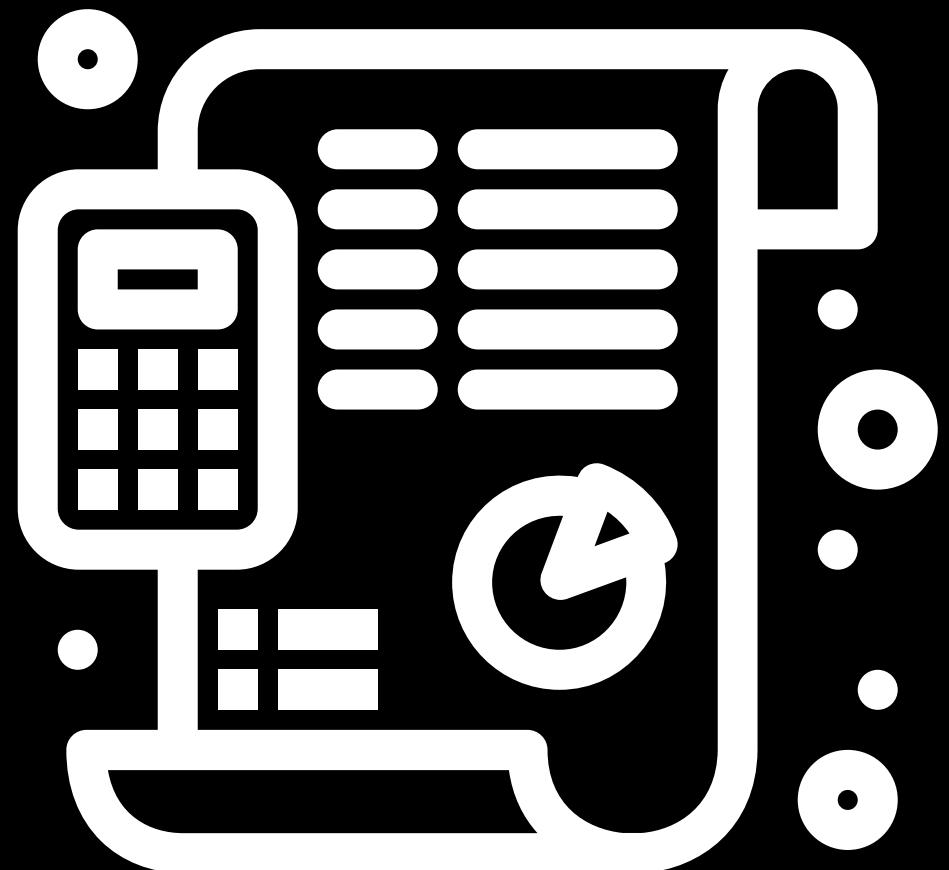


PROJECT PRESENTATION

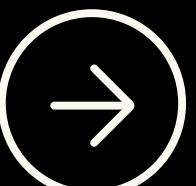
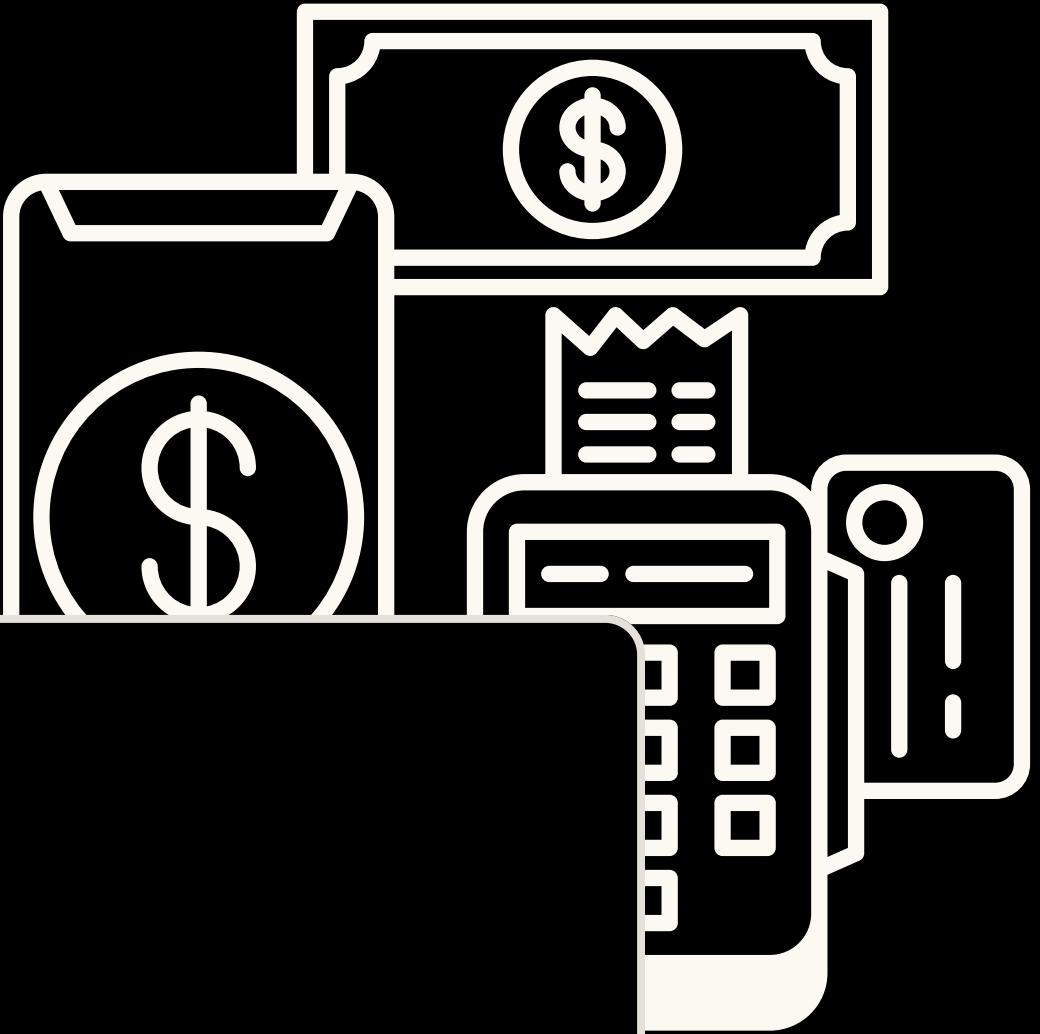
BILLING SYSTEM IN PYTHON



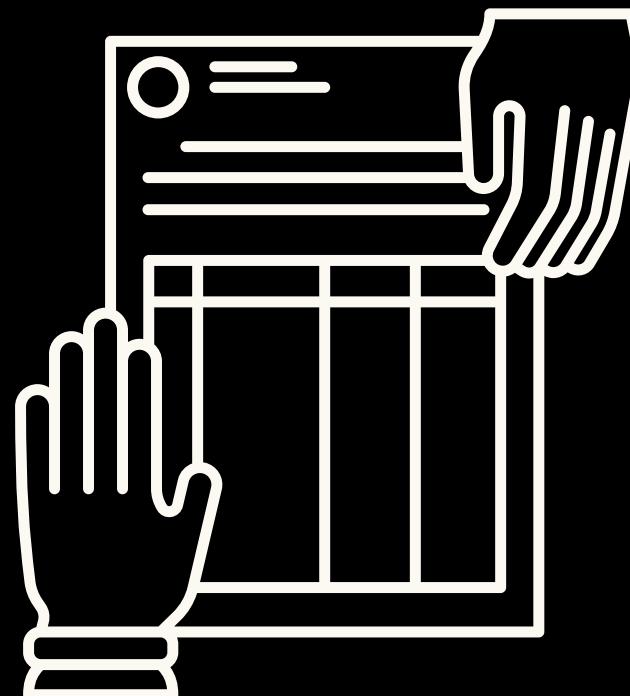
Presented by -
JOEL GEORGE GEMINI
GOURAV L R
AMOGH KRISHNA K
POORNIMA J URS
JEEVANA V

CONTENTS

- 1 Introduction
- 2 Problem Statement
- 3 Key Features
- 4 Objectives
- 5 Technology Stack
- 6 User Interface Design
- 7 System Flow
- 8 Code Overview
- 9 Challenges Faced
- 10 Conclusion & Future Enhancements



INTRODUCTION



Overview: A Python-based billing system designed for retail, using a GUI built with Tkinter.

