

Indian Institute of Technology Gandhinagar



CS 432: Databases

Designing the DBMS

Under the guidance of
Prof. Mayank Singh

20th March, 2023

Submitted By: CDSLite

Following Clauses have been used in the mentioned functions:

1. INSERT

- a. To add a product to Products.

The screenshot shows a web-based application for managing products. The URL in the browser is 127.0.0.1:5000/product. The interface is titled 'CDSLITE'. The main area contains a form for adding a new product. The fields include:

- ProductName:** Dell Laptop
- Description:** Dell G3
- is Merchandise:** No
- Base Price:** 85000
- is Barter:** Yes (radio button selected)
- Category:** Computers
- Sub Category:** A dropdown menu showing options like Video Game Consoles, Video Games, Gaming Accessories (selected), Televisions, and Home Theater Systems.

At the bottom of the form is a green 'Submit' button. Below the form, a red banner displays the message 'Product added successfully'.

Hoodie Rs. 750 Edit Delete	Shoes Rs. 1999 Edit Delete	Puma collar less tshirt Rs. 400 Edit Delete	Dell Laptop Rs. 85000 Edit Delete
--	--	---	---

```
q = f"INSERT INTO Products VALUES ('{product_id}', '{seller_id}')"
```

- b. To add a user in User.

We take Bank details from the user after registration for the first time and then populate the details in the seller table:

```
mysql> SELECT * FROM Seller;
+-----+-----+-----+-----+-----+
| SellerID | UserID | SoldProducts | BankName | AccountNo | IFSC_Code |
+-----+-----+-----+-----+-----+
| SE1188343744 | 8722531813 | 0 | Bank of Baroda | 6543126541 | IFSC 456 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
query = f"INSERT INTO User VALUES ({user_id},'{first_name}', '{last_name}', '{email}', {mob_number}, '{password}', '{dob}', '{gender}', '{bank}', {acc}, '{ifsc}' )"
```

- c. To add a user in the Seller.

```
q = f"INSERT INTO Seller VALUES ('{seller_id}', '{user_id}', 0 ,'{bank}', {acc}, '{ifsc}')"
```

- d. To add the products in VP_Products.

```
q2 = f"INSERT INTO VP_Products VALUES ('{product_id}', '{pdt_name}', '{desc}', 'Yes', {base_price}, '{is_barter}', '{creation_date}' )"
```

- e. To add the products in FP_Products.

```
q2 = f"INSERT INTO FP_Products VALUES ('{product_id}', '{pdt_name}', '{desc}', 'Yes', {mrp}, {quantity}, '{creation_date}', '{creation_date}' )"
```

- f. To add the products in FPhasSubCat.

```
q3 = f"INSERT INTO FPhasSubCat VALUES ('{product_id}', '{i}')"
```

- g. To add the products in VPhasSubCat.

```
q3 = f"INSERT INTO VPhasSubCat VALUES ('{product_id}', '{i}')"
```

- h. To add products in the ShoppingCart.

```
query = f"INSERT INTO ShoppingCart VALUES ('{user_id}', '{product_id}', '{quantity}', DEFAULT)"
```

- n. To add products in Unlisted_Products.

```
q = f"INSERT INTO Unlisted_Products VALUES ('{product_id}', '{prod_name}', '{desc}', '{creation_date}')"
```

- p. To add a bid on the Product in BidTable.

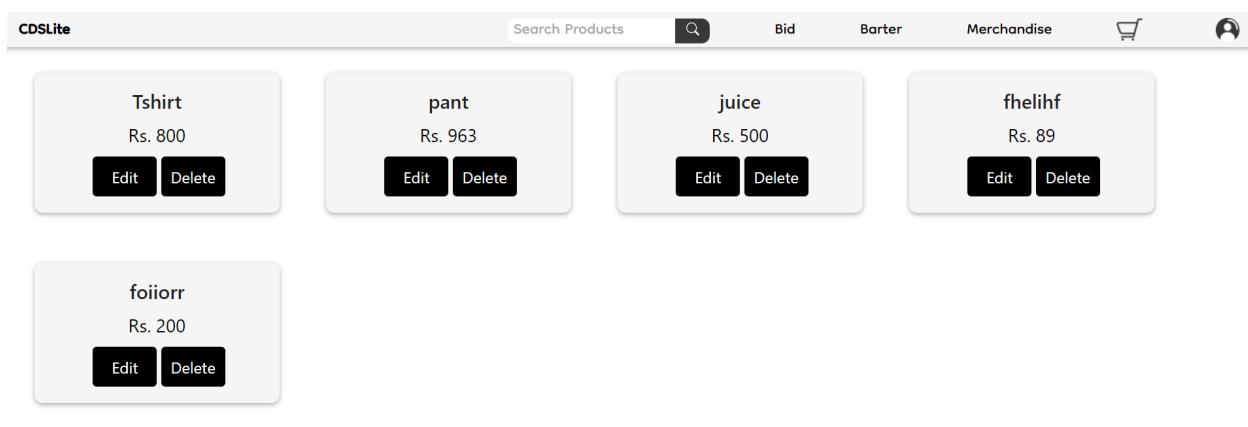
```
q= f"INSERT INTO BidTable VALUES ('{bid_id}', '{id_}', '{user_id}', DEFAULT, {bid_price}, 'Pending')"
```

- r. To add a barter on the Product in Barter.

```
q4 = f"INSERT INTO Barter (BarterID, P1ID, P2ID) VALUES ('{Barter_id}', '{id}', '{product_id}')";
```

2. UPDATE

- a. To edit FP Product details in FP_Products. (Not added search-box as while editing product, the user cannot search for any other products and leave the page.)



