College management system

Submitted to the

saveetha institute of medical and technical sciences

By

P. Hemanth Reddy (192365046)

Under the guidance of

DR.R. GEETHA

professor

Computer Science Engineering (Cyber Security)

SAVEETHA SCHOOL OF ENGINEERING CHENNAI –602105 TAMIL NADU, INDIA

AUGUST 2024

Title: Making monthly expenditure using Python

Table of contents: -

Abstract:

To manage monthly expenditures using Python, you can create a script that tracks and analyses your spending

patterns. First, design a simple data structure, such as a dictionary or a CSV file, to store expense categories and amounts. Develop functions to input and categorise expenses, allowing for real-time updates. Use Python libraries like pandas for data manipulation and matplotlib for visualising spending trends. Implement features to calculate total expenses, compare them against budgets, and generate summaries. This approach provides a clear overview of your financial habits, helping you make informed decisions and adjustments to manage your budget effectively.

INTRODUCTION:

Managing monthly expenditures is crucial for maintaining financial health and achieving long-term goals. Python, with its versatile libraries and simple syntax, offers an effective solution for tracking and analysing spending habits. By leveraging Python's capabilities, you can automate the process of recording expenses, categorise them, and generate insightful reports. This not only saves time but also provides a clearer picture of where your money is going, enabling you to make more informed financial decisions.

Implementing a monthly expenditure tracker in Python involves creating a structured approach to data management. Using libraries such as pandas for handling data and matplotlib for visualisation, you can build a tool that categorises expenses, tracks trends, and compares spending against set budgets. This system can be tailored to individual needs, providing

personalised insights and helping users develop better budgeting practices. Ultimately, a Python-based expenditure tracker simplifies financial management and enhances your ability to control and optimise your spending.

Advantages:

Utilising Python for managing monthly expenditures streamlines the entire process, offering significant efficiency improvements. Automation of data entry and categorization reduces the risk of human error and saves time compared to manual tracking methods. Python's powerful libraries, such as pandas for data manipulation and matplotlib for visualisation, allow users to handle large datasets and perform complex analyses with ease. Customizable reports and visualisations provide clear insights into spending patterns, helping users identify trends and areas for improvement.

Moreover, Python's flexibility and extensibility make it an ideal tool for personal finance management. Integration with various data sources, such as CSV files or online financial APIs, ensures that your expenditure tracker can adapt to different needs and sources of data. Additionally, Python scripts can be easily updated or expanded to incorporate new features or adapt to changing financial goals. This adaptability not only enhances the accuracy of financial tracking but also empowers users to make more informed and strategic financial decisions.

Modules:

Modules Used:

In a Python-based monthly expenditure tracker, several key modules are utilised to streamline functionality and enhance user experience. The panda's module is essential for data manipulation and analysis, offering powerful tools to handle expense records, categorise transactions, and perform budget calculations efficiently. numpy complements pandas by providing advanced numerical operations and transformations, crucial for handling complex calculations and data processing.

For visualising spending trends, matplotlib and seaborn are invaluable. matplotlib provides a wide range of plotting options, including bar charts, line graphs, and pie charts, while seaborn enhances these visualisations with more sophisticated statistical graphics and aesthetic improvements. The csv module simplifies the import and export of data, making it easy to manage expense records and integrate them with other tools or platforms.

datetime is used to manage and format date-related information, ensuring accurate tracking of expenses over time. For creating a more interactive user

experience, tkinter or PyQt can be employed to develop a graphical user interface, allowing users to input, view, and analyse their expenditures in a user-friendly environment. These modules collectively contribute to a comprehensive, efficient, and intuitive expenditure management system, tailored to meet individual financial tracking needs.

Key Components:

In a Python-based monthly expenditure tracker, key components include

1. Data Storage: Utilises data structures like pandas Data Frames or CSV files to store and manage expense records and budget information.

