MINI PROJECT REPORT

on

RECEIPT GENERATOR using Python

Submitted by

POOVARASAN.V(RA2311053010007)
TARUN.C. D (RA2311053010054)

Semester – II

Academic Year: 2023-24 Even

Under the guidance of

Dr. D. Tharani
Assistant Professor, Department of ECE

Inpartial fulfillment for the Course

of

21CSS101J -PROGRAMMING FOR PROBLEM SOLVING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
College of Engineering and Technology,
SRM Institute of Science and Technology
SRM Nagar, Kattankulathur – 603203, Kancheepuram District, Tamil Nadu.

April 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act,1956)

BONAFIDECERTIFICATE

Certified that this activity report for the course 21CSS101J -PROGRAMMING FOR PROBLEM SOLVING is the bonafide work of **Poovarasan.V(RA2311053010007), Tarun.C.D(RA2311053010054) and** who carried out the work under my supervision.

SIGNATURE

Dr. D. Tharani

Assistant Professor

Department of ECE

SRMIST

Kattankulathur

SIGNATURE

Academic Coordinator

SRMIST

Kattankulathur

TABLEOFCONTENTS

ABSTRACT:

The receipt generator project represents a streamlined, digital solution aimed at enhancing the billing process for sellers and vendors across various business sectors. Developed using Python, this tool allows for the quick generation of bills in an online format, addressing the immediate needs of both sellers and customers during sales transactions. The system simplifies the traditional billing process by automating the generation of itemized bills and calculates totals with accuracy, thereby reducing human errors and increasing efficiency. The application is designed to be user-friendly, providing an intuitive interface that requires minimal training for users. This receipt generator not only expedites the billing process but also supports environmental initiatives by reducing the need for paper-based receipts. Furthermore, the system includes features such as saving and retrieving past transactions, which enhances customer service and aids in business management.

The Receipt Generator Project is a sophisticated digital tool designed to modernize the point-of-sale interactions between sellers and buyers across diverse commercial environments. Utilizing Python's robust programming capabilities, this tool facilitates the instantaneous creation and distribution of digital receipts, which not only streamlines sales operations but also enhances customer satisfaction by significantly reducing wait times during transactions.

This innovative project directly addresses the practical challenges associated with traditional billing methods, which are often cumbersome and error-prone. By digitizing the entire process, the Receipt Generator eliminates common errors associated with manual data entry, such as miscalculations or misspellings, and ensures a high degree of accuracy in financial transactions. Moreover, the system supports dynamic updates and instant

modifications, allowing vendors to easily add discounts, taxes, and service charges in real time, reflecting changes immediately in the customer's bill.

An additional significant feature of the receipt generator is its ability to integrate with existing inventory and accounting systems. This seamless integration facilitates real-time tracking of inventory levels, ensuring that stock quantities are automatically updated following each transaction, which is crucial for maintaining accurate business records. Furthermore, the tool's capability to generate detailed sales reports enhances strategic decision-making by providing business owners with insights into sales trends, customer preferences, and revenue streams.

The environmental impact of transitioning to a digital receipt system cannot be overstated. By eliminating the need for paper-based receipts, the project contributes significantly to environmental conservation efforts, aligning with global sustainability goals. Digital receipts can be emailed directly to customers or accessed through secure web portals, ensuring they are readily available without contributing to paper waste.

This tool represents a significant advancement in how businesses handle financial transactions, offering a scalable, reliable, and environmentally friendly solution that meets the evolving demands of modern commerce.

OBJECTIVE:

- **Enhance Transaction Efficiency:** To decrease the time spent on creating and issuing bills, thereby speeding up the transaction process which is crucial during peak business hours.

- **Reduce Errors in Billing:** To minimize human errors in calculations and itemization, which can lead to financial discrepancies and customer dissatisfaction.

- **Improve Customer Experience:** To provide customers with clear, concise, and immediately accessible digital receipts which can be easily stored and retrieved for future reference, thereby enhancing the overall shopping experience.

- **Support for Environmental Goals:** To contribute to environmental sustainability by promoting a paperless process, reducing the need for physical copies of receipts.

- **Data Management and Security:** To ensure secure storage and management of transaction data, facilitating easy access to sales histories and improving the ability to track business performance over time.

- **Scalability and Customization:** To design the system with scalability in mind, ensuring it can easily be adapted to different business sizes and types, and to include features that allow for customization of receipts (like adding a business logo and terms & conditions).

