EXPENSE TRACKER

Name: Shaik Shafiya

Roll Number: 23491A4753

Course / Department: CSIOT & BCT

College: QIS College of Engineering and Technology

Batch Number: 14

Mentor: [Mentor Name]

Date of Submission: May 13, 2025

# Certificate

This is to certify that the project entitled "Expense Tracker" is a bona fide work carried out by Shaik Shafiya (Roll No: 23491A4753), submitted in partial fulfillment for the award of degree in CSIOT & BCT at QIS College of Engineering and Technology. This project was carried out under my supervision and guidance.

Signature of Student Signature of Mentor

# Acknowledgement

I, **Shaik Shafiya**, student of **QIS College of Engineering and Technology**, bearing Roll No. **23491A4751**, would like to express my heartfelt gratitude to all those who have supported and guided me throughout the successful completion of my project titled **"Expense Tracker."**

First and foremost, I extend my sincere thanks to my project guide and mentor for their continuous encouragement, valuable suggestions, and unwavering support during each phase of the project. Their guidance played a crucial role in shaping this project into its final form.

I am also grateful to the management and faculty of **QIS College of Engineering and Technology** for providing the necessary resources, academic support, and an environment conducive to learning and innovation.

My special thanks go to my friends and family, whose moral support and motivation have been instrumental throughout my project journey.

Finally, I acknowledge the use of various online resources, development tools, and open-source technologies that helped me in designing and building this Expense Tracker efficiently.

This project has been a valuable learning experience and has helped me enhance my technical, problem-solving, and project management skills.

**Abstract**

The **Expense Tracker** is a user-friendly application designed to help individuals efficiently manage their personal finances. In today's fast-paced world, keeping track of daily expenses and income is essential for maintaining financial stability. This project aims to provide a digital solution that enables users to record, categorize, and analyze their financial transactions in a simple and effective manner.

The system allows users to add their income and expenses under various categories such as food, transport, utilities, and more. It provides visual summaries through charts and graphs, offering insights into spending patterns and budget control. The application is designed with a focus on ease of use, data accuracy, and accessibility.

Built using modern web technologies such as HTML, CSS, and JavaScript (or any framework/language you used), the Expense Tracker ensures real-time updates and responsive design for both desktop and mobile users. Optional features such as expense limits, monthly summaries, and alerts further enhance the user experience.

Table of Contents

1. 1. Introduction
2. 2. Problem Statement
3. 3. Objectives
4. 4. Existing System
5. 5. Proposed System
6. 6. System Design / Architecture
7. 7. Technologies Used
8. 8. Implementation
9. 9. Testing
10. 10. Results and Discussion
11. 11. Conclusion
12. 12. Future Scope
13. 13. References
14. 14. Appendix

# Introduction

In the modern era, where financial transactions occur almost daily, managing personal expenses has become a vital part of an individual's routine. Without a proper tracking system, it is easy to lose control over spending habits, which may lead to financial instability. To address this issue, the **Expense Tracker** project is developed as a simple yet effective tool to help users monitor and manage their income and expenditures.

The Expense Tracker serves as a digital assistant that enables users to keep a detailed record of all their financial transactions. It helps users categorize their expenses (e.g., food, transport, utilities, entertainment) and visualize spending trends over a specific period. With the help of intuitive charts and summaries, users can make informed decisions about their budgeting and saving practices.

This project is particularly beneficial for students, working professionals, and anyone aiming to gain control over their personal finances. By using this tool regularly, users can develop better financial discipline and avoid unnecessary expenditures.

The system is designed with a focus on usability, accessibility, and clarity. It utilizes technologies such as HTML, CSS, and JavaScript (or any relevant tools you used) to ensure a responsive and interactive interface. Whether it's daily tracking or monthly budgeting, this application provides a complete solution for personal finance management.

Problem Statement

In today’s fast-paced lifestyle, individuals—especially students and young professionals—often struggle to maintain a clear understanding of their personal finances. With multiple daily transactions and no structured approach to monitoring spending habits, managing income and expenses becomes a challenging task. Traditional methods like handwritten logs or static spreadsheets are time-consuming, error-prone, and lack the real-time insights necessary for effective financial planning.

