

Warehouse Management System

CMPT 308N

Section 200

The Big Boyz Team



Marist College

School Of Computer Science and Mathematics

Submitted to:
Dr. Reza Sadeghi

Fall 2022

Table of Content

Group Project Selected: Project 4 (Warehouse Management System)	4
Review the related work	4
Entity Relationship Diagram (ER Diagram)	5
Enhanced Entity Relationship Diagram (EER diagram).....	7
Database Development	9
Importing Data	13
Data Manipulation	16
Graphical User Experience	18
Graphical user experience design	18
Reference	28

Table of Figures

Figure 1: The Big Boyz Warehouse Management System ER Diagram	6
Figure 2: The Big Boyz Warehouse Management System EER Diagram.....	8

Progress Report: The Big Boyz Warehouse Management System

Team Name

The Big Boys

Team Members:

Ricky Junior Isheja Rickyjunior.isheja1@marist.edu

My name is Ricky Junior Isheja, I am an international student from Rwanda, and I am a sophomore majoring in computer science, my general passions are coding, basketball, and soccer. I chose my teammates on the basis that we all believe in hard work, and we are all willing to invest in the necessary efforts to finish this project together and succeed in the course.

Saeed Abdilahi Saeed.abdilahi1@marist.edu

My name is Saeed Abdilahi, and I am from Somaliland. I am Junior, studying computer science. The way I selected my current teammates was by asking them if we could be in a group team.

Descartes Tuyishime descartes.tuyishime1@marist.edu (Team Leader)

My name is Descartes Tuyishime, and I am an international student from Rwanda. I am a senior, majoring in Computer Science and Data Analytics. I am interested in the use of machine learning to improve agricultural and medical field. I selected my team based on flexibility, motivation, and responsibility. I admire people who are flexible and motivated to do what needs to be done. I trust my team members to be responsible enough to take this course and project seriously.

Group Project Selected: Project 4 (Warehouse Management System)

The Big boys warehouse management system is a system that will help people to know, manage and maintain materials kept in the warehouse. The System will be managing details like the number of products, their store time, the price and the weight of the products, The system will also help the user to search a specific product. The system will make easier the accounting processes of the user's business which might in the overall increase the profit. The objective of the big boys is to make a warehouse management system by the end of this course that can help an admin user to enter the food types, books, carts and any products, the store time in the warehouse, the pick out time, see the prices and search a product. And we shall also create a user's page for the user to buy and check out products.

Review the related work

❖ Amazon Warehouse

Amazon Warehouse works by label and packaging item and order that came through electronic orders that people who connect with Amazon made. The processes that the packages are labeled and boxed are physically by using human hands [1].

❖ Target Warehouse

Target Warehouses uses similar processes of packaging and distribution and then delivering to the right destination, the same as Amazon. However, the only difference is the brand of the item or product and its price sector [2].

❖ Apple Warehouse

Apple Warehouse is the same as the other two warehouses above, but the difference is that apple mainly focuses on electronics whereas the two provide multiple products. All in all, Apple, Amazon, and Target use the same warehouse management system. All of these are held by humans [3].

Advantages of a Warehouse Management System

- Reduced Operating & Processing Expenses
- Reduced Mispicks
- Improved Customer Relations

Costs that come with having a Warehouse Management Systems

- Requires Expert Knowledge
- Requires Tight Security
- High Initial Investment

Merits Of the Project

1. Our Warehouse Management System (WMS) will lead to a drastic change in operating expenses since it reduces the manpower and other resources that are needed to manage and control the products that are in the warehouse.
2. The Labor will be better allocated with our system since most of the work is done by the system so the labor will have time to do their work in a more effective way while also being efficient.
3. WMS will increase the overall customer satisfaction, since the Service provided will be faster and easier for both the seller and the buyer, which will also increase the customers loyalty to the business thus increased profitability on the businesses side.
4. WMS will facilitate and help to manage the inventory that comes in and gets out (2), which overall will facilitate the accounting process at the end of the fiscal year.

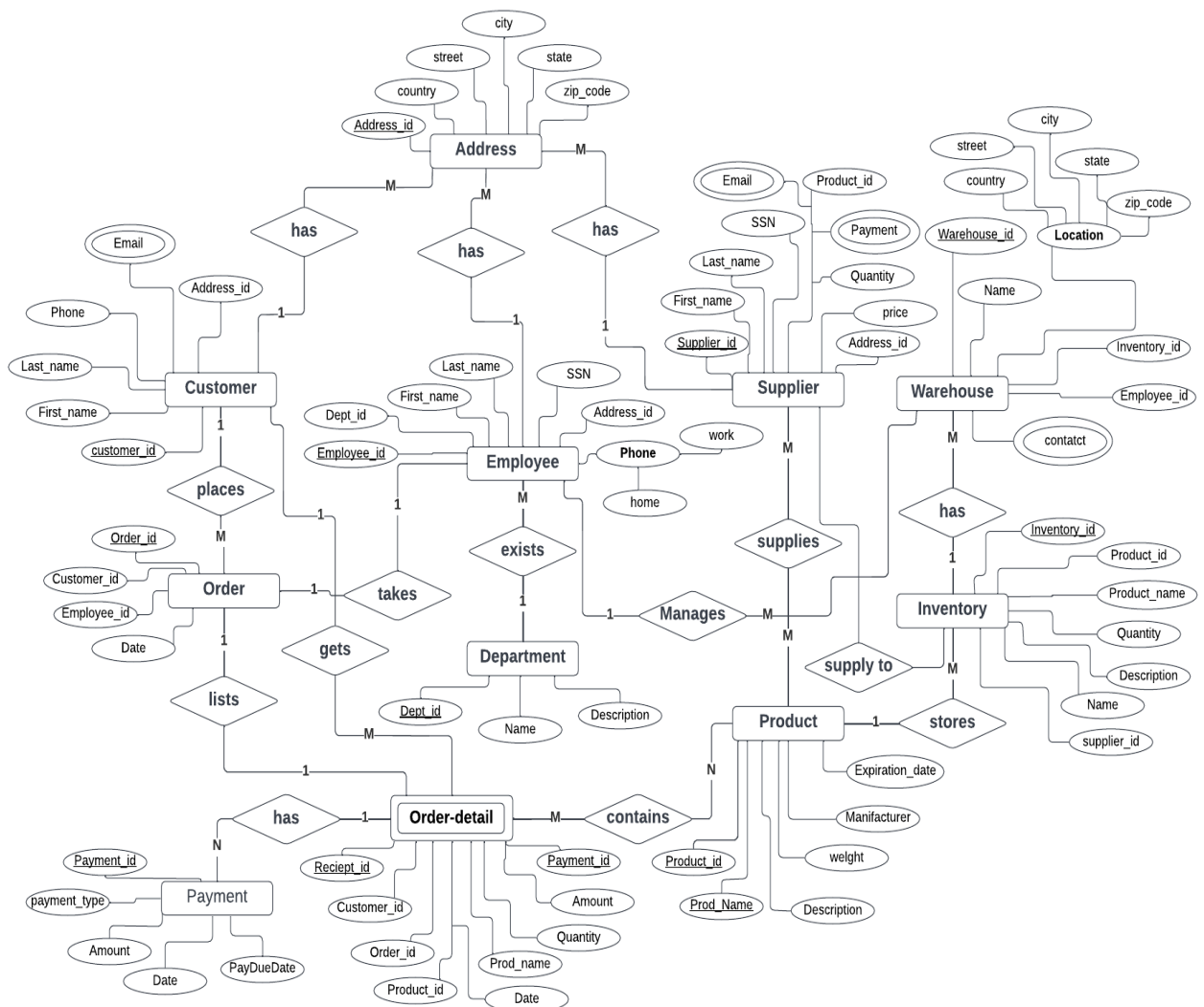
Entity Relationship Diagram (ER Diagram)

Our Warehouse Management System (WMS) consists of 11 entities which are: Warehouse, inventory, supplier, product, employee, department, customer, address, order, order detail, and payment. Below are the business rules that determined the entities in our WMS (Check out figure 1 below.).

