# 

**CSES003**

# Expense Tracker Project Report

Submitted by: Mohammad Kavish

Reg. No- 12211011

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab

# Abstract

The Expense Tracker project is a web-based application designed to help users manage their personal finances by tracking their expenses. Utilizing HTML, CSS, and JavaScript, this project demonstrates the implementation of core data structures and algorithms to handle expense data efficiently. Features include adding, viewing, sorting, and filtering expenses, along with a visual representation using Chart.js. The project also incorporates a user-friendly interface with dark mode functionality and local storage for data persistence.

# Table of Contents

1. Introduction

1.1 Background

1.2 Objectives

1.3 Significance

2. Problem Definition and Requirements

2.1 Problem Statement

2.2 Software Requirements

3. Proposed Design / Methodology

3.1 Schematic Diagram

3.2 File Structure

3.3 Algorithms Used

4. Results

4.1 Screenshots

4.2 Metrics

5. References

# 1. Introduction

## 1.1 Background

Managing personal finances is a crucial aspect of modern life, with many individuals seeking efficient tools to track their expenditures. The Expense Tracker project addresses this need by providing a simple yet powerful application that allows users to record and monitor their expenses seamlessly. This project leverages web technologies and data structures to deliver a practical solution for personal finance management.

## 1.2 Objectives

- Develop a user-friendly web application to track expenses.  
- Implement core data structures and algorithms for efficient data handling.  
- Provide features for adding, viewing, sorting, and filtering expenses.  
- Integrate visual representation of expenses using Chart.js.  
- Ensure data persistence through local storage.

## 1.3 Significance

The Expense Tracker project not only aids users in managing their finances but also showcases practical applications of data structures and algorithms. It highlights the importance of efficient data handling and user interface design in developing web-based applications.

# 2. Problem Definition and Requirements

## 2.1 Problem Statement

In today's fast-paced world, managing personal finances efficiently is crucial for maintaining financial health. However, many individuals struggle with tracking their expenses, leading to poor financial decisions and stress. Traditional methods of expense tracking, such as pen and paper or basic spreadsheets, can be cumbersome and prone to errors. Additionally, these methods lack features that can provide meaningful insights into spending habits.

The Expense Tracker Website project aims to address these issues by providing a robust, user-friendly platform for tracking personal expenses. This project integrates advanced features such as sorting, filtering, and visual representation of expenses to offer a comprehensive view of one's financial status. By utilizing modern web development technologies and data structures, this project ensures a seamless and efficient user experience.

## 2.2 Software Requirements

- HTML  
- CSS  
- JavaScript  
- Chart.js library  
- Local storage (browser)

# 3. Proposed Design / Methodology

## 3.1 Schematic Diagram

User Interface (HTML/CSS)  
 |  
JavaScript (Functional Logic)  
 |  
Local Storage (Data Persistence)  
 |  
Chart.js (Data Visualization)

