**Restaurant Billing System Documentation**

**Table of Contents**

1. Introduction
2. Objectives
3. Technology Stack
4. System Architecture
5. Features
6. Implementation Details
7. Deployment Guide
8. Challenges and Solutions
9. Future Enhancements
10. Conclusion
11. Appendix

**1. Introduction**

The "Restaurant Billing System" is an automated solution designed to simplify and enhance the billing process in restaurants. The system calculates the total bill, generates a detailed receipt in PDF format, and sends the receipt via SMS to the customer. The primary goal is to reduce manual effort, minimize errors, and improve customer satisfaction.

**2. Objectives**

* Automate the billing process.
* Generate professional receipts in PDF format.
* Send receipts via SMS to customers.
* Provide a user-friendly interface.
* Ensure easy deployment and scalability.

**3. Technology Stack**

* **Python**: Core programming language for backend logic.
* **Flask**: Web framework for building the application.
* **Twilio API**: Service for sending SMS notifications.
* **ReportLab**: Library for generating PDF documents.
* **HTML/CSS**: For creating the frontend interface.
* **Git**: Version control.
* **Docker**: Containerization (optional).
* **Heroku**: Cloud platform for deployment (optional).

**4. System Architecture**

The system architecture consists of the following components:

* **Frontend**: HTML/CSS forms for user input.
* **Backend**: Flask routes to handle requests, process data, and integrate with Twilio and ReportLab.
* **PDF Generation**: Using ReportLab to create formatted receipts.
* **SMS Notification**: Using Twilio API to send receipts via SMS.
* **Database**: (Optional) To store receipts and customer details.

**5. Features**

* **User Interface**: Form to input food items, quantities, and customer details.
* **Receipt Generation**: Detailed receipt in PDF format.
* **SMS Notification**: Sends receipt to customer via Twilio API.
* **GST Calculation**: Adds GST to the total bill.
* **Sequential Billing**: Generates sequential bill numbers.
* **Customer Details**: Automatically includes "+91" for phone numbers, displays customer name, address, and contact details on the receipt.

**6. Implementation Details**

**Flask Application Structure**

* **app.py**: Main application file containing routes and logic.
* **templates/**: Folder containing HTML templates.
  + **index.html**: Form for user input.
  + **bill.html**: Displays the generated receipt.
* **static/**: Folder for static assets like CSS.

**7. Challenges and Solutions**

* **PDF Generation**: Formatting text and handling line breaks was a challenge, solved by using ReportLab's text object.
* **SMS Formatting**: Ensuring the SMS receipt was readable required careful formatting and testing.

**8. Future Enhancements**

* **Database Integration**: Store customer and transaction data in a database.
* **Enhanced UI**: Improve the user interface with modern design elements.
* **Advanced Receipt Features**: Add more details like discounts, offers, and loyalty points.

**9. Conclusion**

The Restaurant Billing System automates the billing process, generates professional receipts, and enhances customer satisfaction through SMS notifications. It leverages Python, Flask, Twilio, and ReportLab to deliver a robust and scalable solution.

**10. Appendix**

* **Twilio Documentation**: Twilio API
* **Flask Documentation**: Flask
* **ReportLab Documentation**: ReportLab
* **Python Documentation**: [Python](https://docs.python.org/3/)

**Documented by –** Sambhab Sahoo ;