Img: My Products Page

Editing product “flhelihf” to “cake”. We can edit all details of the product. The default values remain the inserted values.

CDSLITE

ProductName
Cake

Description
if

MRP
89

Quantity
897

Category
Clothing and Accessories

Sub Category
Tops
Bottoms
Dresses and Jumpsuits
Activewear
Underwear

Submit

CDSLITE

		Search Products		Bid	Barter	Merchandise	Cart	User							
Tshirt	Rs. 800	Edit	Delete	pant	Rs. 963	Edit	Delete	juice	Rs. 500	Edit	Delete	Cake	Rs. 89	Edit	Delete
foiiorr	Rs. 200	Edit	Delete												

b.

```
q2 = f"UPDATE fp_products SET ProductName = '{pdt_name}', Description_ = '{desc}', MRP = {MRP}, Quantity = {Quantity}, CategoryID = '{category}'"
```

c. To edit VP Product details in VP_Products.

```
q2 = f"UPDATE vp_products SET ProductName = '{pdt_name}', Description_ = '{desc}', BasePrice = '{base_price}', CategoryID = '{category}'"
```

d. To alter the bid status of a product in BidTable.

```
q = f"UPDATE BidTable SET BidStatus = 'Confirmed' where BidID='{bid_id}'"
```

e. To alter barter status of a product

g. To edit account details of the User.

FIRST NAME:
Siddhesh

LAST NAME:
asdfasd

EMAIL:
siddhesh.kanawade@litgn.ac.in

MOBILE NUMBER:
7896541230

ADDRESS LINE:
K 420

CITY:
Gandhinagar

PINCODE:
321456

Update Profile


```
mysql> SELECT * FROM User;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| UserID | FirstName | LastName | Email_ID          | MobileNo | Password_ | DOB       | Gender | AddressLine | City      | PinCode |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1494644685 | Siddhesh | Kanawade | siddhesh.kanawade@litgn.ac.in | 7896541230 | asdf      | 2023-03-17 | F       | K 420        | Gandhinagar | 422065   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```


Your profile has been updated

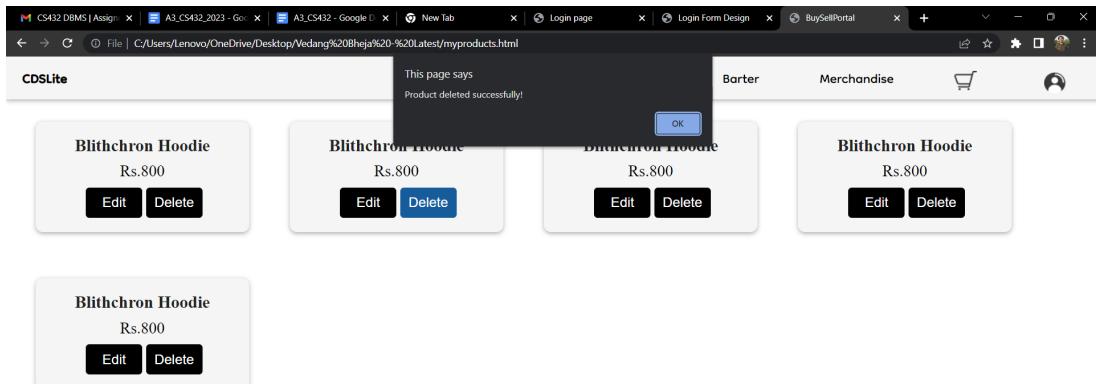
FIRST NAME:
Siddhesh


```
mysql> SELECT * FROM User;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| UserID | FirstName | LastName | Email_ID          | MobileNo | Password_ | DOB       | Gender | AddressLine | City      | PinCode |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1494644685 | Siddhesh | Kanawade | siddhesh.kanawade@litgn.ac.in | 7896541230 | asdf      | 2023-03-17 | F       | K 420        | Gandhinagar | 422065   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

query = f"UPDATE User SET FirstName='{first_name}', LastName='{last_name}', Email_ID='{email}', MobileNo='{mob_number}', AddressLine='{address}', City='{city}', PinCode={pincode}"