Despite the increasing need for financial awareness, many individuals lack access to a simple, user-friendly, and centralized digital platform to track and analyze their daily expenditures. This leads to poor budgeting, uncontrolled spending, and missed savings opportunities.

Therefore, the problem addressed in this project is the absence of an intuitive, customizable, and accessible system that allows users to efficiently record, categorize, and evaluate their financial activities.

This project aims to solve the problem by designing and developing an **Expense Tracker** that is modern, responsive, and insightful—empowering users like **Shaik Shafiya** to take control of their personal finances, make informed decisions, and develop better money management habits.

Objectives

The main objective of this project is to design and develop a user-friendly **Expense Tracker** application that helps individuals manage their personal finances effectively. The specific goals of this project include:

**1.To provide a centralized platform** for recording daily income and expense transactions in a structured and organized manner.

**2.To categorize financial transactions** under customizable headings such as food, transport, utilities, entertainment, etc., for better analysis.

**3.To offer visual insights** through graphs, charts, and summaries that help users understand their spending patterns and make informed financial decisions.

**4.To ensure accessibility and responsiveness**, allowing the application to be used seamlessly across various devices including desktops, tablets, and smartphones.

**5.To implement budget monitoring features**, enabling users to set spending limits and receive alerts when approaching or exceeding those limits.

**6.To promote financial discipline** among users, especially students and professionals, by encouraging regular tracking and review of their financial activities.

**7.To enhance technical skills** in web development (HTML, CSS, JavaScript or any applicable tools) through the practical implementation of a real-world problem.

Existing System

In the current scenario, individuals use a variety of traditional and digital methods to track their expenses. However, these methods come with several limitations in terms of efficiency, accuracy, and user experience. The most commonly used existing methods include:

**Manual Record Keeping**  
Many people still rely on writing their daily expenses in notebooks or diaries. While this method is simple, it is time-consuming, prone to errors, and lacks data analysis capabilities.

**Spreadsheets (e.g., Microsoft Excel, Google Sheets)**  
Some users create custom templates in spreadsheets to record and calculate their income and expenses. Though more structured than manual methods, this approach requires technical knowledge and lacks automation, mobile accessibility, and real-time insights.

**Bank Statements and SMS Alerts**  
Users often refer to their bank statements or transaction SMS alerts to monitor their spending. However, these methods do not provide categorized views or summaries and are not suitable for proactive budget planning.

**Generic Mobile Applications**  
There are several existing expense tracking apps available in app stores. While some offer good features, many are either too complex, come with subscription fees, or do not allow customization according to individual needs.

### ****Limitations of Existing Methods****

Lack of real-time tracking and visualization.

No customization or categorization options.

Limited accessibility across devices.

Poor user interface or overwhelming features.

Dependence on external tools or premium features.

Proposed System

To overcome the limitations of the existing methods, this project proposes the development of a **Personal Expense Tracker System** that is simple, efficient, and tailored to individual needs. The proposed system is a web-based or mobile-compatible application that allows users to easily record, monitor, and analyze their financial transactions in real time.

### ****Key Features of the Proposed System****

**User-Friendly Interface**  
A clean and intuitive UI that makes it easy for users to add, view, and manage their income and expenses without technical knowledge.

**Categorized Transactions**  
Users can assign each transaction to specific categories (e.g., Food, Travel, Bills, Entertainment) for organized tracking.

**Visual Reports & Graphs**  
The system will generate charts and summaries to show monthly expenditure, savings, and category-wise spending for better financial insights.

**Responsive Design**  
The application is designed to work seamlessly on different devices such as desktops, tablets, and smartphones.

**Budgeting & Alerts**  
Users can set monthly budget limits for specific categories and receive alerts when they approach or exceed their budget.

**Search & Filter Options**  
Users can easily search for past transactions or filter them based on date, category, or amount.

**Data Storage**  
Transactions will be stored in a structured format (locally or in a database) ensuring quick access and data security.

System Design / Architecture

#### ****1. Presentation Layer (Front-End)****

This is the user interface of the application through which users interact. It handles all user input and displays relevant data in a structured and visually appealing format.

**Technologies Used**: HTML, CSS, JavaScript

**Components**:

Dashboard with expense summary

Forms to input income and expenses

Category selection dropdown

Charts/graphs (e.g., pie charts, bar graphs)

Budget alerts and notifications

#### ****2. Application Layer (Business Logic)****

This layer processes user inputs, performs calculations, and manages the overall flow of data between the front end and the database.

