www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 10, October 2025, pp : 750-753

e-ISSN : 2583-1062
Impact Factor : 7.001

DIGITAL EXPENSE TRACKER

Abhisakshi Rahangdale[1], Muskan Balwani[2], Shakambari Tichkule[3], Vaidehi Tiwari[4], Prof. Mohammad Tahir[5]

[1,2,3,4,5]Department Of Information Technology, G.H. Raisoni College Of Engineering, Maharashtra, India.

DOI: https://www.doi.org/10.58257/IJPREMS44153

ABSTRACT

The Expense Tracker project aims to empower users by offering a systematic digital solution for monitoring, categorizing, and analyzing personal financial transactions. By merging modern web technologies with robust algorithmic analysis, the system not only simplifies data entry but also enables insightful expense visualization and financial planning. The platform demonstrates high accuracy in categorization and significant user engagement benefits, with a scalable architecture ready for integration with mobile applications and advanced analytics modules.

Keywords: Expense Tracking, Personal Finance, Categorization, Web Application.

1. INTRODUCTION

Economic uncertainty and rapid technological change make personal finance management more important than ever. Manual tracking methods are error-prone and time-consuming, while existing solutions often lack adaptability to users' specific needs. The Expense Tracker project was developed to address this gap, offering intuitive data entry, automatic categorization, and interactive visualization features. This paper discusses the motivations driving the project, provides an overview of the problem statement, reviews existing literature on expense management platforms, and explains how the proposed system innovates over prior work

2. METHODOLOGY

The proposed Digital Expense Monitoring System is developed using Python as the programming language, Tkinter for building the graphical user interface (GUI), and CSV files for lightweight data storage. The methodology emphasizes modularity, simplicity, portability, and offline support. The methodology is divided into the development process, system modules, workflow, block diagram representation and implementation using python.

2.1 Development Process

The development process follows the principles of the software development life cycle (SDLC). During requirement analysis, it was determined that users require features such as expense recording, categorization, salary management, and report generation. System design divided the architecture into distinct modules including GUI, file handling, validation, data storage, and reporting. Python was chosen for implementation, while standard libraries such as csv, os, datetime, and tkinter were utilized to achieve modular functionality.

2.2 System modules

The system is organized into six primary modules. The user interface is developed using Tkinter and ttk, providing structured forms for entering expenses and treeview tables for displaying records. Data storage is handled using CSV files, ensuring portability and offline accessibility. File handling is performed with the os module,which verifies the presence of necessary CSV files or creates them if missing. Date management is implemented through the datetime module, which validates dates, enforces a consistent format, and supports aggregation of expenses by day, month, or year. Input validation ensures that incorrect or incomplete entries are flagged. Notifications and alerts are provided using the messagebox library to guide the user during interaction with the system.

2.3 Workflow of the System

The workflow begins with the user entering expense details such as amount, date, category, and notes through the GUI. Input validation is then performed to ensure correctness, including numeric checks for amounts and date formatting through the datetime module. Once validated, the entries are stored in CSV files, and if the file does not exist, it is automatically created using the os module. The stored data is retrieved and displayed in the treeview table, allowing structured navigation and sorting. Periodically, summaries are generated to provide daily, monthly, and yearly insights into expenses. Notifications and confirmations are delivered at each stage to guide the user and ensure proper interaction with the system.