A warehouse has a unique identification, a name, location, an inventory, a manager, and contacts for communications. A warehouse has one or more inventories, and an inventory belongs to one and only one warehouse. An inventory has a unique identification, name, product identification, product name, quantity, and product description, supplier identification. Inventory contains stores products, and products have unique identification, product name, description, weight, manufacturer, and expiration date. A warehouse is managed by an employee, and an employee has employee unique identification, department identification, first name, last name, Social Security number, Address identification, and phone number. Each employee belongs to a certain department, and a department has department unique identification, name, and Description. Each product is supplied by a supplier, and a supplier has unique identification, first name, last name, social security number, email, product identification, payment, quantity, and Address identification. Our WMS has customers, and a customer is represented by customer unique identification, first name, last name, email, and address identification. Customer, Employee, and Supplier

all have address, and the Address has address unique identification, country, street, city, state, zip code. A Customer can place an order, and an order is taken by the employee. An order has a unique identification, customer identification, employee identification, and date. Each order has an order detail, and an order detail contains products. Each order detail list receipt unique identification, customer identification, order identification, product identification, date, product name, quantity, amount, and payment identification. Each detail has payment, and each payment consists of unique payment identification, payment type, amount, date, and pay due date. To

Figure 1: The Big Boyz Warehouse Management System ER Diagram



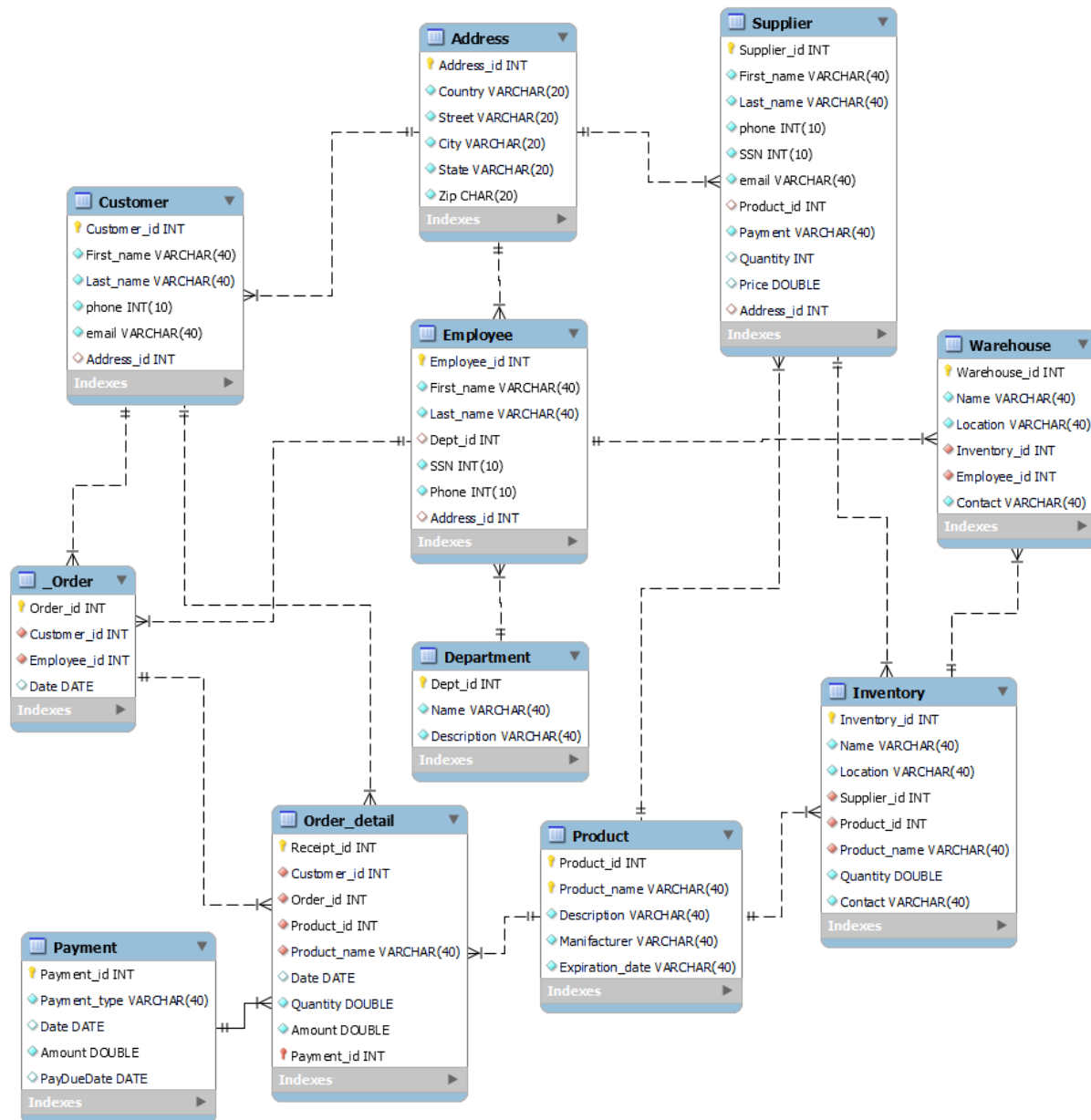
Designed using Lucid.app.

Enhanced Entity Relationship Diagram (EER diagram)

As described above, our warehouse management system is composed of 11 entities which are: Warehouse, inventory, supplier, product, employee, department, customer, address, order, order detail, and payment (Check figure 2).

A warehouse has many inventories, and an inventory belongs to one warehouse. A warehouse entity has warehouse identification (warehouse_id) as the primary key and has foreign keys inventory identification (inventory_id), and employee identification (employee_id) from inventory and employee respectively. The inventory stores many products, and a product is in an inventory. The suppliers supply products to one to more inventory. The inventory entity has primary key in inventory identification (inventory_id), and it has foreign keys in product identification (product_id), product name (product_name), and supplier identification (supplier_id). Supplier entity has supplier identification as a primary key, and it has foreign keys in product identification (product_id) and address identification (address_id). The product entity has product identification (product_id) and product name (product_name) as a composite primary key. The employee entity has employee identification (employee_id) as a primary key. Department entity has department identification (dept_id) as a primary key. Customer entity has customer identification (customer_id) as a primary key. Address entity has address identification (address_id) as a primary key. Order entity has order identification (order_id) as a primary key. Order detail has receipt identification (receipt_id) and payment identification (payment_id) as composite primary keys. Every order detail has a payment, and every payment corresponds to one and only one order detail. The payment entity has payment identification (payment_id) as a primary key.

