Expense Tracker Report

1. Introduction

In an era of increasing financial complexity, managing personal expenses efficiently is crucial for financial well-being. An Expense Tracker serves as a digital tool to record, categorize, and monitor daily expenditures. This project aims to build an interactive Expense Tracker desktop application using JavaFX, providing users with a seamless and responsive interface to manage their financial data. JavaFX is chosen for its rich GUI capabilities and ease of integration with backend data sources, making it suitable for building cross-platform desktop applications.

2. Abstract

The goal of this project is to develop a desktop-based Expense Tracker Application using JavaFX, allowing users to log income and expenses, categorize transactions, and visualize financial trends through charts. The application stores data locally or in a lightweight database (like SQLite or MySQL), and provides users with features like date filtering, data summaries, and graphical insights. The JavaFX framework enables us to design a modern, visually appealing user interface with built-in components for layout, controls, and data visualization. The final product is a practical and user-friendly tool that encourages better financial habits and improved budgeting.

3. Tools Used

Tool/Technology	Purpose
Java	Core programming language
JavaFX	GUI development (FXML, Scene Builder)
Springboot	For backend / server side code for api connection
SQLite / MySQL	Lightweight database to store expense records
Scene Builder	For designing GUI without writing raw FXML
IntelliJ IDEA	IDE used for development
JDBC	Java Database Connectivity for linking JavaFX to database
JFreeChart / JavaFX Charts	For generating pie charts or bar graphs to visualize expenses

4. Steps Involved in Building the Project

Step 1: Requirements Gathering

Define the core features: add/edit/delete expenses, categorize expenses, show summaries, generate charts, filter by date/month.

Step 2: Designing the UI

Use JavaFX with FXML and Scene Builder to create a modular and clean layout.

UI includes:

Dashboard (Total balance, income, expenses)

Form to input transactions

Table to display transaction history

Graphs for data visualization

Step 3: Backend & Database Setup

Design a simple schema: transactions(id, title, amount, type, category, date)

Use MySQL for data storage.

Step 4: Implementing Core Functionalities

Add Transaction: Input fields with validation, save to database.

Display Transactions: Use TableView with observable lists to display data.

Filter/Search: Implement date pickers or combo boxes for filtering data.

Delete/Edit: Right-click context menu or button action for modifying entries.

Step 5: Data Visualization

Use JavaFX PieChart/BarChart to visualize expenses by category or date.

Update charts dynamically as the data changes.

Step 6: Testing & Debugging

Unit test core functions (database operations, UI components).

Validate input to prevent runtime crashes.

Ensure responsiveness and UI consistency.

Step 7: Packaging and Deployment

Use JavaFX packaging tools or jlink to create a standalone desktop application.

Provide .jar or native installers for distribution.

5. Conclusion

The JavaFX Expense Tracker project demonstrates how desktop applications can be effectively used to solve real-world financial tracking problems. The application provides a clean interface, real-time updates, and visual analytics, helping users monitor their financial behavior. JavaFX allows for rapid development of rich UIs and smooth integration with databases. This project not only reinforces object-oriented programming principles but also provides hands-on experience with UI design, database connectivity, and data handling. In the future, this application could be enhanced with cloud sync, multi-user support, or mobile versions using JavaFX Gluon.