Billing System

Implement a simple Django/Flask/FastAPI application for the following problem.

Problem Description

1. Design a database schema for products with the fields (name, product ID, available stocks, price of one unit (float) and tax percentage(float)), and you can seed the values. You can add a CRUD page to add these details or use Django or any related admin.
2. Design a billing calculation page as **Page 1**.
    1. The form should include entering the customer email who purchased the items.
    2. There should be a provision to add all products that the client bought to the bill by clicking on **Add New** button, and it should dynamically add a field to the form to enter a *Product ID* and the quantity of the purchased product.
    3. Create a separator, as shown in the image next to the product section. You can create the default denominations as shown in the image to collect the denominations count in each of the available values. **Note:** The denominations are the values that are available in the shop.
    4. There should be another input field to collect the customer's paid amount for the bill.
    5. Clicking **Generates Bill** button should calculate the values, show the information as shown on **Page 2** and send the invoice to the customer's email ID in the background (asynchronously).
    6. Calculate the balance denomination that needs to be given to the customer based on the denominations available in the shop, as shown on page 2.
3. Add an option to view previous purchases by the customer and list all the purchases, and selecting one from it should show what items were purchased in that purchase.

Notes

1. If there is any doubt, please reach us or implement based on assumptions and mention those assumptions.
2. You should implement code by following best practices, and it should be production-ready.
3. We give more importance to **Model** and **View** related concepts. The basic level of **Template** logic is good; no need for fancy CSS.
4. There is no predefined solution on the internet, so open to using any approach for designing database schema and Django/Flask/FastAPI logic.
5. Add all project dependencies in the **requirements.txt** file in the project directory. We will use this to install all dependencies when evaluating your task.
6. Mention any special instructions in the **readme** file to run and test the project. The final project you sent to us should run on our machine without issues.

Page 1

Billing Page

Customer Email: Email ID

Add New

Bill section

Product ID	Quantity
Product ID	Quantity
Product ID	Quantity

Denominations

500	Count
50	Count
20	Count
10	Count
5	Count
2	Count
1	Count

Cash paid by customer: Amount

Cancel | Generate Bill

Page 2

Billing Page

Customer Email: Email Collected in the previous page

Bill section

Product ID	Unit Price	Quantity	Purchase Price	Tax % for item	Tax payable for item	Total price of the item
Product ID	Unit Price	Quantity	Purchase price	Tax % for item	Tax payable for item	Total price of the item
Product ID	Unit Price	Quantity	Purchase price	Tax % for item	Tax payable for item	Total price of the item

Total price without tax:	2102.00
Total tax payable:	255.60
Net price of the purchased item:	2357.60
Rounded down value of the purchased items net price:	2357.00
Balance payable to the customer:	643.00

Balance Denomination:

500:	1
50:	2
20:	2
2:	1
1:	1