## 3.2 File Structure

index.html  
pep.css  
pep.js

## 3.3 Algorithms Used

1. Expense Storage (Array/Vector):

A black and pink text

Description automatically generated

A black and white screen with white text

Description automatically generated

2. Expense Definition (Object/Struct):

A black and white screen with white text

Description automatically generated

A screen shot of a computer program

Description automatically generated

3. Summary Calculation (Hash Map):

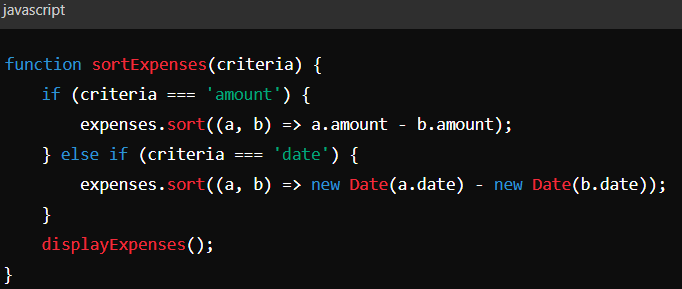
A computer screen with text on it

Description automatically generated

A computer screen shot of a code

Description automatically generated

4. Sorting:



A computer screen with text and symbols

Description automatically generated

5. Searching:

A screen shot of a computer code

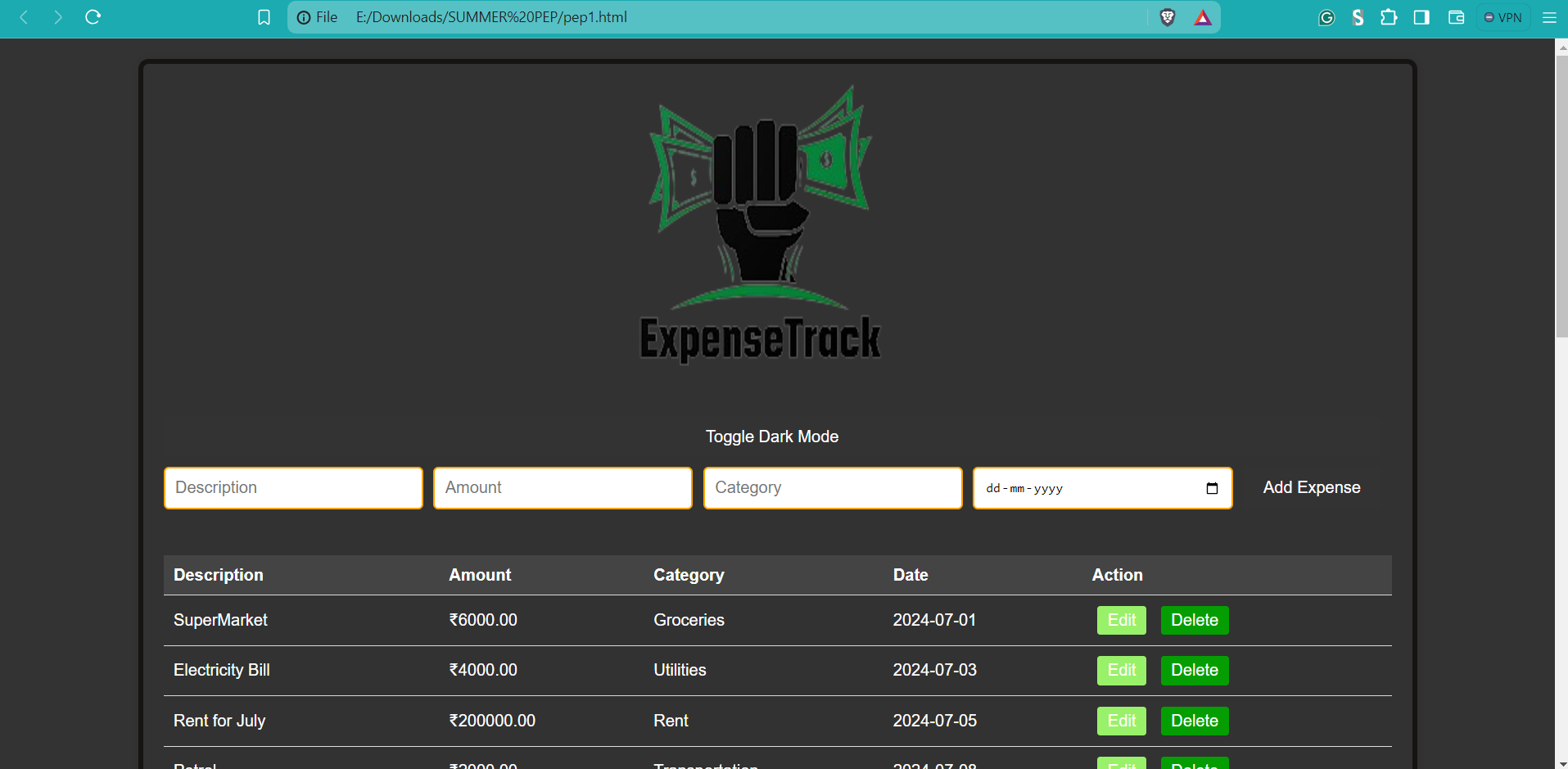
Description automatically generated

A computer screen with white text

Description automatically generated

# 4. Results

## Screenshots



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## 4.2 Metrics

## **Performance Metrics:**

* + Time complexity of adding an expense: O(1)
  + Time complexity of sorting expenses: O(n log n)
  + Time complexity of searching expenses: O(n)
  + Time complexity of summarizing expenses: O(n)

**Conclusion**

The Expense Tracker Website project addresses the need for efficient and user-friendly personal finance management. By leveraging modern web development technologies and implementing advanced features such as sorting, filtering, and visualization, this project provides users with a comprehensive tool for tracking and analyzing their expenses. The intuitive interface ensures ease of use, while the robust data storage and real-time updates guarantee a seamless user experience.

This project demonstrates the practical application of computer science principles in solving real-world problems. It not only simplifies the process of expense tracking but also empowers users to make informed financial decisions. Whether used for personal finance management, budget planning, or financial education, the Expense Tracker Website is a valuable tool that addresses a common and critical need in today's society

**References**

 MDN **Web Docs**: For JavaScript and Web APIs documentation.

* [JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript)
* [Web APIs](https://developer.mozilla.org/en-US/docs/Web/API)

 W**3Schools**: For HTML, CSS, and JavaScript tutorials and references.

* HTML
* CSS
* JavaScript

 Chart.js: For documentation on creating charts.

* Chart.js Documentation

 **GeeksforGeeks**: For C++ data structures and algorithms.

* C++ Data Structures
* C++ Algorithms