IITG BUY & SELL

Database Management System: MA518
Course Project Work

by

Chandresh Joshi : 232123007
Sayed Parvej Rahaman : 232123126

Under the guidance of

Prof. Ashok Singh Sairam

DEPARTMENT of MATHEMATICS

INDIAN INSTITUTE of TECHNOLOGY GUWAHATI

CONTENTS

1 Introduction:

The IITG Buy-and-Sell platform is a purpose-built online marketplace for the IITG campus community, encompassing students, faculty, and staff. This platform makes it easy to exchange second-hand items securely and conveniently within the campus. Developed using HTML and CSS for the frontend, PHP for backend operations, and MySQL for data management, the platform enhances the user experience with a streamlined interface, efficient product listings, advanced search functions, and secure user authentication. It aims to simplify and enrich the process of buying and selling pre-owned goods in a user-friendly environment.

Key Features:

Admin Functionality: Admin is responsible for maintaining platform integrity through user and product management

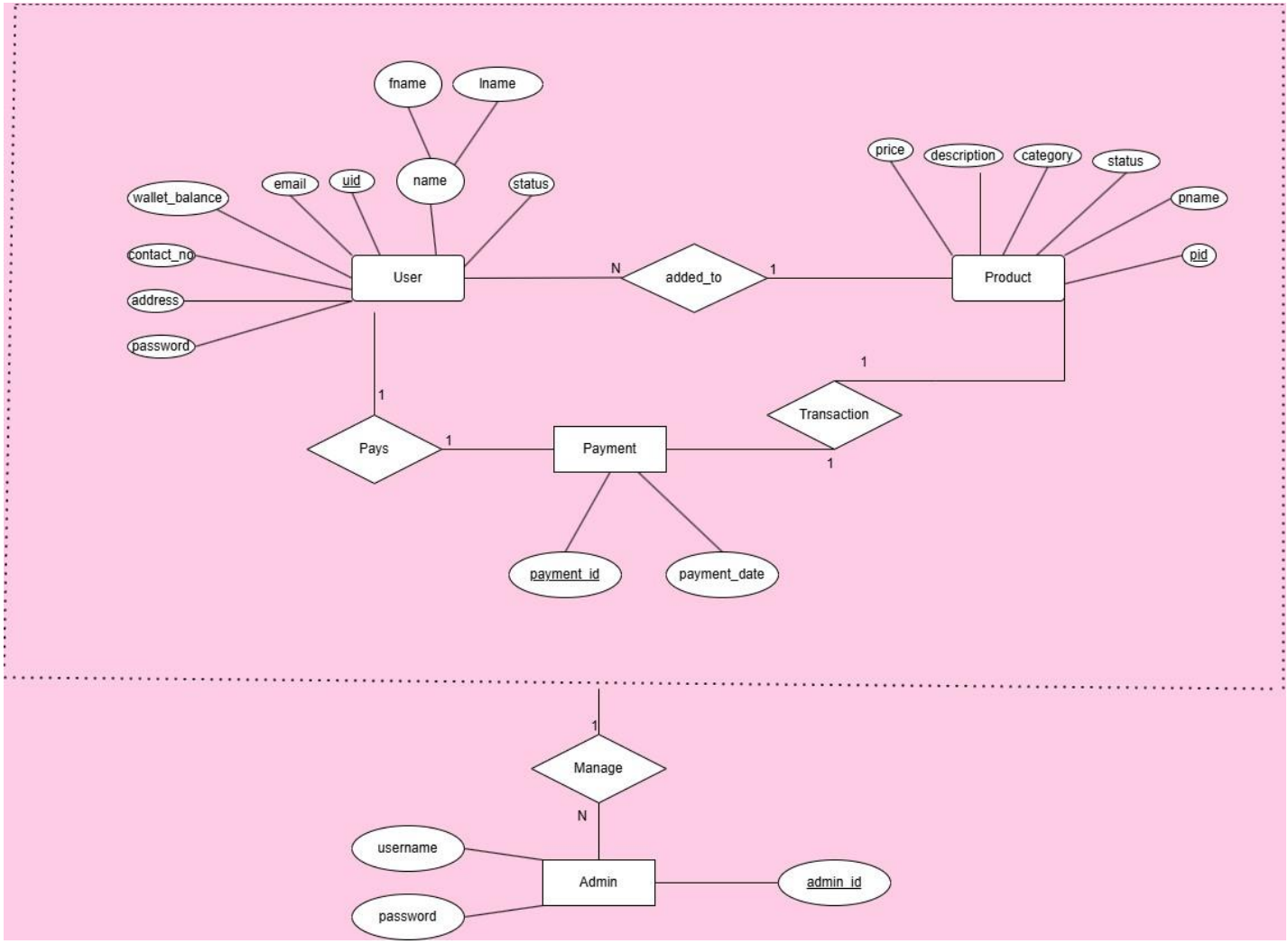
- **User Monitoring and Blocking:** If any suspicious activity or discrepancy is detected, Admin can block the user to prevent further activity.
- **Product Removal:** Admin has the authority to delete any product associated with the detected discrepancies.
- **User Appeal and Unblocking:** If a blocked user contacts Admin with a valid explanation, Admin can review the appeal and unblock the user if deemed appropriate.

