A project

Automated Business Solution with QR code Integration

CSE 200: Software Development Project **II**

SUBMITTED BY :

Name	ID	Intake
Md Soykot Ahamed	**22235103247**	51
Ali Akbar Rakib	**22235103237**	51
Utsob	**22235103256**	51
MD Jubayer Hossain	**22235103236**	51
MD Sadat Islam	**22235103269**	51

SUPERVISED BY:

Samin Yasar

Lecturer
Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)

25th, December, 2024

Abstract

This project is a comprehensive **Business Management Application** designed to streamline customer and product data management while providing advanced features like analysis and reporting. Built with Python and Tkinter, the application allows users to manage customer information, generate QR codes and PDFs for products, and perform data analysis using Google Sheets integration. The app provides tools for searching customer data by ID, generating expense reports, and saving them as PDF files. It also offers functionality to create QR codes containing product details, which can be used for marketing or inventory purposes.

By integrating Google Sheets for data storage, the application ensures easy accessibility and scalability. With additional features like customer discount analysis and product estimation trends, businesses can gain valuable insights into sales performance and customer behavior. This user-friendly tool is ideal for small to medium-sized businesses looking to improve data organization, automate report generation, and enhance decision-making processes.

Approval

We do hereby declare that the project works presented here entitled as, "**Automated Business Solution with QR code Integration**" are the results of our own works carried out by MD Soykot Ahmed (ID-22235103247), Ali Akbar Rakib (ID-22235103237),Utsob (ID-22235103256), MD Jubayer Hossain (ID-22235103236), MD Sadat islam (ID-22235103269) under my supervision. I further declare that no part of this project has been submitted else where for the requirements of any degree, award diploma, or any other purposes except for publications. I further certify that the dissertation meets the requirements and standards for the degree of Bachelor of Science in Computer Science and Engineering.

Supervisor:
Samin Yasar (Lecturer)
Department of Computer Science & Engineering
Bangladesh University of Business and Technology
Dhaka, Bangladesh

Chairman:
Md. Saifur Rahman
Assistant Professor & Chairman
Department of Computer Science & Engineering
Bangladesh University of Business and Technology
Dhaka, Bangladesh

Acknowledgement

We express our heartfelt gratitude to everyone who contributed to the success of the "Automated Business Solution with QR Code Integration" project. Special thanks to our project supervisor, Samin Yasar (Lecturer), for her invaluable guidance and constructive feedback throughout the development process. We are also deeply grateful to the faculty members of BUBT for their mentorship, which played a pivotal role in achieving our project goals. Additionally, we extend our appreciation to our peers and colleagues for their assistance, feedback, and encouragement, fostering an environment of collaboration and innovation. This project stands as a testament to the collective efforts and support of all involved, enhancing our learning experience and contributing to advancements in automated business solutions and QR code integration.

List of Figures

Table of Contents

Chapter 1
INTRODUCTION

1.1. Introduction

This project, "Automated Business Solution with QR Code Integration," is a comprehensive system designed to simplify and enhance business processes. The application focuses on managing customer and product data efficiently, providing tools for analysis, and automating tasks like generating reports and QR codes. Leveraging Python and Tkinter for the user interface and Google Sheets for data storage, the system ensures seamless integration and accessibility. By incorporating advanced features like data analysis and reporting, the project aims to empower businesses to make informed decisions and streamline their operations.

1.2. Objectives

The primary goal of this project is to develop a comprehensive and user-friendly platform that simplifies business data management while automating key processes. The system is designed to cater to small and medium-sized businesses by offering the following functionalities:

- **Efficient Data Management:** Users can easily add, search, edit, and delete customer and product information. This ensures that all business data is organized and up-to-date, reducing the chances of errors or inconsistencies.
- **In-depth Analysis:** The platform provides tools for analyzing customer behavior, tracking discount usage, and evaluating overall business performance. These insights help businesses make informed decisions and identify areas for improvement.
- **Automated QR Code and PDF Generation:** Businesses can generate QR codes for products or customers, scan QR codes to update information instantly, and create PDF reports for record-keeping or sharing. This reduces manual workload and improves accuracy.
- **Real-time Google Sheets Integration:** By leveraging the Google Sheets API, the system ensures that all data is stored securely and updated in real time. This makes data accessible anytime and provides a scalable solution for growing businesses.
- **User-friendly Interface:** The platform is designed to be intuitive and easy to navigate, allowing users with minimal technical skills to operate it effectively.

This project aims to enhance efficiency by automating repetitive tasks, reducing manual data entry, and providing actionable insights. By streamlining operations, the system empowers businesses to focus on growth and strategic decision-making while maintaining accurate and organized records.

1.3. Motivation

The motivation behind this project stems from the growing need for businesses to adopt digital tools for improved efficiency and competitiveness. Manual processes for managing customer and product data are often error-prone and time-consuming. By integrating automated solutions like QR code generation and data analytics, businesses can save time, reduce errors, and gain valuable insights into their operations. Additionally, the use of Google Sheets ensures scalability and easy collaboration, further motivating the adoption of this solution.

1.4. Related Works

While many existing business management solutions focus on automation, they often fall short in providing flexibility, customization, and seamless integration with other tools. Commonly used platforms like Microsoft Excel or standalone inventory systems offer limited support for automation, real-time updates, or advanced analysis. This project fills these gaps by combining cutting-edge QR code technology, Google Sheets integration, and an intuitive interface to create a versatile and accessible platform for comprehensive business management.

How This Project Stands Out:

1. **Customization:** Unlike rigid off-the-shelf tools, this platform allows businesses to tailor the solution to their specific needs, making it suitable for diverse industries and workflows.
2. **Real-time Integration:** By leveraging Google Sheets API, businesses benefit from scalable, cloud-based data storage with real-time updates, accessible from anywhere.
3. **Automation with QR Codes:** QR code technology is seamlessly integrated into the system, enabling automated data updates, quick retrieval, and enhanced operational efficiency.
4. **Ease of Use:** The user-friendly interface ensures that even users with minimal technical expertise can navigate and utilize the platform effectively.