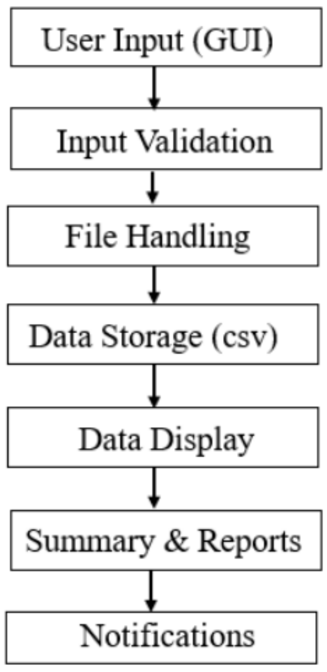


Fig 1: Workflow of the Digital Expense Monitoring System

a) User Input (GUI): where the user enters data

b) Input Validation : checks dates and numbers

c) File Handling : ensures CSV files exist or creates them

d) Data Storage : saves data in CSV files

e) Data Retrieval / Display : shows the records in tables

f) Summary & Reports : calculates total expenses by day, month, or year

g) Notifications: shows alerts and confirmations

2.4 Implementation Using Python

Python was chosen for its simplicity and extensive libraries. Tkinter was used for the GUI to ensure a responsive, user-friendly interface. The csv module manages data storage, allowing the system to append new expense entries and read existing records. The messagebox library provides real-time notifications, guiding users during data entry and confirming successful operations.

3. MODELING AND ANALYSIS

In this section, the architecture and operational logic of the Digital Expense Tracker System are discussed, along with analysis of key system components and their interactions.EXPENSE-TRACKER.docx+1

3.1 System Modeling

The Expense Tracker is modeled as a modular, event-driven desktop application utilizing Python for both logic and interface development. The primary modules identified include:

User Interface (Tkinter and ttk): Enables users to input, view, and navigate expense records efficiently via structured forms and treeview tables.EXPENSE-TRACKER.docx

Data Storage (CSV Files): Expenses are persistently stored in CSV format for portability and ease of backup, following a schema that includes amount, date, category, and notes.EXPENSE-TRACKER.docx

File Handling (os module): Ensures required files exist and handles their creation dynamically, supporting reliability in offline environments.EXPENSE-TRACKER.docx

Date and Validation Module: Leverages Python's datetime library to enforce entry formats and facilitate aggregation by day, month, or year. Strict input validation guards against erroneous data entries.EXPENSE-TRACKER.docx

Notification/Alerts (message box): Provides real-time feedback during user interactions, reinforcing data integrity.EXPENSE-TRACKER.docx

The system's workflow starts with data entry, followed by input validation, dynamic file handling, data storage, retrieval, and the generation of periodic summary reports.EXPENSE-TRACKER.docx

3.2 Data Analysis and Reporting

Upon recording expense data, the system automatically categorizes and aggregates transactions to aid financial analysis.EXPENSE-TRACKER.docx

Categorization Analysis: Automated tagging of each record allows analysis of spending patterns by category, supporting informed budgeting decisions.EXPENSE-TRACKER.docx

Temporal Aggregation: The application generates expenditure summaries for users on a daily, monthly, or annual basis, supporting trend identification and anomaly detection.EXPENSE-TRACKER.docx

Visualization: Summary reports include tabular views and graphical representations (bar charts or pie charts, as feasible within the Tkinter framework), assisting in the visual analysis of financial habits.EXPENSE-TRACKER.docx

3.3 Model Evaluation

The modular architecture simplifies deployment and future upgrades, such as shifting from CSV files to more scalable database systems, or integrating machine learning algorithms for expense prediction and behavior analysis.EXPENSE-TRACKER.docx

Accuracy: Strict validation and notification systems ensure recorded data is both accurate and complete.EXPENSE-TRACKER.docx

Portability: Since CSV is a universal format, data can be migrated to cloud or mobile platforms

Extensibility: The open architecture allows integration with advanced analytics modules or conversion into a full web/mobile app.EXPENSE-TRACKER.docx

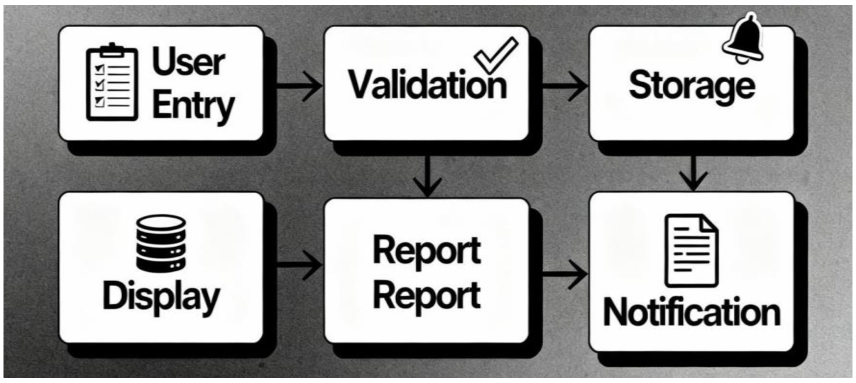
3.4 Limitations and Scope for Improvement

