Sathish Kumar R

IV ECE ‘B’

Karpagam Institute of Technology

**Billing Software using Flask ,MySQL**

Flask is a micro web framework for Python. It is designed to be lightweight and flexible, providing the essentials needed to build web applications without imposing a lot of overhead or unnecessary features. Flask is often referred to as a "micro" framework because it focuses on the core functionality of routing HTTP requests and handling responses while leaving many other tasks, like database integration and form validation, to be implemented using third-party extensions as needed.

Here are some key features and characteristics of Flask:

1. Routing: Flask provides a simple way to map URLs to Python functions or views. You can define routes and specify what should happen when a user accesses a particular URL.

2. Templates: Flask includes a templating engine (Jinja2 by default) for rendering HTML pages with dynamic content. This allows you to generate HTML pages by filling in data from your Python code.

3. Request and Response Handling: Flask simplifies handling incoming HTTP requests and generating responses. It provides easy access to request data, form data, and cookies.

4. Werkzeug: Flask is built on top of the Werkzeug WSGI toolkit, which helps with request and response handling, routing, and other web-related tasks.

5. Extensible: Flask is designed to be extensible, meaning you can add functionality to your application by incorporating various Flask extensions. These extensions cover a wide range of tasks, including database integration (SQLAlchemy, Flask-SQLAlchemy), authentication (Flask-Login), form handling (Flask-WTF), and more.

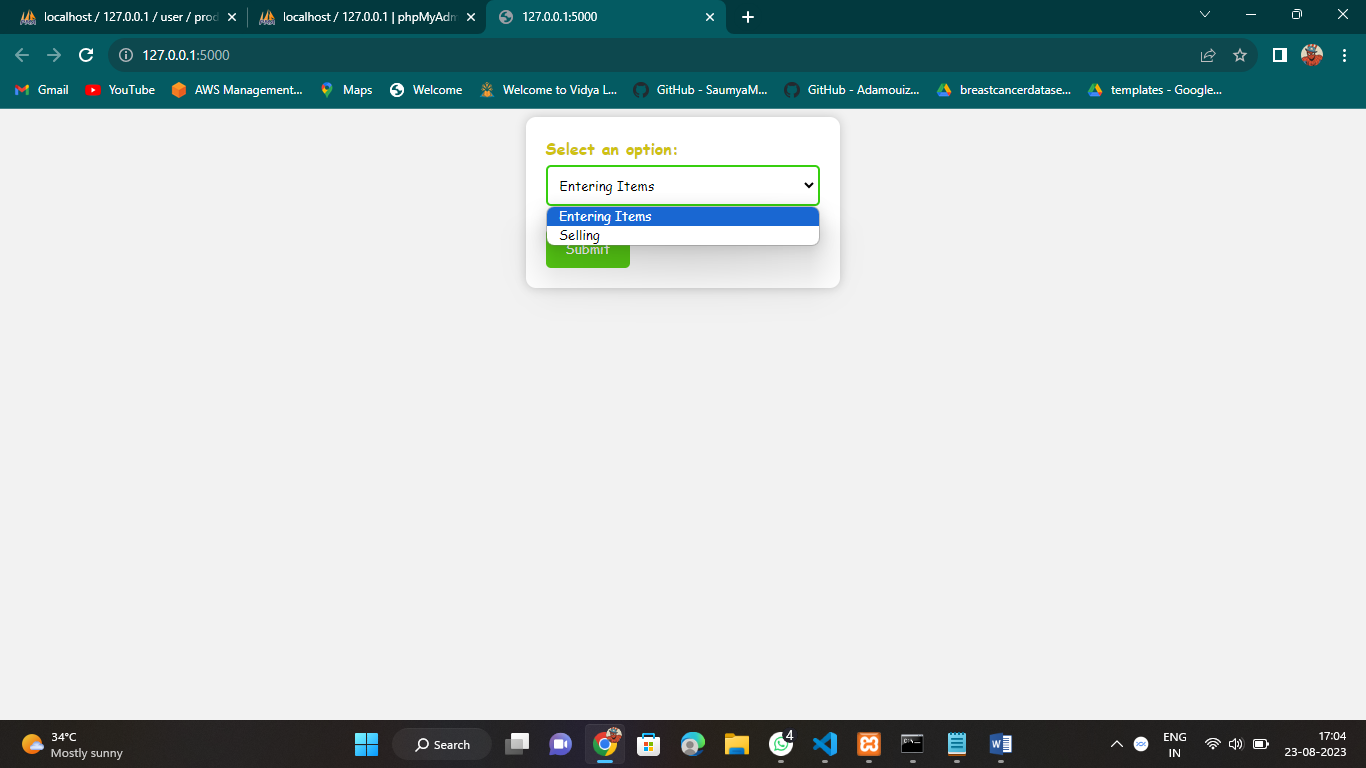
6. Minimalistic: Flask intentionally keeps its core small and unopinionated, which means it doesn't prescribe a specific way to structure your application or dictate the choice of libraries for various tasks. This flexibility allows developers to choose the components and libraries that best fit their project's needs.