Key Use Cases and Applications:

1. **Car Rental Management Systems:**
    - QR codes simplify the booking process by linking to customer profiles or rental agreements.
    - Automated check-in/check-out processes improve efficiency and reduce waiting times.
    - QR codes can track vehicle availability and maintenance history in real time.
2. **Inventory Management Systems:**
    - Retail stores and warehouses use QR codes to tag items for easy tracking and stock updates.
    - This approach eliminates manual counting errors and ensures inventory levels are optimized.
    - Generate actionable insights like demand forecasting and reorder alerts for operational efficiency.
3. **Point of Sale (POS) Systems:**
    - QR codes enable secure and contactless payment solutions.
    - Quickly retrieve customer-specific discounts or loyalty rewards, improving the checkout experience.
    - Integrated sales tracking helps businesses analyze performance and identify trends.
4. **Mobile Payment Systems:**
    - QR codes facilitate fast, secure transactions across multiple payment platforms.
    - Widely applicable in retail, e-commerce, and food delivery, ensuring a seamless payment experience for customers.
5. **Supply Chain & Logistics:**
    - QR codes enable product tracking across the supply chain, offering visibility from production to delivery.
    - Real-time updates on shipment locations and delivery schedules improve operational accuracy.
    - Reduces errors in inventory transfers and ensures product traceability.
6. **Hospitality Industry:**
    - Hotels and restaurants utilize QR codes for express check-ins, accessing menus, and placing orders.
    - QR codes can be used for event management, allowing attendees to scan for entry or view schedules and updates.
    - Enhances customer satisfaction by reducing wait times and providing a seamless experience.

7. **Customer Relationship Management (CRM) Systems:**
    - QR codes allow businesses to personalize customer interactions by linking to profiles, purchase history, or preferences.
    - Used in marketing campaigns, QR codes can provide customers with exclusive offers or instant access to support.
    - Streamlines service delivery, ensuring consistent and high-quality customer interactions.
8. **Healthcare Industry:**
    - QR codes can be used for patient check-ins, medical record access, and appointment scheduling.
    - Streamlines billing processes by linking to insurance details or invoices.
    - Enhances data security and accessibility while improving overall patient experience.
9. **Education Sector:**
    - Educational institutions can use QR codes for attendance tracking, accessing digital resources, or managing events.
    - QR codes simplify library management by linking to books or borrower records.
    - Promotes digital learning by linking to multimedia content or online courses.
10. **Real Estate and Property Management:**

- QR codes streamline property viewings by linking to virtual tours or property details.
- Facilitates tenant management, allowing quick access to lease agreements or maintenance requests.
- Enhances transparency and accessibility in property dealings.

Chapter 2
Background

2.1 Registration

The registration module enables businesses to register customer details securely. The user interface is designed to be intuitive, with input fields for customer ID, name, and other essential details. Data entered during registration is stored in Google Sheets, ensuring easy access and scalability.

2.2 Login

The login module ensures secure access to the system. Users can log in using credentials, granting access to features like customer search, report generation, and data analysis. The interface emphasizes simplicity and security to enhance user experience.

2.3 Feature

The application offers diverse features, including:

- QR code generation for products.
- PDF report creation for customer data and expense analysis.
- Visualization of sales trends and customer spending patterns.
- Easy access to all customer and product data.
- These features aim to streamline business operations and provide actionable insights.

2.4 Information Search

The information search module allows users to search for specific customer details using a unique ID. The results display comprehensive data, including individual expenses and product details, with options to save reports as PDFs.

2.5 Database Design

The database design is centered around Google Sheets, ensuring real-time updates and accessibility. The structured format includes tables for customers, products, expenses, and sales data, enabling efficient data retrieval and analysis.

-- Customers Table

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100) NOT NULL,
    Email VARCHAR(100),
    Phone VARCHAR(15),
    Address TEXT,
    RegistrationDate DATE DEFAULT CURRENT_DATE
);

-- Products Table
CREATE TABLE Products (
    ProductID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100) NOT NULL,
    Category VARCHAR(50),
    Price DECIMAL(10, 2),
    Stock INT DEFAULT 0
);

-- Expenses Table
CREATE TABLE Expenses (
    ExpenseID INT PRIMARY KEY AUTO_INCREMENT,
    Description VARCHAR(255),
    Amount DECIMAL(10, 2) NOT NULL,
    ExpenseDate DATE DEFAULT CURRENT_DATE
);

-- Sales Table
CREATE TABLE Sales (
    SaleID INT PRIMARY KEY AUTO_INCREMENT,
    CustomerID INT,
    ProductID INT,
    Quantity INT NOT NULL,
    SaleDate DATE DEFAULT CURRENT_DATE,
    TotalPrice DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);


SELECT Customers.Name, SUM(Sales.TotalPrice) AS TotalSpent
```

```
FROM Customers
JOIN Sales ON Customers.CustomerID = Sales.CustomerID
GROUP BY Customers.Name;

SELECT Name, Stock
FROM Products
WHERE Stock < 10;

SELECT SUM(Amount) AS TotalExpenses
FROM Expenses;
```

Explanation of the Code:

1. **Customers Table:**
    - Stores customer details such as their name, email, phone number, and address.
    - Includes RegistrationDate, which automatically captures when the customer was added.
    - The primary key (CustomerID) uniquely identifies each customer.
2. **Products Table:**
    - Contains product information, including the product name, category, price, and stock quantity.
    - The Stock column helps track inventory.
    - The primary key (ProductID) uniquely identifies each product.
3. **Expenses Table:**
    - Records business expenses with details like description, amount, and date.
    - The primary key (ExpenseID) ensures each expense entry is unique.
4. **Sales Table:**
    - Tracks each sale, including which customer bought which product and in what quantity.
    - Uses foreign keys (CustomerID and ProductID) to link sales data to customers and products.
    - Automatically calculates the total price (TotalPrice) based on the product price and quantity.

Comparison with Google Sheets:

- **SQL Advantage:**
  - Structured and normalized data storage with powerful query capabilities for complex analysis.
  - Strong integrity rules (e.g., foreign keys) ensure data consistency.
- **Google Sheets Advantage:**
  - Real-time collaboration and accessibility.
  - Easier for non-technical users to view and update data without knowledge of SQL.

In your project, Google Sheets can simulate these relational structures by using separate tabs for tables and linking them with formulas or scripts. Let me know if you need the Google Sheets equivalent of this design!

Chapter 3
System Analysis & Design

3.1 UI Design

The UI design prioritizes simplicity and functionality. Tkinter widgets are used to create a clean layout with buttons, entry fields, and labels for seamless interaction. Each module is designed for specific tasks, ensuring clarity and ease of use.

Let's walk through a simple example of how to use Tkinter to create a basic window with a button and a label.

Step-by-Step Explanation:

1. **Import Tkinter**: To use Tkinter, you need to import it. Tkinter is usually included in standard Python distributions, so no additional installation is needed.
2. **Create a Window**: You will create a main window where all the widgets (like buttons, labels, etc.) will be placed.
3. **Add Widgets**: Widgets are the elements like buttons, text fields, labels, etc. You can add them to the window.
4. **Define Functions**: Functions are often needed to respond to events like clicking a button. In this case, we'll define a function to update the label when the button is clicked.
5. **Main Loop**: The Tkinter event loop (mainloop()) keeps the window running and waits for user interactions.

Here is an example code that demonstrates these steps:

Example Code:

```
import tkinter as tk

# Create a function to change the label text
def change_text():
    label.config(text="Hello, Tkinter!")

# Create the main window
root = tk.Tk()
root.title("Tkinter Example")  # Set window title
```

```
root.geometry("300x200")  # Set window size

# Create a label widget
label = tk.Label(root, text="Click the button below", font=("Helvetica", 14))
label.pack(pady=20)  # Pack the label into the window with some padding

# Create a button widget
button = tk.Button(root, text="Click Me", command=change_text, font=("Helvetica",
14))
button.pack(pady=10)  # Pack the button into the window

# Start the Tkinter event loop
root.mainloop()
```

Explanation of the Code:

- **Import Tkinter**:
  ```
  import tkinter as tk
  ```

  This imports the Tkinter library, which provides all the GUI elements.

- **Create a Function**:

  ```
  def change_text():
      label.config(text="Hello, Tkinter!")
  ```

  This function changes the text of the label when called. We use the config() method to update the text of the label.

- **Create the Main Window**:

  ```
  root = tk.Tk()
  root.title("Tkinter Example")
  root.geometry("300x200")
  ```

  root = tk.Tk() creates the main window. title() sets the window's title, and geometry() sets the window's size to 300x200 pixels.

- **Create a Label**:

  ```
  label = tk.Label(root, text="Click the button below", font=("Helvetica", 14))
  label.pack(pady=20)
  ```

The Label widget displays text. label.pack() places the label inside the window. The pady=20 adds vertical padding to space it out.

- **Create a Button**:

  button = tk.Button(root, text="Click Me", command=change_text, font=("Helvetica", 14))
  button.pack(pady=10)