2. Expense Input: Provides functionality for users to enter and categorise expenses, often through a form or input interface.

3. Data Processing: Leverages pandas and numpy for sorting, filtering, and aggregating expense data to generate meaningful insights.

4. Budget Calculation: Includes functions to compare actual expenses against set budgets and compute variances or overspending.

5. Visualisation: Employs matplotlib or seaborn to create charts and graphs that illustrate spending patterns and trends.

6. Date Management: Uses the datetime module to handle and format dates, ensuring accurate tracking and reporting of expenses over time.

7. Reporting: Generates summaries and detailed reports, providing an overview of spending habits and financial status.

8. User Interface: Implements a graphical user interface with tkinter or PyQt to facilitate interactive data entry and visualisation.

9. Data Export/Import: Allows for the export and import of data using CSV or other formats to ensure compatibility with other tools or systems.

10. Error Handling: Incorporates validation checks and error handling to ensure data integrity and a smooth user experience.

Tools Description:

1. Pandas :

A powerful library for data manipulation and analysis, pandas provide data structures like Data Frames that facilitate the easy handling, processing, and analysis of tabular data. It is ideal for organising expense records and performing calculations such as totals and averages.

2. Numpy:

This library supports numerical operations and array manipulations, enhancing the capabilities of pandas. It is useful for performing mathematical operations on large datasets and transforming data efficiently.

3. Matplotlib:

A comprehensive plotting library that enables the creation of static, animated, and interactive visualisations. It helps users generate various types of charts and graphs to visualise spending patterns and financial trends.

4. Seaborn:

Built on top of matplotlib, seaborn provides a high-level interface for drawing attractive and informative statistical graphics. It simplifies the creation of complex visualisations, such as heatmaps and pair plots, enhancing the presentation of financial data.

4. CSV:

A module for reading from and writing to CSV files, facilitating the import and export of data. It is essential for managing expense records in a format that can be easily shared or integrated with other applications.