7. Python: Flask follows Python principles, making it intuitive and easy for Python developers to work with. It emphasizes clean and readable code.

8. Community and Ecosystem: Flask has a vibrant community, and there is a wide range of third-party extensions and resources available for it. This means you can find solutions and guidance for various use cases.

While Flask is great for small to medium-sized projects and prototyping, it may require more manual configuration and decision-making compared to larger, more opinionated frameworks like Django. Flask's simplicity and minimalism make it a popular choice for developers who prefer to have more control over their application's components and design decisions.

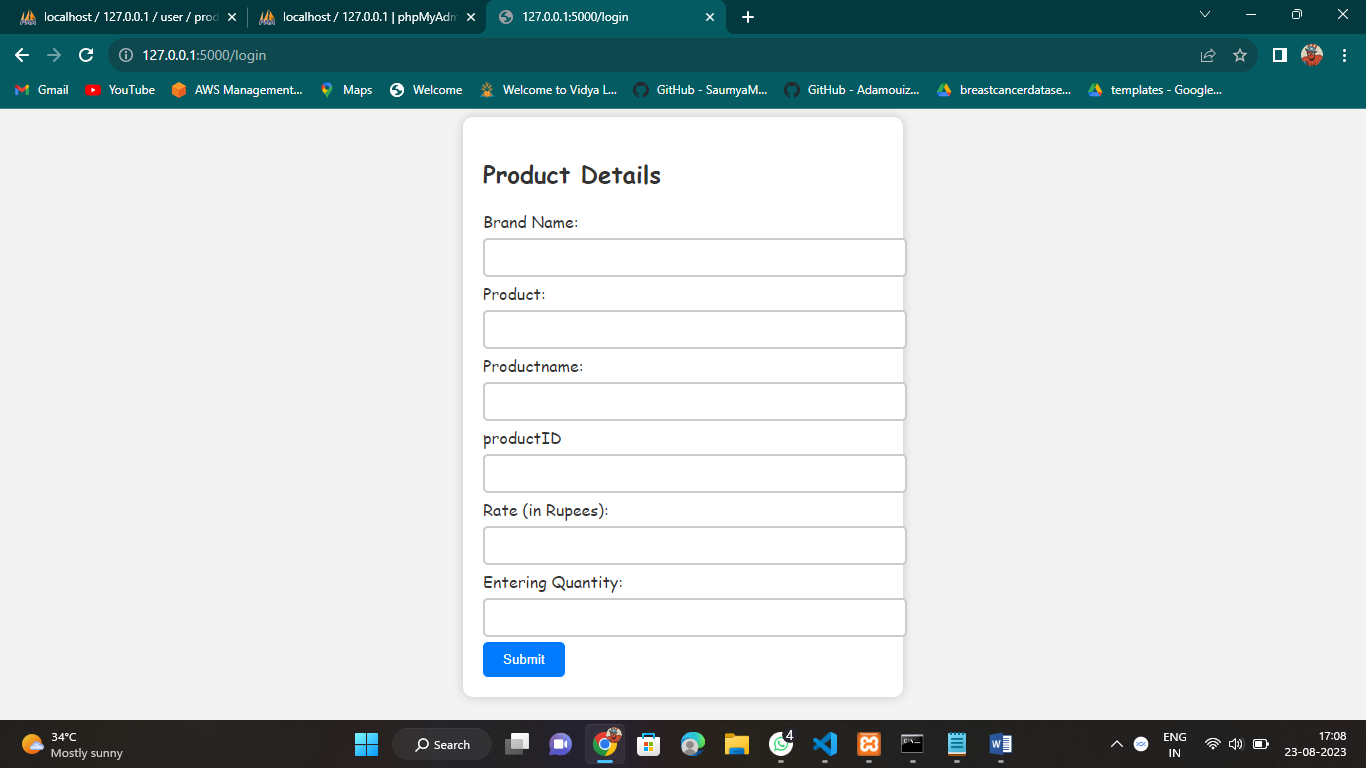
**Login Page:**



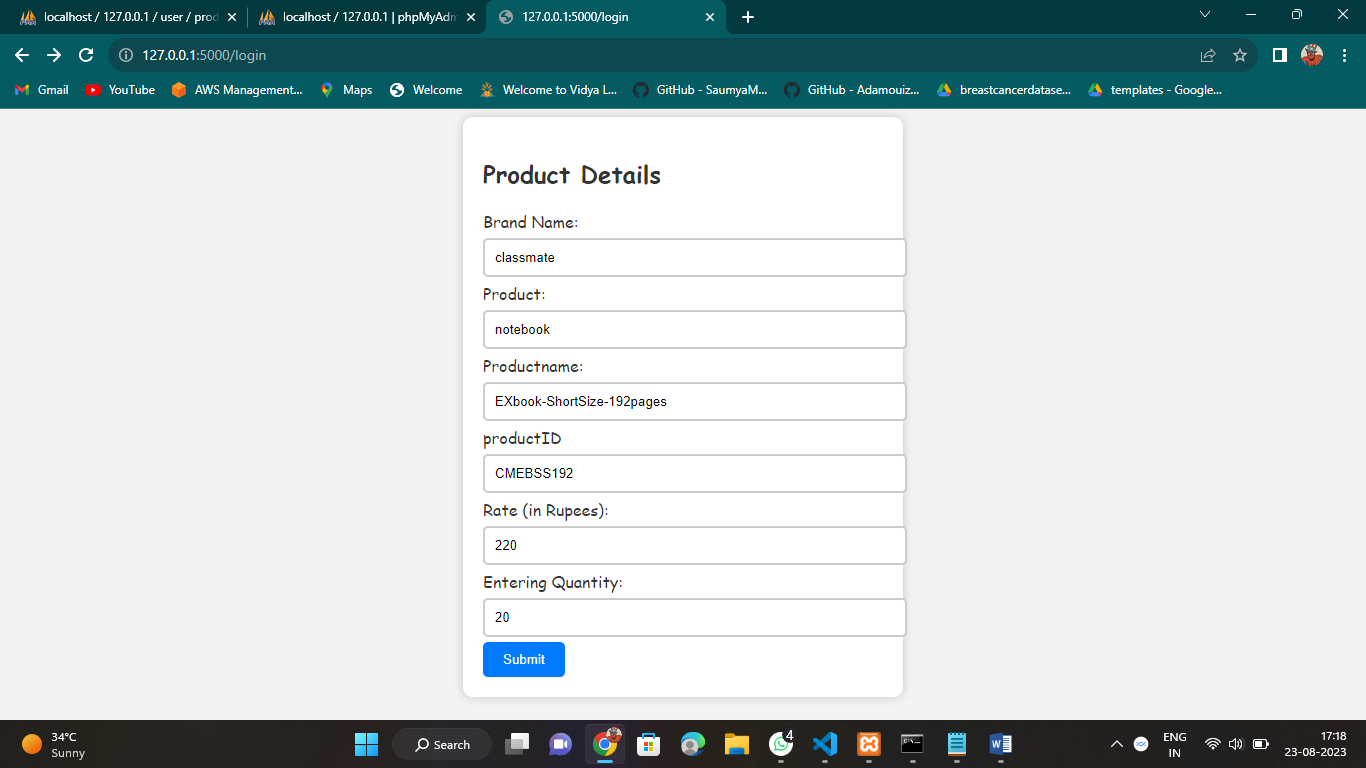
In the Login page I have specified the service into Entering items and Selling.

In selling page if we enter the Entering items it will lead to the entering items page.

