**Expense Tracker Report**

**1. Introduction**

This project is a desktop-based Expense Tracker built using Python and the Tkinter library. It enables users to record, manage, and visualize their daily expenditures with a simple and intuitive graphical interface. The application is developed and tested in **PyCharm IDE**, making it an ideal beginner-to-intermediate level project for learning file handling, GUI programming, and basic data visualization in Python.

**2. Objectives**

The primary objective of the project is to provide users with a tool that can:

* Log their expenses with key details.
* Display all recorded expenses in a tabular format.
* Delete any recorded expense.
* Generate visual reports (Pie and Bar charts) to analyze spending patterns.

**3. Technologies Used**

| **Component** | **Description** |
| --- | --- |
| **Python 3** | Programming language |
| **PyCharm** | Integrated Development Environment (IDE) |
| **Tkinter** | GUI development framework |
| **tkcalendar** | Date selection widget |
| **Matplotlib** | Graph plotting library |
| **CSV** | File format for local data storage |
| **OS/Datetime** | Utility modules for file and date operations |

**4. Key Features**

**✔ Add Expense**

Users can input:

* Date (via calendar picker)
* Category
* Amount
* Description

The data is saved into a CSV file (expenses.csv), with validations to ensure completeness and correctness.

**✔ Delete Expense**

Users can select a row from the displayed table and delete it after confirmation. This updates the CSV file accordingly.

**✔ View Expenses**

All recorded expenses are displayed in a Tkinter Treeview widget, making it easy to browse and verify entries.

**✔ Generate Reports**

Two types of visual reports are generated using **Matplotlib**:

* **Pie Chart**: Shows distribution of expenses by category.
* **Bar Chart**: Monthly trend of total expenses.

**✅ Adding Expense**

add\_expense()

* Validates fields.
* Converts amount to float.
* Appends a new row to the CSV file.

**✅ Deleting Expense**

delete\_expense()

* Requires a row selection.
* Removes the selected row from the file after confirmation.

**✅ Viewing Expenses**

view\_expenses()

* Reads data from the CSV file.
* Displays it using ttk.Treeview.

**✅ Generating Reports**

generate\_report()

* Aggregates expenses by category and month.
* Uses matplotlib to display:
  + Pie chart of category-wise spending.
  + Bar chart of monthly totals.

**5. GUI Design**

* Clean, user-friendly layout using **Tkinter**.
* Background: #f0f8ff (light blueish tone).
* Buttons: Colored for intuitive use (Add = Green, Delete = Red, Report = Blue).
* Responsive layout with dynamic updates.

**6. Conclusion**

This Expense Tracker is a complete, functional, and well-structured desktop application built using Python and Tkinter. It provides essential features for managing daily expenses and visualizing spending patterns. The use of CSV files for storage makes it lightweight, while the integration of charts adds significant analytical value. It's a perfect example of a real-world utility project that demonstrates core programming and GUI design skills in Python.