**Emerging Technologies CSD3444(Project Part -B)**

**Smart-Mart Project**

|  |  |
| --- | --- |
| **Samanth Sidhabathuni** | **(C0773700)** |
| **Rahul Valupadasu** | **(C0768694)** |
| **Bharadwaj Nerella** | **(C0771698)** |

Contents

[SOFTWARE REQUIREMENTS: 3](#_Toc48762009)

[HARDWARE REQUIREMENT: 3](#_Toc48762010)

[APPLICATIONS FILES DESCRIPTION: 3](#_Toc48762011)

[STEP BY STEP USER GUIDE: 5](#_Toc48762012)

[DETAILS ABOUT DATABASE: 11](#_Toc48762013)

# SOFTWARE REQUIREMENTS:

1. Python (version 3.8 or above)
2. package installer for Python (pip)
3. Mysql Server
4. Operating System(Windows 7 or 10, or Linux or Mac)
5. MySql Workbench (Optional)
6. Pycharm

# HARDWARE REQUIREMENT:

1. RAM: 4-6GB
2. Number of Core: Recommended to have a dual core but the minimum is single core
3. Processor: Minimum Intel ® core(™) dual core 2Ghz
4. Hard Disk Drive: Minimum 30 GB of the hard disk drive to store the data.

# APPLICATIONS FILES DESCRIPTION:

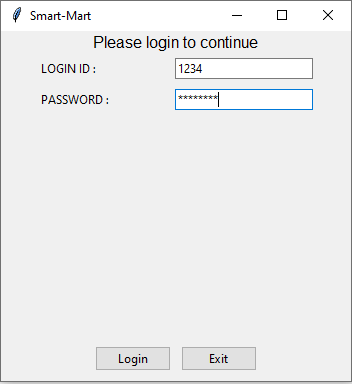
1. App.py:
2. This file contains App class, which is the main class
3. This class contains all the code which is used to generate the application interface, this interface responds to all the events (such as button click)
4. HomePage.py:
5. This file contains code for displaying buttons, by clicking which the respective individual window’s will open to proceed further
6. It has functionality to logout and close the application
7. RegistartionProducts.py:
8. This file contains the code which captures the user data of registering the product, validates the data, then inserts the data into “products” table in database
9. Exception handling is implemented for recording exceptions
10. ShippingProducts.py:
11. This file contains the code which captures the data entered and inserts in “product\_store” table
12. It validates the data by checking whether the product is already registered or not.
13. After successful validations, it inserts data into products\_store table.
14. Exception handling is implemented for recording exceptions
15. IndivualBilling.py:
16. This file contains the code which captures user data entered in billing interface
17. Then it does few validations such as checking whether the product is in the store, checking whether it contains enough number of units(stock).
18. The code for “Submit and Add More” will help user to enter different products data repeatedly, until he wants to stop shopping and pay the bill by generating bill by pressing “Generate Bill” button
19. When a user selects “Generate Bill” all the data captured will be inserted into billing and store\_products\_billing respectively.
20. Exception handling is implemented for recording exceptions
21. SalesReport.py:
22. This file contains code which generates the sales report.
23. StockingReport.py:
24. This file contains code that generate the stocking report.
25. Login.py:
26. This file contains code which captures user credentials and then validates it with the data in login table.
27. After logging in with correct credentials home page will open.
28. MySql.py:
29. This file contains code to connect to the MySQL database.
30. Mysql.connector.connect(host=””, user=””, passwd=””) syntax is used to connect to the database
31. Mysql connection is configured in one class file and reused throughout the application wherever needed
32. Please use the required host name, user and passwd to connect to the database based on individual’s mysql server