**Technologies Used**: JavaScript (or any framework like React.js, Angular, etc.)

**Responsibilities**:

Handle form validations and input formatting

Add, update, and delete transactions

Calculate totals (e.g., total income, total expense, balance)

Generate visual reports

Trigger budget alerts

#### ****3. Data Layer (Storage / Backend)****

This layer is responsible for storing and retrieving user data. Depending on the application type, data can be stored locally or in a cloud database.

**Options**:

**Local Storage** (for simple projects without login)

**Firebase / MongoDB / MySQL** (for full-stack applications with user accounts)

**Data Handled**:

Transaction details (amount, category, date, description)

User information (if login feature is included)

Budget limits and preferences

Technologies Used

The **Expense Tracker** system utilizes various modern technologies to ensure its functionality, performance, and user experience. Below are the key technologies used in the development of the project:

#### ****1. Front-End Technologies****

**HTML**:  
Used for structuring the content of the web application, creating the essential elements like forms, tables, buttons, and links.

**CSS**:  
Provides styling to the HTML elements, ensuring the application is visually appealing. It also ensures the responsiveness of the design across different devices and screen sizes.

**JavaScript**:  
Used for the dynamic behavior of the application, such as handling user inputs, processing data (calculations), generating charts, and displaying alerts. It also helps with form validation and interaction between different components.

**Bootstrap / Tailwind CSS (Optional)**:  
If used, these frameworks assist in quickly building a responsive and clean design with pre-designed components like navigation bars, buttons, and forms.

**Chart.js / D3.js (Optional)**:  
These JavaScript libraries are used to create dynamic and interactive charts and graphs, such as pie charts and bar graphs, to visually represent expenses and income.

#### ****2. Back-End Technologies (Optional)****

**Node.js (Optional)**:  
If a back-end system is implemented for storing data, Node.js can be used to build the server-side logic. It allows interaction with databases, user authentication, and API handling.

**Express.js (Optional)**:  
A web application framework for Node.js that simplifies routing and handling HTTP requests.

**Firebase / MongoDB / MySQL (Optional)**:  
These databases are used to store user data, transactions, and settings.

**Firebase**: A cloud-based NoSQL database that is easy to integrate with real-time features and user authentication.

**MongoDB**: A NoSQL database, used for storing documents in a flexible JSON-like format.

**MySQL**: A relational database for storing structured data with SQL queries.

#### ****3. Data Storage and Sync (Optional)****

**LocalStorage / SessionStorage**:  
For simple applications, LocalStorage or SessionStorage can be used to store transaction data on the user's browser. This option works well when the app does not require user authentication or real-time synchronization.

**Cloud Storage (Firebase, AWS, Google Cloud)**:  
Used to store user data securely and allow access across different devices, enabling users to track their expenses seamlessly from anywhere.

#### ****4. Version Control****

**Git**:  
Version control system used to track changes and collaborate with other developers. Git helps in managing the project's source code efficiently.

**GitHub / GitLab**:  
Platforms used to host the project repository, making it accessible to collaborators and enabling easy version control.

#### ****5. Development Environment****

**VS Code**:  
A code editor used for writing and managing the project’s code efficiently. It offers debugging tools, extensions, and integrations for a smooth development experience.

**Postman (Optional)**:  
If the project involves APIs (like fetching or sending data to a server), Postman is a useful tool for testing and managing API requests.

# Implementation

The **Expense Tracker** application was developed using a simple approach, focusing on ease of use and functionality. Here’s a breakdown of how the system was implemented:

#### ****1. User Interface (UI)****

**HTML**: The basic structure of the application was built using HTML, including sections for the transaction form, expense list, and charts.

**CSS**: The design was made visually appealing and responsive so it works well on both mobile and desktop devices.

**JavaScript**: Used to handle actions like adding expenses, calculating totals, and displaying the charts. It also updates the page in real-time without needing to refresh it.

#### ****2. Data Handling****

**Local Storage**: Transactions are saved in the browser’s LocalStorage, allowing data to persist even after the page is refreshed or closed. This ensures that the user's expense data is always available when they return to the app.