  The Button widget is created with the text "Click Me". When the button is clicked, the change_text() function is called (command=change_text). The pack() method places the button in the window.

- **Main Event Loop**:

  root.mainloop()

  This starts the Tkinter event loop, which waits for the user to interact with the window (e.g., clicking the button). The program will continue running until the window is closed.

Running the Program:

1. When you run this program, a window will appear with the text "Click the button below".
2. When you click the "Click Me" button, the label will update to say "Hello, Tkinter!".

3.2 E-R Model Diagram

The E-R model diagram illustrates the relationships between key entities: Customers, Products, Sales, and Expenses. Each entity is represented with attributes like customer ID, product name, quantity sold, and expense, ensuring a clear understanding of data flow.

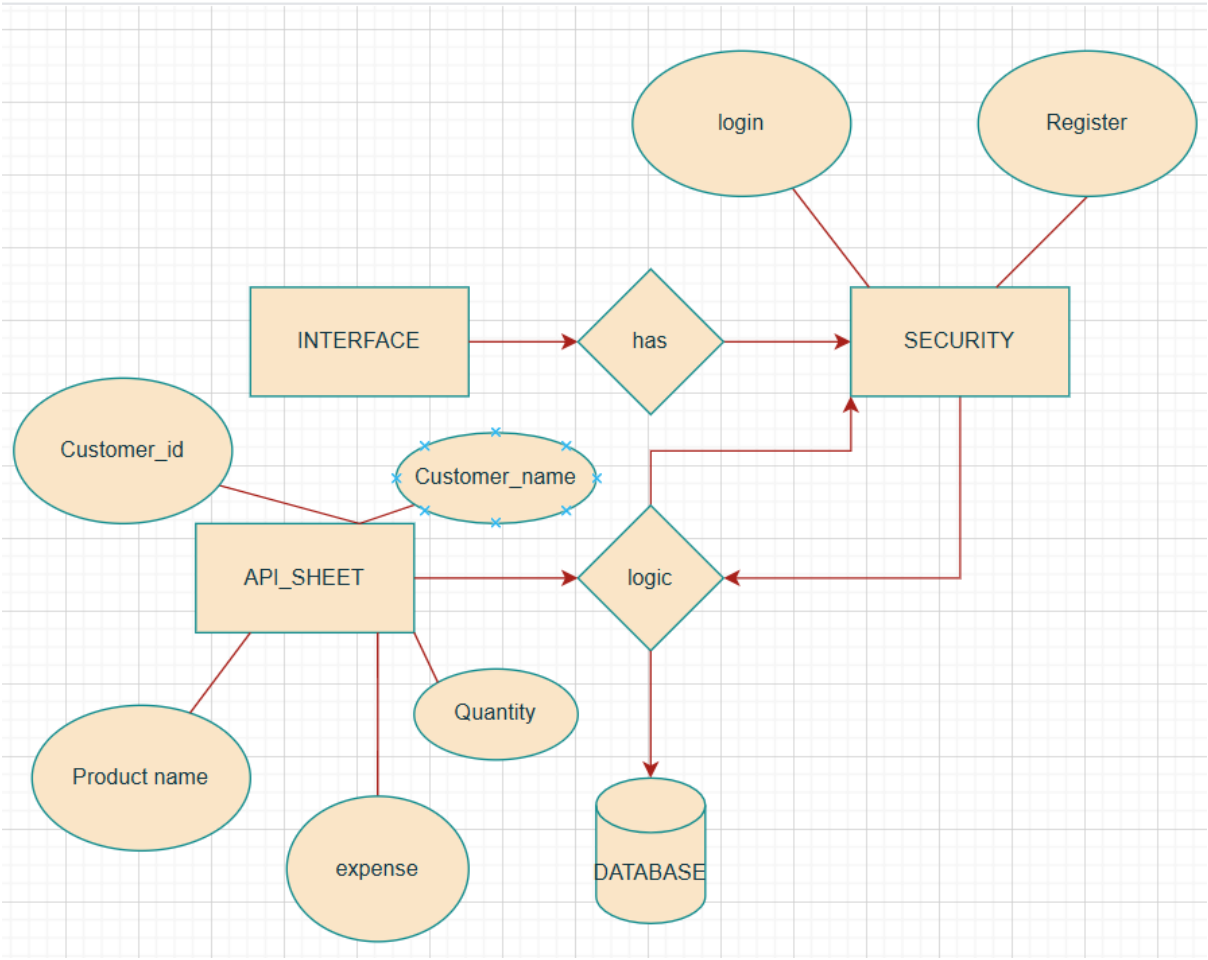

Fig 3.1 : ER Diagram

Chapter 4
IMPLEMENTATION

4.1 Technology

The project uses Python for backend development and Tkinter for the user interface. Integration with Google Sheets is achieved through the gspread library, enabling real-time data synchronization. Additional technologies include:

- QR Code generation using the qrcode library.
- PDF report creation using FPDF.

Data visualization using pandas and matplotlib.
These technologies ensure robust functionality and scalability.

1.Python (Tkinter):

- Used for the core programming and GUI development.
- Provides a user-friendly interface for customers and administrators.

2. MySQL:

- Relational database system for storing and managing data.
- Ensures secure and scalable data handling.

3. JavaScript:

- Adds interactivity to the system, especially for web-based features.
