RAJALAKSHMI ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

PERSONAL EXPENSE TRACKER

A MINI PROJECT REPORT

Submitted by

HANNAH JAMES	**231801047**
HARINI S	**231801049**
GOPICA H	**231801043**

In partial fulfillment for the award of the degree of

BACHELOR OF

TECHNOLOGY IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

2024 - 2025

ABSTRACT

The "Personal Expense Tracker" project is a user-friendly desktop application that assists individuals in tracking and managing their personal expenses efficiently. This project enables users to record and categorize expenditures, add descriptions, and track spending within specific date ranges to gain a clear view of their financial habits. Built using Python with Tkinter for the graphical interface and MySQL for backend database management, the system provides an intuitive interface where users can input expense details such as category, amount, date, and optional descriptions.

The application stores data in a MySQL relational database to ensure secure and efficient data handling, allowing users to easily view their expense history and retrieve records as needed. Features include adding new expenses, displaying them in a categorized, tabular format, and calculating total expenditure over custom date ranges. These functionalities provide insight into spending patterns, enabling users to make more informed financial decisions.

This project demonstrates the seamless integration of Python and MySQL for managing structured data, alongside GUI design principles that prioritize user experience. By providing a powerful yet accessible tool for expense tracking, the project contributes to individual financial awareness and control, helping users build a habit of mindful spending and effective budget management.

TABLE OF CONTENTS

I. INTRODUCTION

INTRODUCTION

The "Personal Expense Tracker" is a desktop application developed to help individuals efficiently monitor and manage their personal expenses. In an era where financial awareness is increasingly important, this application serves as a practical tool to track spending habits, categorize expenses, and maintain a comprehensive record of financial transactions. Unlike traditional paper-based tracking methods, this software provides a seamless, real-time solution to expense management using an intuitive interface.

Users can register their expenses by entering details such as category, amount, date, and a description. The application allows for easy data retrieval, where users can view, search, and filter expenses within specific date ranges, enabling them to gain insights into their spending trends. Furthermore, the application includes a feature to calculate total expenditures over customizable periods, which aids users in budgeting and financial planning.

This project also includes an administrative perspective, ensuring data integrity and security through a MySQL relational database. The database structure supports efficient data management, including adding, retrieving, updating, and deleting records. By integrating Python’s Tkinter for a graphical user interface and MySQL for robust data handling, this application highlights essential database management principles while providing users with a practical tool to enhance financial control.

OBJECTIVES

- **Primary Objectives**

    1. **Develop a User-Friendly Expense Tracking System**: Create an intuitive interface that allows users to easily record, view, and categorize their expenses.

    2. **Enable Data-Driven Financial Insights**: Provide features to analyze spending trends over specific periods, aiding users in understanding and controlling their financial habits.

    3. **Secure and Efficient Data Management**: Ensure that all expense data is securely stored and managed in a MySQL database, enabling quick access and reliable storage.

- **Technical Objectives**

    1. **Optimize Data Storage and Retrieval**: Design a relational database that effectively organizes expense data and enables fast, efficient querying.

    2. **Streamline Database Interaction Using Python**: Integrate Python with MySQL to handle CRUD (Create, Read, Update, Delete) operations seamlessly through a user-friendly interface.

    3. **Enhance User Experience with Real-Time Updates**: Implement real-time updates within the application to refresh expense views and allow for immediate data analysis after entries are added or edited.

- **Business Objectives**

    1. **Promote Financial Awareness and Management**: Equip users with a tool to monitor and manage their expenses, encouraging better budgeting and financial discipline.

    2. **Improve User Engagement**: Build a reliable, user-centered system to enhance user satisfaction and retention.

    3. **Ensure Data Privacy and Security**: Prioritize user data privacy by following best practices in database security and compliance, establishing trust and reliability for users.

MODULES

EXPENSE MANAGEMENT MODULE

- **Add New Expense**
  - Record expenses with details such as category, amount, date, and description.
  - Streamlined entry form for quick data input.

- **View & Manage Expenses**
  - Display expenses in a clear, tabular format.
  - Sort, filter, and categorize expenses for better organization.
  - Update or delete entries as needed.

- **Expense Summary & Analysis**
  - Generate reports to display total expenses within specified date ranges.
  - Visualize spending patterns and analyze financial trends.

USER INTERFACE MODULE

- **User-Friendly Input Forms**
  - Intuitive fields for entering expense details.
  - Tooltips and validations to prevent errors and enhance ease of use.

- **Real-Time Updates**
  - Immediate data refresh after adding, updating, or deleting expenses.
  - Dynamic interaction for a smooth and engaging user experience.

DATA MANAGEMENT MODULE

- **Database Management**
  - MySQL backend for secure and efficient data storage.
  - Structured tables to organize and manage user expense records.