**Categorization**: Expenses are categorized (e.g., food, transport) so users can see how much they are spending in each category.

#### ****3. Budgeting and Alerts****

**Setting Budget Limits**: Users can set spending limits for different categories (like food, entertainment).

**Alert System**: When a user exceeds their budget in any category, the application alerts them so they can stay on track.

#### ****4. Responsive Design****

The app was designed to work smoothly on all devices. The layout adjusts automatically for mobile, tablet, and desktop screens, ensuring a good user experience on any device.

#### ****5. Testing****

The application was tested to make sure all features worked correctly, like adding expenses, calculating totals, and showing alerts. Testing was done on multiple browsers to ensure compatibility.

### ****Challenges Faced****

**Saving Data**: Initially, there were issues with data not being saved properly after the page refreshed, but using LocalStorage solved that.

**Design on Mobile**: Making sure the design worked well on different screen sizes was tricky, but it was fixed by using responsive design techniques.

### ****Future Improvements****

**Cloud Storage**: Data could be saved on the cloud for access across multiple devices.

**User Accounts**: Adding login functionality to let users have personal accounts and securely store their data.

# Testing

Testing is an important step to ensure that the **Expense Tracker** works correctly and as expected. Here’s how testing was done for the application:

#### ****1. Functional Testing****

Functional testing ensures that all features of the application work properly:

**Adding Expenses**: Testing was done to check if users can add new expenses and they show up in the transaction list.

**Calculations**: The totals for income, expenses, and balance are calculated correctly.

**Categories**: Testing whether users can correctly categorize their expenses (e.g., food, transport).

**Budget Alerts**: Ensuring that the budget alert system works correctly by notifying users when they exceed their budget.

#### ****2. Usability Testing****

Usability testing ensures the app is easy to use:

**User Interface (UI)**: We tested the UI to make sure it’s simple, intuitive, and easy for users to understand.

**Responsiveness**: The app was tested on different screen sizes (mobile, tablet, and desktop) to ensure it adapts properly.

#### ****3. Cross-Browser Testing****

The application was tested on different browsers (Chrome, Firefox, Safari) to make sure it works consistently across all platforms.

#### ****4. Data Storage Testing****

Testing was done to check if the application properly saves and retrieves data from the browser’s **LocalStorage**:

**Persisting Data**: Data was checked to ensure it remains saved even after closing and reopening the browser.

**Data Retrieval**: Verifying that the correct transaction data is displayed when the app is opened again.

#### ****5. Error Handling****

**Form Validation**: We tested to ensure that users cannot submit incomplete or incorrect data (e.g., leaving the amount field empty).

**Edge Cases**: Tested edge cases such as entering very high or low values to ensure the app handles them correctly.

### ****Challenges Faced During Testing****

**Data Persistence**: Initially, there was an issue where data wasn’t saving correctly after refreshing the page. This was fixed by improving the LocalStorage handling.

**UI Layout**: Ensuring the layout worked well across all devices was challenging, but it was resolved by refining the responsive design.

### ****Future Testing Improvements****

**Automated Testing**: In the future, automated testing could be implemented to make the testing process faster and more efficient.

**Cloud Sync Testing**: If cloud storage is added, testing would also need to check data synchronization between devices.

# Results and Discussion

The **Expense Tracker** was implemented successfully, and various tests were conducted to ensure its functionality. The results showed that the application meets its objectives, helping users efficiently track and manage their expenses. Below is a discussion of the results and key takeaways from the implementation:

#### ****1. Functionality****

**Expense Tracking**: The application allows users to add, view, and manage their expenses without any issues. Users can add expenses by entering the amount, category, and description, and these are displayed correctly in the transaction list.

**Expense Calculation**: The app correctly calculates total income, total expenses, and the balance after each transaction. These calculations are updated in real-time, providing users with immediate feedback on their spending habits.

**Categorization**: The categorization feature works well, helping users track their expenses by type (e.g., food, transportation). This feature allows users to gain insights into where their money is being spent most.

**Budget Alerts**: The budgeting system is functioning as expected. Users can set a spending limit for each category, and the application alerts them when they exceed their budget. This feature encourages users to stick to their financial plans.