- Facilitates QR code rendering in the interface.

4. Matplotlib:

- Used to create charts and graphs for data visualization.
- Helps administrators track performance metrics.

5. Google Sheets API:

- Enables synchronization of data to Google Sheets.

- Provides real-time access to logs and transaction records.

4.2 Result

The implemented system successfully manages customer and product data, generates QR codes and reports, and provides actionable insights through data analysis. Users can search for customers, visualize sales trends, and save expense reports, making it a valuable tool for businesses.

QR Code:

Fig 4.1 : Generated QR Code

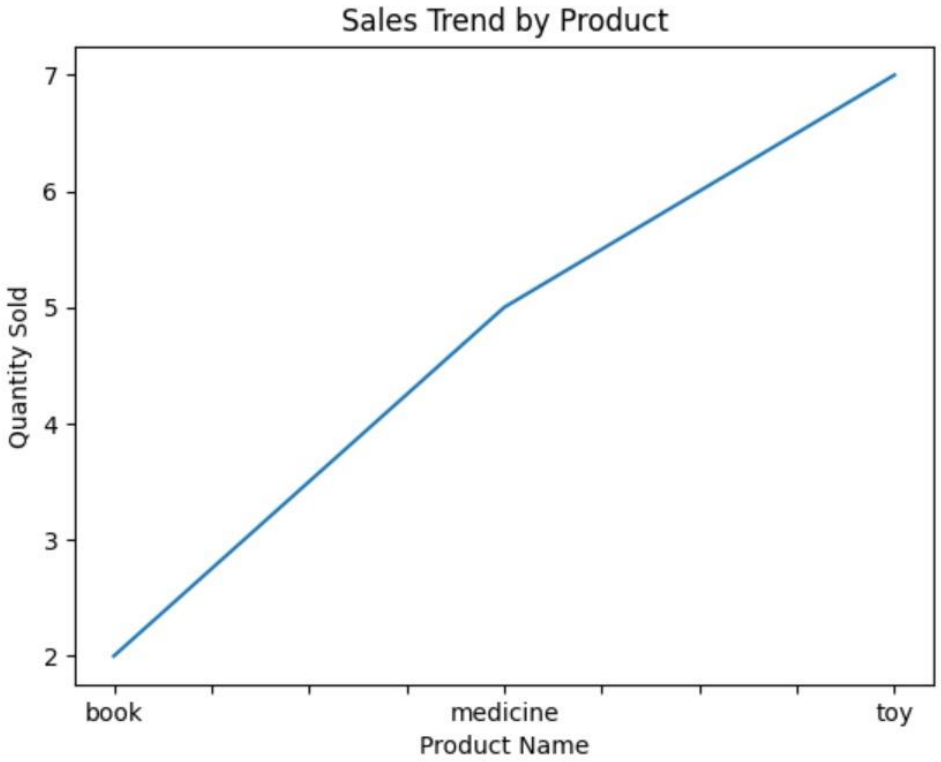

Fig 4.2 : Sales Analysis Graph

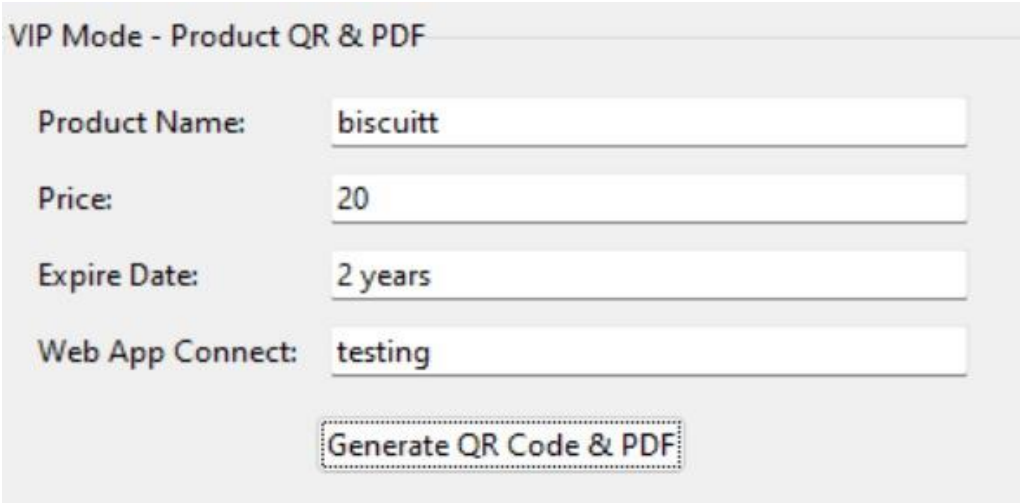

Fig 4.3 : VIP Section

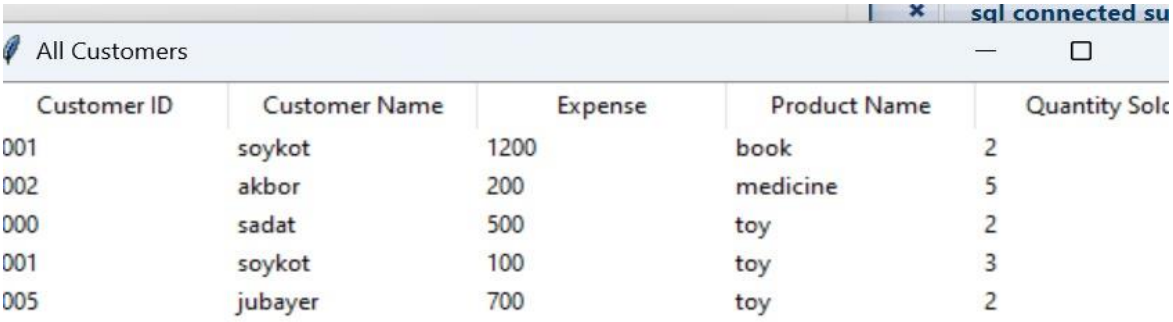

Fig 4.4 : Temporary Data

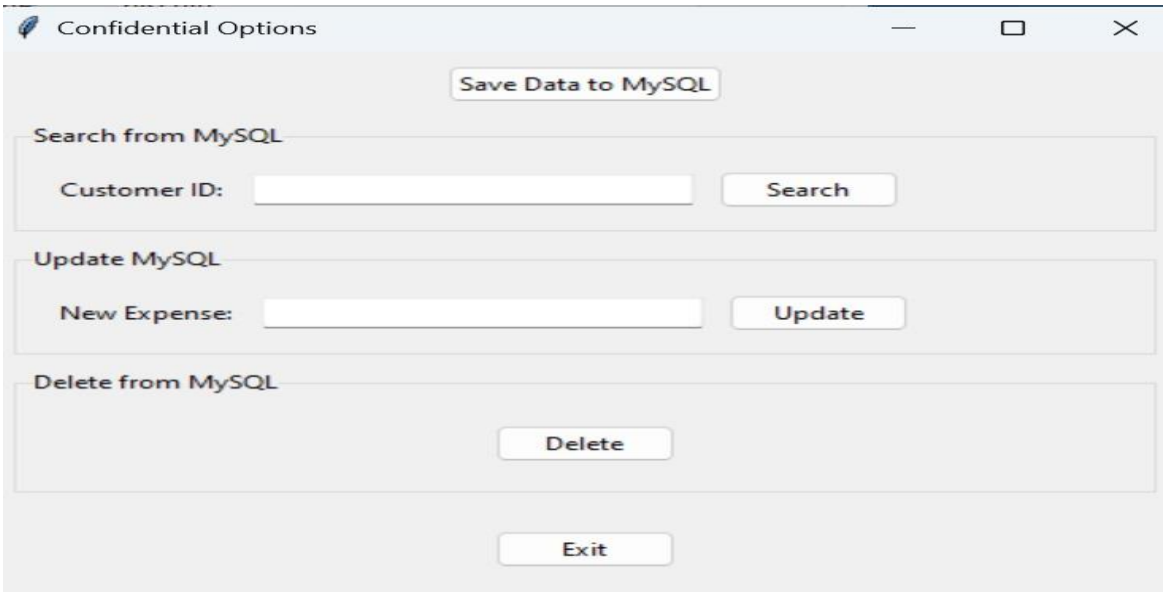

Fig 4.5 : SQL Section

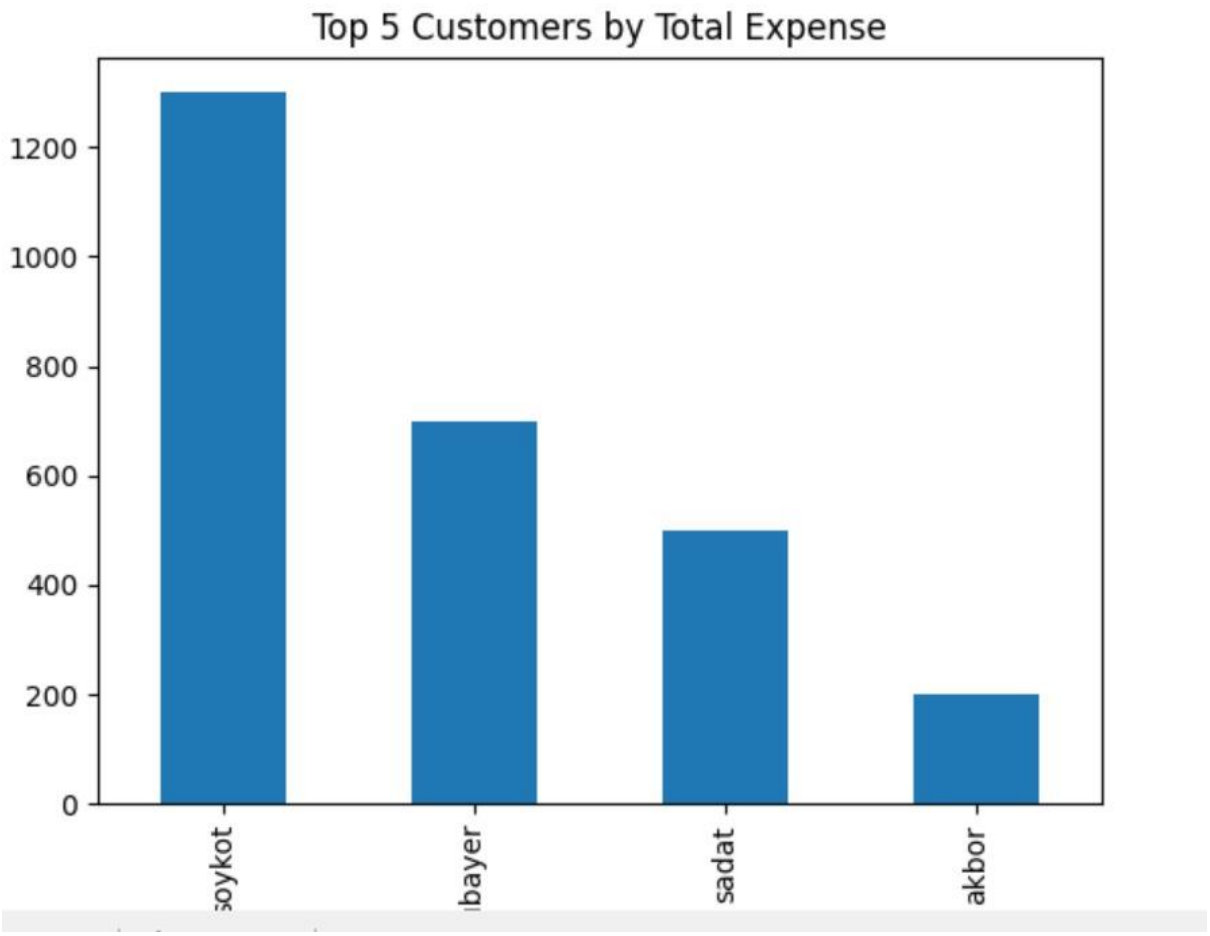

Fig 4.6 : Discount Analysis

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	customer_name	product_name	expense	quantity_sold
1002	akbor	medicine	200	10
1000	sadat	toy	500	4
1005	jubayer	toy	500	4
1001	soykot	book	1200	2
1001	soykot	toy	100	3

Fig 4.7 : Parmanent Data (SQL)

ERROR HANDLE:

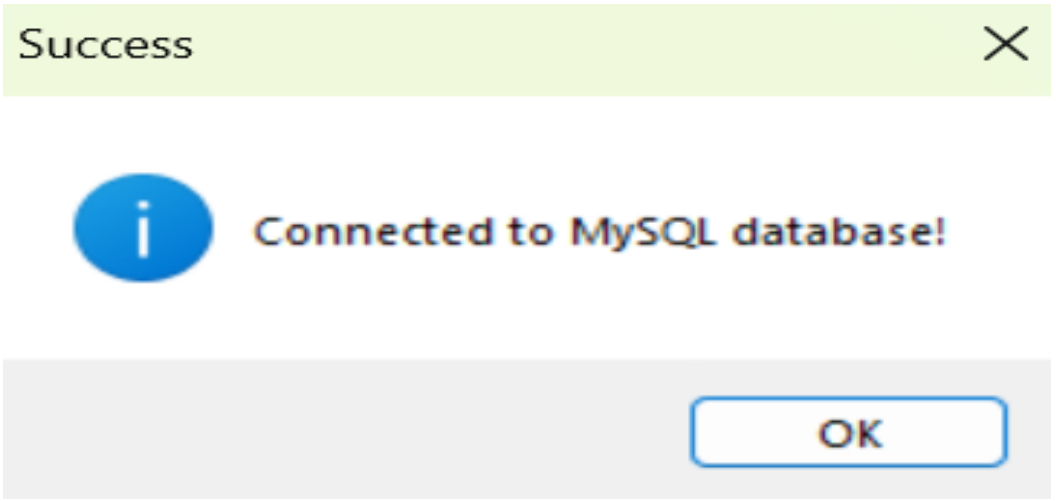

Fig 4.8 : MySQL Connection Check

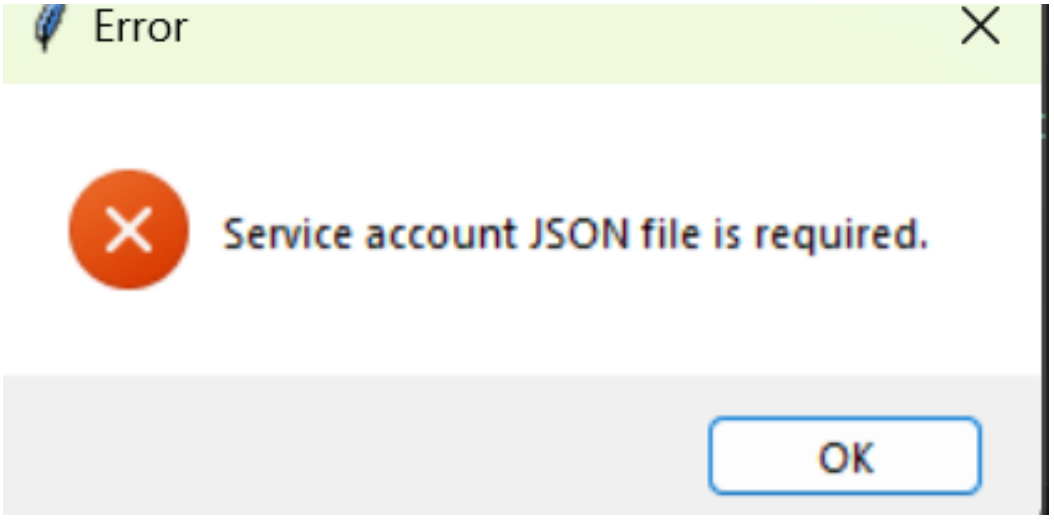

Fig 4.9 : Api Connection Failed

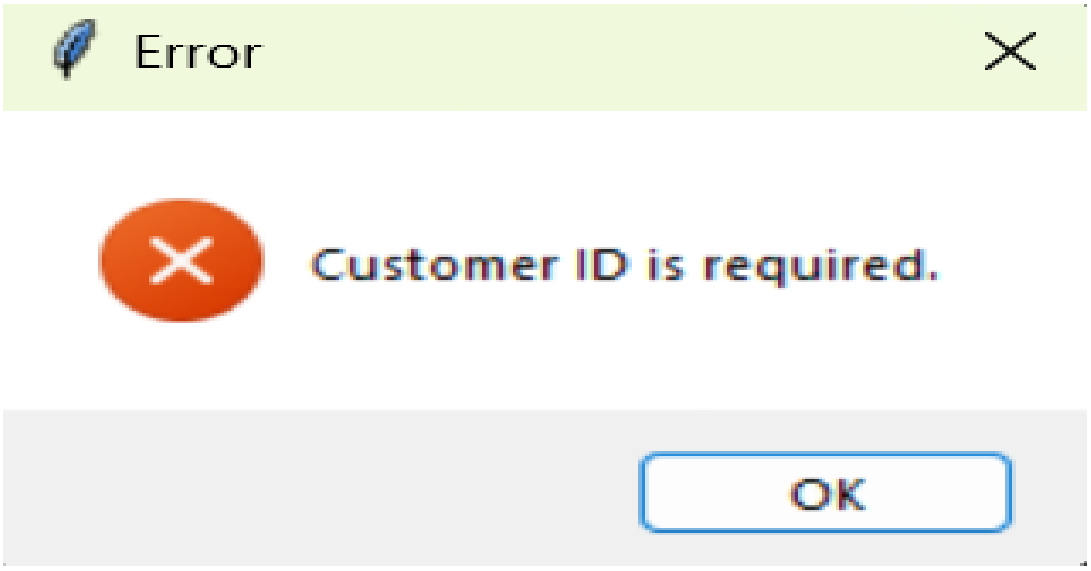

Fig 4.10 : Wrong Input

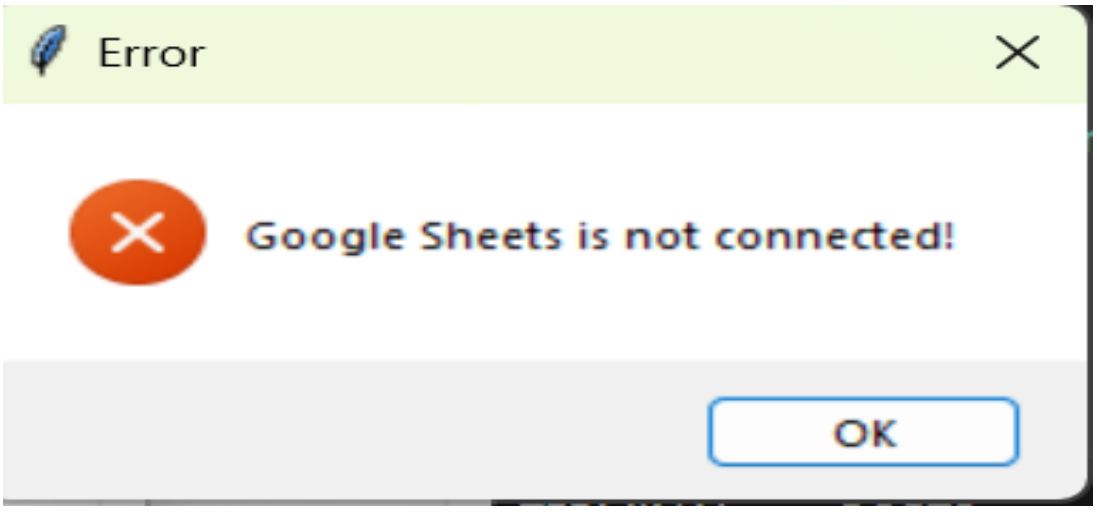

Fig 4.11 : Sheet Connection Failed

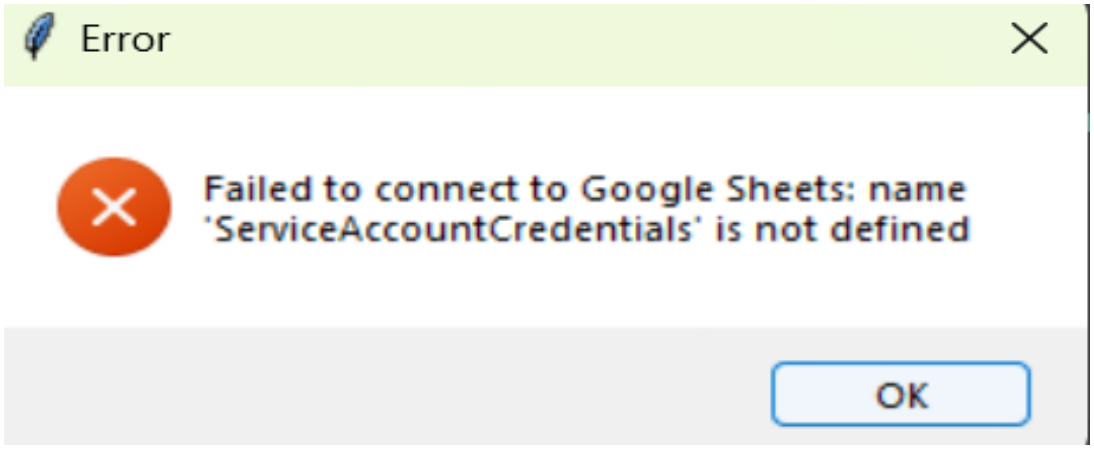

Fig 4.12 : Api Miss Match

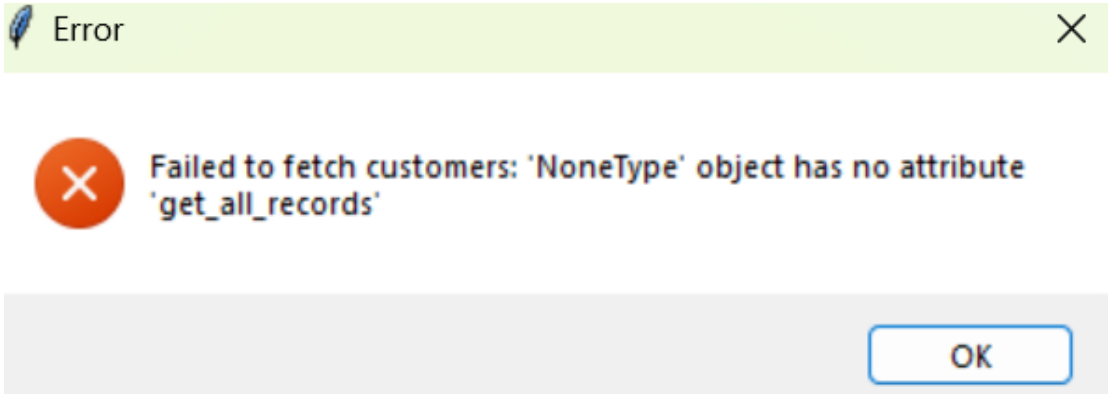

Fig 4.13 : Internal Error

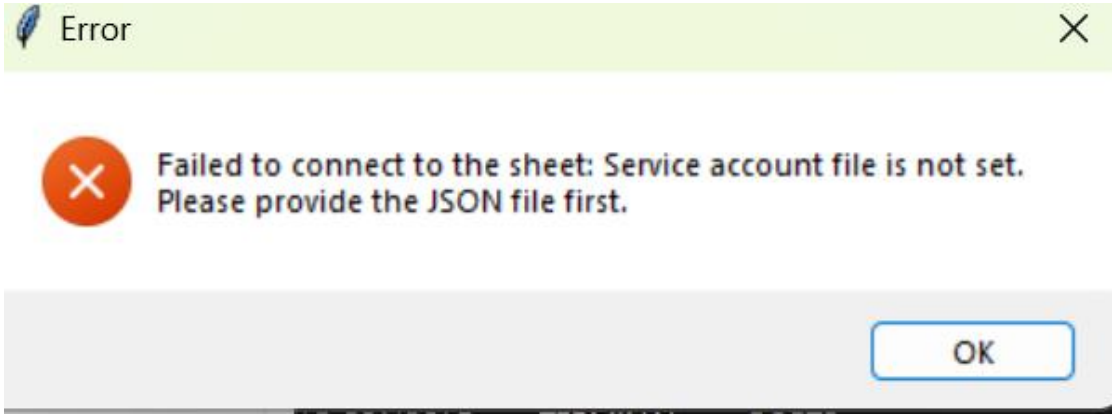