# STEP BY STEP USER GUIDE:

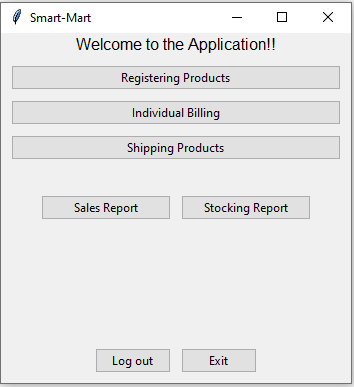
**Login:**

For a user to login into the application, he has to type in

LOGIN ID : 1234, PASSWORD : password



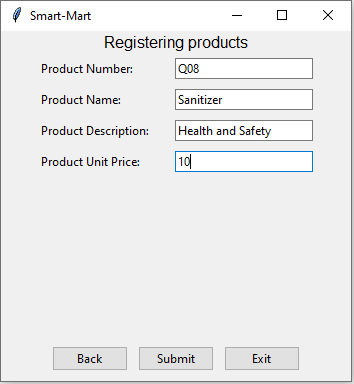
**Home Page:**

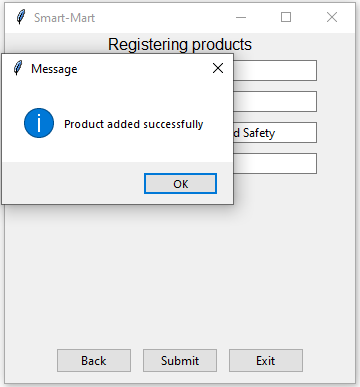
****

After login, user will see a Home Page as shown below

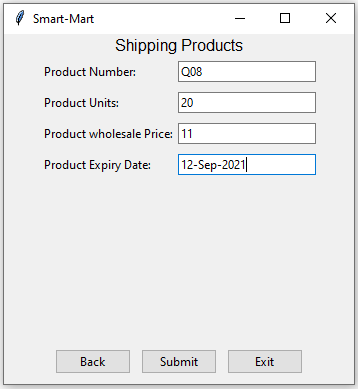
Home page has Seven buttons:

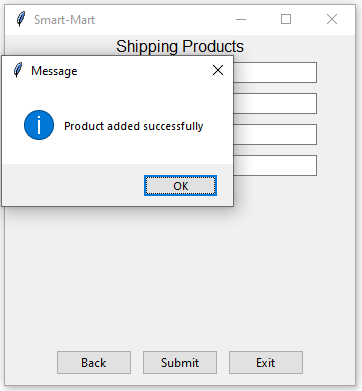
1. **Registering Products:**
   1. All the products our application is dealing with will be registered using this page





1. **Shipping Products:**
   1. All the registered products will be added to the store after being shipped to the store



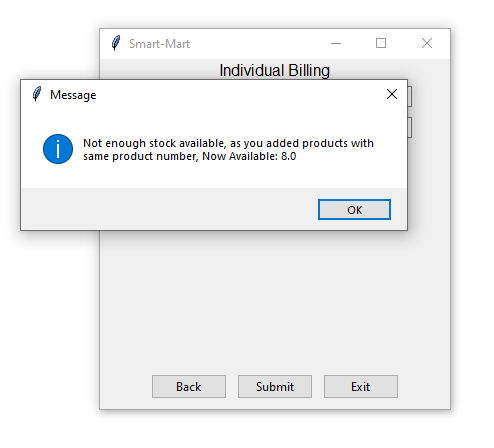


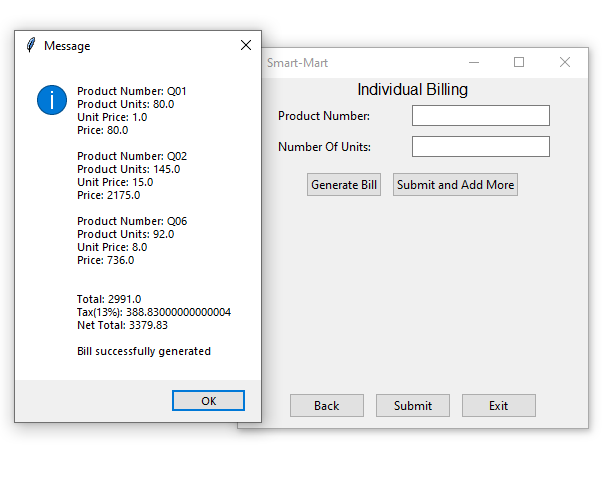
1. **Individual Billing:**
   1. User will input the product number and the number of units of each product until all the products are entered, using “Submit and Add More” button
   2. At the end “Generate Bill” button is used to generate the bill

