2025

Expense Tracker

INTERNSHIP PROJECT REPORT

MALCOLM CEPHAS

Chapter 1: Introduction

This document delineates the unfolding of an Expense Tracker Application, which was developed as a part of a student's internship. The primary aim was to come up with and develop a desktop application that would take care of daily expenses; categorize the transactions made; produce monthly analytics and make the data easy to comprehend via charts.

Managing one's personal finances has turned into a crucial competence, and digital devices make the task easier by performing the tracking, categorization, and reporting automatically. This project is all about creating such a solution using technologies that are very popular and commonly used, such as Java, JavaFX, SQLite, and JFreeChart.

Chapter 2: Project Overview

2.1 Project Overview

The main objective of this project was to create a user-friendly tool that allows users to:

- Keep a daily record of their expenses
- Classify their expenses according to categories
- Access analytics of their spending for a month
- Use pie and bar charts for data visualization
- Report through data export
- Use SQLite for data persistence

2.2 Scope

The project includes the following components in its scope:

- An expense manager that is based on a GUI
- SQLite persistent storage
- Data filtering on a monthly basis
- Automatic creation of database tables

- Statistical charts generation
- Exporting expenses to CSV format

Chapter 3: Objective

The project was developed with the following objectives:

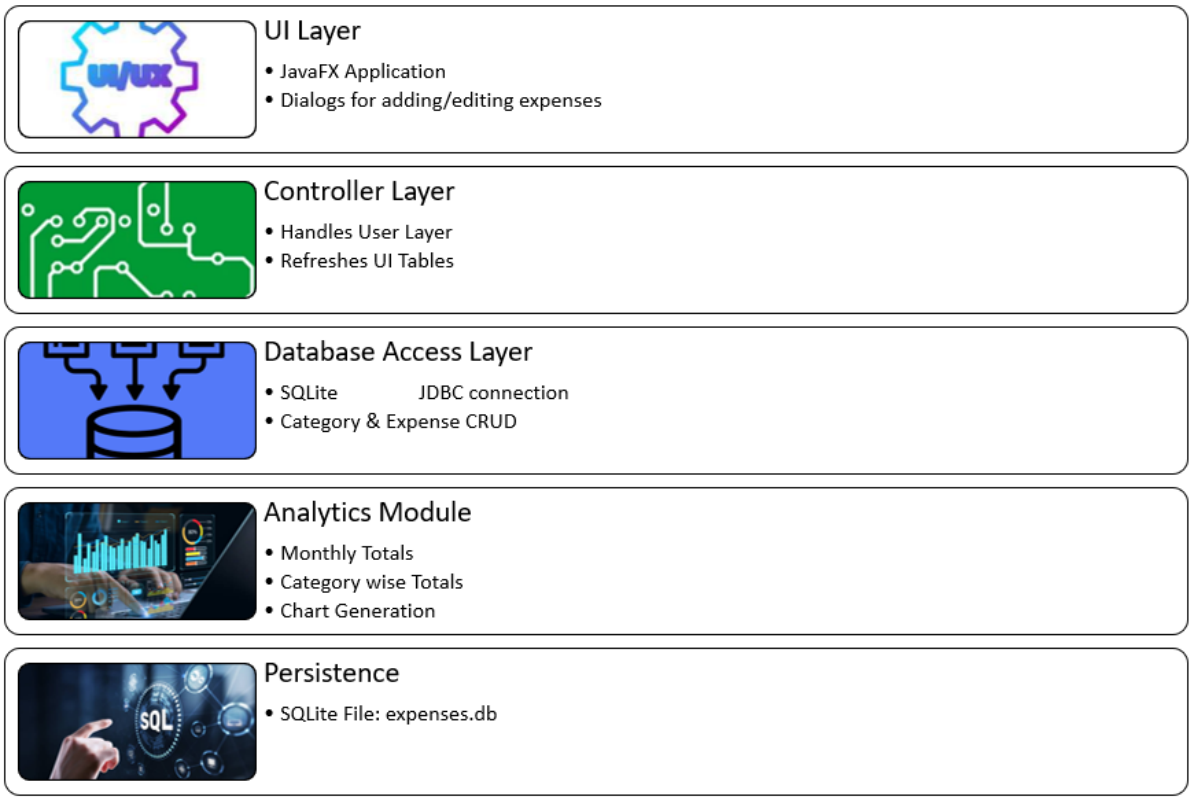
- Build a simple and interactive GUI using JavaFX.
- Implement CRUD (Create, Read, Update, Delete) functionality for expenses.
- Design a lightweight local database using SQLite.
- Provide monthly analytics with category-wise and day-wise totals.
- Visualize data using charts (pie chart and bar chart).
- Allow users to export expenses to CSV.
- Provide editable categories and allow automatic category creation.
- Package the application with an easy-to-run batch file.