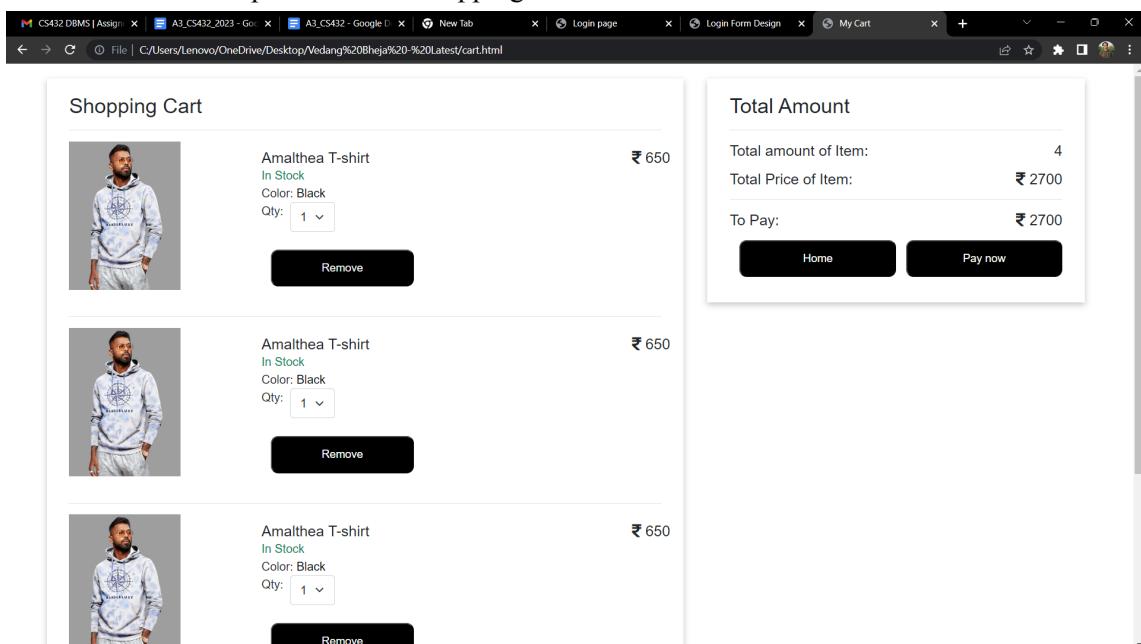
3. DELETE

a. To delete a product from Products.



```
q2 = f"Delete from Products where ProductID = '{param1}'"
```

b. To remove a product from Shopping cart.



```
query = f"DELETE FROM ShoppingCart WHERE UserID = '{user_id}' and ProductID = '{product_id}'"
```

- c. To delete a user account from User.

The screenshot shows a browser window with multiple tabs open. The active tab is titled 'Buy Sell Portal Login' and contains a form with 'Email Address' and 'Password' fields, both filled with placeholder text. Below the form is a blue 'Login' button. At the bottom of the page, there is a red text message: 'Your account has been deleted'.

```
query = f"DELETE FROM User WHERE UserID={user_id}"
```

- d. To delete a product from VPhasSubCat.

```
q2 = f"DELETE FROM vphassubcat WHERE ProductID = '{id}'"
```

4. RENAME

	barter
▶	barterhistory
	bidtable
	category
	fp_products
	fphassubcat
	ismerchant
	myimages
	order_
	payments
	products
	seller
	shoppingcart
	subcategory
	unlisted_products
	user
	variant

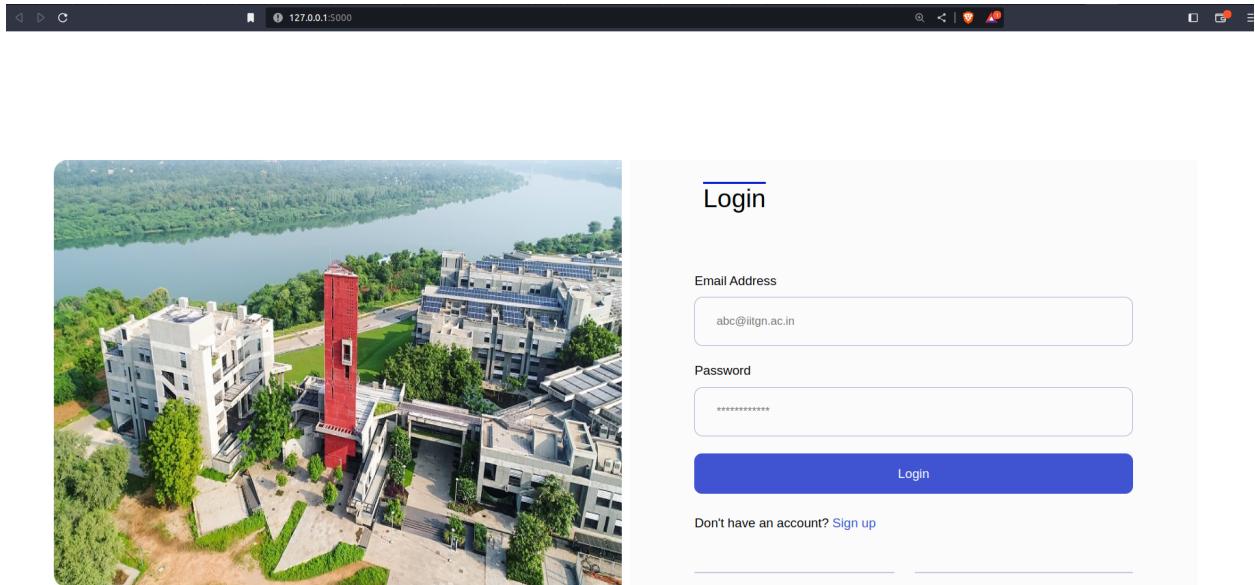
```
RENAME TABLE BarterHistory TO Accepted_Barter;
```

▶	accepted_barter
	barter
	bidtable
	category
	fp_products
	fphassubcat
	ismerchant
	myimages
	order_
	payments
	products
	seller
	shoppingcart
	subcategory
	unlisted_products
	user
	variant

We did not find any relevant USER related function or feature where we could use RENAME and display it on the frontend because showing TABLE names to users does not make sense. Though we have implemented RENAME on backend to rename ‘BarterHistory’ to ‘Accepted_Barter’.