|  |  |
| --- | --- |
| C:\Users\vishw\Desktop\Python_project_screenshot\Indi_bill_1.PNG | C:\Users\vishw\Desktop\Python_project_screenshot\Indi_bill_2.PNG |

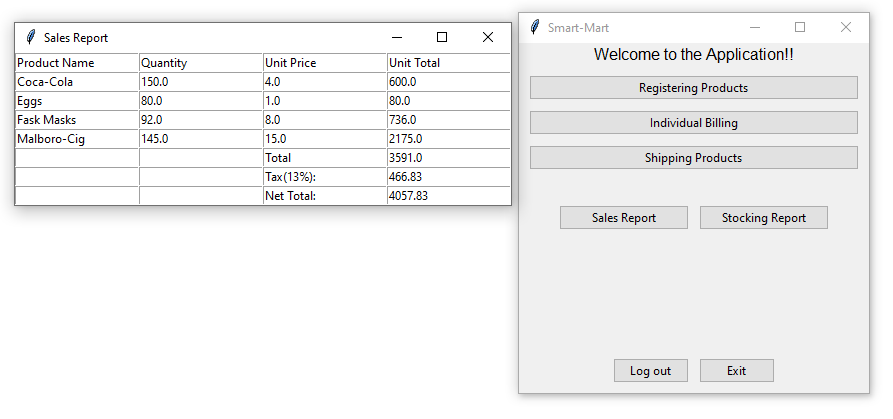
|  |
| --- |
| **C:\Users\vishw\Desktop\Python_project_screenshot\Indi_bill_3.PNG** |

* 1. An error message is shown if stock is not available

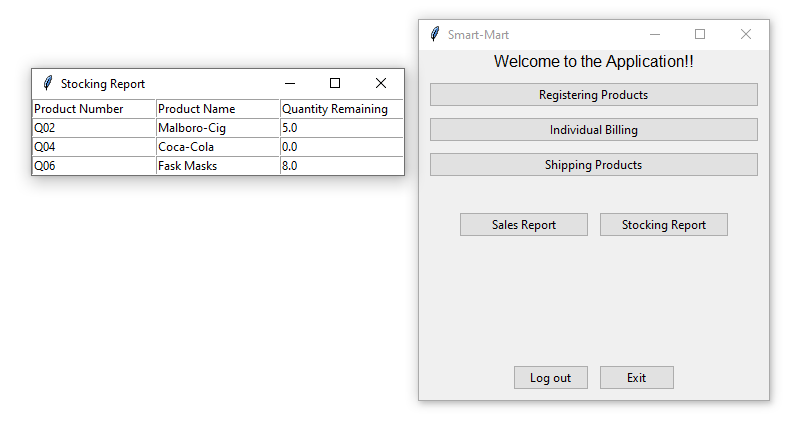
****

* 1. Finally, bill is generated after adding all required products****

1. **Sales Report:**
   1. This report will show the details of all the sales of the day, it contains Product Name, Quantity, Unit Price, Total