User Functionality: A user can log in as either a buyer or a seller.

- **Seller Workflow**: Upon logging in, the seller completes a form to upload product details, including an image. After submission, they are redirected to my_items.php, where a table displays all their listed products (both sold and unsold). The table includes a "status" column and other essential details for efficient product management.

- **Buyer:** Upon logging in, the buyer is redirected to a page showing all available products for purchase, excluding any products the buyer (**since here a buyer can be itself a seller also, so whatever products buyer itself listed for sell will not be visible to him**) has listed and also excluding the sold products. If interested, the buyer can proceed with the purchasing process for selected products

2. ENTITY-RELATIONSHIP MODEL and DESCRIPTION



Entities and Attributes

1. Admin Entity:

- **Attributes:** admin_id, name, password, email
- **Description:** Represents the system administrators who manage users and products.

```
MariaDB [buyandsell]> describe admin;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| admin_ID | int(11)      | NO   | PRI | NULL    | auto_increment |
| username | varchar(50)  | NO   | UNI | NULL    |                |
| Password | varchar(255) | NO   |     | NULL    |                |
| name     | varchar(20)  | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
4 rows in set (0.016 sec)

MariaDB [buyandsell]> describe payment;
```

2. User Entity:

- **Attributes:** uid, fname, lname, email, wallet_balance, contact_no, address, password, status
- **Description:** Represents users who buy or sell products.

```
Database changed
MariaDB [buyandsell]> describe user;
+----------------+---------------+------+-----+---------+----------------+
| Field          | Type          | Null | Key | Default | Extra          |
+----------------+---------------+------+-----+---------+----------------+
| uid            | int(11)       | NO   | UNI | NULL    | auto_increment |
| fname          | varchar(50)   | NO   |     | NULL    |                |
| lname          | varchar(50)   | YES  |     | NULL    |                |
| email          | varchar(100)  | NO   | PRI | NULL    |                |
| contact_no     | varchar(15)   | YES  |     | NULL    |                |
| address        | text          | YES  |     | NULL    |                |
| password       | varchar(200)  | YES  |     | NULL    |                |
| wallet_balance | decimal(10,2) | YES  |     | 0.00    |                |
| status         | varchar(20)   | YES  |     | active  |                |
+----------------+---------------+------+-----+---------+----------------+
9 rows in set (0.017 sec)
```

