes the `Expense` class and the `DatabaseHelper` class for database operations.
- **View**: The user interface components, including forms like `MainForm`, `AddUpdateForm`, `SearchForm`, and `SummaryForm`.
- **Controller**: Handles the logic for interacting between the Model and the View. This is implemented within the event handlers of the forms.

3. Namespaces and Classes

3.1 `PersonalExpenseTracker` Namespace

3.1.1 `AddUpdateForm`

- **Purpose**: A form for adding new expenses or updating existing ones.
- **Key Methods**:
 - `PopulateCategories()`: Populates the category ComboBox with predefined categories.
 - `LoadExpenseData()`: Loads data for an expense to be updated.
 - `txtAmount_KeyPress()`: Ensures that only valid numeric input (digits, decimal point, and optional negative sign) is entered in the Amount field.
 - `btnSave_Click()`: Saves the expense data to the database.

3.1.2 `MainForm`

- **Purpose**: The main form of the application, displaying a list of expenses and providing options to add, update, delete, search, and view summaries.

- **Key Methods:**

- `LoadExpenses()`: Loads expense data from the database into a DataGridView.
- `btnAdd_Click()`: Opens the `AddUpdateForm` for adding a new expense.
- `btnUpdate_Click()`: Opens the `AddUpdateForm` for updating an existing expense.
- `btnDelete_Click()`: Deletes an expense from the database.
- `btnSearch_Click()`: Opens the `SearchForm` for searching expenses.
- `btnSummary_Click()`: Opens the `SummaryForm` for viewing expense summaries.

3.1.3 SearchForm

- **Purpose:** A form for searching expenses based on description, category, and date range.
- **Key Methods:**
 - `PopulateCategories()`: Populates the category ComboBox with predefined categories.
 - `btnSearch_Click()`: Retrieves and displays expenses that match the search criteria.

3.1.4 SummaryForm

- **Purpose:** A form for viewing a summary of expenses grouped by category.
- **Key Methods:**
 - `SummaryForm_Load()`: Loads and displays the expense summary data and the total expenses.

3.1.5 Categories

- **Purpose:** A static class that provides a list of predefined expense categories.
- **Key Methods:**
 - `GetCategories()`: Returns an array of category strings.

3.1.6 DatabaseHelper

- **Purpose:** A utility class for database operations, including adding, updating, deleting, and retrieving expenses.
- **Key Methods:**
 - `AddExpense()`: Adds a new expense to the database.
 - `UpdateExpense()`: Updates an existing expense in the database.
 - `DeleteExpense()`: Deletes an expense from the database.
 - `GetExpenses()`: Retrieves all expenses from the database.
 - `GetExpenseById()`: Retrieves a single expense by its ID.
 - `SearchExpenses()`: Retrieves expenses based on search criteria.
 - `GetExpenseSummary()`: Retrieves a summary of expenses grouped by category.

3.1.7 Expense

- **Purpose:** A class representing an expense entity with properties for ID, description, category, amount, and date.

3.1.8 Logger

- **Purpose:** A static class for logging errors to a text file.
- **Key Methods:**
 - `LogError()`: Logs an exception and an optional message to a file in the `Logs` directory.

3.1.9 Program

- **Purpose:** The entry point of the application.
- **Key Methods:**
 - `Main()`: Initializes and runs the application, starting with the `MainForm`.

4. Database Setup

The application uses a SQL Server LocalDB database named `ExpenseTrackerDB`. The database schema is defined in the `ExpenseDB.sql` script.

Steps to Set Up the Database:

1. Create the Database:

- Open SQL Server Management Studio (SSMS).
- Execute the `ExpenseDB.sql` script to create the `Expenses` and `Categories` tables and insert initial category data.

2. Update Connection String:

- If necessary, update the connection string in `DatabaseHelper.cs` to match your database setup.
-

5. User Guide

5.1 Main Form

- Add Expense:

- Click the **Add Expense** button.
- Fill in the description, category, amount, and date.
- Click **Save** to add the expense.

- Update Expense:

- Select an expense in the DataGridView.
- Click the **Update Expense** button.
- Make changes to the expense details.
- Click **Save** to update the expense.

- **Delete Expense:**
 - Select an expense in the DataGridView.
 - Click the **Delete Expense** button.
 - Confirm the deletion in the dialog box.
- **Search Expenses:**
 - Click the **Search Expense** button.
 - Enter search criteria (description, category, date range).
 - Click **Search** to view the results.
- **View Summary:**
 - Click the **View Summary** button.
 - View the summary of expenses grouped by category and the total expenses.

5.2 Add/Update Form

- **Fields:**
 - **Description:** A text box for the expense description.
 - **Category:** A ComboBox for selecting the expense category.
 - **Amount:** A text box for entering the expense amount (numeric input only).
 - **Date:** A DateTimePicker for selecting the expense date.
- **Save:**
 - Click **Save** to add or update the expense.

5.3 Search Form

- **Fields:**
 - **Description:** A text box for entering part of the expense description.
 - **Category:** A ComboBox for selecting the expense category.
 - **Start Date** and **End Date:** DateTimePickers for selecting the date range.

- **Search:**
- Click **Search** to retrieve expenses that match the criteria.

5.4 Summary Form

- **Display:**
 - Shows a DataGridView with expenses grouped by category.
 - Displays the total expenses at the bottom.
-

6. Troubleshooting

- **Database Connection Issues:**
 - Ensure that the SQL Server LocalDB is installed and running.
 - Verify that the connection string in `DatabaseHelper.cs` is correct.
 - Make sure that the database and tables are created successfully.
 - **Error Logging:**
 - Errors are logged to a file named `error_log.txt` in the `Logs` directory within the application's folder.
 - **User Permissions:**
 - Ensure that the application has the necessary permissions to read and write to the database and log files.
-

This documentation provides a comprehensive overview of the Personal Expense Tracker application, its architecture, key components, and how to use it effectively.