#### ****2. Usability****

**User Interface**: The user interface is clean, intuitive, and easy to navigate. Users can quickly add transactions, view reports, and set budget limits without confusion.

**Responsive Design**: The application adapts well across different devices (mobile, tablet, desktop), ensuring that users can track their expenses on the go.

**Ease of Use**: The overall ease of use has been rated positively. Users can perform all necessary actions (adding expenses, viewing totals, setting alerts) without needing any prior technical knowledge.

#### ****3. Performance****

**Speed**: The application performs well even with a large number of transactions. Adding or removing transactions doesn’t significantly impact the speed of the app.

**Data Storage**: Data persistence works smoothly using LocalStorage. Transaction data is retained even after the app is closed and reopened.

#### ****4. Challenges and Solutions****

**Data Persistence Issue**: Initially, there was an issue with data not being saved after refreshing the page. This was resolved by using LocalStorage, which ensured the data remained even after a refresh.

**Design Responsiveness**: Ensuring the design worked well on all screen sizes posed a challenge. However, by implementing media queries and adjusting the layout, the issue was resolved, ensuring a seamless experience on all devices.

#### ****5. User Feedback****

Users have provided positive feedback on the application’s simplicity and effectiveness in tracking expenses. Many have appreciated the budgeting feature, as it helps them stay within their spending limits.

However, some users suggested adding features like export options (CSV/PDF) for their transaction data and adding a login system to securely store and access their data across multiple devices.

### ****Discussion****

The **Expense Tracker** fulfills its purpose by offering a simple, easy-to-use platform for tracking expenses. The application provides essential features such as expense categorization, budget alerts, and real-time calculations, which help users manage their finances effectively.

However, as the app evolves, additional features could be added to improve user experience and functionality:

**Cloud Syncing**: Syncing data across multiple devices and providing access to user data from anywhere.

**Export Features**: Allowing users to export their data for reports or record-keeping.

**User Accounts**: Adding login functionality would allow users to store their data securely and access it from any device.

Overall, the application has met its initial goals and shows potential for future enhancements.

Let me know if you want to dive deeper into any of the results or need further clarification!

4o mini

##### You said:

i need in a simple way and short

The **Expense Tracker** application works as expected and helps users manage their expenses efficiently. Here's a summary of the results:

#### ****1. Functionality****

**Expense Tracking**: Users can add and view expenses, and the app calculates totals correctly.

**Categorization**: Expenses are categorized (e.g., food, transport) to help users track spending.

**Budget Alerts**: The app alerts users when they exceed their set budget for any category.

#### ****2. Usability****

The app is easy to use with a simple interface, allowing users to add transactions and view reports without confusion.

It is **responsive**, working well on mobile, tablet, and desktop devices.

#### ****3. Performance****

The app runs smoothly with quick updates to the data, even when adding or removing transactions.

**Data Persistence**: Transaction data is saved and retrieved from LocalStorage, so it remains available even after closing the app.

#### ****4. Challenges****

There was an issue with saving data after refreshing the page, which was fixed using LocalStorage.

Making the design responsive on all devices took some adjustments, but it was successfully implemented.

#### ****5. User Feedback****

Users like the budgeting feature and the simplicity of the app. Some suggested adding features like data export and user login for security.

# Conclusion

The **Expense Tracker** successfully meets its goal of helping users track and manage their expenses efficiently. The application is easy to use, responsive, and provides essential features like expense categorization, budget alerts, and real-time calculations. It stores data locally, ensuring that users' transactions are saved even after refreshing the page.

While the app is functional, future improvements could include features like cloud syncing, data export options, and user login for a more personalized and secure experience.

# Future Scope

The **Expense Tracker** can be enhanced further with the following features:

**Cloud Syncing**: Allow users to store their data in the cloud, enabling access from multiple devices.

**User Accounts**: Implement a login system so users can securely access their expense data across devices.

**Data Export**: Provide options to export transaction data in formats like CSV or PDF for better reporting and analysis.

**Advanced Analytics**: Add features like monthly spending reports, spending trends, and financial forecasting to help users understand their financial habits better.