Chapter 4: Technologies Used

Component	**Technology**
Programming Language	Java (JDK 17)
GUI Framework	JavaFX 21
Database	SQLite
Charting Library	JFreeChart
Build Tool	Maven
Packaging	Batch File using JavaFX module path

Chapter 5: System Architecture

5.1 Architecture Diagram



Chapter 6: Database Design

6.1 Tables

Categories

Column	Type	Description
Id	Integer **Primary Key**	Category ID
name	Text	Category Name

Expenses

Column	**Type**	**Description**
id	Integer **Primary Key**	Expense ID
amount	Real	Expense amount

date	Text	ISO date format
category_id	Integer	Foreign key referencing categories
note	Text	Optional comments

6.2 Features

- Tables automatically created at runtime
- Default categories inserted if empty
- New categories auto-created when the user types one

Chapter 7: Implementation Details

7.1 Major Modules

7.1.1 Main Application

Handles the JavaFX window, initialization, table loading, and linking UI actions.

7.1.2 ExpenseDAO

Interacts with the database:

- Insert expense
- Update expense
- Delete expense
- Query expenses for selected month
- Generate monthly totals
- Auto-create categories

7.1.3 SQLite Database

A lightweight file-based database stored locally as: expenses.db

7.1.4 UI Dialogs

Custom dialogs for:

- Adding expenses
- Editing expenses
- Displaying error messages

7.2 Chart Generation

Uses JFreeChart to generate:

- Pie chart (category-wise totals)
- Bar chart (daily totals)

Charts are integrated into JavaFX using SwingNode.

Chapter 8: Running the Application

The project includes a **run.bat** file so the user does not need to configure JavaFX manually.

Steps to Run

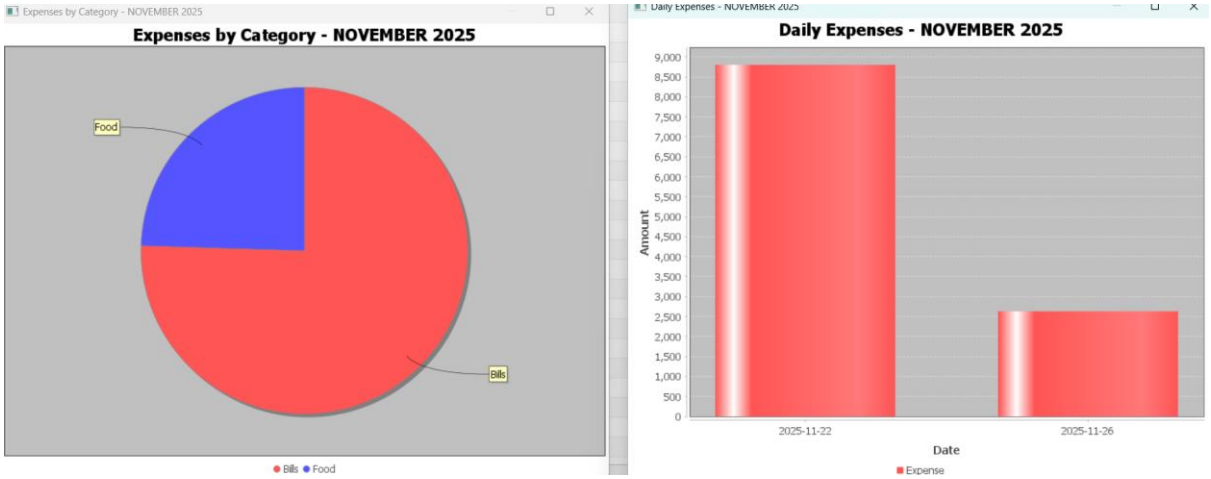
1. Install **Java 17+**
2. Download **JavaFX SDK**
3. Extract it to: C:\javafx-sdk-21.0.9\
4. Double-click the file: run.bat
5. The Expense Tracker window opens.