3. Product Entity:

- **Attributes:** pid, pname, description, category, price, status
- **Description:** Represents products that are either available for sale or already sold.

```
MariaDB [buyandsell]> describe product;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| pid         | int(11)       | NO   | PRI | NULL    | auto_increment |
| pname       | varchar(100)  | NO   |     | NULL    |                |
| price       | decimal(10,2) | NO   |     | NULL    |                |
| description | text          | NO   |     | NULL    |                |
| category    | varchar(50)   | YES  |     | NULL    |                |
| seller_id   | int(11)       | YES  | MUL | NULL    |                |
| buyer_id    | int(11)       | YES  | MUL | NULL    |                |
| status      | varchar(10)   | NO   |     | unsold  |                |
| image_path  | varchar(255)  | YES  |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
9 rows in set (0.015 sec)
```

4. Payment Entity:

- **Attributes:** payment_id, payment_date, amount

- **Description:** Represents the payment transactions made for product purchases.

```
MariaDB [buyandsell]> describe payment;
+----------------+---------------+------+-----+---------------------+----------------+
| Field          | Type          | Null | Key | Default             | Extra          |
+----------------+---------------+------+-----+---------------------+----------------+
| payment_id     | int(11)       | NO   | PRI | NULL                | auto_increment |
| transaction_id | varchar(255)  | NO   |     | NULL                |                |
| pid            | int(11)       | NO   | MUL | NULL                |                |
| seller_id      | int(11)       | NO   | MUL | NULL                |                |
| buyer_id       | int(11)       | NO   | MUL | NULL                |                |
| amount         | decimal(10,2) | NO   |     | NULL                |                |
| payment_date   | timestamp     | NO   |     | current_timestamp() |                |
+----------------+---------------+------+-----+---------------------+----------------+
7 rows in set (0.008 sec)

MariaDB [buyandsell]>
```

Relationships and Cardinality:

1. added_to (User-Product):

- Type: N:1
- Description: Represents the relationship where a user can add multiple products, but each product is added by only one user.

2. Transaction (Product-Payment):

- Type: 1:1
- Description: Represents the relationship where each product is associated with a single payment, and each payment corresponds to a single product.

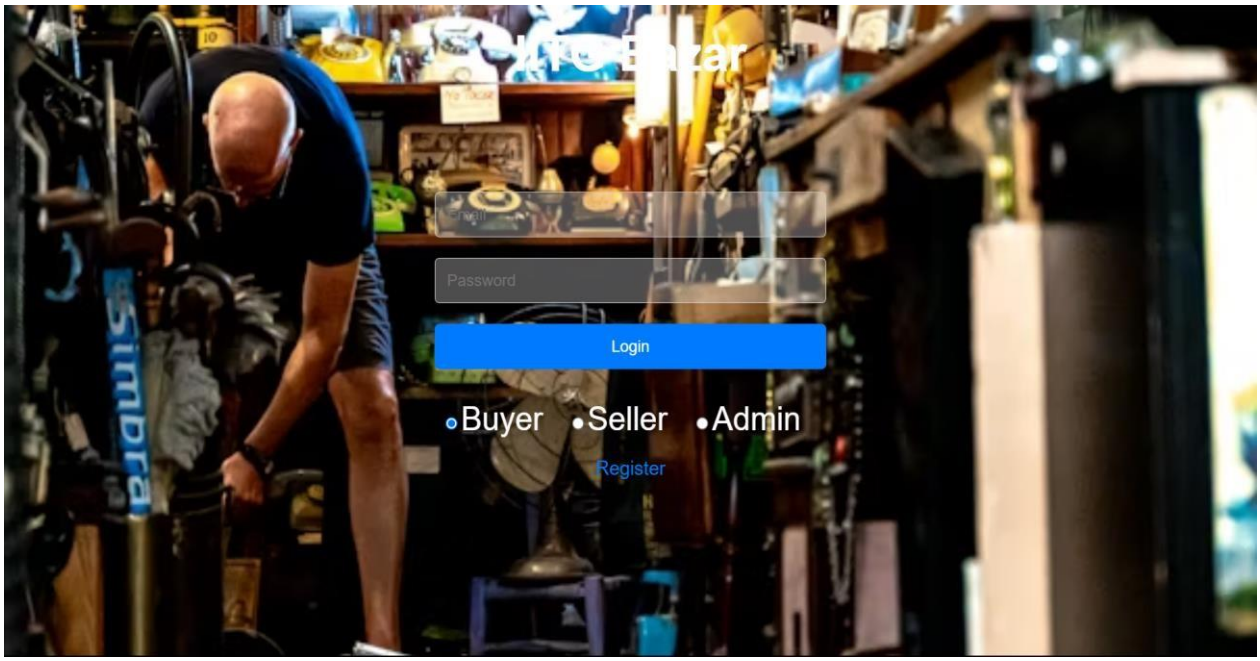
3. Pays (User-Payment):