Figure 2: The Big Boyz Warehouse Management System EER Diagram



Database Development

```
drop database Warehouse;  
create database if not exists Warehouse;  
show databases;  
use Warehouse;
```

Here we just created a database called warehouse but we first checked if there is no other database with the same name

```
CREATE TABLE Address (  
  Address_id INT AUTO_INCREMENT, ,  
  Country varchar(20) NOT NULL,  
  Street varchar(20) NOT NULL ,  
  City varchar(20) NOT NULL ,  
  State varchar(20) NOT NULL ,  
  Zip char(20) NOT NULL,  
  PRIMARY KEY (Address_id)  
);
```

Here we just created a table called address with 5 attributes namely address id (primary key), Country, street, city, state and zip. This table is used to record the addresses of three tables the customer, the supplier and the employee and it will have address_id as foreign key in all of the three tables.

```
CREATE TABLE Customer (  
  Customer_id INT AUTO_INCREMENT,  
  First_name varchar(40) NOT NULL,  
  Last_name varchar(40) NOT NULL ,  
  phone int(10) NOT NULL ,  
  email varchar(40) NOT NULL ,  
  Address_id INT,  
  PRIMARY KEY (Customer_id),  
  FOREIGN KEY (Address_id) REFERENCES Address(Address_id)  
);
```

Here we jus created table customer which will be recording customer details. It has 6 attributes including Customer_id (Primary key), first_name, Last_name, phone, email and address_id (foreign key) from address table. This table will have 3 main connections with the order details table, the order table and the adrerss table.

```
CREATE TABLE Department (  
  Dept_id INT AUTO_INCREMENT,  
  Name varchar(40) NOT NULL,  
  Description varchar(40) NOT NULL ,
```

```
PRIMARY KEY (Dept_id)
);
```

Here we just created the table of Department, this table will be recording the various departments we have in our Warehouse management company. It has 3 main attributes namely dept_id(Primary key), name and description. This table is directly related to the employee table

```
CREATE TABLE Employee (
  Employee_id INT AUTO_INCREMENT,
  First_name varchar(40) NOT NULL,
  Last_name varchar(40) NOT NULL,
  Dept_id INT,
  SSN int(10) NOT NULL ,
  Phone int(10) NOT NULL ,
  Address_id INT,
  PRIMARY KEY (Employee_id),
  FOREIGN KEY (Address_id) REFERENCES Address(Address_id),
  FOREIGN KEY (Dept_id) REFERENCES Department(Dept_id)
);
```

Here we just created a table called employee and its in charge of keeping data of our employees. It has 7 attributes, employee_id (primary_key), first_name, Last_name, dept_id (foreign key), ssn, phone, address_id (foreign_key). This table is related directly to 4 tables, the warehouse table, the address table, the department table and the order table.

```
CREATE TABLE Product (
  Product_id INT AUTO_INCREMENT,
  Product_name varchar(40) NOT NULL,
  Description varchar(40) NOT NULL ,
  Manufacturer varchar(40) NOT NULL ,
  Expiration_date varchar(40) NOT NULL ,
  PRIMARY KEY (Product_id, Product_name )
);
select * from Product;
```

Here we just created the product table, this table records a specific product that is being taken into the inventory or out of the inventory. It has 5 attributes, the product_id (Primary key), product_name, description, Manufacturer and expiration date. This table is related to three other tables, the order_detail,

```
CREATE TABLE Supplier (
  Supplier_id INT AUTO_INCREMENT,
  First_name varchar(40) NOT NULL,
  Last_name varchar(40) NOT NULL ,
  phone int(10) NOT NULL ,
  SSN int(10) NOT NULL ,
  email varchar(40) NOT NULL ,
  Product_id INT,
  Payment varchar(40) NOT NULL,
```

```
Quantity INT,  
Price DOUBLE,  
Address_id INT,  
PRIMARY KEY (Supplier_id),  
FOREIGN KEY (Address_id) REFERENCES Address(Address_id),  
FOREIGN KEY (Product_id) REFERENCES Product(Product_id)  
);
```

Here we just created a table called supplier which deals with the suppliers' information and the information of what they supplied. It has 11 attributes, supplier_id (Primary Key), first_name, last_name, phone, ssn, email, Product_id (Foreign Key), Payment, quantity, price, address (Foreign Key). This table is directly connected to three tables, Address, Product and inventory.

```
CREATE TABLE _Order (  
  Order_id INT AUTO_INCREMENT,  
  Customer_id INT NOT NULL,  
  Employee_id INT NOT NULL ,  
  Date date DEFAULT NULL,  
  PRIMARY KEY (Order_id),  
  FOREIGN KEY (Customer_id) REFERENCES Customer(Customer_id),  
  FOREIGN KEY (Employee_id) REFERENCES Employee(Employee_id)  
);
```

Here we jus created a table of order, it handles the order information and saves it. This table has 4 attributes, Order_id (Primary_Key) , customer_id (Foreign_Key), Employee_Id (Foreign Key) and the date. This table is directly related to the customer table, the order detail table and the employee table.

```
CREATE TABLE Payment (  
  Payment_id INT AUTO_INCREMENT,  
  Payment_type varchar(40) NOT NULL,  
  Date date DEFAULT NULL,  
  Amount double NOT NULL,  
  PayDueDate date DEFAULT NULL,  
  PRIMARY KEY (Payment_id)  
);
```

The payment table that we just created records the payment details related to the customer and the order detail. It has five attributes, Payment_id(Primary Key), payment_type, date, amount and paydueDate. This table is related to order detail table.

```
CREATE TABLE Order_detail (  
  Receipt_id INT AUTO_INCREMENT,  
  Customer_id INT NOT NULL,  
  Order_id INT NOT NULL,  
  Product_id INT NOT NULL,  
  Product_name varchar(40) NOT NULL,  
  Date date DEFAULT NULL,  
  Quantity double NOT NULL,  
  Amount double NOT NULL,  
  Payment_id INT NOT NULL,  
  FOREIGN KEY (Product_id, Product_name) REFERENCES Product(Product_id, Product_name),
```

```
FOREIGN KEY (Customer_id) REFERENCES Customer(Address_id),  
FOREIGN KEY (Order_id) REFERENCES _Order(Order_id),  
FOREIGN KEY (Payment_id) REFERENCES Payment(Payment_id),  
PRIMARY KEY (Receipt_id, Payment_id)  
);
```

The order detail table is a table that records information of the order that has been made. It has nine attributes, Receipt_id (Primary Key), customer_id (Foreign_key), Order_id(Foreign Key), Product_id (Foreign_key), Product_name(Foreign_key), date, quantity,amount, payment_id(Primary Key). This table is directly related to Customer table, Payment table, order table, product table.

```
CREATE TABLE Inventory (  
Inventory_id INT AUTO_INCREMENT,  
Name varchar(40) NOT NULL,  
Location varchar(40) NOT NULL ,  
Supplier_id INT NOT NULL ,  
Product_id INT NOT NULL ,  
Product_name varchar(40) NOT NULL ,  
Quantity double NOT NULL,  
Contact varchar(40) NOT NULL ,  
PRIMARY KEY (Inventory_id),  
FOREIGN KEY (Supplier_id) REFERENCES Supplier(Supplier_id),  
FOREIGN KEY (Product_id, Product_name) REFERENCES Product(Product_id, Product_name)  
);
```

Here we just created a table called inventory, it will hold details of the products that are in the inventory. This table has eight attributes, Inventory_id(Primary_key), name,location, supplier_id (Foreign_key), product_id(Foreign Key), Product_name(Foreign_Key), quantity, contact. This table is related to the supplier table, the warehouse supplier and the products table.

```
CREATE TABLE Warehouse (  
Warehouse_id INT AUTO_INCREMENT,  
Name varchar(40) NOT NULL,  
Location varchar(40) NOT NULL ,  
Inventory_id INT NOT NULL ,  
Employee_id INT NOT NULL ,  
Contact varchar(40) NOT NULL ,  
PRIMARY KEY (Warehouse_id),  
FOREIGN KEY (Employee_id) REFERENCES Employee(Employee_id),  
FOREIGN KEY (Inventory_id) REFERENCES Inventory(Inventory_id)  
);
```

We just created a table called Warehouse, this table is of recording what enters and gets out of our warehouse. It has six attributes, warehouse_id (Primary_key), name, location, Inventory (Foreign Key), Employee(Foreign Key), Contact. This table is connected directly to the employee table and to the inventory.

```
create table roles( Role_name varchar(80), Role_specification varchar(80));
```

Here we just created a table called roles that will be recording a user id whether the user is a supplier, Admin, or buyer.