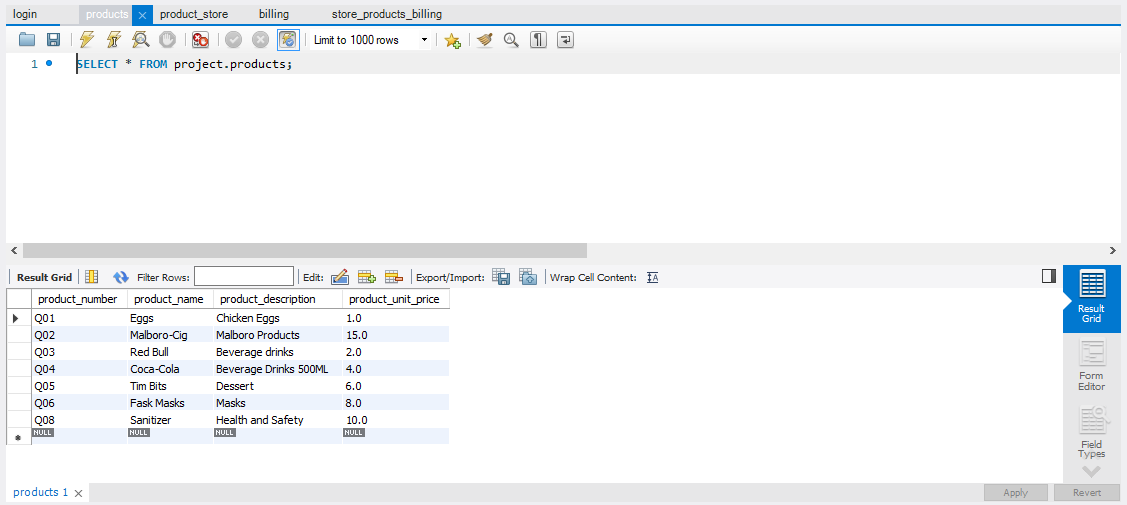
1. **Stocking Report:**
   1. This report shows the products which are about to finish ( < 10 products)



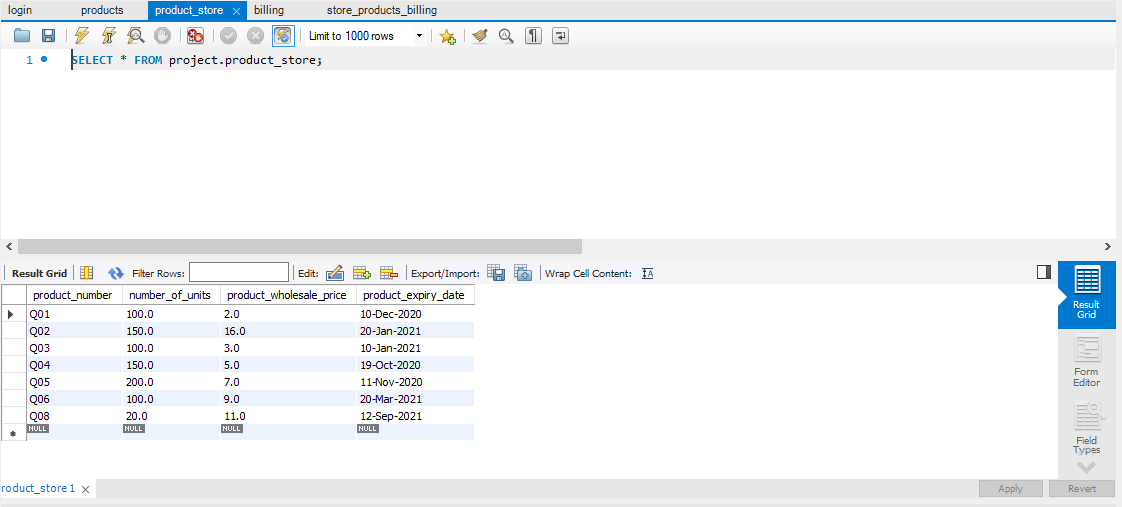
1. **Logout and Exit:**
   1. Logout button is used to come out of the application, if user wants to use the application after logout, he has to login again
   2. Exit button is used to close the application completely