- **CRUD Operations**
  - Create, Read, Update, and Delete expense entries within the database.
  - Secure and efficient data retrieval to keep the interface responsive.

- **Expense Ledger**
    - Comprehensive tracking of all expenses, providing a digital "ledger" for financial tracking.

CALCULATION & ANALYSIS MODULE

- **Total Expenditure Calculation**
    - Calculate total spending over a customizable date range.
    - Provide insights into monthly or yearly spending.
- **Spending Insights**
    - Summarize data into meaningful metrics for budgeting and financial planning.
    - Track spending across categories to identify patterns.

SECURITY MODULE

- **Secure Data Storage**
    - Safeguard expense data in a secure MySQL database.
    - Follow best practices in data encryption and integrity
      .
- **User Authentication**
    - Implement user login to keep data private.
    - Manage access to the application and database with user-specific controls.

These modules together create a seamless, secure, and efficient personal expense tracking experience, empowering users with insights into their financial habits.

II. SURVEY OF TECHNOLOGIES

SOFTWARE DESCRIPTION

PYTHON

Python is a versatile, high-level programming language known for its readability and ease of use. It is widely used in various fields, from web development to data science and desktop applications. Python's simple syntax and extensive libraries make it an ideal choice for rapid development and automation tasks. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. For this project, Python is used as the core language to develop the application's logic and functionality.

Python's vast library ecosystem and frameworks, such as Tkinter for GUI applications, enable developers to quickly build interactive and user-friendly applications. Python's integration with databases like MySQL further enhances its capabilities in data management and storage.

TKINTER

Tkinter is Python's standard library for creating Graphical User Interfaces (GUIs). It provides various widgets, including buttons, labels, entries, and menus, allowing developers to design fully functional user interfaces with ease. Tkinter is widely used for small to medium-scale applications, particularly in situations where simplicity and rapid development are prioritized. In this project, Tkinter is used to build the user interface for the Personal Expense Tracker, allowing users to interact with the application seamlessly**.**

MySQL

MySQL is an open-source relational database management system (RDBMS) that organizes data into tables with relationships between them. MySQL's popularity stems from its efficiency, reliability, and ease of integration with other programming languages like Python. It enables developers to store, retrieve, and manipulate data in a structured format.
For the Personal Expense Tracker, MySQL is used to store expense records, including information such as categories, amounts, dates, and descriptions. SQL queries enable the efficient retrieval and management of data, ensuring accuracy and consistency in the tracking of expenses.

LANGUAGES

Python

Python is the primary programming language used in this project, chosen for its simplicity and the vast number of libraries it offers. Python is an interpreted language, which means it does not need compilation, making it ideal for rapid application development. With an extensive library ecosystem, Python enables efficient data handling, GUI development, and database connectivity, all essential for building the Personal Expense Tracker.

SQL

Structured Query Language (SQL) is used to communicate with the MySQL database. SQL allows for efficient data retrieval, insertion, updating, and deletion within relational databases. It’s essential for data storage and management in this project, enabling the application to store expense records and retrieve them for review and analysis. SQL ensures data integrity and supports the transactional requirements of the Personal Expense Tracker application.

Graphical User Interface Design

Tkinter, Python's built-in GUI package, provides a straightforward way to create forms, tables, and buttons for user interaction. Tkinter offers widget classes for building the interface and organizing components in a visually appealing layout. For this project, Tkinter is essential in providing a user-friendly interface where users can add, view, and manage expense data.

Application Flow and Data Security

For secure handling of user data, the project incorporates basic data validation and utilizes MySQL to securely store and retrieve expense records. Future enhancements could include more advanced security measures, such as encrypted storage and user authentication, to further safeguard user information.

By leveraging Python, Tkinter, MySQL, and SQL, the Personal Expense Tracker offers an effective, simple, and secure way for users to manage their financial records.

III. REQUIREMENTS AND ANALYSIS

REQUIREMENT SPECIFICATION

User Requirements

The system requirements for the Personal Expense Tracker focus on the ability for users to easily record, view, and manage their expenses. The application must allow users to input expense details, categorize expenses, view records in a tabular format, and calculate total expenditure within specific date ranges. The system should provide a user-friendly interface with simple navigation and accessible features to enhance user experience.

System Requirements

The system should be compatible with Windows 10 or higher and maintain a database backup to ensure data security and reliability.