5. Datetime:

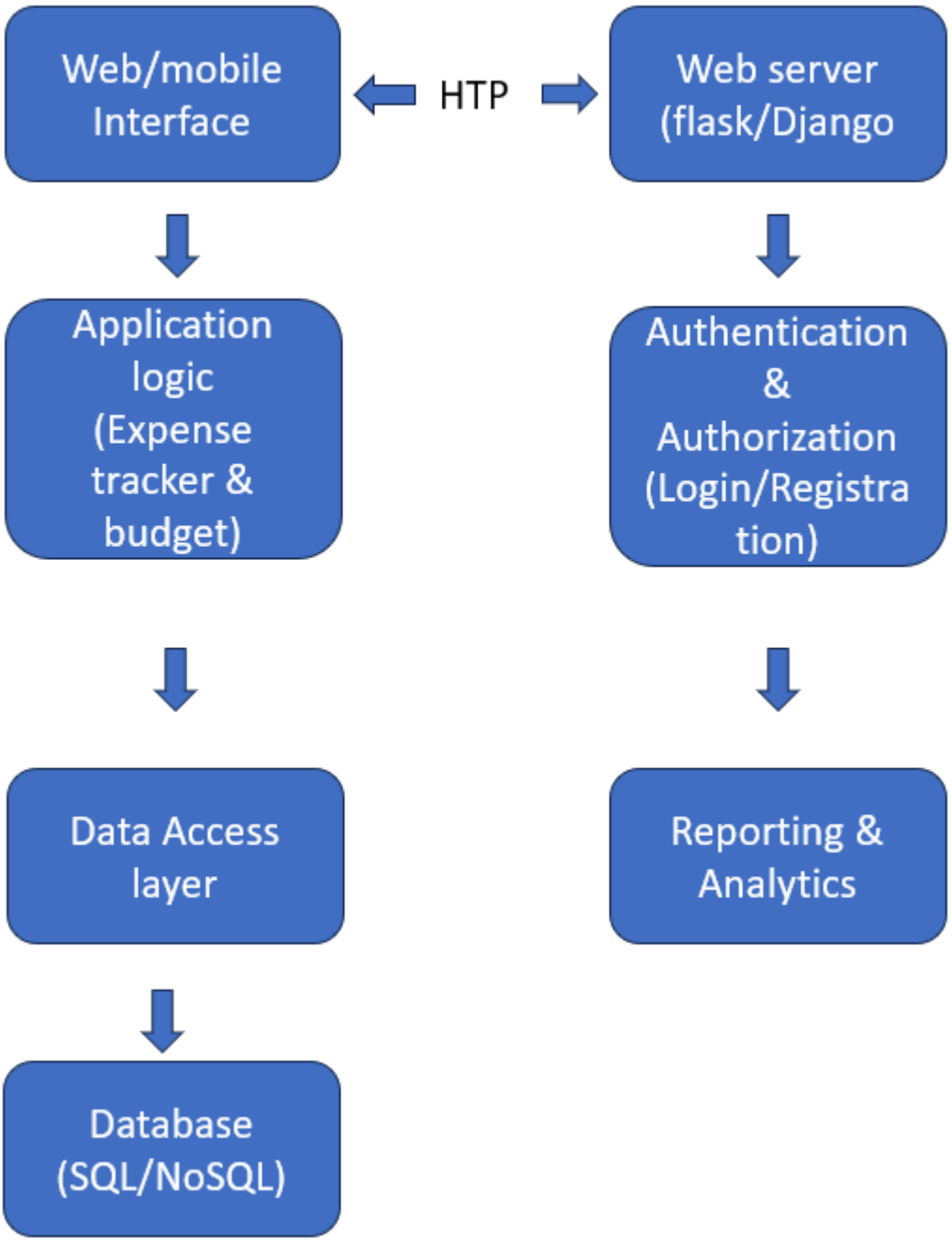
A module for handling dates and times, enabling accurate tracking of expenses over different periods. It assists in formatting, calculating date ranges, and ensuring the proper organisation of time-related data.

6.SQLite:

- Purpose: SQLite is a lightweight, disk-based database that doesn't require a separate server process. It's ideal for small to medium-sized applications where a full-fledged database system might be overkill. It supports SQL queries and transactions.

- **Architecture diagram**:

Block diagram:



Source code:

```
import json
import os

# Define file path for saving data
DATA_FILE = 'budget_data.json'

def load_data():
    """Load data from the JSON file."""
    if os.path.exists(DATA_FILE):
    with open(DATA_FILE, 'r') as file:
        return json.load(file)
    return {"income": [], "expenses": []}

def save_data(data):
    """Save data to the JSON file."""
    with open(DATA_FILE, 'w') as file:
    json.dump(data, file, indent=4)

def add_income(data):
    """Add income to the budget."""
    source = input("Enter income source: ")
```

```
        amount = float(input("Enter income amount: "))
    data['income'].append({"source": source,
"amount": amount})
        print(f"Income from '{source}' of amount
${amount:.2f} added.")


def add_expense(data):
        """Add expense to the budget."""
        description = input("Enter expense description:
")
        amount = float(input("Enter expense amount:
"))
    data['expenses'].append({"description":
description, "amount": amount})
        print(f"Expense '{description}' of amount
${amount:.2f} added.")


def view_summary(data):
        """View the budget summary."""
        total_income = sum(item['amount'] for item in
data['income'])
        total_expenses = sum(item['amount'] for item in
data['expenses'])
        balance = total_income - total_expenses
```

```
        print("\n--- Budget Summary ---")
        print(f"Total Income: ${total_income:.2f}")
        for income in data['income']:
        print(f"  {income['source']}:
${income['amount']:.2f}")

        print(f"Total Expenses: ${total_expenses:.2f}")
        for expense in data['expenses']:
        print(f"  {expense['description']}:
${expense['amount']:.2f}")

        print(f"Balance: ${balance:.2f}")

def main():
        data = load_data()

        while True:
        print("\n--- Budget Planner ---")
        print("1. Add Income")
        print("2. Add Expense")
        print("3. View Summary")
        print("4. Exit")
```

```
        choice = input("Choose an option (1-4): ")

        if choice == '1':
            add_income(data)
        elif choice == '2':
            add_expense(data)
        elif choice == '3':
            view_summary(data)
        elif choice == '4':
            save_data(data)
            print("Data saved. Exiting...")
            break
        else:
            print("Invalid option. Please choose a
number between 1 and 4.")

if _name_ == "_main_":
        main()
```