# DETAILS ABOUT DATABASE:

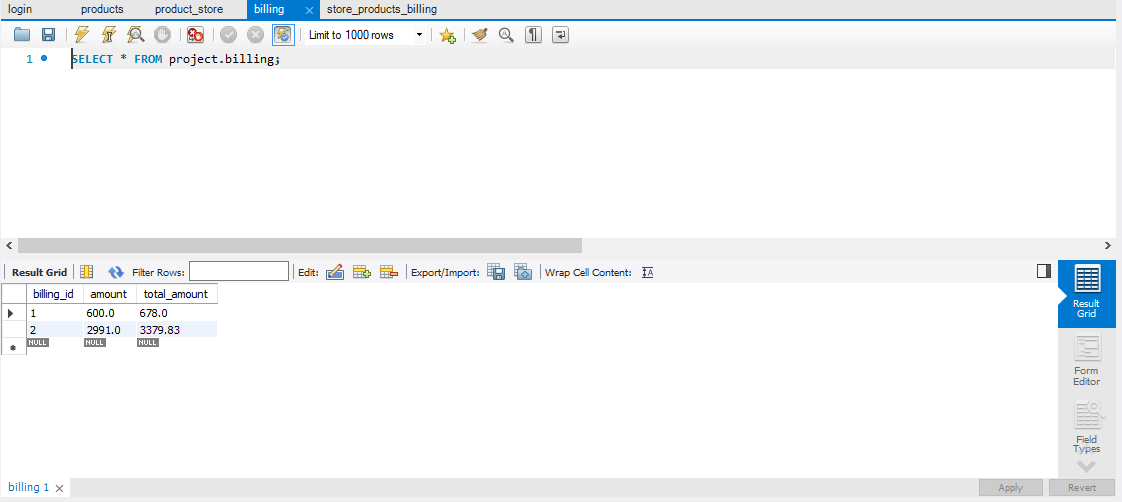
1. Schema Name: project
2. Tables:
3. products:
4. This table consists of all the information regarding registration of the product.



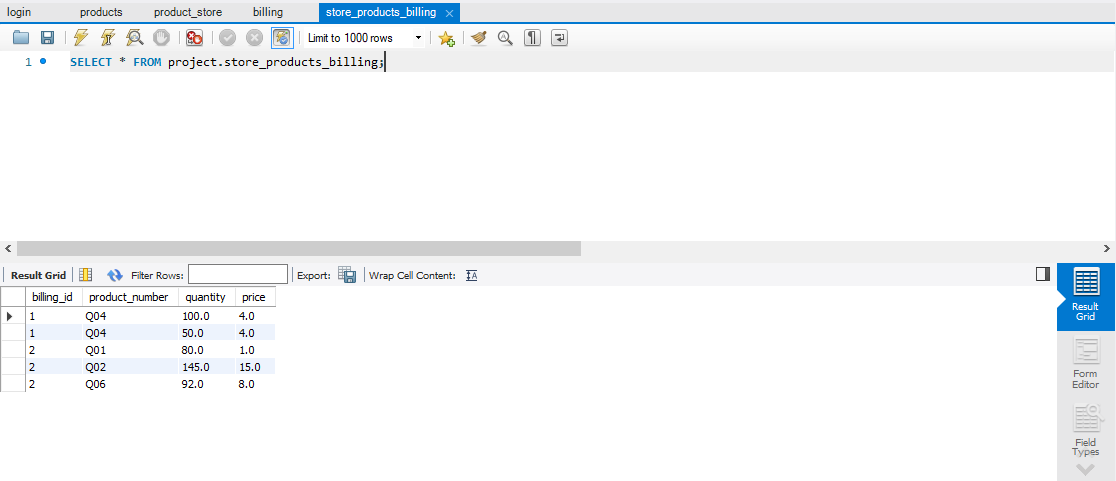
1. product\_store:
2. This table consists of all information regarding products added to the store that are shipped to the store.



1. billing
2. This table consists information regarding each bill
3. billing\_id is auto generated primary key



1. store\_products\_billing
2. This table consists of information of each product for respective bill, billing\_id is the foreign key referencing billing table



1. login:
2. This table consists of information of the user credentials.