5. WHERE

- a. Logging into the User account.



b.

```
query = f"select * from vp_products where isBarter='No'"
```

Selects all Variable Price Products which are not available for Barter (isBarter = 'No')

c.

```
query = f"SELECT * from User WHERE User.UserID='{user_id}'"
```

Selects all Users whose UserID is equal to the given user_id

d.

```
query = f"SELECT * from User WHERE User.Email_ID='{email_id}' and User.Password_='{password}'"
```

Selects all users where User's email and password match for a specific set of email and password.
(Query for login)

And many of the following instances of the where clause were used in the SQL queries in our backend and reflected in frontend. These are stated below:

f.

```
e. query      =      f"UPDATE      User      SET      FirstName='{first_name}',  
LastName='{last_name}',Email_ID='{email}',MobileNo='{mob_number}',Address  
Line='{addressline}',City='{city}',PinCode='{pincode}'      WHERE      UserID      =  
'{user_id}'"
```

```

g. query = f"DELETE FROM User WHERE UserID={user_id}"
h. query = f"SELECT * from Seller WHERE Seller.SellerID='{gen_id}'"
i. query = f"SELECT SellerID from Seller WHERE Seller.UserID='{user_id}'"
j. query = f"SELECT * from Products WHERE Products.ProductID='{product_id}'"
k. q1 = f"SELECT * from VP_Products WHERE VP_Products.isBarter='Yes'"
l. query = f"select * from vp_products where isBarter='Yes'"
m. query = f"SELECT * from Barter WHERE Barter.BarterID='{Barter_id}'"
n. q2 = f"select * from(select * from Products natural join FP_Products
   where Products.ProductID = FP_Products.ProductID) as P where P.sellerid =
   '{seller_id}'"
o.
p. q2 = f"Delete from Products where ProductID = '{param1}'"
q. q2 = f"select * from VP_Products where Productid = '{id}'"
r. q2 = f"DELETE FROM vphassubcat WHERE ProductID = '{id}'"
s. q2 = f"select * from FP_Products where Productid = '{id}'"
t. q2 = f"UPDATE fp_products SET ProductName = '{pdt_name}', Description_ =
   '{desc}', MRP = {MRP}, Quantity = {Quantity}, CategoryID =
   '{category_id}', UpdationDate = '{updation_date}' WHERE ProductID =
   '{id}'"
u.
v. query = f"SELECT * from BidTable WHERE BidTable.BidID='{bid_id}'"
w. q = f"UPDATE BidTable SET BidStatus = 'Confirmed' where BidID='{bid_id}'"
x. query = f"Select * from (Select * from BidTable natural join vp_products
   where VP_products.ProductID=BidTable.ProductID) as P where
   P.ProductID='{id_}'"
y.

```

```

z.      q2 = f"Select ProductID from BidTable where BidID='{bid_id}'"
aa.     q3 = f"UPDATE BidTable SET BidStatus = 'Declined' where ProductID = '{p_id}' and BidStatus != 'Confirmed'"
bb.     query = f"SELECT * FROM ShoppingCart WHERE UserID = '{user_id}' and ProductID = '{product_id}'"
cc.     query = f"SELECT * FROM ShoppingCart LEFT JOIN ( SELECT ProductID,
    ProductName, Price, Description_, CreationDate, UpdationDate, CategoryID
    FROM (SELECT BidTable.ProductID, ProductName, BidPrice AS Price,
    Description_, CreationDate, UpdationDate, CategoryID FROM BidTable LEFT
    JOIN VP_Products ON BidTable.productID = VP_Products.productID WHERE
    BidStatus = 'Confirmed') as temp UNION SELECT ProductID, ProductName, MRP
    AS Price, Description_, CreationDate, UpdationDate, CategoryID FROM
    FP_Products ) AS temp2 ON ShoppingCart.ProductID = temp2.ProductID WHERE
    (UserID = {user_id});"

```

We are creating a single user account to demonstrate the CRUD operation on User. CRUD operations include Adding user details, Reading user details, Updating user details and Deleting the User. We have used Cascading in deleting. Following shows an example schema with Cascading implementation.

```

create table Seller(
SellerID varchar(15) PRIMARY KEY,
UserID varchar(15),
SoldProducts int DEFAULT 0,
BankName varchar(25),
AccountNo char(18),
IFSC_Code char(11),
Foreign Key(UserID) References User(UserID) ON DELETE CASCADE
);

```

Logging out of the website:

1. Selecting Log OUT:

Edit Profile

Add Product

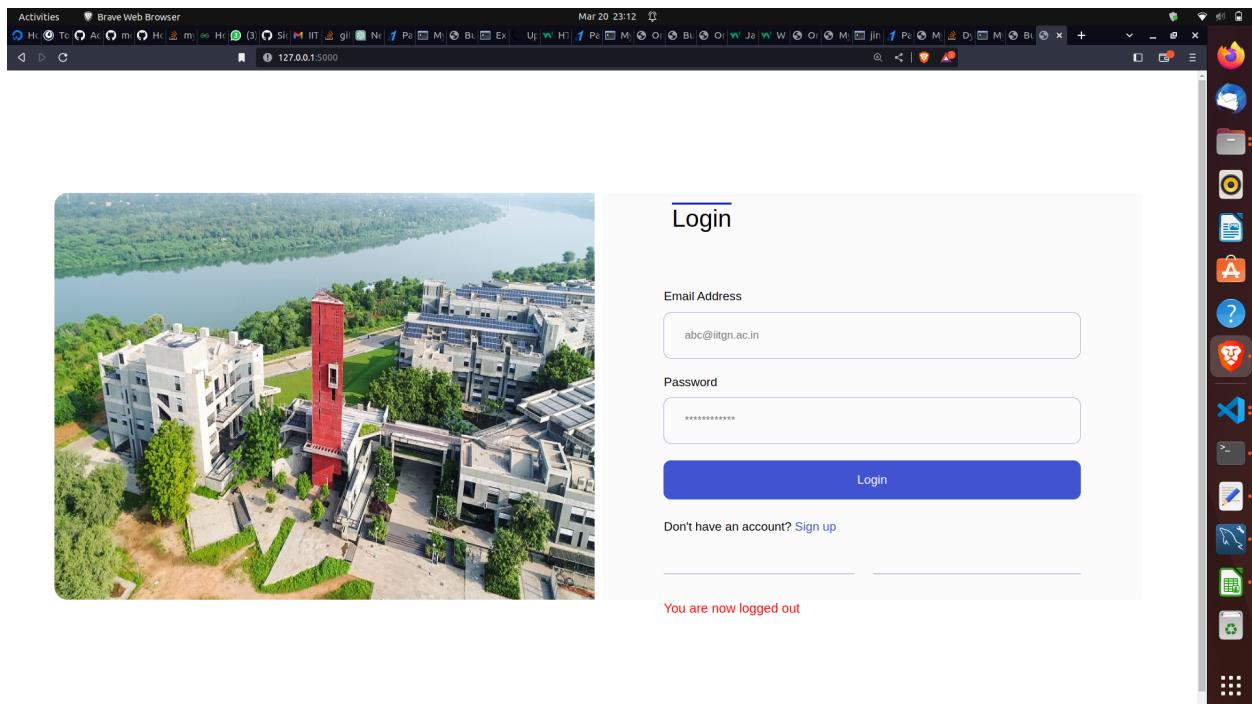
My Product

My order

log Out

Delete Account

2. Logged out successfully.



Deleting an account:

1. Selecting Delete Account

Edit Profile

Add Product

My Product

My order

log Out

Delete Account

2. Account Deleted Successfully.



Login

Email Address

abc@iitgn.ac.in

Password

Login

Don't have an account? [Sign up](#)

Your account has been deleted

```
mysql> SELECT * FROM User;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| UserID | FirstName | LastName | Email_ID           | MobileNo | Password_ | DOB      | Gender | AddressLine | City      | PinCode |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1494644685 | Siddhesh | Kanawade | stddhesh.kanawade@lttgn.ac.in | 7896541230 | asdf     | 2023-03-17 | F       | K 420      | Gandhinagar | 422065   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM User;
Empty set (0.00 sec)
```

Empty set shows that there is no user.

My Products Page:

Tshirt Rs. 800 Edit Delete	pant Rs. 963 Edit Delete	juice Rs. 500 Edit Delete	fhelihf Rs. 89 Edit Delete
foiiorr Rs. 200 Edit Delete			

Bid Page:

Tshirt Rs. 800 Bid	Tshirt Rs. 963 Bid
--	--

Placing a Bid:

1. Entering the amount of the Bid.

PUMA COLLAR LESS TSHIRT

Grey color t-shirt

Base Price: Rs. 400

PLACE BID: Rs.

Submit

2. Confirmation to place a Bid.

127.0.0.1:5000 says
Are you sure?

Cancel **OK**

PUMA COLLAR LESS TSHIRT

Grey color t-shirt

Base Price: Rs. 400

PLACE BID: Rs.

Submit

3. Bid Placed Successfully.

Selecting a Bid:

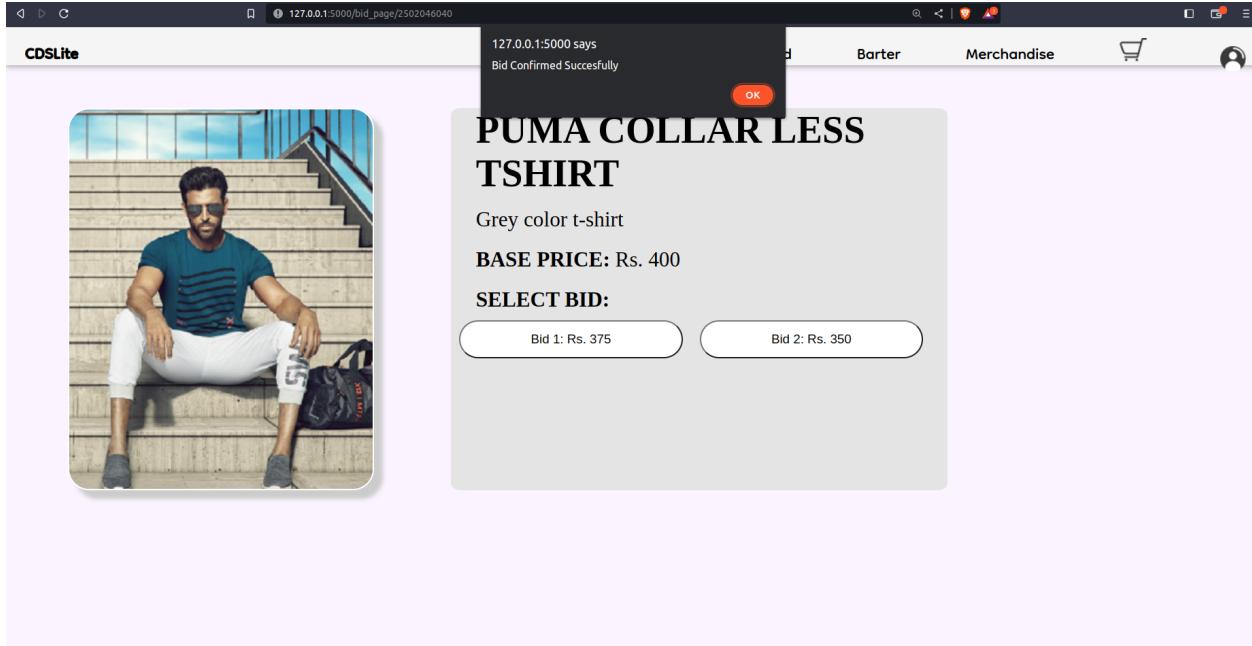
1. Choose a Bid

The screenshot shows a web browser window for 'CDSLite' at the URL 127.0.0.1:5000/bid_page/2502046040. The page displays a product listing for a 'PUMA COLLAR LESS TSHIRT'. On the left, there is a large image of a man sitting on wooden steps, wearing a teal t-shirt with horizontal stripes and white pants. The main content area contains the product title 'PUMA COLLAR LESS TSHIRT', a description 'Grey color t-shirt', and a 'BASE PRICE: Rs. 400'. Below this, a section titled 'SELECT BID:' offers two options: 'Bid 1: Rs. 375' and 'Bid 2: Rs. 350'. The top navigation bar includes links for 'Search Products', 'Bid', 'Barter', 'Merchandise', and a user profile icon.

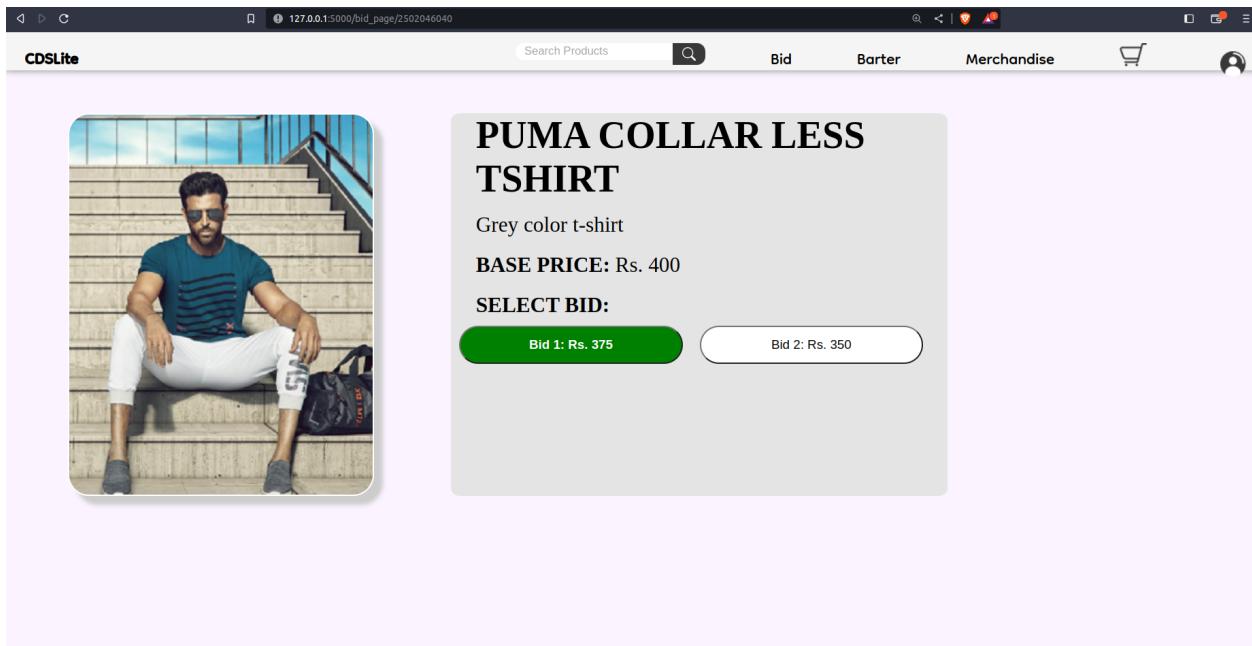
2. Confirmation of the Bid

The screenshot shows a web browser window for 'CDSLite' at the URL 127.0.0.1:5000/bid_page/2502046040. A confirmation dialog box is displayed in the center, asking 'Are you Sure to confirm the Bid?' with 'Cancel' and 'OK' buttons. The background page shows the same product listing for the 'PUMA COLLAR LESS TSHIRT' as in the previous screenshot. The product details, including the image of the man on the steps, the title, description, base price, and bid selection options, are visible.

3. Bid Confirmation



4. Bid Selected





Now the product is populated in the shopping cart and the flow will be demonstrated below.

Ordering a Product:

1. Adding Product to Shopping Cart:

A screenshot of a web browser window. The address bar shows "127.0.0.1:5000/merchandise/". The page title is "CDSLte". There is a search bar with "Search Products" and a magnifying glass icon. Below the search bar are two product cards. The first card for a "T shirt" is priced at "400" with an "Add to Cart" button. The second card for a "Hoodie" is priced at "700" with an "Add to Cart" button. Both cards have a light gray background with rounded corners.

2. Shopping Cart:

A screenshot of a web browser window. The address bar shows "127.0.0.1:5000/index/shopping_cart". The page title is "CDSLte". The main content area is titled "Shopping Cart" and displays a message "Your Product has been added to Cart". It shows a thumbnail of a person wearing a hoodie, the product name "Hoodie", "In Stock", "Qty: 1", and the price "₹ 700". Below the product are "Remove" and "Buy Now" buttons. To the right, there is a "Total Amount" section with a table:

Total amount of Item:	1
Total Price of Item:	₹ 700
To Pay:	₹ 700

Below the table are "Home" and "Pay now" buttons.