- Type: 1:1
- Description: Represents the relationship where each user makes a single payment for a product, and each payment is associated with only one user.

4. Manage (Admin):

- Type: Aggregation with User, Product, Payment
- Description: Represents the relationship where an admin oversees the management of users, products, and payments in the system.

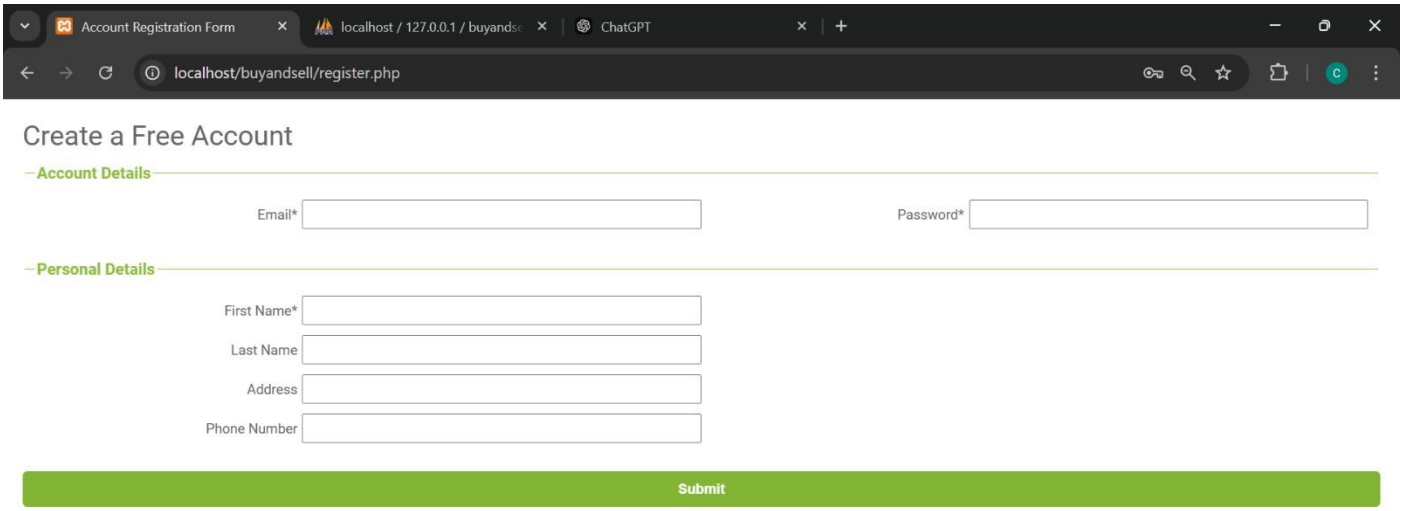
3.a.Home Page



Here a buyer, seller and admin can login and also a user is not registered then he can also click on register and will redirect to “register.php”.

- IMPORTANT POINT TO NOTICE HERE- A session is initiated to ensure that only logged-in users can access specific pages. If a user attempts to navigate directly to a restricted page without logging in, the session variables will be uninitialized. This triggers an automatic redirection to the home.html page, ensuring secure access control.