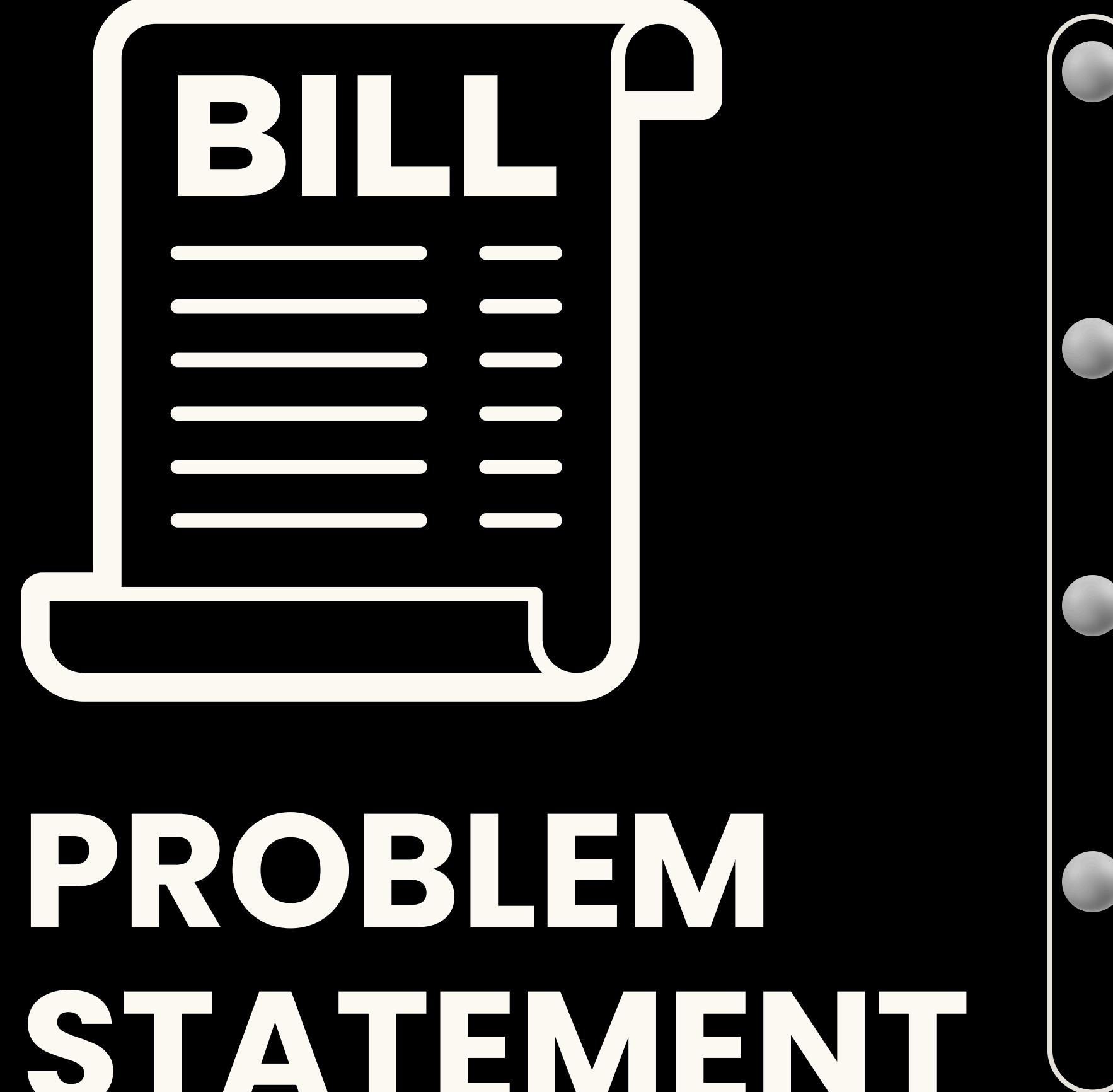
Purpose: Streamlines the billing process, automates calculations, and generates bills efficiently.

Key Features:

- Dynamic quantity adjustment.
- Bill generation in PDF format.
- Easy management of customer transactions.

Ideal for small businesses seeking a simple and efficient billing solution.





PROBLEM STATEMENT

- **Manual Billing Inefficiencies:** Traditional billing processes in small retail stores are often time-consuming and prone to errors.
- **Need for Automation:** Calculating total amounts, applying taxes, and generating bills manually can lead to inaccuracies and delays.
- **User Convenience:** Lack of an easy-to-use system for store owners to quickly generate and save bills.
- **Solution:** A simple, automated billing software that streamlines the entire billing process with dynamic product management and automated calculations.