**Currency Support**: Enable users to track expenses in different currencies for broader global usage.

# References

1.W3Schools. (n.d.). **HTML Tutorial**. Retrieved from https://www.w3schools.com/html/

2.MDN Web Docs. (n.d.). **CSS Reference**. Retrieved from [https://developer.mozilla.org/en-US/docs/Web/CSS/Reference](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference" \t "_new)

3.Mozilla Developer Network (MDN). (n.d.). **JavaScript Guide**. Retrieved from [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide" \t "_new)

4.W3Schools. (n.d.). **JavaScript Tutorial**. Retrieved from https://www.w3schools.com/js/

5.Stack Overflow. (n.d.). **JavaScript LocalStorage Guide**. Retrieved from [https://stackoverflow.com/questions/6407425/how-to-use-localstorage](https://stackoverflow.com/questions/6407425/how-to-use-localstorage" \t "_new)

# Source code:

HTML CODE:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Expense Tracker</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h2>Expense Tracker</h2>

<form id="transaction-form">

<input type="date" id="date" required>

<input type="text" id="description" placeholder="Description" required>

<input type="text" id="category" placeholder="Category" required>

<input type="number" id="amount" placeholder="Amount" required>

<select id="type">

<option value="income">Income</option>

<option value="expense">Expense</option>

</select>

<button type="submit">Add Transaction</button>

</form>

<h3>Transactions</h3>

<table id="transaction-table">

<thead>

<tr>

<th>Date</th>

<th>Description</th>

<th>Category</th>

<th>Amount</th>

<th>Type</th>

<th>Action</th>

</tr>

</thead>

<tbody></tbody>

</table>

<div id="summary">

<p>Total Income: ₹<span id="total-income">0</span></p>

<p>Total Expense: ₹<span id="total-expense">0</span></p>

<p>Net Balance: ₹<span id="net-balance">0</span></p>

</div>

<script src="script.js"></script>

</body>

</html>

CSS CODE:

body {

font-family: Arial, sans-serif;

padding: 20px;

max-width: 800px;

margin: auto;

background-color: #f5f5f5;

}

h2, h3 {

text-align: center;

}

form {

background: #fff;

padding: 10px;

border-radius: 5px;

margin-bottom: 20px;

}

input, select, button {

margin: 5px;

padding: 8px;

width: 18%;

}

button {

background-color: #28a745;

color: white;

border: none;

}

table {

width: 100%;

border-collapse: collapse;

background: #fff;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: center;

}

#summary {

background-color: #fff;

padding: 10px;

margin-top: 20px;

border-radius: 5px;

text-align: center;

}

JAVASCRIPT CODE:

let transactions = JSON.parse(localStorage.getItem("transactions")) || [];

const form = document.getElementById("transaction-form");

const tableBody = document.querySelector("#transaction-table tbody");

const totalIncome = document.getElementById("total-income");

const totalExpense = document.getElementById("total-expense");

const netBalance = document.getElementById("net-balance");

form.addEventListener("submit", function (e) {

e.preventDefault();

const date = document.getElementById("date").value;

const description = document.getElementById("description").value;

const category = document.getElementById("category").value;

const amount = parseFloat(document.getElementById("amount").value);

const type = document.getElementById("type").value;

if (!date || !description || !category || isNaN(amount)) {

alert("Please fill out all fields correctly.");

return;

}

const transaction = { id: Date.now(), date, description, category, amount, type };

transactions.push(transaction);

localStorage.setItem("transactions", JSON.stringify(transactions));

form.reset();

renderTable();

});

function renderTable() {

tableBody.innerHTML = "";

let income = 0;

let expense = 0;

transactions.forEach((tx) => {

const row = document.createElement("tr");

row.innerHTML = `

<td>${tx.date}</td>

<td>${tx.description}</td>

<td>${tx.category}</td>

<td>₹${tx.amount.toFixed(2)}</td>

<td>${tx.type}</td>

<td><button onclick="deleteTransaction(${tx.id})">Delete</button></td>

`;

tableBody.appendChild(row);

if (tx.type === "income") income += tx.amount;

else expense += tx.amount;

});

totalIncome.textContent = income.toFixed(2);

totalExpense.textContent = expense.toFixed(2);

netBalance.textContent = (income - expense).toFixed(2);

}

function deleteTransaction(id) {

transactions = transactions.filter((tx) => tx.id !== id);

localStorage.setItem("transactions", JSON.stringify(transactions));

renderTable();

}

renderTable();