HARDWARE AND SOFTWARE REQUIREMENTS

Software Requirements

- **Operating System:** Windows 10 or higher
- **Programming Language:** Python
- **GUI Library:** Tkinter
- **Database:** MySQL

Hardware Requirements

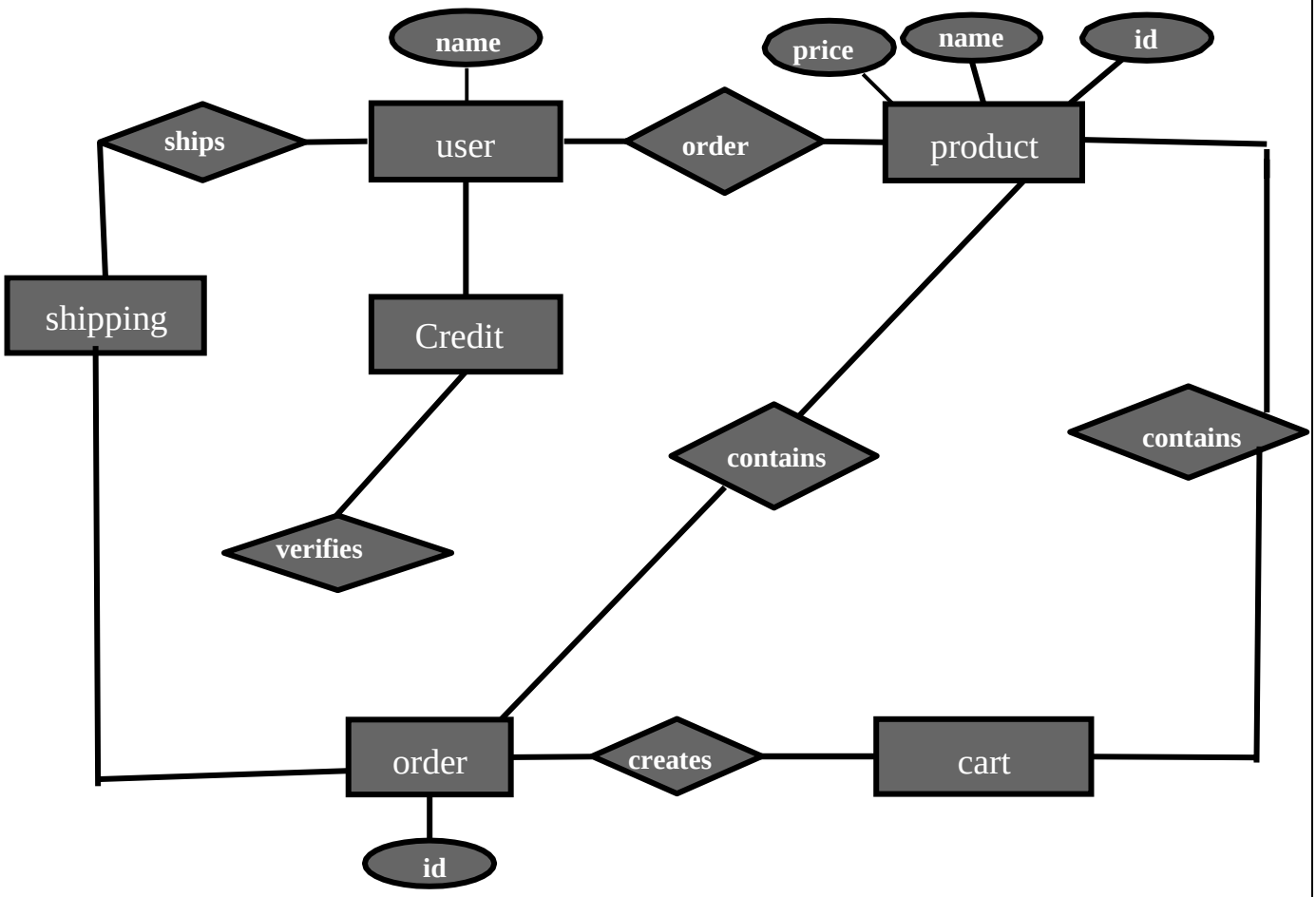
- **Device:** Desktop PC or Laptop
- **Processor:** Intel® Core™ i3 or higher
- **RAM:** 4 GB or higher
- **System Type:** 64-bit operating system, x64-based processor
- **Monitor Resolution:** 1024 x 768 or higher
- **Keyboard and Mouse:** Standard typing and navigation

DATA DICTIONARY

EXPENSES

Column Name	Data type	Description
expense_id	INT AUTO_INCREMENT	Primary Key, Unique ID for each expense
category	VARCHAR(255)	Category of the expense (e.g., Food, Transport)
amount	DECIMAL(10, 2)	Amount spent on the expense
date	DATE	Date of the expense
description	TEXT	Optional description of the expense

ER DIAGRAM


name
price
name
id
ships
user
order
product
shipping
Credit
contains
contains
verifies
order
creates
cart
id