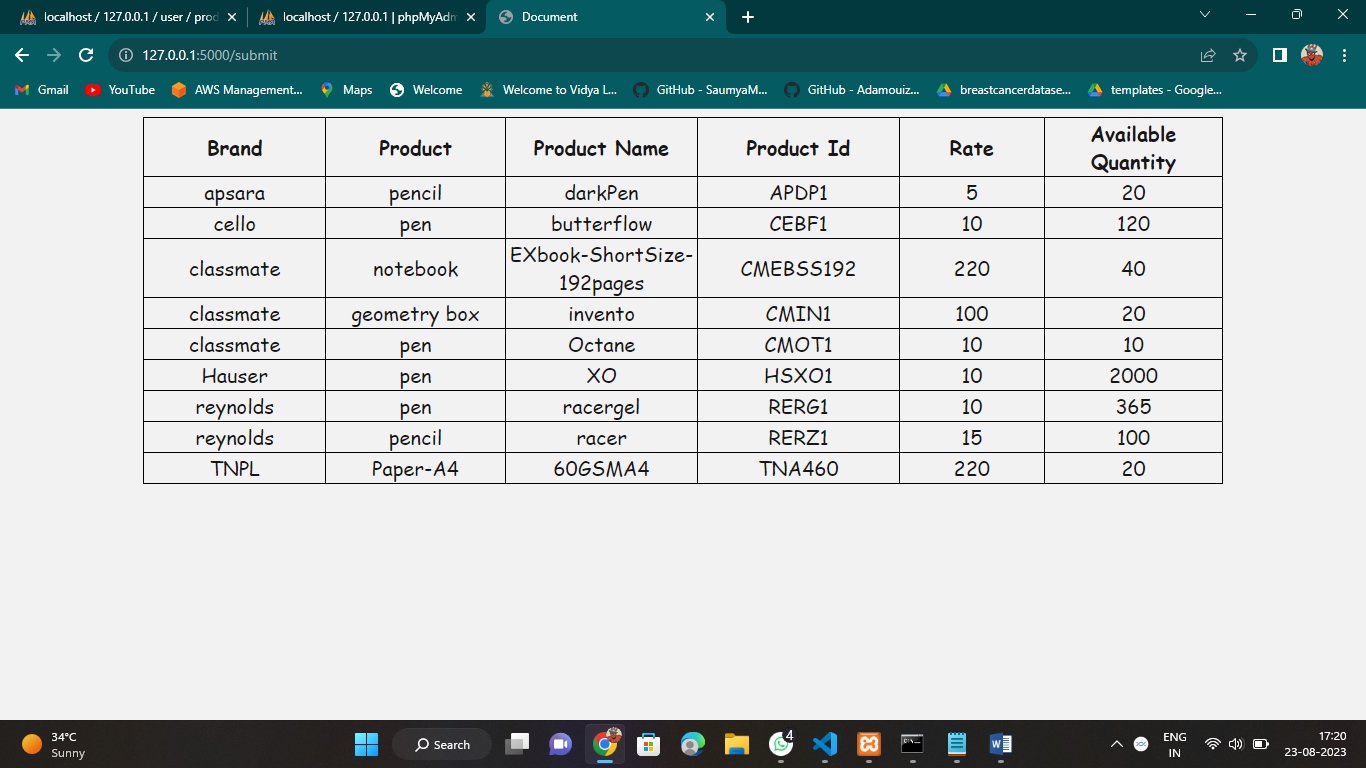
In entering items page I enter the details of the product that is going to enter



After entering the details it gets stored in the database and after submitting it shows the available products.