Importing Data

Inserting data error.

22	16:37:43	insert into Customer values ('Jacob','jane',646785893,'bhac@gmail.com',1), ('Job','Mark',646720893,'bcdc@gmail.com',2), ('Bill',...	Error Code: 1136: Column count doesn't match value count at row 1	0.000 sec
----	----------	---	---	-----------

We set the foreign_key_checks to zero before importing data into the tables. Then we set foreign_key_check to 1 to maintain the integrity of the tables.

```
insert into Address(Country,street,city,state,zip) values ('Rwanda','45lake','kigali','Kimironko',0000),
('USA','45lake','NY','Poughkeepsie',2435),
('USA','75clark','NY','fishkill',7890),
('Belgium','45AVE','La ville','manic',335),
('USA','45lake','NY','Poughkeepsie',2435),
('USA','45lake','NY','Poughkeepsie',2435),
('USA','45lake','NY','Poughkeepsie',2435),
('USA','45lake','NY','Poughkeepsie',2435),
('USA','45lake','NY','Poughkeepsie',2435),
('USA','5lake','LA','newtown',1565);

select * from Address;
```

Here we just inserted data into the Address table in the rows of country, city, state and zip code. We inserted 10 values in each row.

```
set foreign_key_checks=0;
```

```
insert into customer(First_name,Last_name,phone,email,Address_id)
values
('Jacob','jane',646785893,'bhac@gmail.com',1),
('Job','Mark',646720893,'bcdc@gmail.com',2),
('Bill','Toamso',657867786,'hbah@gmail.com',3),
('Nico','jane',646785893,'bhac@gmail.com',6),
('Ricky','Isheja',646785893,'bhac@gmail.com',8),
('Descartes','Abdilahi',646785893,'bhac@gmail.com',9),
('Saeed','Musoni',646785893,'bhac@gmail.com',10);
```

```
set foreign_key_checks=1;
```

```
select * from customer;
```

Here we to avoid the foreign key problem we used the set check method which will deactivate the foreign key check when we enter the data and reactivate it after we enter it. We then entered 10 values in each row of the customer table.

```
insert into department(name,description)values('IT','In charge of IT'),
```

```
('HR','In charge of Human Resources'),('Operations','In charge of all business related Operations');
```

```
select * from department;
```

Here we inserted three departments in the departments table with their descriptions and we displayed the output.

```
insert into employee(First_name,Last_name,Dept_id,SSN,phone,Address_id)
values('John','doe',1,000000222,222111333,9),
```

```
('John','doe',1,000000333,111222333,3),
```

```
('Jo','doe',2,000000444,444222333,4),
```

```
('Jahn','doe',3,000000555,000111222,5),
```

```
('ohn','doe',3,000000678,212323434,6),
```

```
('Ricky','doe',3,000000777,121212121,7),
```

```
('Junior','doe',1,000000888,111000222,8),
```

```
('Descartes','doe',2,000000999,909090909,9),
```

```
('Saeed','doe',2,000000181,756876890,2);
```

```
select * from employee;
```

Here we just inserted 8 employees in the employees' table and we also added information for 6 entities namely firstname, lastname, department id as foreign key, social security number, phone number and address id . We never faced a foreign key problem because we inserted data correctly in the right order.

```
insert into
```

```
Product(Product_name,description,manufacturer,Expiration_date)values('riham','biscuit','axzam',12/3/2021),
```

```
('MT dew','Drink','dew',1/3/2021),
```

```
('T-shirt','Clothes','Gucci',12/4/2021),
```

```
('Hoddie','Clothes','LV',12/3/2021),
```

```
('laptop','Tech','HP',13/4/2022),
```

CMPT308N-200_Project Progress Report Phase #5_The Big Boyz Team

('Macbook pro','tech','Apple',4/3/2022),

('Books','paperterie ','axzam',12/3/2023);

Here we just inserted information in the products table and its 5 entities namely product name, description, manufacturer and expiration date

insert into

Supplier(First_name,Last_name,phone,SSN,email,payment,Quantity,Price,Address_id,Product_id)

values('John','Doe',832456789,000000111,'jaha@gxc.com','Credit card',5,5000,1,1),

('John','Doe',832456908,000000111,'jaha@gxc.com','Debit Card',5,2000,9,5),

('John','Doe',832263876,000000111,'jaha@gxc.com','Credit card',5,3000,3,1),

('John','Doe',832798725,000000111,'jaha@gxc.com','credit card',5,4000,2,2),

('John','Doe',832798723,000000111,'jaha@gxc.com','credit card',5,5000,1,3),

('John','Doe',832798725,000000111,'jaha@gxc.com','credit card',5,7000,4,5),

('John','Doe',832798722,000000111,'jaha@gxc.com','credit card',5,1000,5,7);

select * from Supplier;

Here we just inserted data of the suppliers' details with its entities. We entered 7 values in each entity successfully.

insert into Orders(Customer_id,Employee_id)values(1,2),(3,4),(1,6);

select * from Orders;

We just entered values for the order details table which holds data for the details of every product.

insert into

Payment(Payment_type,Date,Amount,PayDueDate)value('Check',12/3/2021,2000,24/3/2021),

('Debit Card',12/6/2021,2000,24/12/2021),

('Check',12/5/2021,6000,24/11/2021),

('Credit Card',12/7/2021,5000,24/10/2021),

('Check',12/8/2021,200,24/9/2021);

select * from payment;

Here we just entered the way the customers paid for the products that they bought and we entered 5 values for each entity

insert into Inventory(Name,Location,Quantity,Contact)value('biscuits','3rd',2000,387873),

('Drinks','4th',2000,384573),

```
(('Clothes','1st',2000,383873),
('Tech','5th',2000,387323),
('biscuits','3rd',2000,387873),
('biscuits','3rd',2000,387873),
('biscuits','3rd',2000,387873),
('biscuits','3rd',2000,387873);
```

Here we just entered products in the inventory table, each product has its name, location, quantity, and contact and we added 8 products to the inventory.