Note: We can also delete products from the shopping cart.

Shopping Cart

Product deleted from cart

ORDER SUMMARY

Name	Product Name	Quantity	Total Price
Siddhesh	Hoodie	1	700
		Total:	700

K 420
Gandhinagar
456321

Pay Now

Cancel Order

Payment

CDSLite
A Wild Sheep Chase

Amount payable is
₹2,300

Select a payment method

Credit/Debit Card Visa, Mastercard, Maestro, Rupay	Pay with Credit/Debit Card
Netbanking Pay with Internet Banking Account	Card Number <input type="text"/>
Wallet Pay using a Wallet	Expiry Date <input type="text"/> MM / YY
UPI Pay using BHIM, GPay and other UPI apps	CVV <input type="text"/> ***
PayLater Pay using PayLater	Cardholder's Name <input type="text"/>

PAY NOW

Powered by 

CDSLite
A Wild Sheep Chase

Amount payable is
₹2,300

Select a payment method

Credit/Debit Card Visa, Mastercard, Maestro, Rupay	Pay using Netbanking
Netbanking Pay with Internet Banking Account	Choose a bank
Wallet Pay using a Wallet	 ICICI Bank
UPI Pay using BHIM, GPay and other UPI apps	 Axis Bank
PayLater Pay using PayLater	 Kotak Mahindra Bank

PAY NOW



Barter

1. Products available for barter

The screenshot shows a product listing on the CDSLITE platform. At the top, there is a navigation bar with links for "CDSLITE", "Search Products" (with a magnifying glass icon), "Bid", "Barter" (which is highlighted in blue), "Merchandise", a shopping cart icon, and a user profile icon.

The main content area displays a product listing for "BOAT ROCKERZ 103 PRO EARPHONES". On the left, there is an image of the earphones, which are black with a braided cable and a remote control. To the right of the image, the product name is displayed in large, bold, black capital letters. Below the name, the price "Rs.500" is shown in a smaller font. A detailed description follows: "Wireless Earphones with 10 mm Drivers, Single Press Voice Assistant, Used for 5 months, working and in good condition". Below the description, there is a small link labeled "Barter".

2. INFO OF PRODUCT, Barter against the selected product 'BOAT ROCKERZ 103 PRO EARPHONES'

Product Name
SHIRT

Description
T-SHIRT OF SIZE EXCEL

Upload Product Images
Choose files hoodie.jpg

hoodie.jpg

Upload

Submit

3. Product with previous info added successfully against barter

Your Barter Added Successfully

4. Profile -> myproducts

Nike Shoes Rs. 750	Edit Delete
SG Cricket Bat Rs. 1500	Edit Delete

5. Profile -> myproducts -> SG Cricket Bat

CDSLite

Search Products

Bid Barter Merchandise



SG CRICKET BAT
SELECT BARTER:

[Yonex GR 303 Badminton Racquet](#)