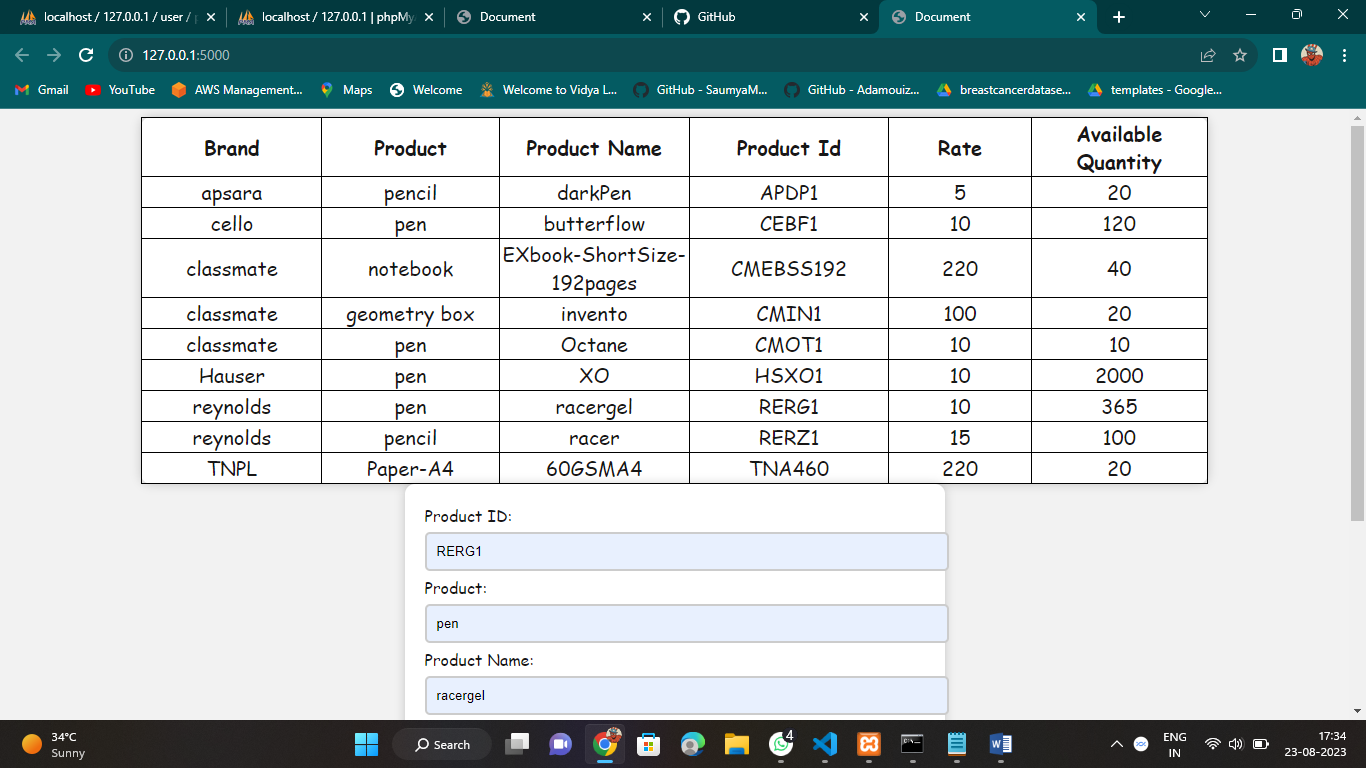
It gets reflected in database. I have created product ID as primary key.So that there will be no duplicate values in table and valuation done using primary key.