Fig 4.14 : JSON File Miss Match

Chapter 5
CONCLUSION

5.1 Future Work

As the system evolves, several enhancements can be implemented to broaden its capabilities and make it even more efficient and user-friendly. These improvements include:

- **Integration with Cloud Databases:**
  - By integrating with advanced cloud database solutions like Firebase, AWS RDS, or Google Cloud SQL, the system can handle larger datasets and ensure seamless scalability as business needs grow.
  - Cloud databases also offer robust security features and real-time synchronization, enabling businesses to manage data across multiple locations effortlessly.
- **Mobile Application Development:**
  - Developing a companion mobile app would allow users to access, update, and manage data on the go.
  - Features like QR code scanning for instant product or customer information and push notifications for updates or reminders can enhance convenience and user engagement.
  - The app could be cross-platform (e.g., developed using Flutter or React Native) to ensure compatibility with both Android and iOS devices.
- **AI-Based Predictive Analytics:**
  - Incorporating machine learning algorithms can enable predictive analytics for better decision-making.
  - Examples include forecasting sales trends, identifying high-performing products, or predicting customer churn rates.
  - AI can also provide actionable insights by analyzing historical data to suggest optimal discount strategies or inventory levels.
- **Voice Command Integration:**
  - Adding voice command functionality (e.g., using Google Assistant, Amazon Alexa, or Microsoft Cortana) would make the system hands-free and improve accessibility.