INTRODUCTION:

In the digital era, businesses continually seek innovative solutions to enhance operational efficiency and customer service. The receipt generator project was conceived as a response to the growing need for quick and efficient transaction processes in commercial settings. Traditionally, billing involved manual entries and physical receipts, which were not only time-consuming but also prone to errors. Our solution leverages Python, a powerful yet versatile programming language, to create a system that automates these processes.

The receipt generator facilitates the creation of digital receipts that are easy to generate, store, and retrieve. By moving away from manual processes, the system ensures greater accuracy and reduces the time customers spend waiting during checkout. Moreover, digital receipts are easier to manage compared to their paper counterparts, offering both environmental benefits and operational conveniences. This project aligns with the global trend towards digital transformation and sustainability, aiming to make the billing process as painless as possible for both sellers and customers while also supporting green business practices.

SOURCE CODE:

```
# Import the FPDF library for creating PDF documents
from fpdf import FPDF

# Create an instance of the FPDF class
pdf = FPDF()

# Add a page to the PDF document
pdf.add_page()

# Set the font for the text in the PDF
pdf.set_font("Arial", size=15)

# Define the filename for the receipt text file
filename = "receipt.txt"

# Create or clear the content of the receipt text file
with open(filename, "w") as f:
    a = f.write("")

# Open the receipt text file in append mode
f = open(filename, "a")

# Write the store name and header to the receipt text file
f.write("            ************************  Template Store
************************\n\n")

# Initialize variables for calculating the total bill
sum = 0
i = 1
```

```
# Begin a loop to collect product information and calculate the bill
while True:
    # Prompt the user to enter the product name or 'q' to print the final bill
    productPurchased = input(
        "Enter the product name (Press q to print the final bill): \n"
    )

    # If the user enters 'q', break the loop and proceed to printing the bill
    if productPurchased == "q":
        break

    # Prompt the user to enter the product quantity
    productQuantity = int(input("Enter the product quantity : \n"))

    # Prompt the user to enter the product code
    productCode = input("Enter the product code : \n")

    # Prompt the user to enter the unit price of the product
    productPrice = input("Enter the unit price : \n")

    # If the user did not enter 'q', calculate the total cost of the product and add
it to the overall sum
    if productPurchased != "q":
        sum = sum + (float(productPrice) * productQuantity)

        # Create a string to represent the billing information for the current
product
        billing = f"{i}). \t {productPurchased} of code {productCode}, quantity of
{productQuantity} \n billed  {productPrice}x{productQuantity} units  =  \n \t Rs.
{(float(productPrice)) * productQuantity}  "

        # Write the billing information to the receipt text file
        f.write(billing + "\n")
```

```
        # Print the billing information to the console
        print(billing)

        # Increment the counter for the next product
        i += 1

# Print the total order amount calculated so far
print(f"order total so far Rs. {sum}")

# Create a string to thank the customer for shopping and display the total bill amount
thanq = (f"\n\n!!! Your bill total is only Rs. {sum}. Thanks for shopping with us !!!\n\n")

# Write the thank-you message to the receipt text file
f.write(thanq)

# Print the thank-you message to the console
print(thanq)

# Close the receipt text file after writing all the information
f.close()

# Generate the bill in PDF format

# Open the receipt text file in read mode
f = open(filename, "r")

# Iterate through each line of the receipt text file
for x in f:
    # Add each line of text from the receipt to the PDF document, aligned to the left
    pdf.cell(200, 10, txt=x, ln=1, align="L")
```