Output:

--- Budget Planner ---

1. Add Income

2. Add Expense

3. View Summary

4. Exit

Choose an option (1-4): 1

Enter income source: salary

Enter income amount: 1200

Income from 'salary' of $1200.00 added.

--- Budget Planner ---

1. Add Income

2. Add Expense

3. View Summary

4. Exit

Choose an option (1-4): 3

--- Budget Summary ---

Total Income: $1200.00

  salary: $1200.00

Total Expenses: $0.00

Balance: $1200.00

--- Budget Planner ---

1. Add Income

2. Add Expense

3. View Summary

4. Exit

Choose an option (1-4): Exit

Future enhancement:

1. AI-Driven Insights:

Integrate machine learning algorithms to analyse spending patterns and provide predictive insights or personalised financial advice based on historical data.

2. Mobile Integration:

Develop a mobile app or integrate with existing mobile platforms to allow users to track expenses on-the-go, using features like camera-based receipt scanning.

3.Cloud Syncing:

Implement cloud-based storage and synchronisation to enable users to access and update their expense records from multiple devices securely.

4. Automatic Data Import:

Enhance the system to automatically import transaction data from bank accounts or financial institutions through APIs, reducing manual entry.

5.Advanced Budgeting Tools:

Add features for setting and tracking multiple budgets, including savings goals and debt reduction plans, with visual progress indicators.

6. Expense Categorization:

Use natural language processing (NLP) to automatically categorise expenses from transaction descriptions, improving accuracy and reducing manual input.

7. Interactive Dashboards:

Create more dynamic and interactive dashboards with real-time data updates, allowing users to drill down into specific categories and time periods.

8. Voice Command Integration:

incorporate voice recognition technology to allow users to add and manage expenses using voice commands for a more hands-free experience.

9. Customizable Alerts:

Develop a system for setting customizable alerts and notifications for budget limits, unusual spending patterns, or upcoming bills.

10. Data Security Enhancements:

Implement advanced security measures such as encryption, multi-factor authentication, and secure data storage to protect sensitive financial information.

Conclusion:

In conclusion, developing a Python-based monthly expenditure tracker offers a robust and efficient solution for managing personal finances. By

leveraging powerful libraries like pandas and numpy, users can handle and analyse their financial data with precision. Visualisation tools such as matplotlib and seaborn enable clear representation of spending trends, making it easier to understand and adjust financial habits. The integration of modules like datetime for date management and csv for data handling ensures that the system is versatile and user-friendly.

Future enhancements promise to significantly expand the functionality of the tracker. Incorporating artificial intelligence can provide predictive insights and personalised financial advice, while mobile integration and cloud syncing ensure that users can manage their finances conveniently across different platforms. Features like automatic data import, advanced budgeting tools, and interactive dashboards will further enrich the user experience, making financial tracking more intuitive and responsive. Additionally, voice command integration and customizable alerts can streamline expense management and improve accessibility.

Data security remains a critical consideration, and implementing advanced measures will safeguard sensitive financial information. Overall, a Python-based expenditure tracker, with its potential for continuous improvement, represents a valuable tool for individuals seeking to gain better control over their finances, optimise their spending, and

achieve their financial goals with greater confidence and ease.