KEY FEATURES

- **User-Friendly Interface:** Simple, intuitive design using Tkinter for easy navigation.
- **Dynamic Quantity Adjustment:** Users can increase or decrease item quantities in real-time.
- **Automated Calculations:** Total price and tax are calculated automatically based on selected items.
- **Bill Generation:** Final bills are saved directly as PDF files for easy storage and printing.
- **Error Handling:** Ensures valid input for quantities, preventing common user mistakes.



OBJECTIVES

AUTOMATE BILLING

Create a user-friendly system to automate the billing process for small retail stores.

PRODUCT MANAGEMENT

Allow dynamic input of product quantities with automatic total calculation.

ERROR-FREE CALCULATIONS

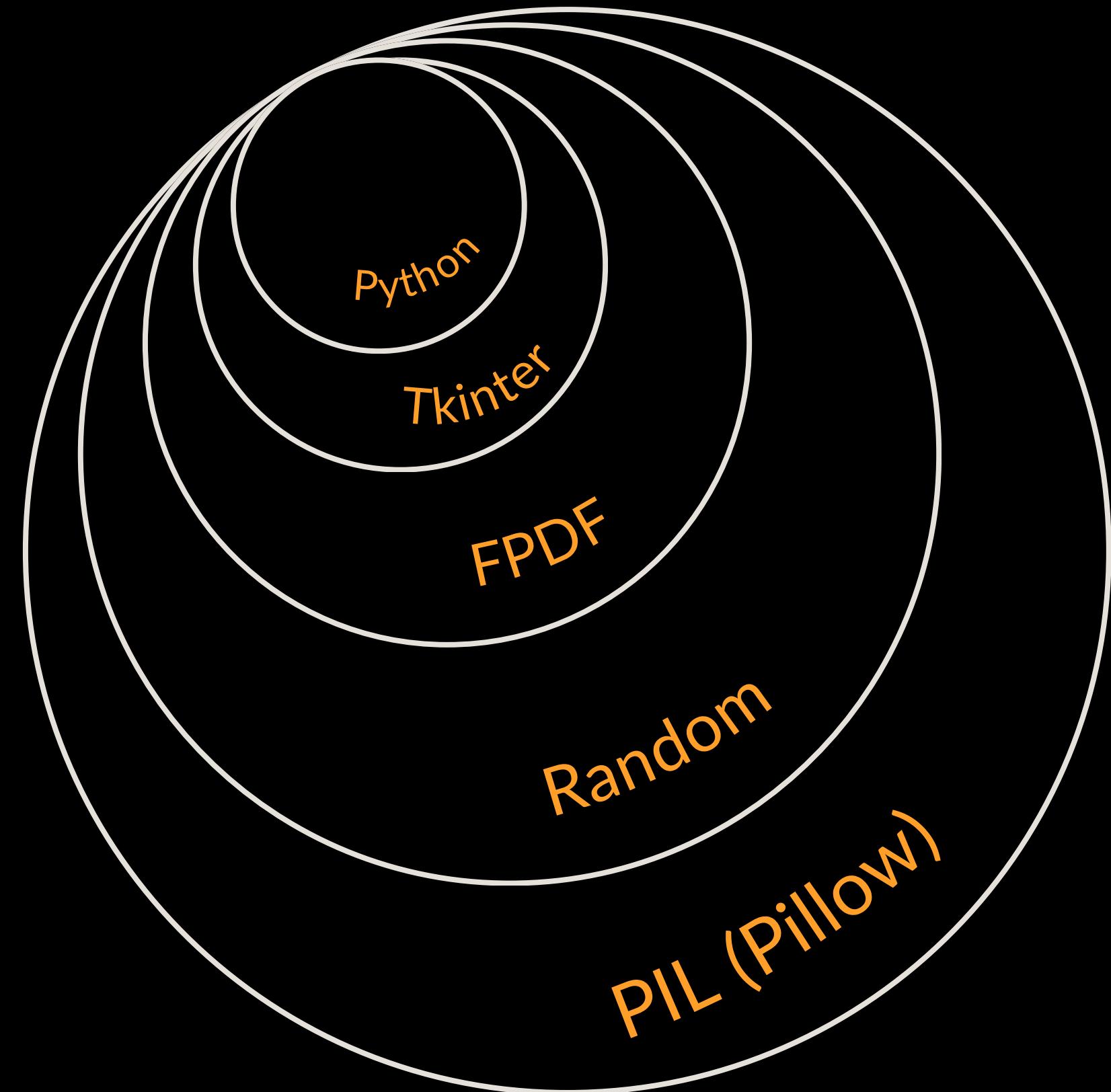
Ensure accurate and real-time updates for total price and tax during billing.