Data Manipulation

	Customer_id	First_name	Last_name	phone	email	Address_id
▶	1	Jacob	jane	646785893	bhac@gmail.com	1
	2	Jacob	jane	646785893	bhac@gmail.com	1
	3	Job	Mark	646720893	bcdc@gmail.com	2
	4	Jacob	jane	646785893	bhac@gmail.com	1
	5	Job	Mark	646720893	bcdc@gmail.com	2
	6	Bill	Toamso	657867786	hbah@gmail.com	3
	7	Nico	jane	646785893	bhac@gmail.com	6
	8	Ricky	Isheja	646785893	bhac@gmail.com	8

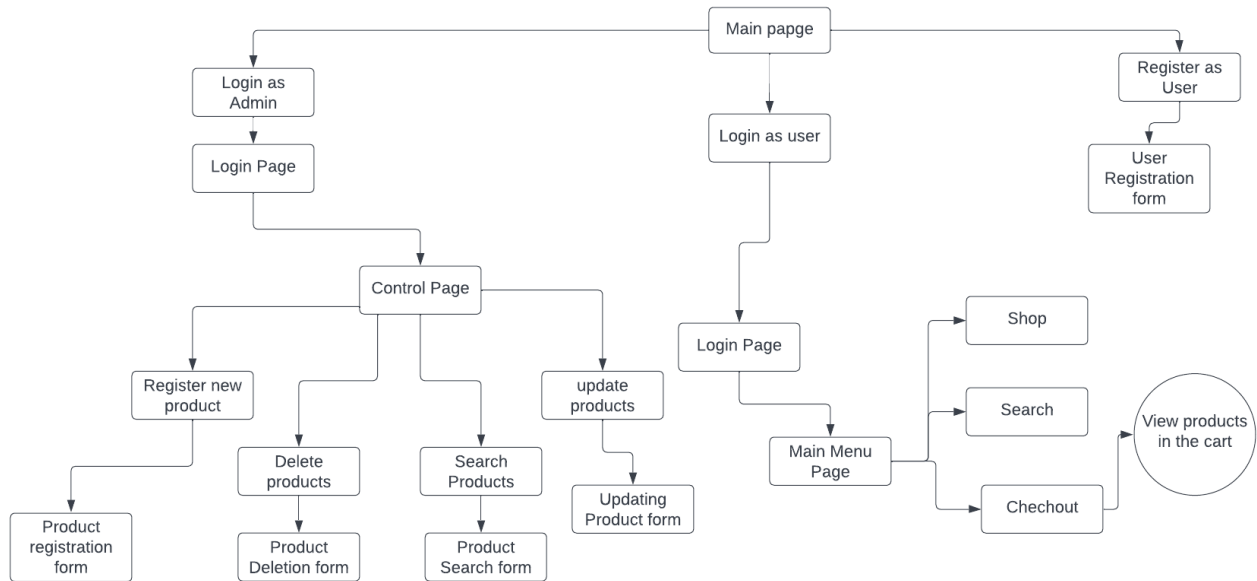
Alter table Customer add column parents varchar(30);

Alter table customer modify column parents int;

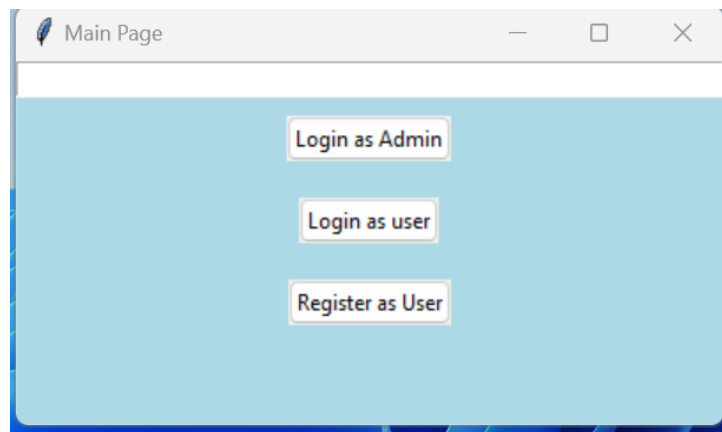
#	Field	Schema	Table	Type
1	Customer_id	warehouse	customer	INT
2	First_name	warehouse	customer	VARCHAR
3	Last_name	warehouse	customer	VARCHAR
4	phone	warehouse	customer	INT
5	email	warehouse	customer	VARCHAR
6	Address_id	warehouse	customer	INT
7	parents	warehouse	customer	INT

Alter table customer drop column parents;

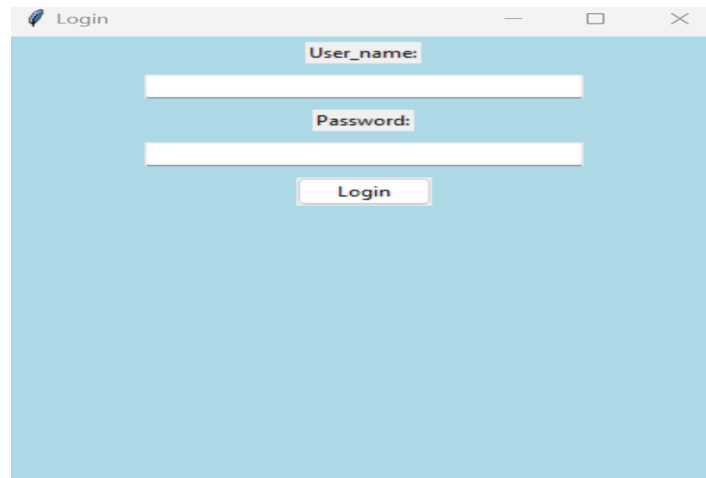
Graphical User Experience



Graphical user experience design

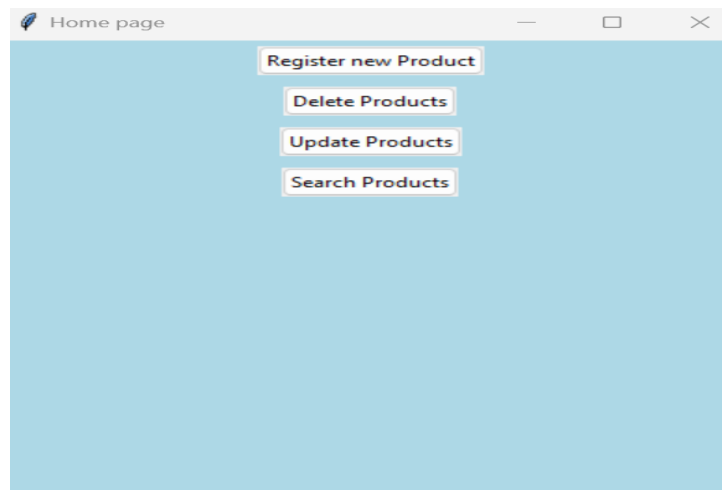


Here we have the main page which asks the user if they want to login as user which means that they are already in database or login as admin which means that they will have the access to delete, add or edit products.



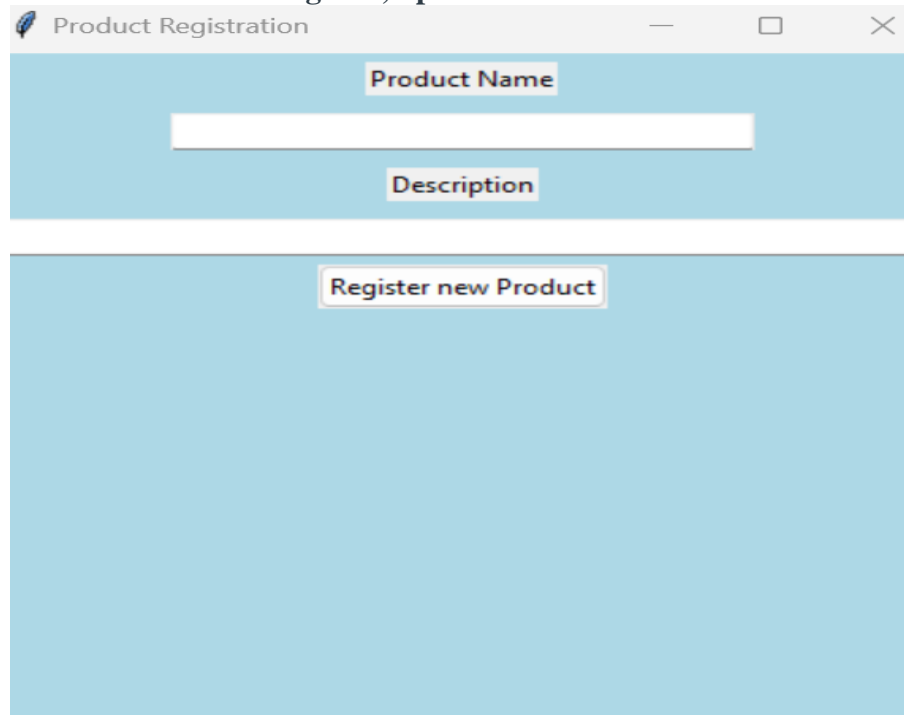
A screenshot of a web browser window titled "Login". The window has a light blue background. At the top, there is a label "User_name:" followed by a white text input field. Below this is a label "Password:" followed by another white text input field. At the bottom of the form is a button labeled "Login".