  - Users could retrieve customer data, update records, or generate reports through simple voice commands.
- **Enhanced Reporting and Visualization Tools:**
  - Advanced dashboards with real-time data visualization can provide insights at a glance.

    - Customizable reports with interactive charts and graphs can help users analyze key performance indicators (KPIs) more effectively.
- **Integration with E-commerce Platforms:**
    - Synchronizing with platforms like Shopify, WooCommerce, or Amazon can automate order management and inventory updates.
    - This integration can also provide valuable insights into online sales performance and customer behavior.
- **Multilingual Support:**
    - Adding support for multiple languages would make the system accessible to a global user base, catering to businesses in different regions.
- **Integration with Accounting Software:**
    - Linking the system with popular accounting software like QuickBooks, Xero, or Zoho Books can automate financial tracking and reporting.
    - This would streamline expense management, invoicing, and tax preparation.
- **IoT Integration for Inventory Management:**
    - Leveraging IoT (Internet of Things) devices like RFID tags or smart sensors can automate stock tracking and provide real-time updates on inventory status.
    - This would be particularly beneficial for warehouses and supply chain operations.
- **Blockchain for Enhanced Security:**
    - Integrating blockchain technology can ensure secure and tamper-proof records, particularly for sensitive data like financial transactions or customer information.
    - Smart contracts could automate workflows, such as triggering product restocks when inventory levels fall below a certain threshold.
- **Custom Workflow Automation:**