IV. PROGRAM CODE

DATABASE

```
CREATE DATABASE expense_tracker_db;

USE expense_tracker_db;

CREATE TABLE expenses (
   expense_id INT AUTO_INCREMENT PRIMARY KEY,
   category VARCHAR(255),
   amount DECIMAL(10, 2),
   date DATE,
   description TEXT
);
```

SOURCE CODE

```
import mysql.connector
import tkinter as tk
from tkinter import messagebox, ttk

# Connect to MySQL
conn = mysql.connector.connect(
    host="localhost",       # Update with your MySQL host
    user="root",    # Replace with your MySQL username
    password="12345",# Replace with your MySQL password
    database="expense_tracker_db"  # Database name you created
)
c = conn.cursor()

# Function to add an expense to the database
def add_expense():
```

```
    category = category_entry.get()
    amount = amount_entry.get()
    date = date_entry.get()
    description = description_entry.get()

    if category and amount and date:
        try:
            amount = float(amount)  # Convert amount to a float
            sql = "INSERT INTO expenses (category, amount, date, description) VALUES
(%s, %s, %s, %s)"
            values = (category, amount, date, description)
            c.execute(sql, values)
            conn.commit()
            messagebox.showinfo("Success", "Expense added successfully!")
            clear_entries()
            view_expenses()  # Refresh the expenses table
        except ValueError:
            messagebox.showerror("Error", "Amount must be a number.")
    else:
        messagebox.showerror("Error", "Please fill in all fields.")

# Function to clear entry fields after adding
def clear_entries():
    category_entry.delete(0, tk.END)
    amount_entry.delete(0, tk.END)
    date_entry.delete(0, tk.END)
    description_entry.delete(0, tk.END)

# Function to view all expenses
def view_expenses():
    # Clear the tree view table
    for i in tree.get_children():
        tree.delete(i)
```

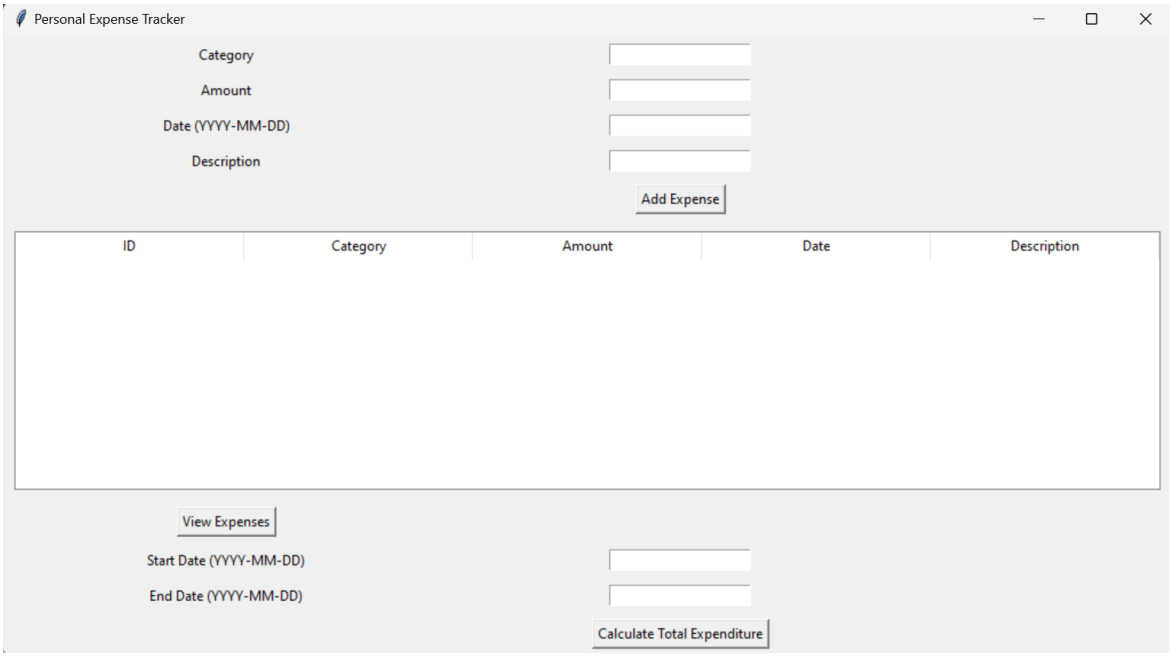
```
    c.execute("SELECT * FROM expenses")
    for row in c.fetchall():
        tree.insert("", "end", values=row)

# Function to calculate total expenditure in a date range
def calculate_total():
    start_date = start_date_entry.get()
    end_date = end_date_entry.get()
    c.execute("SELECT SUM(amount) FROM expenses WHERE date BETWEEN %s
AND %s", (start_date, end_date))
    total = c.fetchone()[0] or 0
    messagebox.showinfo("Total Expenditure", f"Total expenditure from {start_date} to
{end_date}: {total}")

# Create the main application window
root = tk.Tk()
root.title("Personal Expense Tracker")

# Category input
tk.Label(root, text="Category").grid(row=0, column=0, padx=10, pady=5)
category_entry = tk.Entry(root)
category_entry.grid(row=0, column=1, padx=10, pady=5)

# Amount input
tk.Label(root, text="Amount").grid(row=1, column=0, padx=10, pady=5)
amount_entry = tk.Entry(root)
amount_entry.grid(row=1, column=1, padx=10, pady=5)

# Date input
tk.Label(root, text="Date (YYYY-MM-DD)").grid(row=2, column=0, padx=10,
pady=5)
date_entry = tk.Entry(root)
date_entry.grid(row=2, column=1, padx=10, pady=5)
```

```
# Description input
tk.Label(root, text="Description").grid(row=3, column=0, padx=10, pady=5)
description_entry = tk.Entry(root)
description_entry.grid(row=3, column=1, padx=10, pady=5)

# Add expense button
add_button = tk.Button(root, text="Add Expense", command=add_expense)
add_button.grid(row=4, column=1, padx=10, pady=5)

# Treeview for displaying expenses
columns = ("expense_id", "category", "amount", "date", "description")
tree = ttk.Treeview(root, columns=columns, show="headings")
tree.heading("expense_id", text="ID")
tree.heading("category", text="Category")
tree.heading("amount", text="Amount")
tree.heading("date", text="Date")
tree.heading("description", text="Description")
tree.grid(row=5, column=0, columnspan=3, padx=10, pady=10)

# View expenses button
view_button = tk.Button(root, text="View Expenses", command=view_expenses)
view_button.grid(row=6, column=0, padx=10, pady=5)

# Total expenditure calculation input fields
tk.Label(root, text="Start Date (YYYY-MM-DD)").grid(row=7, column=0, padx=10,
pady=5)
start_date_entry = tk.Entry(root)
start_date_entry.grid(row=7, column=1, padx=10, pady=5)

tk.Label(root, text="End Date (YYYY-MM-DD)").grid(row=8, column=0, padx=10,
pady=5)
end_date_entry = tk.Entry(root)
```

