

Expense Tracker

A Desktop Expense Management Application Using Python, Tkinter and SQLite

Viva Questions & Answers

Prepared by: _____
Roll / Enrollment No.: _____
Course / Semester: _____
Department: _____
Institution: _____
Project Guide: _____
Academic Year: _____

1. What is the Expense Tracker?

It is a desktop application for recording and managing personal expenses and generating simple summaries and reports.

2. Which technologies are used?

Python, Tkinter, SQLite, and CSV file handling.

3. Why Python?

Python provides simple syntax, a large standard library, and built-in support for GUI and SQLite-based desktop applications.

4. What is Tkinter?

Tkinter is Python's standard GUI toolkit for creating windows, forms, buttons, tables, and other desktop interface components.

5. What is SQLite?

SQLite is a lightweight relational database engine that stores the database in a local file and does not require a separate database server.

6. Why use SQLite instead of MySQL?

The project is a small single-user desktop application, so SQLite is simpler because it is local, serverless, and included with Python.

7. What is CRUD?

CRUD stands for Create, Read, Update, and Delete. The project primarily demonstrates creating, reading, and deleting expense records.

8. How are expenses stored?

Expense records are stored in a local SQLite database file created by the application.

9. What is a category summary?

It groups expense records by category and calculates the total spending for each category.

10. What is a monthly report?

It summarizes expenses for a selected month and provides the total spending and report data.

11. Why is CSV export useful?

CSV files can be opened in spreadsheet applications and are convenient for sharing, further analysis, or archiving.

12. How does the chart work?

The application calculates category totals and draws a simple visual chart using Tkinter Canvas.

13. Does the project require internet access?

No. The core application runs locally and does not require an internet connection.

14. Does it require MySQL?

No. It uses SQLite, so a separate MySQL server is not required.

15. What happens when the database does not exist?

The application creates the SQLite database and required table automatically during initialization.

16. What is the advantage of local storage?

It keeps expense data on the user's device and makes the application simple to deploy for a small desktop project.

17. What are the limitations?

It is designed as a small local desktop application and does not provide multi-user cloud synchronization, authentication, or enterprise financial controls.

18. How could the project be improved?

Add editing, recurring expenses, budgets, authentication, cloud synchronization, richer charts, PDF reports, and automated tests.

19. What is data persistence?

Persistence means data remains stored after the application is closed and can be loaded again later.

20. What is the purpose of the requirements.txt file?

It documents external Python dependencies. In this project, the main application uses Python's standard library, so no third-party package is required.