    - Introducing customizable workflows can allow businesses to automate repetitive tasks like email follow-ups, generating invoices, or sending reminders for unpaid dues.
    - Integration with tools like Zapier or Microsoft Power Automate can extend these capabilities further.

- **Advanced QR Code Features:**

    - Expanding QR code functionalities, such as embedding dynamic data that updates automatically or adding customer-specific QR codes for loyalty programs.

- QR codes could also be used for attendance tracking, event registrations, or secure access to digital resources.

By implementing these future enhancements, the system can evolve into a comprehensive, highly adaptable solution that caters to the diverse needs of businesses. These improvements will not only boost usability and efficiency but also position the system as a cutting-edge tool for business management and analytics.

5.2 Limitation

Current limitations include :

- Dependency on Google Sheets for data storage, which may not be suitable for large-scale operations.
- Limited to desktop applications due to the use of Tkinter.
- Addressing these limitations will broaden the project's applicability.

References:

1. Python Software Foundation. *Python Documentation*. Python.org, https://docs.python.org/. Accessed 24 Dec. 2024.
2. Google Developers. *Google Sheets API*. Google, https://developers.google.com/sheets. Accessed 24 Dec. 2024.
3. Python Software Foundation. *Tkinter GUI*. Python.org, https://docs.python.org/3/library/tkinter.html. Accessed 24 Dec. 2024.
4. Python Package Index. *QR Code Library*. PyPI, https://pypi.org/project/qrcode/. Accessed 24 Dec. 2024.
5. PyFPDF. *FPDF Documentation*. PyFPDF, https://pyfpdf.github.io/. Accessed 24 Dec. 2024.