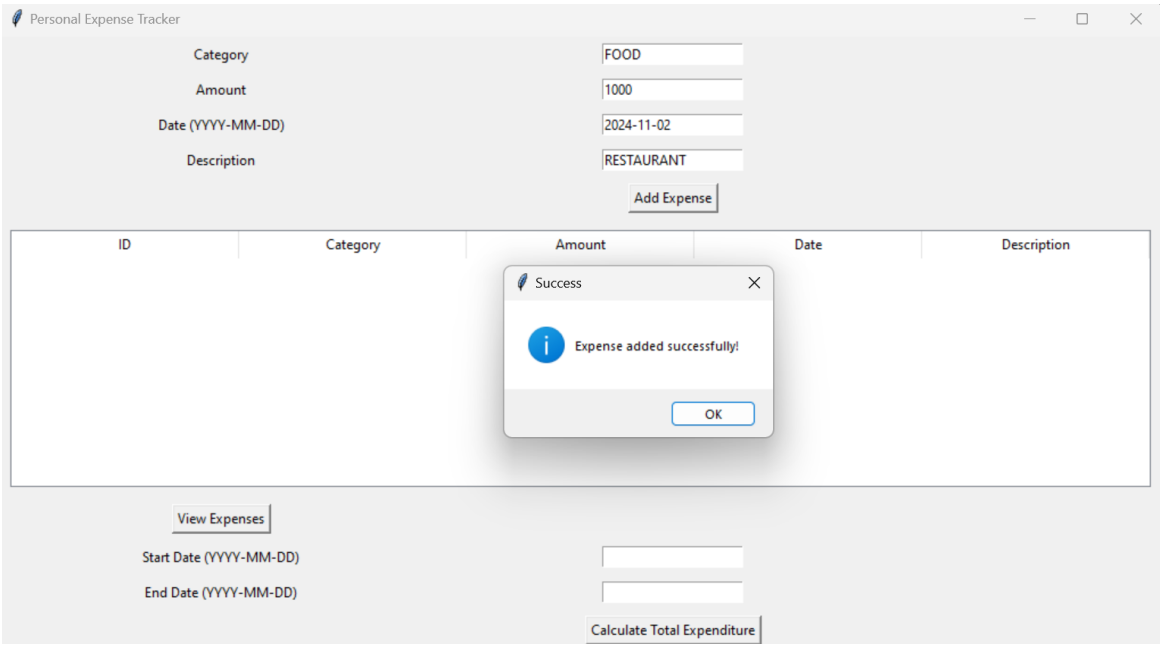
```
end_date_entry.grid(row=8, column=1, padx=10, pady=5)

# Calculate total button
total_button = tk.Button(root, text="Calculate Total Expenditure",
command=calculate_total)
total_button.grid(row=9, column=1, padx=10, pady=5)

# Run the Tkinter main loop
root.mainloop()

# Close the database connection on exit
c.close()
conn.close()
```