The next page is the admin login page which checks the data that is in the users table and compares it to the roles table and sees if the person who logs in is the Admin.



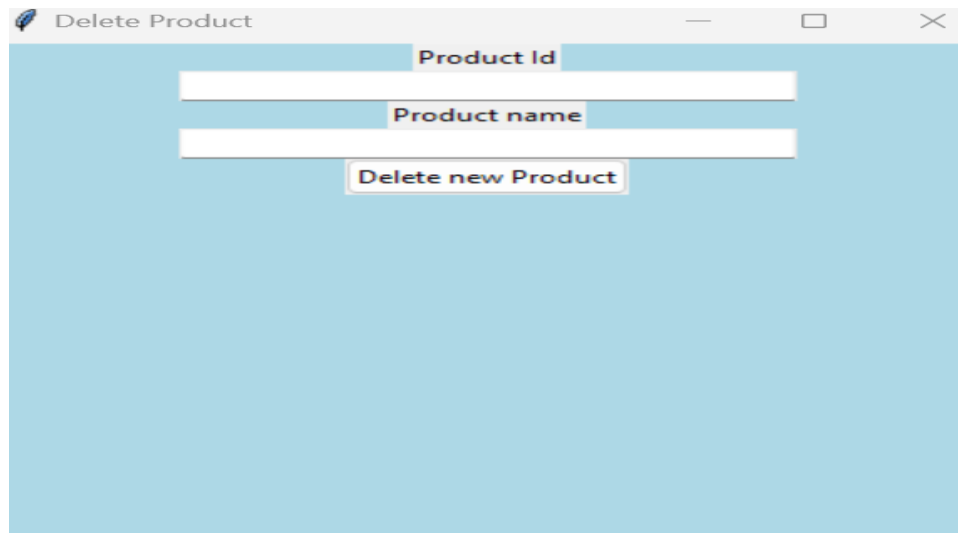
A screenshot of a web browser window titled "Home page". The window has a light blue background. In the center, there is a vertical stack of four buttons: "Register new Product", "Delete Products", "Update Products", and "Search Products".

The admin is then prompted to continue to the control page which allows them to delete, register, update and search



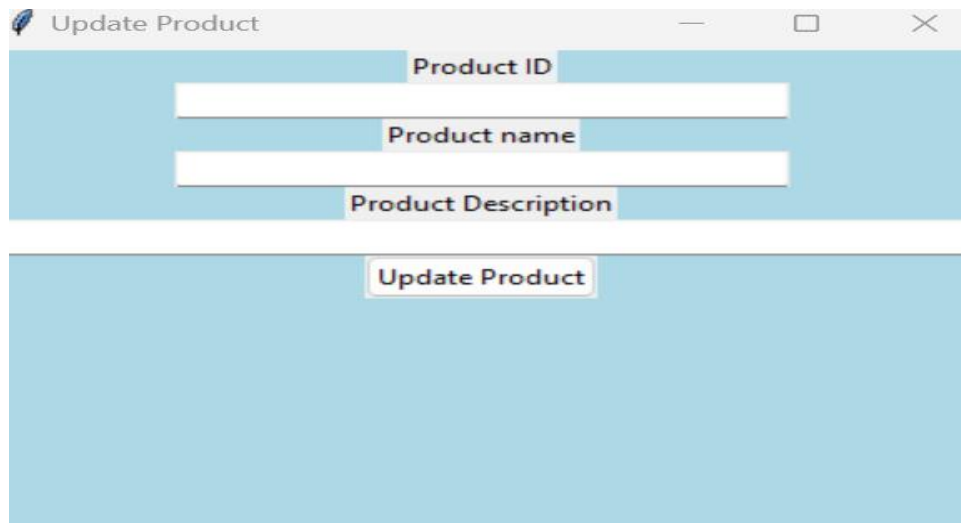
A screenshot of a web application window titled "Product Registration". The window has a light blue background and a white title bar with standard window controls. It contains two text input fields: "Product Name" and "Description". Below these fields is a button labeled "Register new Product".

The admin can add a new product using the following interface



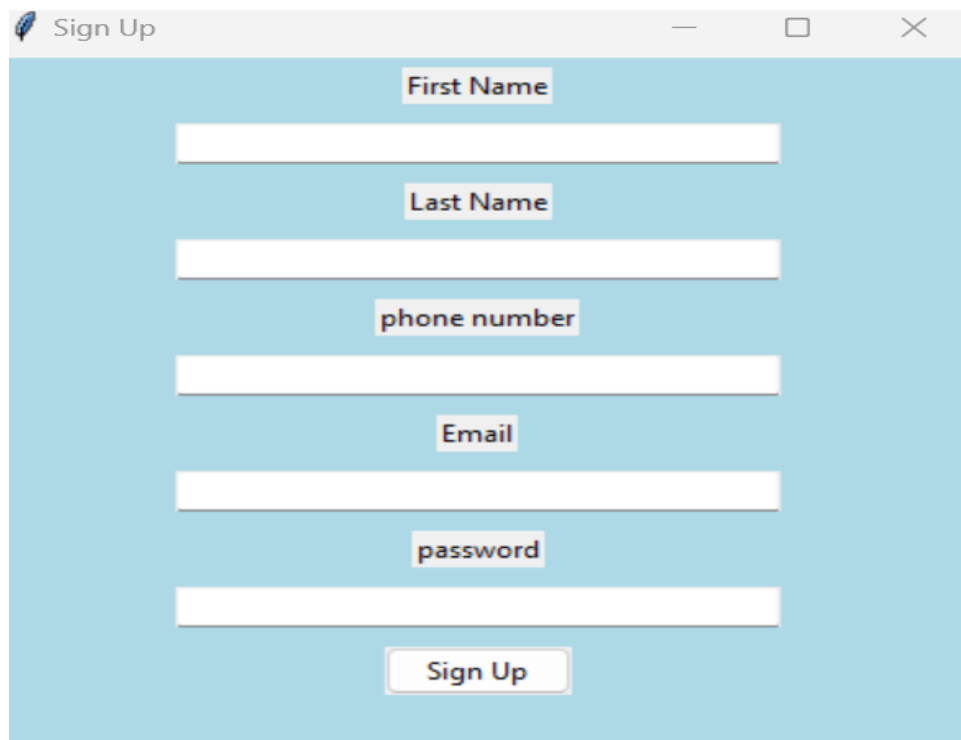
A screenshot of a web application window titled "Delete Product". The window has a light blue background and a white title bar with standard window controls. It contains two text input fields: "Product Id" and "Product name". Below these fields is a button labeled "Delete new Product".

