**SpillBill**

Key Aspects of SpillBill

1. Overview and Purpose

* What is SpillBill - SpillBill is an inventory management and billing system designed to help users keep track of products, generate QR codes, scan products, and create bills.
* Purpose: It streamlines the billing process, ensures accurate inventory management, and provides a user-friendly interface for both customers and administrators.

2. Technologies Used

* Backend: Python with Flask framework for web development.
* Frontend: HTML, CSS, and JavaScript for the user interface.
* Database: SQLAlchemy with SQLite for database management.
* QR Code Generation:

Library: qrcode

Purpose: This library is used to generate QR codes for products. When you add a new product, a QR code is automatically created that contains the product details.

QR Code Scanning:

Library: html5-qrcode

Purpose: This JavaScript library is used on the frontend to scan QR codes. It integrates with the HTML5 video element to read QR codes using a device's camera.

* 3. Key Features
* Add New Products: Admins can add new products with names and prices.
* QR Code Generation: Automatically generate QR codes for new products.
* Scan Products: Scan QR codes to add products to the customer’s cart.
* Manual Product Entry: Option to manually enter product names if QR codes are unavailable.
* Generate Bill: Calculate the total bill, including taxes, and save it to a file.
* Inventory Management: View and manage the inventory, including deleting products.
* Customer Details: Collect customer information during the billing process.

4. Project Structure

* [app.py](https://app.py/?form=MG0AV3): Initializes Flask app and database, and handles routing.
* [models.py](https://models.py/?form=MG0AV3): Defines the database model for products.
* [routes.py](https://routes.py/?form=MG0AV3): Contains the main application routes and logic.
* templates/: Folder containing HTML templates (e.g., home.html, add\_items.html, scan\_products.html, total\_bill.html).
* static/: Folder for static files such as CSS, JavaScript, and QR code images.
* instance/: Folder containing the SQLite database.

5. Detailed Workflow

* Adding Products:
  + Admins can add new products via the add\_items.html page.
  + Products are stored in the SQLite database, and QR codes are generated.
* Scanning and Checkout:
  + Customers' details are collected and stored temporarily.
  + Products can be scanned using a QR code scanner, or entered manually if needed.
  + Products are added to the cart, and quantities can be adjusted.
* Generating Bill:
  + Once all products are added, the bill is calculated, including taxes.
  + The bill is saved to a file and displayed to the user.