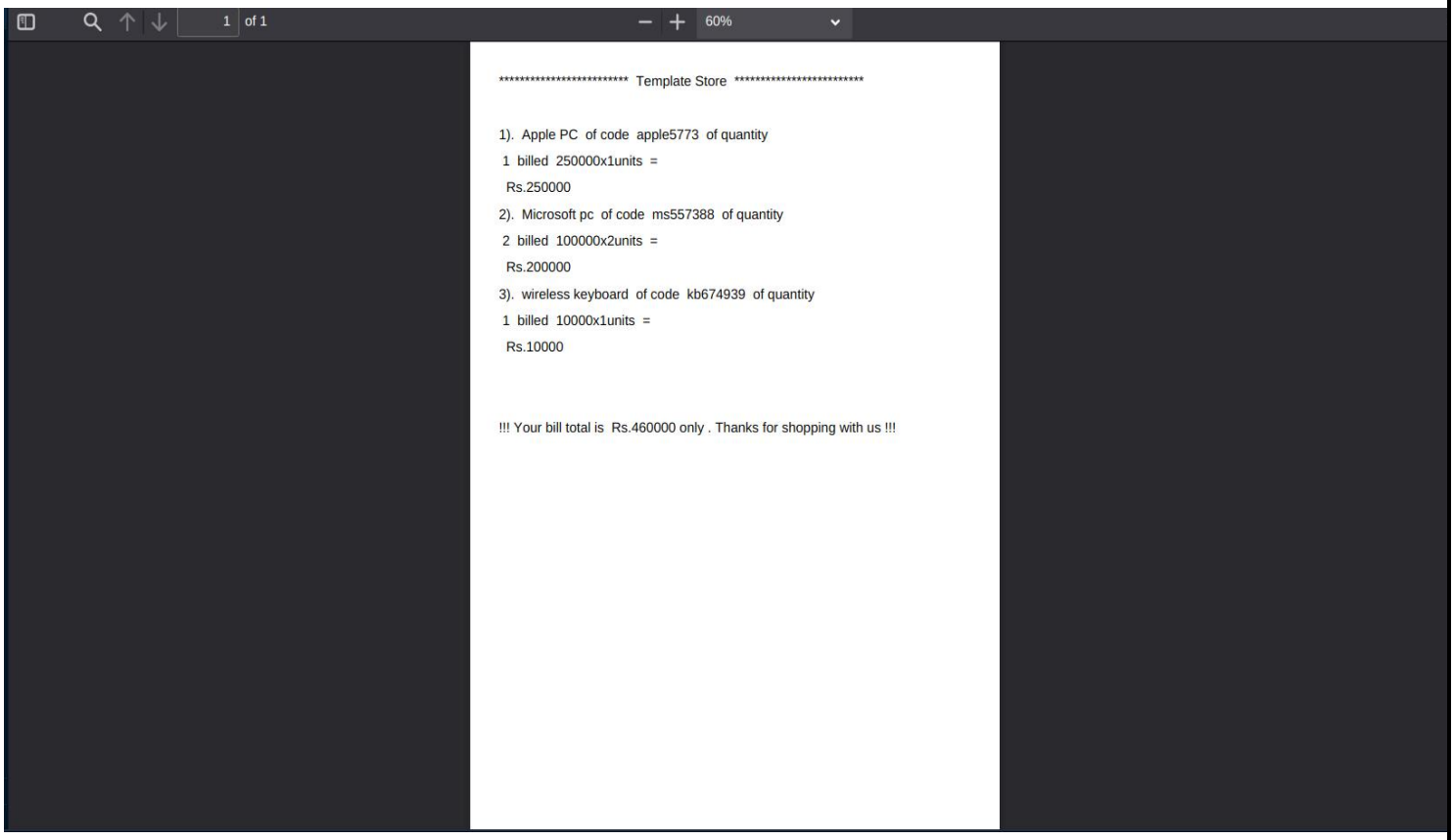
```
# Save the generated PDF file with the name "billGenerated.pdf"
pdf.output("billGenerated.pdf")

# Close the receipt text file
f.close()
```

RESULT:

```
Enter the product name (Press q to print the final bill):
Apple PC
Enter the product quantity :
1
Enter the product code :
apple5773
Enter the unit price :
250000
1).      Apple PC  of code  apple5773  of quantity
 1  billed  250000x1units  =
         Rs.250000
Enter the product name (Press q to print the final bill):
Microsoft pc
Enter the product quantity :
2
Enter the product code :
ms557388
Enter the unit price :
100000
2).      Microsoft pc  of code  ms557388  of quantity
 2  billed  100000x2units  =
         Rs.200000
Enter the product name (Press q to print the final bill):
wireless keyboard
Enter the product quantity :
1
Enter the product code :
kb674939
Enter the unit price :
10000
3).      wireless keyboard  of code  kb674939  of quantity
 1  billed  10000x1units  =
         Rs.10000
Enter the product name (Press q to print the final bill):
q
order total so far Rs.460000


!!! Your bill total is  Rs.460000 only . Thanks for shopping with us !!!
```



REFERENCES:

W3schools.com

Geeksforgeeks.org

Pyfpdf.readthedocs.io