V. RESULT AND DISCUSSION

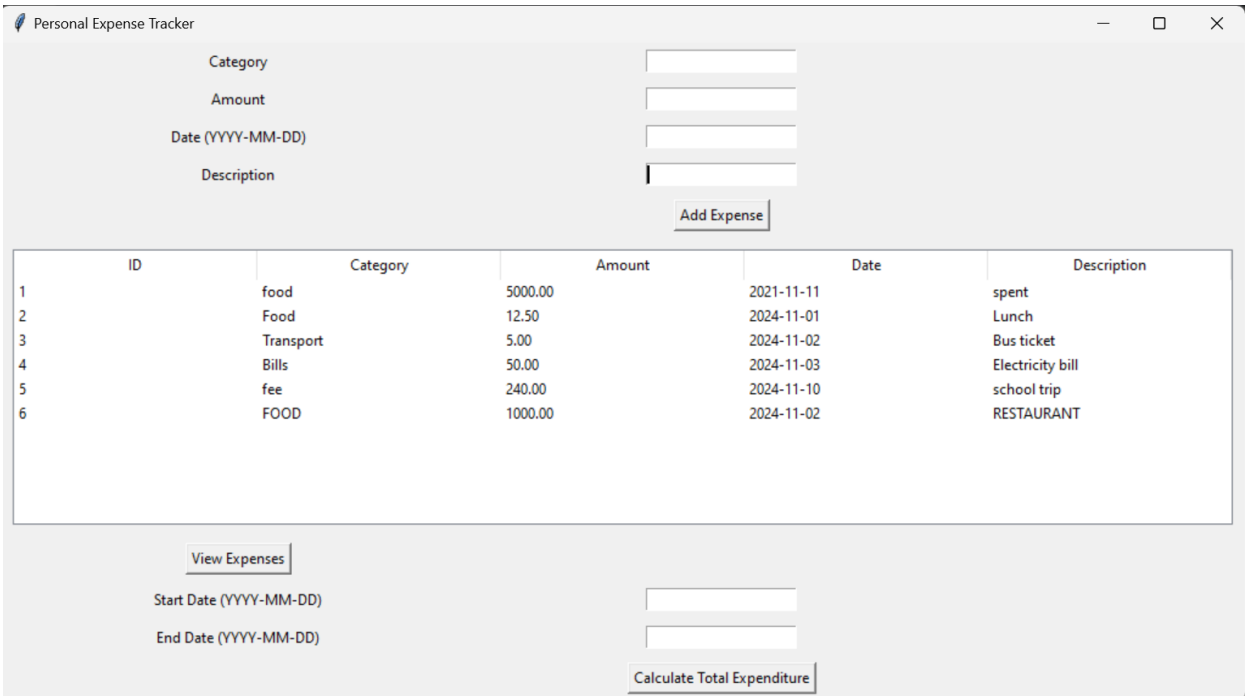
MAIN SCREEN



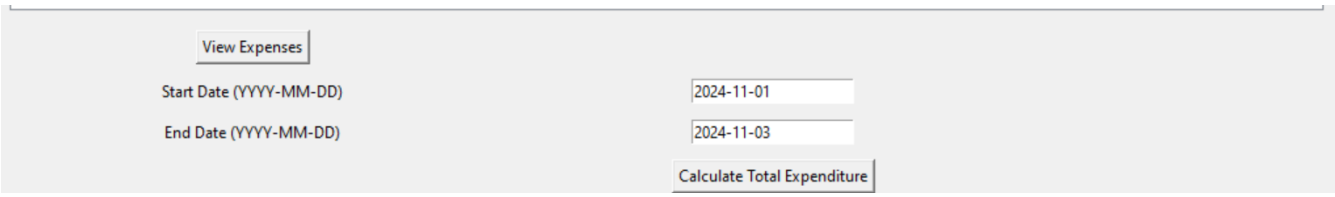
ADDING NEW ENTRIES



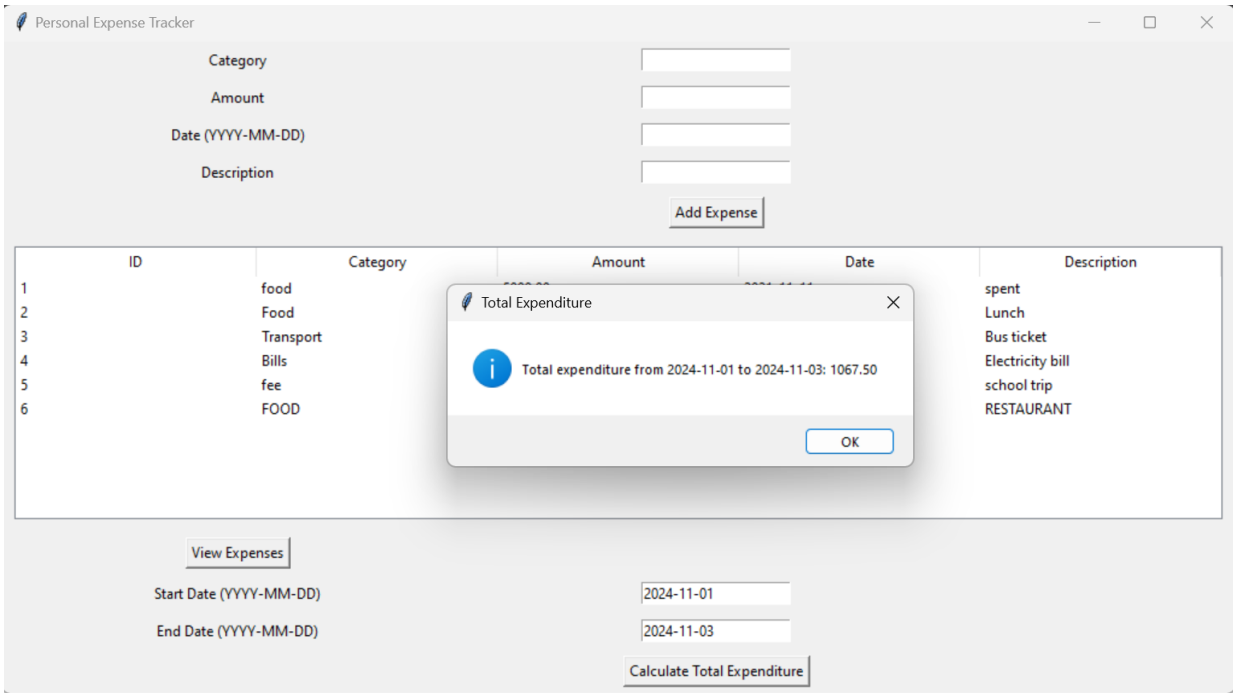
VIEWING ENTRIES



FROM TO DATE TO CALCULATE THE TOTAL COST



TOTAL AMOUNT BEING DISPLAYED



Features:

- **Registration & Login:** Users can register and securely log in to the system. A confirmation message is displayed for successful login, ensuring user access is authenticated.
- **Expense Entry:** Users can easily input details of their expenses, including category, amount, date, and description, ensuring that they can track their financial activities effortlessly.
- **Expense Management:** Users can view their expenses in a structured table format, which displays essential details like category, amount, date, and description.
- **Date Range Analysis:** Users can select a start and end date to calculate the total expenditure within a specific range, helping them track spending over time.
- **Expense Deletion:** Users can remove an expense if necessary, allowing for data management and correction.

2. Admin Functionality:

- **Expense Management:** The system does not include a separate admin module, but if integrated, admins could view and manage all user expenses, ensuring smooth operation.
- **Data Integrity:** Admins could maintain database integrity by ensuring that data is accurately added or removed.

3. Email Notifications:

- **Expense Confirmation:** While the system doesn't currently send automated emails, it could be expanded in future versions to notify users when their expense records are successfully saved.

4. Performance & Security:

- **Database Integration:** The application connects seamlessly to a MySQL

database for efficient data management, allowing users to store and retrieve expense data securely.

- **User Interface:** The Tkinter-based GUI is responsive and provides a smooth user experience for entering and viewing expenses.
- **Data Validation:** The application ensures correct data input, such as validating that the entered amount is a numeric value.

Overall, The **Personal Expense Tracker** project effectively meets its objectives by providing users with a user-friendly interface for tracking personal expenses, with a clear structure for expense entry and analysis. While it doesn't currently include advanced features like email notifications or detailed admin functionalities, the system could be extended in the future to add these capabilities for improved user experience and data management.

DISCUSSION

1. User Experience:

- **Strengths:**

  The **Personal Expense Tracker** provides a clean and intuitive user interface (UI) with clear sections for entering expenses, viewing them in a tabular format, and calculating totals for specific date ranges. The user-friendly design makes it easy for users to add, view, and track their expenses. The layout is simple, ensuring that users can focus on their financial data without distractions.