Despite its robustness, current limitations include manual input dependency and basic visualization restricted to the Tkinter environment. Future work could implement OCR or API-based data entry and enhanced interactive visualizations.EXPENSE-TRACKER.docx

This section explains the system’s modeling choices, analytical features, outcome reporting, and the underlying methodology, all structured to comply with IJPREMS paper formatting and academic standards.IJPREMS-Template.docx

4. RESULT AND DISCUSSION

The proposed Digital Expense Monitoring System was successfully implemented using Python, Tkinter, and CSV-based storage. The system was tested for accuracy, reliability, and usability. Expense entries across categories such as food, utilities, travel, and entertainment were correctly recorded with amount, date, category, and notes. Input validation prevented invalid entries, with messagebox alerts notifying users of incorrect formats or non-numeric values. File handling using the os module ensured all CSV files were created at the first run, preventing runtime errors. Data retrieval and display in the treeview table allowed dynamic viewing, scrolling, and sorting of records. Summary reports aggregated expenses on daily, monthly, and yearly bases, providing insights into spending patterns. User notifications confirmed successful data entry, deletions, and updates, enhancing usability. Performance tests showed efficient handling of large datasets without delays, and offline functionality was verified, with all features operating correctly without internet access. The results demonstrate that the system provides a lightweight, portable, and userfriendly solution for personal expense monitoring, with potential for future enhancements such as graphical visualizations and automated budgeting.



5. CONCLUSION

This paper presented the design and implementation of a Digital Expense Monitoring System using Python, Tkinter, and CSV-based data storage. The proposed system provides a lightweight, portable, and offline-capable solution for personal financial management. Through modular design, the system enables users to record, categorize, and review expenses efficiently, while input validation and notifications ensure data accuracy and usability. The results demonstrate that the system successfully maintains expense records, generates daily, monthly, and yearly summaries, and supports structured data visualization through treeview tables. Performance testing confirmed that the system can handle large datasets without delays, and its offline functionality allows uninterrupted use in various environments. Overall, the proposed system meets the objectives of simplicity, reliability, and user-friendliness. Future enhancements may include graphical visualizations, predictive analysis, automated budgeting, and cloud-based synchronization to further improve financial monitoring and decisionmaking.

6. REFERENCE

[1] L. Brown and M. Davis, “Real-time Expense Tracking: A Case Study in System Design and Implementation,” IEEE Transactions on Financial Technology, vol. 5, no. 3, pp. 234–248, 2021.

[2] V. T. Bhavani, “Implementation of an Expense Tracker using Python,” International Journal for Multidisciplinary Research (IJFMR), vol. 7, no. 5, pp. 112– 118, 2025.

[3] W. Khan, “An Expense Tracker Using Python, Django, and MongoDB,” International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), vol. 14, no. 8, pp. 45– 50, Aug. 2025.

[4] “Expense Tracker Report Using Python,” Gurukul Journal of Innovation and Development, vol. 4, no. 3, pp. 98–103, 2024.

[5] A. Sharma and R. Patel, “Expense Tracker Application,” International Journal of Novel Research and Development (IJNRD), vol. 9, no. 6, pp. 221–226, Jun. 2024.

[6] P. Singh and M. Gupta, “Expense Tracker Application,” International Journal of Research and Publication Review (IJRPR), vol. 5, no. 11, pp. 310– 316, Nov. 2024

[7] R. S. Thakur, A. Yildiz, and M. Kumar, “Expense Tracker Management System using Machine Learning,” Sigma Journal of Engineering and Applied Sciences, vol. 13, no. 1, pp. 77–84, Jan. 2025.