BILL GENERATION

Generate and save bills quickly in PDF format for easy record-keeping.

TECHNOLOGY STACK

- **Python**: Core programming language for the application.
- **Tkinter**: GUI library used for creating the application interface.
- **FPDF**: Python library used to generate PDF bills.
- **Random**: Python module to generate unique bill numbers.
- **PIL (Pillow)**: For saving the generated PDF file.



USER INTERFACE DESIGN

Tkinter GUI

- Simple, easy-to-navigate interface for billing.
- Fields for customer details and product quantities.
- Bill display area with automatic scrolling.

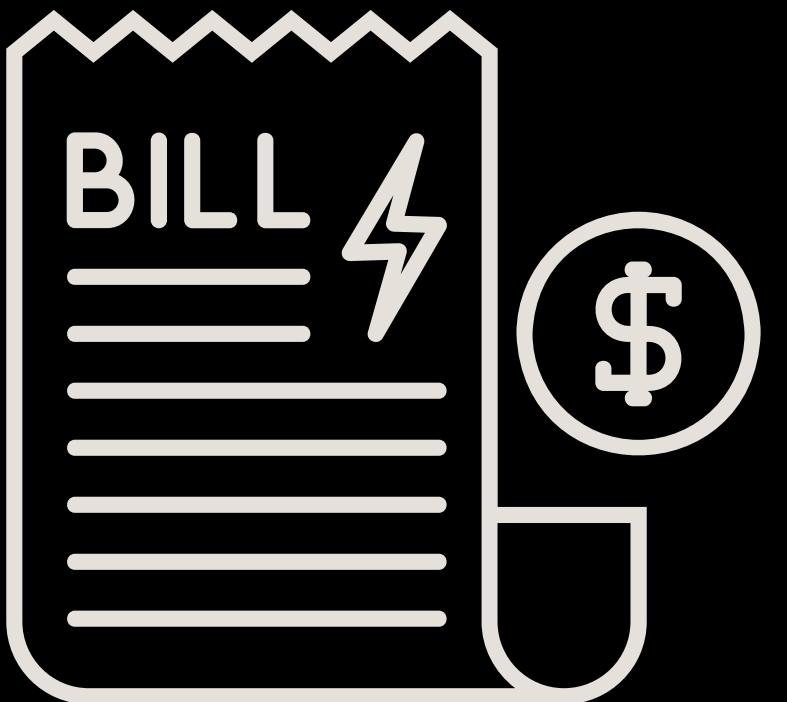
Input Fields

- Customer name, phone number, and bill number.
- Product quantities with add/remove buttons.