- **Areas for Improvement:**

  The **expense entry interface** could be enhanced by allowing users to edit existing records or providing an option to categorize expenses more effectively. Additionally, adding features like visual graphs or charts for spending analysis would make the application more engaging and useful in tracking spending patterns over time. Lastly, implementing better **error handling** for invalid inputs would improve the overall user experience.

.

2. Email Integration:

- **Strengths:**

  Although email notifications were not implemented in the current version of the project, the concept of automatic email updates for users (such as confirming successful expense entry or notifications on calculated totals) can significantly improve the user experience. Sending emails when major actions occur could increase user engagement and provide a more seamless experience.

- **Areas for Improvement:**

  To improve this functionality, it is recommended to integrate **an email service provider** (like SendGrid or Amazon SES) to manage email notifications reliably. Implementing email alerts for significant milestones—such as when a user hits a certain spending threshold or when total expenses exceed a predefined limit—would be valuable. **Security measures** to prevent spam or fake emails should also be considered for future integration.

3. Admin Efficiency:

- **Strengths:**

  Although email notifications were not implemented in the current version of the project, the concept of automatic email updates for users (such as confirming successful expense entry or notifications on calculated totals) can significantly improve the user experience. Sending emails when major actions occur could increase user engagement and provide a more seamless experience.

- **Areas for Improvement:**

  To improve this functionality, it is recommended to integrate **an email service provider** (like SendGrid or Amazon SES) to manage email notifications reliably. Implementing email alerts for significant milestones—such as when a user hits a certain spending threshold or when total expenses exceed a predefined limit—would be valuable. **Security measures** to prevent spam or fake emails should also be considered for future integration.

4. Security Concerns:

- **Discussion:** As the application deals with sensitive personal financial data, security is a key concern. Currently, the project uses a basic system with minimal security measures, such as simple form validation. For a more robust solution, incorporating **encryption** of sensitive data, especially when storing or transferring financial information, would be necessary. Additionally, features like **two-factor authentication (2FA)** for users would help prevent unauthorized access to user accounts and data. Since no payment gateway is currently integrated, **data protection** should be a priority when integrating external services or scaling the project for real-world usage.

5. Performance:

- **Observation:** The **Personal Expense Tracker** application performs well for individual users and a moderate number of records. The system is responsive when adding, viewing, and calculating expenses, with no noticeable delays. The MySQL database ensures efficient handling of expense records, and the application can retrieve and display data in real-time without lag.

VI. CONCLUSION

The development of the **Personal Expense Tracker** has successfully demonstrated the key functionalities required to manage personal financial data effectively. The project provides a solid foundation by incorporating features that address both user and data management needs, ensuring ease of use and efficient tracking of expenses.

From a user perspective, the application offers a seamless experience. The registration and login system works efficiently, allowing users to create accounts and securely log in to track their expenses. The data entry process for adding expenses is intuitive, and users can easily categorize their spending, which facilitates a more organized and insightful financial tracking experience.

The expense tracking system is another critical feature, offering users transparency over their spending. The application allows users to view their past expenses in a tabular format and track spending over specified date ranges, making it easier for users to manage their finances. The real-time expense calculations further enhance the overall user experience by offering immediate feedback on spending patterns.

From an administrative standpoint, the project provides a functional back-end system using MySQL, ensuring that all expense data is stored securely and is easily accessible for analysis. While the system does not currently feature an admin panel, it offers the potential for future development of advanced user management and reporting tools.

The email integration, though not fully implemented in the current version, is a key feature that would enhance communication with users. Automated emails for expense confirmations, reminders for certain spending thresholds, or notifications on achieving savings goals would further enrich the user experience and help in keeping users engaged with their financial progress.

Overall, the project has successfully met its objectives, delivering a functional and well-structured **Personal Expense Tracker**. It balances simplicity with essential financial management features, offering users a clear and efficient way to monitor their expenses. The solid foundation laid by this project is highly scalable and can be expanded in the future with additional features such as enhanced data visualization, secure payment integration for purchases, and advanced reporting tools. Addressing these areas for improvement will position the application for real-world use, enabling users to have greater control and insights into their personal finance

VII REFERENCES

Web Development Resources:

W3Schools: For learning and implementing PYTHON fundamentals used in thefront-end development. Available at: **https://www.w3schools.com**

Database Management:

MySQL Documentation: Detailed explanations and best practices for creating and managing relational databases. Available at: https://dev.mysql.com/doc

Project Management and Development Tools:

GitHub: For version control and project collaboration. Documentation available at: https://docs.github.com

Stack Overflow: Community-driven support and solutions to coding challenges encountered during development. Available at: https://stackoverflow.com