They can also delete the products in the table



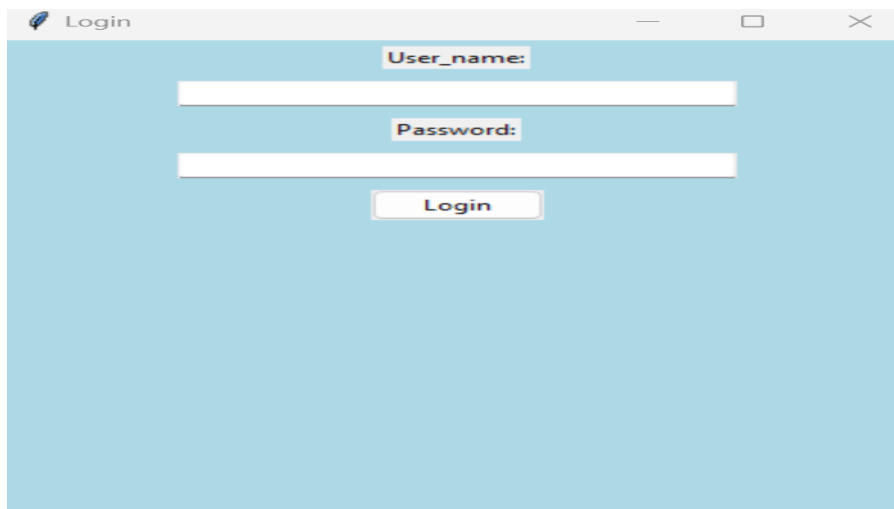
A screenshot of a web application window titled "Update Product". The window has a light blue background and a white title bar with standard window controls. The form contains three text input fields stacked vertically, each with a label above it: "Product ID", "Product name", and "Product Description". Below these fields is a single "Update Product" button.

The admin can also update products



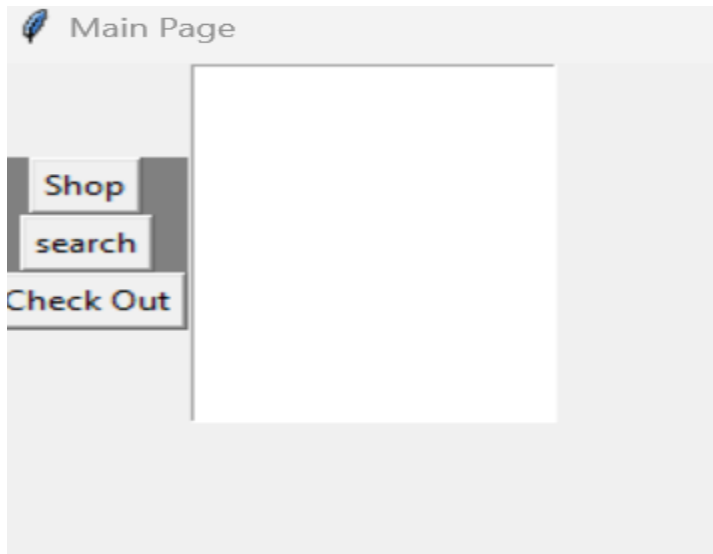
A screenshot of a web application window titled "Sign Up". The window has a light blue background and a white title bar with standard window controls. The form contains five text input fields stacked vertically, each with a label above it: "First Name", "Last Name", "phone number", "Email", and "password". Below these fields is a single "Sign Up" button.

Besides that we can also register a new customer and send them to the customer table where their information will be recorded for login purposes



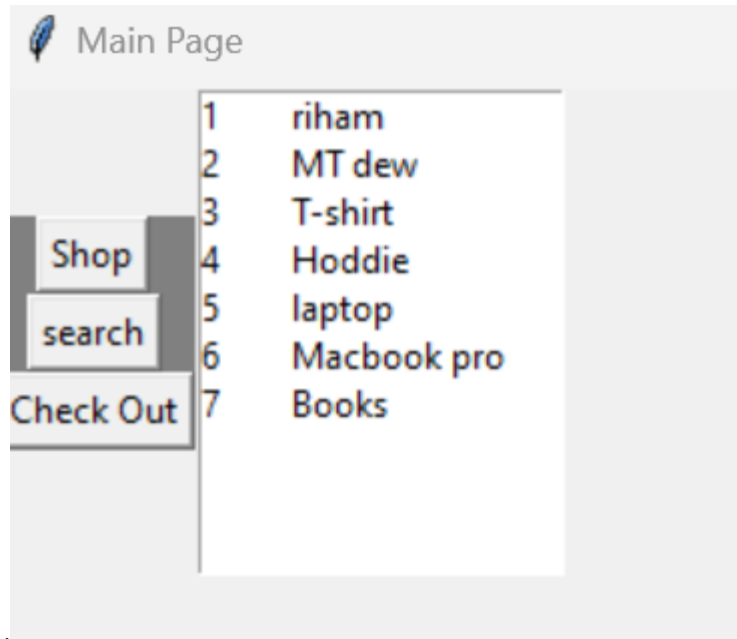
A screenshot of a web application window titled "Login". The window has a light blue background. It contains three input fields: a "User_name:" field, a "Password:" field, and a "Login" button. The "User_name:" field is highlighted with a red border. The "Password:" field is also highlighted with a red border. The "Login" button is highlighted with a red border.

They can then login and start buying products.



A screenshot of a web application window titled "Main Page". The window has a light gray background. On the left side, there is a vertical sidebar with three buttons: "Shop", "search", and "Check Out". The "Shop" button is highlighted with a red border. The "search" button is also highlighted with a red border. The "Check Out" button is highlighted with a red border. The main area of the window is a large white rectangle, which is currently empty.

This is the interface that allows them to shop and when they click on shop they can see the products in the list box



As seen in the list box now products appear and the user list box.

User Interface and Connection Code

```
import tkinter
from tkinter import *
from tkinter.ttk import *
import mysql.connector
import tkinter.messagebox as messagebox

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="rootroot",
    port="3306",
    database="Warehouse"
)
cursor = mydb.cursor()

main_window= Tk()
main_window.geometry("400x200")
main_window.title("Main Page")
main_window.configure(background='lightblue')
t=[]

def Login_page():
```

```
global new_window
new_window = Toplevel(main_window)
new_window.geometry("400x400")
new_window.title("Login")
new_window.configure(background='lightblue')
Username_Label = Label(new_window, text="User_name:")
Uname_Entry = Entry(new_window, width=40)
Password_Label = Label(new_window, text="Password:")
Password_Entry= Entry(new_window, width=40)
Login_btn = Button(new_window, text="Login", command=customerview)
Username_Label.pack(padx=5,pady=5)
Uname_Entry.pack(padx=5,pady=5)
Password_Label.pack(padx=5,pady=5)
Password_Entry.pack(padx=5,pady=5)
Login_btn.pack(padx=5,pady=5)
def Login_admin():
    global new_window
    new_window = Toplevel(main_window)
    new_window.geometry("400x400")
    new_window.title("Login")
    new_window.configure(background='lightblue')
    Username_Label = Label(new_window, text="User_name:")
    Uname_Entry = Entry(new_window, width=40)
    Password_Label = Label(new_window, text="Password:")
    Password_Entry= Entry(new_window, width=40)
    Login_btn = Button(new_window, text="Login", command=main)
    Username_Label.pack(padx=5,pady=5)
    Uname_Entry.pack(padx=5,pady=5)
    Password_Label.pack(padx=5,pady=5)
    Password_Entry.pack(padx=5,pady=5)
    Login_btn.pack(padx=5,pady=5)

def Sign_up():
    global new_window2
    global Fname_Entry
    global Lname_textbox
    global Phone_textbox
    global Email_textbox
    global Pass_textbox
    new_window2= Toplevel(main_window)
    new_window2.geometry("400x400")
    new_window2.title("Sign Up")
    new_window2.configure(background="lightblue")
    Fname_label=Label(new_window2,text="First Name")
    Fname_Entry=Entry(new_window2,width=40)
    Lname_label= Label(new_window2, text="Last Name")
    Lname_textbox = Entry(new_window2, width=40)
    Phone_label = Label(new_window2, text="phone number")
    Phone_textbox = Entry(new_window2, width=40)
    Email_label = Label(new_window2, text="Email")
    Email_textbox = Entry(new_window2, width=40)
    Pass_label= Label(new_window2, text="password")
```