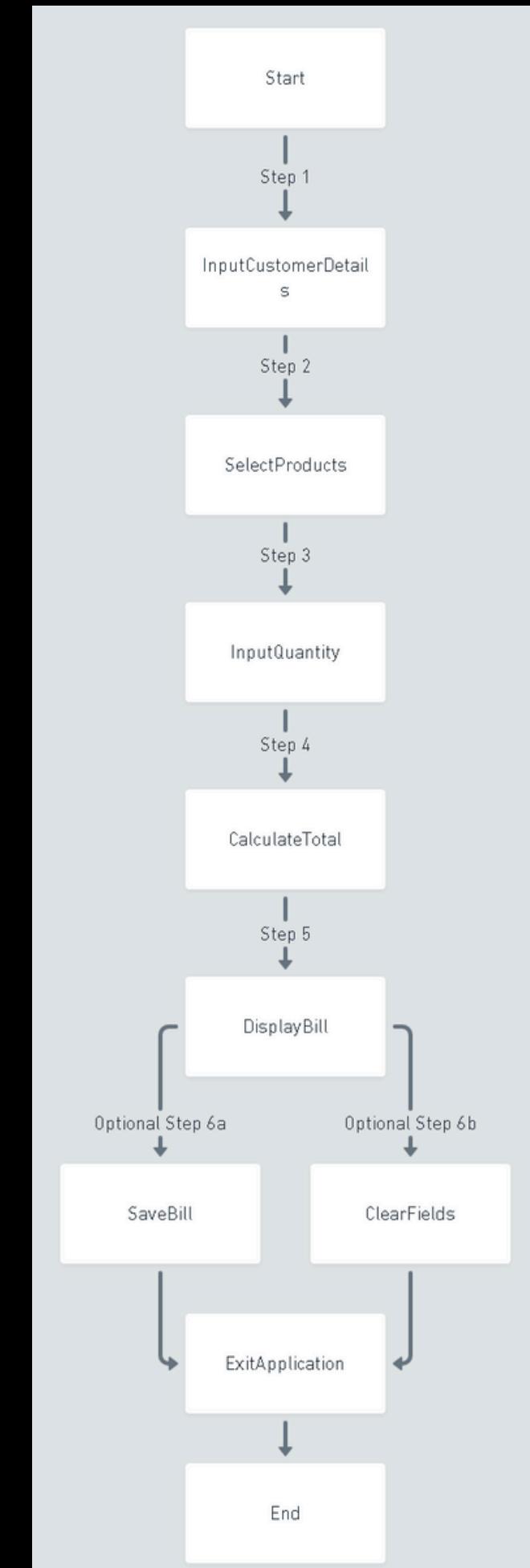
Output

- Real-time bill preview in the bill area.
- Option to generate and download the final bill as a PDF.

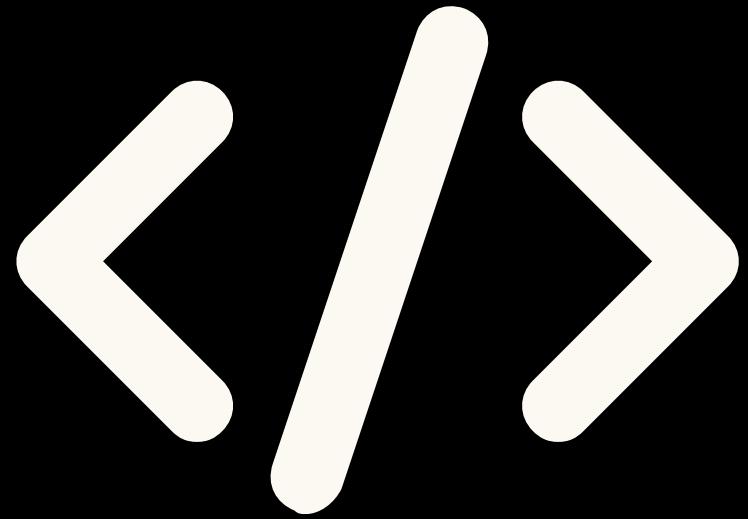
SYSTEM FLOW



- 1. Start:**
 - Application launches.
- 2. Customer Input:**
 - Enter customer details: name, phone number, and bill number.
- 3. Product Selection:**
 - Choose products from the list.
 - Adjust quantities using increment/decrement buttons.
- 4. Bill Generation:**
 - Click on "Total Bill" to calculate total price and tax.
 - Display detailed bill in the bill area.
- 5. Save/Export:**
 - Option to save the bill as a text file or download as PDF.
- 6. Clear/Exit:**
 - Clear fields for a new transaction or exit the application.



CODE OVERVIEW



1. Main Class - `Bill_App`:

- The core functionality is encapsulated in the `Bill_App` class, where the entire billing system logic is implemented.
- It initializes the graphical user interface (GUI) and handles user interactions.

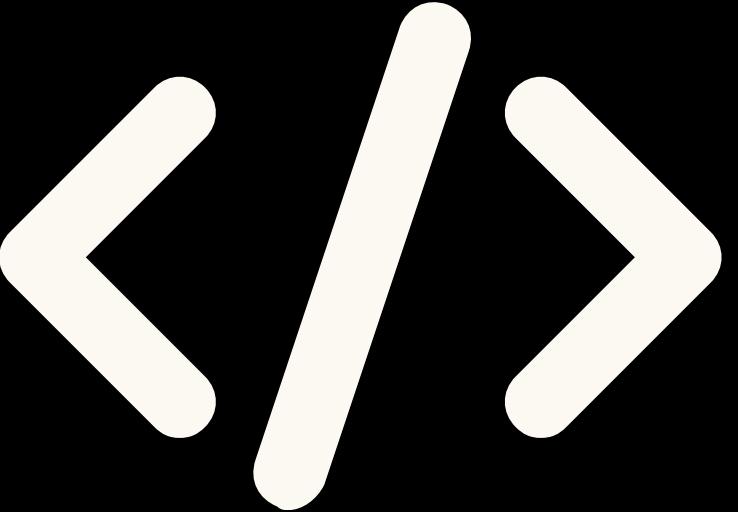
2. GUI Setup:

- The app uses `Tkinter` for the user interface.
- It consists of different sections: title, customer details, product list, bill area, and billing summary.