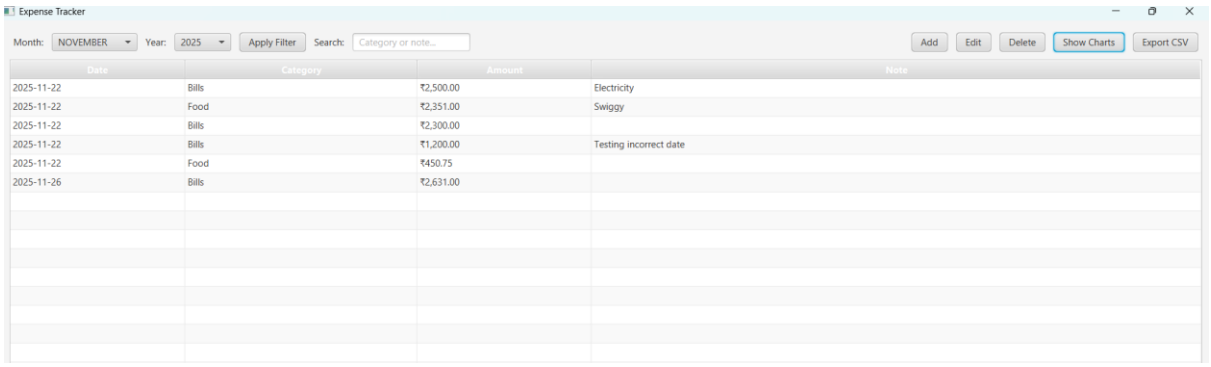
Chapter 9: Results

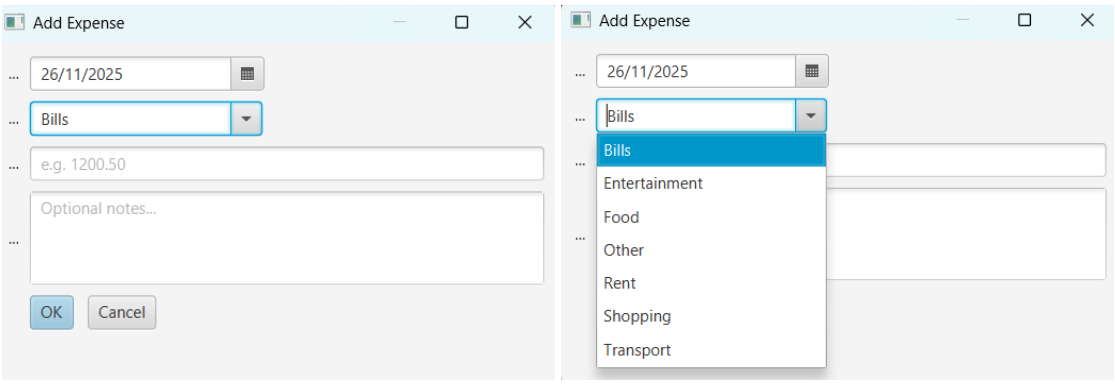
The application successfully demonstrates:

- Smooth GUI operation using JavaFX
- Automatic category creation

- Storage and retrieval from SQLite
- Clean table-based expense listing
- Exporting expenses to CSV
- Working pie chart and bar chart
- Accurate monthly filtering







Chapter 10: Advantages

- Lightweight and offline (no internet needed)
- Easy to use for beginners
- Zero-config database
- Editable categories
- Visual analytics included
- Cross-platform support (Windows, Linux, Mac with adjustments)

Chapter 11: Limitations

- No user login or authentication
- Desktop-only application
- No cloud backup
- Cannot sync across devices
- Manual setup of JavaFX required on other systems

Chapter 12: Future Enhancements

Planned improvements include:

- Adding login module (optional encryption)
- Cloud sync using Firebase or REST API
- Android version using Flutter or React Native
- Export charts as images
- Dark mode UI
- Budget alerts and notifications
- Sorting and filtering enhancements
- Dashboard with multiple charts

Chapter 13: Conclusion

The internship project has been an accomplishment, bringing A to Z the Expense Tracker Application that was equipped with the database, presented with chart visualization, and had a user-friendly graphical interface. Financial needs have been satisfied with the app by accomplishing its objectives of tracking, sorting, and analysis.

Partially the above-mentioned things I have gained as my practical experience:

designing with JavaFX UI

working with SQLite database

using Maven for managing dependencies

Java packaging and application deployment

problem-solving and debugging in real-life situations

The project not only sets up the user's demand but also provides an excellent base to build even more sophisticated finance-related applications on top of it.

Chapter 14: References

- JavaFX Documentation
- SQLite JDBC Driver
- JFreeChart
- Maven User Guide
- Oracle Java Documentation