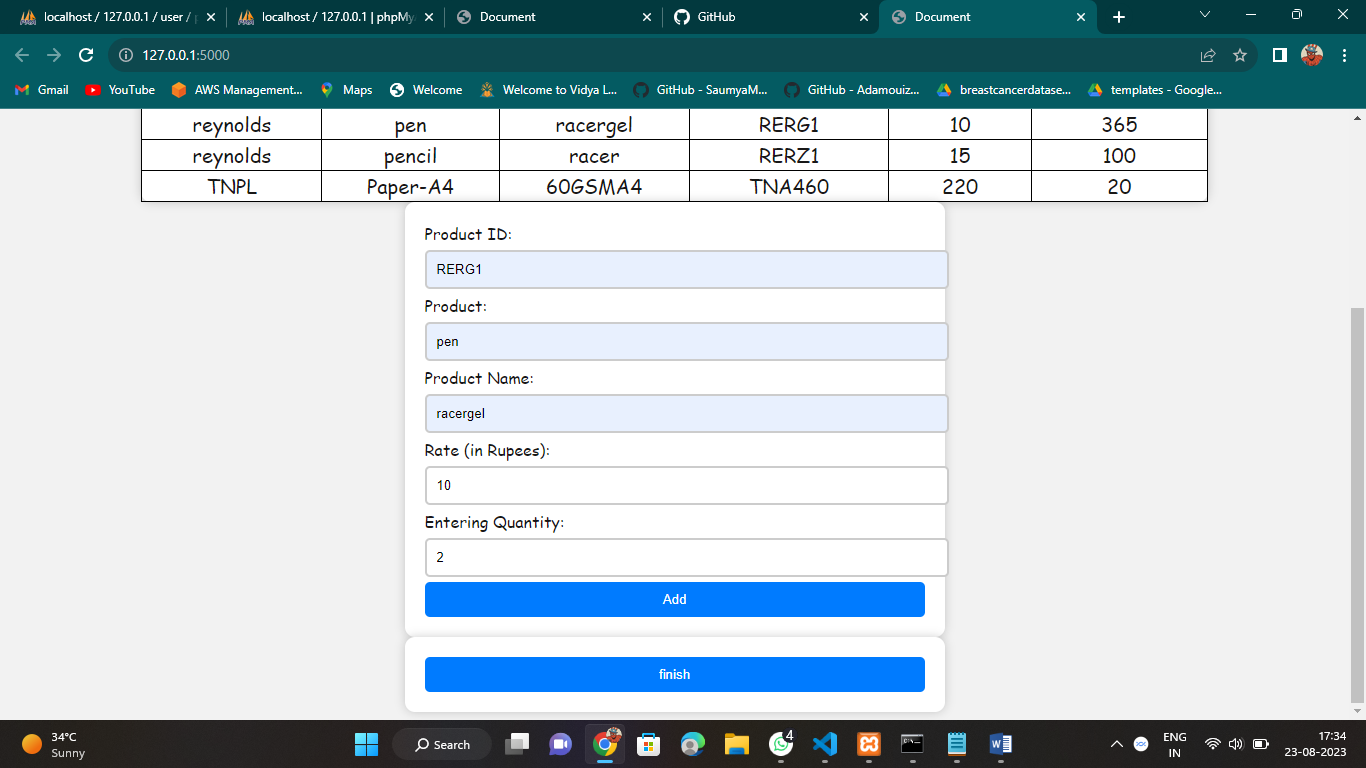


Using product ID if the product ID already exists,it gets updated in quantity only

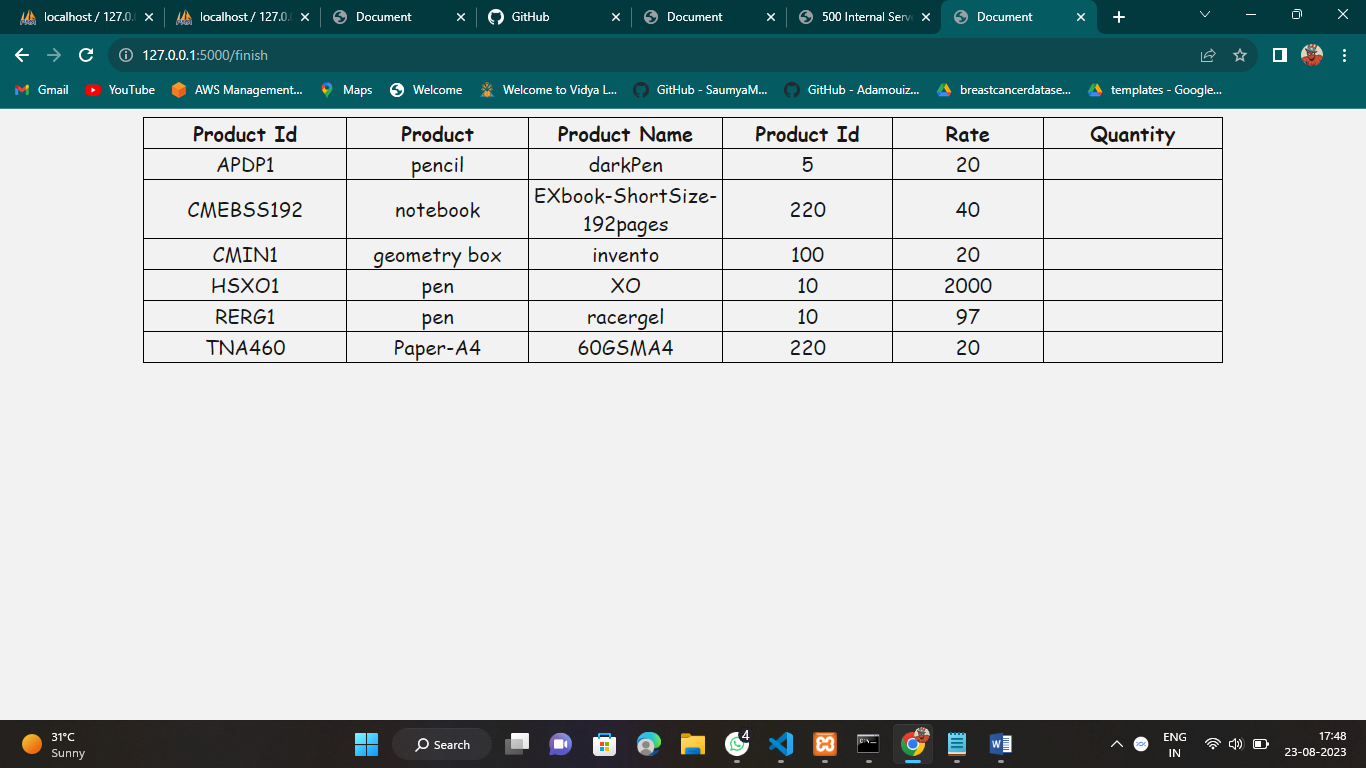
i.e..,it gets incremented in the quantity.

**Billing/Selling page:**

In selling page****



By clicking add the values in the primary database gets reduced and billing database gets updated.

after everything it gives the bill