3. Product Management:

- A dictionary is used to store product names along with their respective quantity and price.
- Users can input the quantity for each product, and the system will calculate the total based on predefined prices.

CODE OVERVIEW



4. Bill Area:

- The bill area displays all the products, quantities, prices, and the final bill, including tax.
- Real-time bill updates are shown as users enter the quantities.

5. Total Calculation:

- The `total` method calculates the total price and applies a 5% tax.
- The total amount is displayed in the billing summary and printed in the bill area.

6. Saving the Bill:

- The `save_bill` method allows users to save the bill as a `'.pdf` file with the unique bill number.

7. Other Functionalities:

- Clear function resets all fields for new billing.
- Exit function provides safe closure with user confirmation.

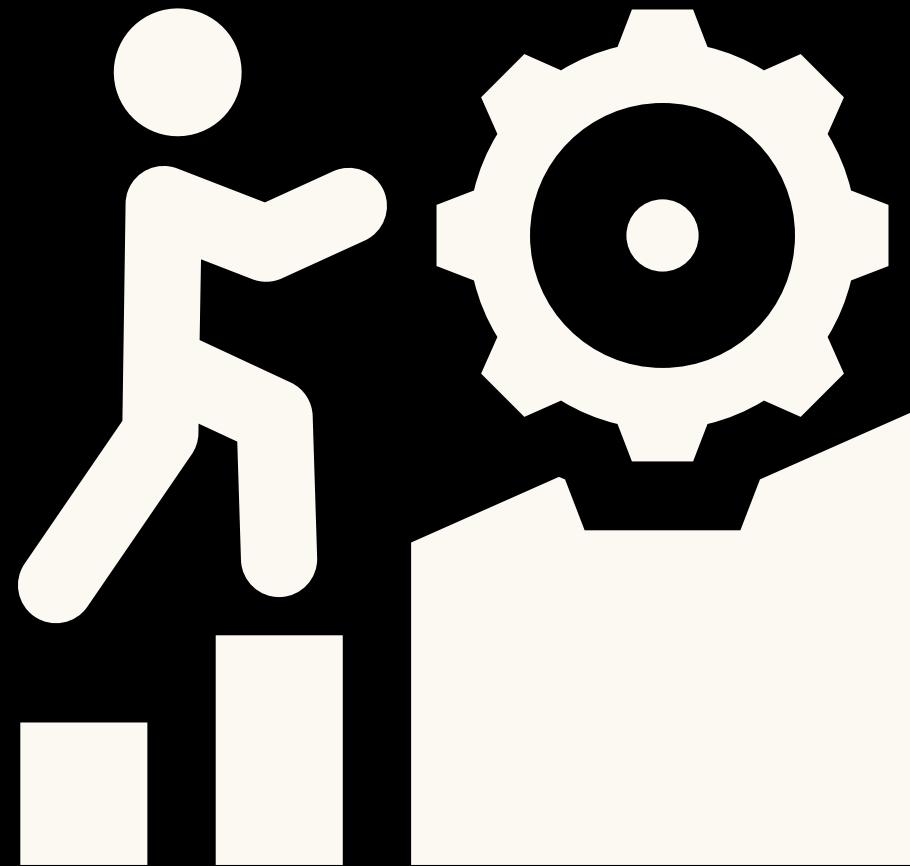
CHALLENGES FACED

UI Design & Layout:

- Organizing multiple components in a user-friendly layout and ensuring the design was responsive to various screen sizes.

Dynamic Data Management

- Handling real-time updates for product quantities, calculating totals, and saving bills accurately posed technical challenges.



CONCLUSION AND FUTURE ENHANCEMENTS:

- **CONCLUSION-** This billing software simplifies retail transactions by automating the billing process, providing accurate calculations, and generating downloadable bills. It's an efficient tool for small businesses.
- **FUTURE ENHANCEMENTS-**
 - **Database Integration:** Implement a database to store customer and product details, making it more scalable.
 - **Barcode Scanning:** Integrate barcode scanning for quicker product entry and improved accuracy.

THANK YOU

for your time and attention