CMPT308N-200_Project Progress Report Phase #5_The Big Boyz Team

```
Pass_textbox = Entry(new_window2, width=40)
sign_btn = Button(new_window2, text="Sign Up", command=insert)
Fname_label.pack(padx=5,pady=5)
Fname_Entry.pack(padx=5,pady=5)
Lname_label.pack(padx=5,pady=5)
Lname_textbox.pack(padx=5,pady=5)
Phone_label.pack(padx=5,pady=5)
Phone_textbox.pack(padx=5,pady=5)
Email_label.pack(padx=5,pady=5)
Email_textbox.pack(padx=5,pady=5)
Pass_label.pack(padx=5,pady=5)
Pass_textbox.pack(padx=5,pady=5)
sign_btn.pack(padx=5,pady=5)

def insert():
    First_name=Fname_Entry.get()
    Last_name=Lname_textbox.get()
    Phone=Phone_textbox.get()
    Email=Email_textbox.get()
    C_password=Pass_textbox.get()

    if (First_name=="" or Last_name=="" or C_password==""):
        messagebox.showinfo("Insert satus", "Missing Fields")
    else:
        cursor.execute("insert into Customer
values('"+First_name+"','"+Last_name+"','"+Phone+"','"+Email+"','"+C_password+"')")
        cursor.execute("commit");
        messagebox.showinfo("Insert status", "Inserted Succesfully")
        cursor.close();

def main():
    new_window3 = Toplevel(main_window)
    new_window3.geometry("400x400")
    new_window3.title("Home page")
    new_window3.configure(background="lightblue")

    register_product=Button(new_window3,text="Register new Product", command=register_p)
    delete_product=Button(new_window3,text="Delete Products",command=delete_p)
    Update_product=Button(new_window3,text="Update Products",command=update_p)
    search_products=Button(new_window3,text="Search Products")
    register_product.pack(padx=5,pady=5)
    delete_product.pack(padx=5,pady=5)
    Update_product.pack(padx=5,pady=5)
    search_products.pack(padx=5,pady=5)

def register_p():
    global pname_box
    global pdesc_box
    new_window4= Toplevel(main_window)
    new_window4.geometry("400x400")
    new_window4.title("Product Registration")
    new_window4.configure(background="lightblue")
    p_name=Label(new_window4,text="Product Name")
```

CMPT308N-200_Project Progress Report Phase #5_The Big Boyz Team

```

pname_box=Entry(new_window4,width=40)
p_desc=Label(new_window4,text="Description")
pdesc_box=Entry(new_window4,width=70)
p_name.pack(padx=5,pady=5)
pname_box.pack(padx=5,pady=5)
p_desc.pack(padx=5,pady=5)
pdesc_box.pack(padx=5,pady=5)
register_product = Button(new_window4, text="Register new Product",
command=Register_con)
register_product.pack()
def Register_con():
    Product_name=pname_box.get()
    Description=pdesc_box.get()

    if (Product_name==" " or Description==""):
        messagebox.showinfo("Insert satus", "Missing Fields")
    else:
        cursor.execute("Update Product set description='"+Description+"' where
Product_id='"+Product_name+"'")
        cursor.execute("commit");
        messagebox.showinfo("Insert status", "Inserted Succesfully")
        cursor.close();

def delete_p():
    new_window5= Toplevel(main_window)
    new_window5.geometry("400x400")
    new_window5.title("Delete Product")
    new_window5.configure(background="lightblue")
    idlabel=Label(new_window5,text="Product Id")
    p_id=Entry(new_window5,width=40)
    name_label=Label(new_window5,text="Product name")
    pname=Entry(new_window5,width=40)
    idlabel.pack()
    p_id.pack()
    name_label.pack()
    pname.pack()
    delete_product = Button(new_window5, text="Delete new Product", command='')
    delete_product.pack()

def update_p():
    new_window6 = Toplevel(main_window)
    new_window6.geometry("400x400")
    new_window6.title("Update Product")
    new_window6.configure(background="lightblue")
    pid=Label(new_window6,text="Product ID")
    id_box=Entry(new_window6,width=40)
    namelabel= Label(new_window6, text="Product name")
    pname= Entry(new_window6, width=40)
    productdesc = Label(new_window6, text="Product Description")
    pdesc = Entry(new_window6, width=70)
    pid.pack()
    id_box.pack()
    namelabel.pack()
```

CMPT308N-200_Project Progress Report Phase #5_The Big Boyz Team

```
pname.pack()
productdesc.pack()
pdsc.pack()
Update_product = Button(new_window6, text="Update Product", command='')
Update_product.pack()

def customerview():
    global list
    new_window7= Toplevel(main_window)
    new_window7.geometry("700x700")
    frmleft = tkinter.Frame(new_window7, bg='grey', width=300, height=1000)
    frmright = tkinter.Frame(new_window7, bg='lightblue', width=1000, height=1000)
    frmleft.grid(row=0, column=0)
    frmright.grid(row=0, column=1)
    products_btn=tkinter.Button(frmleft,text="Shop")
    products_btn.grid(row=2,column=1)
    search_btn = tkinter.Button(frmleft, text="search")
    search_btn.grid(row=3,column=1)
    checkout_btn=tkinter.Button(frmleft,text="Check Out",command=show)
    checkout_btn.grid(row=4,column=1,)
    list=Listbox(frmright,)
    list.grid(row=1,column=2)

def show():
    cursor.execute("Select * from Product")
    Products = cursor.fetchall();

    for product in Products:
        insertdata=str(product[0])+ ' '+product[1]
        list.insert(list.size()+1,insertdata)
        cursor.close()

Title = Text(main_window, height =1, width = 50)
login=Button(main_window, text="Login as Admin", command=Login_admin)
login_user=Button(main_window, text="Login as user", command=Login_page)
sign_up=Button(main_window, text="Register as User", command=Sign_up)
Title.pack()
login.pack(padx=30,pady=10)
login_user.pack(padx=30,pady=10)
sign_up.pack(padx=30,pady=10)
mainloop()
```

Reference

1. https://wdgcorp.com/request-more-information/?gclid=EAIaIQobChMI7q_xi5DM-gIVEr3ICh0r2gvBEAAYBCAAEgI8OfD_BwE
2. <https://www.netsuite.com/portal/resource/articles/erp/warehouse-management.shtml>
3. <https://www.jrcded.com/advantages-and-disadvantages-of-warehousing/>

GitHub Resources¹

<https://github.com/The-Big-Boys-Inc/Warehouse-Management-System.git>

5.ER Diagram

https://lucid.app/lucidchart/0cd5589d-adbc-4fc7-8cde-2a64a2257a23/edit?viewport_loc=45%2C179%2C1897%2C1057%2C0_0&invitationId=inv_89f1f654-3612-447c-b252-7a72f98b60e0#

¹<https://www.quora.com/What-are-the-pros-and-cons-of-Apple>