6. For current product

CDSLite

Search Products

Bid Barter Merchandise

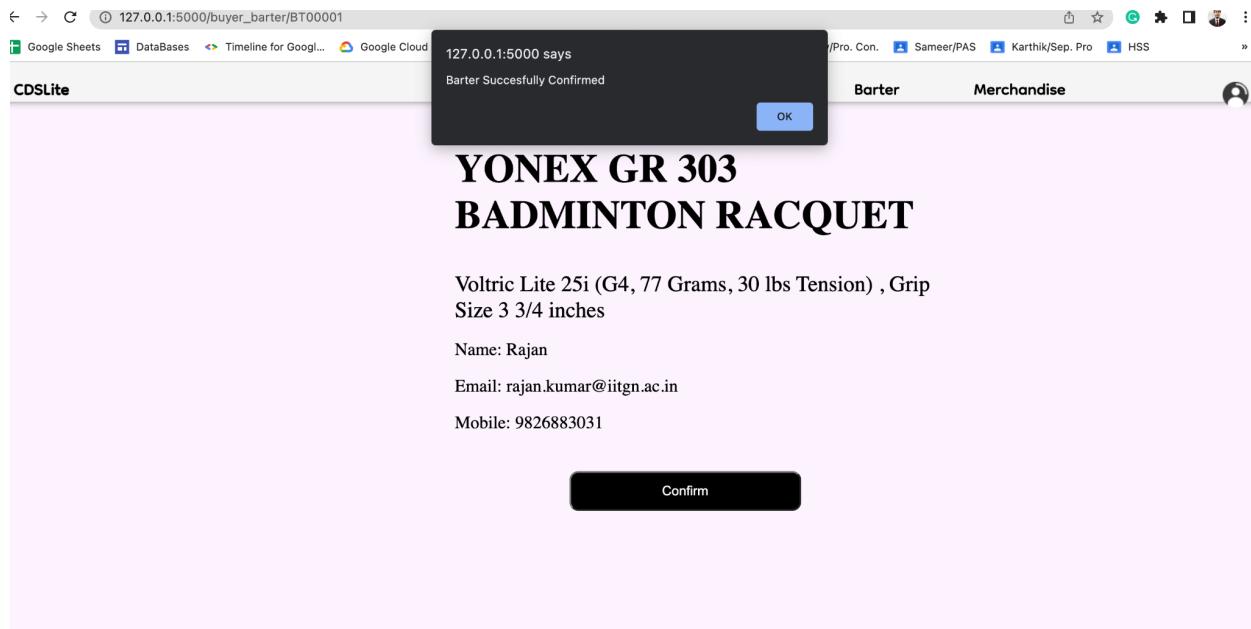
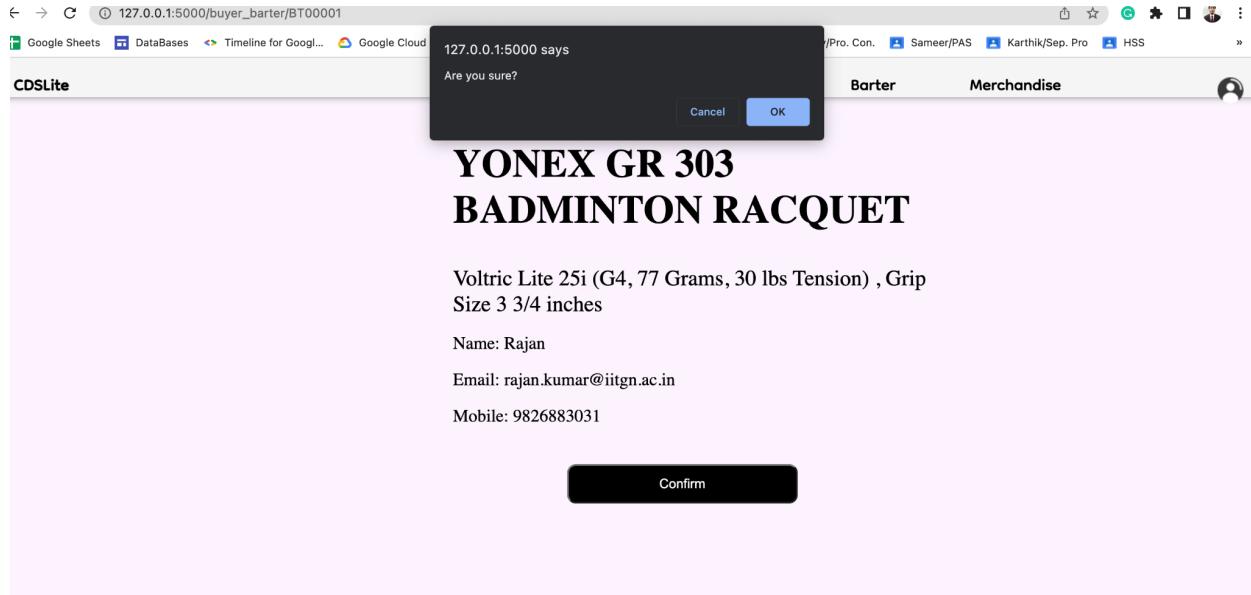
**YONEX GR 303
BADMINTON RACQUET**

Voltric Lite 25i (G4, 77 Grams, 30 lbs Tension) , Grip Size 3 3/4 inches

Name: Rajan
Email: rajan.kumar@iitgn.ac.in
Mobile: 9826883031

[Confirm](#)

7. Confirm barter



Work Distribution

Member	Roll Number	Group	Contribution
Vedang Chavan	20110222	G1	<ul style="list-style-type: none"> Created HTML, CSS files for Barter and all product details page in Bid, Barter, My

			<p>Products page.</p> <ul style="list-style-type: none"> ● Integrated frontend with backend ● Created JavaScript functions for dynamically adding products in all the websites ● Added screenshots and answers in PDF documentation
Ayush Gupta	20110031	G1	<ul style="list-style-type: none"> ● Created HTML, CSS files for Merchandise, Add Products and Edit products pages. ● Added README.md ● Helped in PDF documentation in adding screenshots and adding answers.
Ayush Chaudhari	20110042	G1	<ul style="list-style-type: none"> ● Helped Create HTML file of product page.
Ekansh Bansal	20110243	G1	<ul style="list-style-type: none"> ● Created HTML, CSS files for bid, profile, shopping cart and login pages.
Kushagra Jain	20110099	G1	<ul style="list-style-type: none"> ● Helped create HTML file of Barter page.
Nesar Joshi	20110120	G1	<ul style="list-style-type: none"> ● Created CSS files and dynamically added categories and subcategories.
Sakshi Jain	20110181	G1	<ul style="list-style-type: none"> ● Created HTML, CSS files for register, bid and order and some product description pages. ● Wrote Javascript functions for product description pages
Aditya Bhujbal	20110040	G2	<ul style="list-style-type: none"> ● Added barter back-end files ● Wrote SQL queries and schema cascading ● Documentation and added screenshots ● Helped in debugging the error handling
Mann Jain	20110108	G2	<ul style="list-style-type: none"> ● Added backend in flask for products, login, register, edit, delete, bid and barter. ● Integrated frontend with backend. ● Wrote corresponding sql queries. ● Debugging the errors and testing ● Documentation and added screenshots
Piyush Dhirwani	20110138	G2	<ul style="list-style-type: none"> ● Added barter back-end files and SQL queries ● Handled git repository and managed conflicts ● Documentation and added screenshots
Ronak Kalra	20110171	G2	<ul style="list-style-type: none"> ● Debugging HTML file routing ● Wrote corresponding sql queries. ● Worked on debugging the errors and testing ● Added screenshots and corresponding texts to report

Siddhesh Kanawade	20110199	G2	<ul style="list-style-type: none">• Debugging HTML file routing• Build backend for shopping cart• Integrated Login, Register and Edit Profile• Integrated payment gateway (Razorpay)
-------------------	----------	----	---