3.b. Registration Page



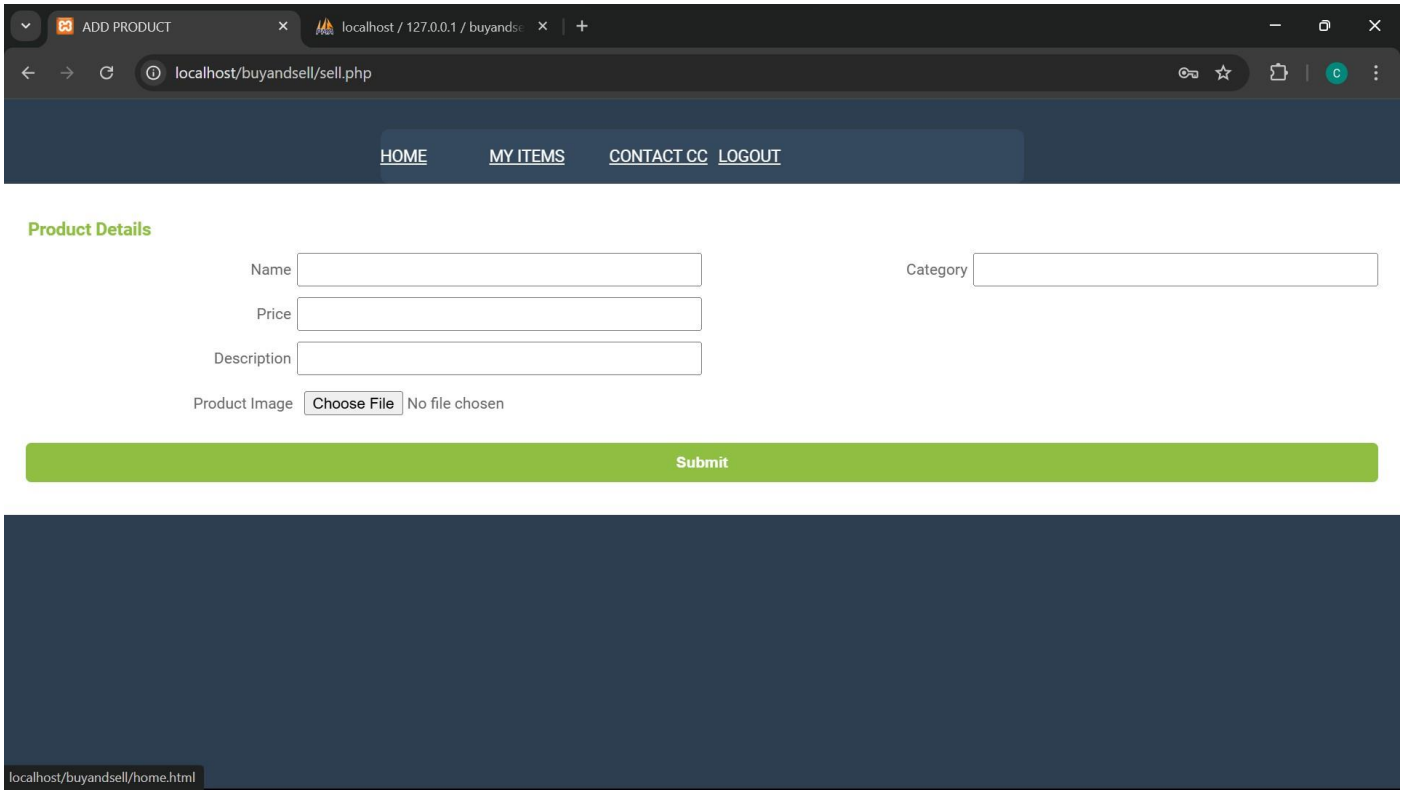
Users can register for an account on IITG Bazar and will be redirected to the home page upon successful registration. The form enforces the following validations:

1. **Mandatory Fields**: Email, password, and first name are required.
2. **Email Validation**: The email must be in the correct format (e.g., containing @).
3. **Password Strength**: Passwords must be at least 8 characters long.
4. **Phone Number Validation**: The phone number must contain exactly 10 digits.

If any of these conditions are violated, an appropriate warning message will be displayed to the user, prompting them to correct the inputs.

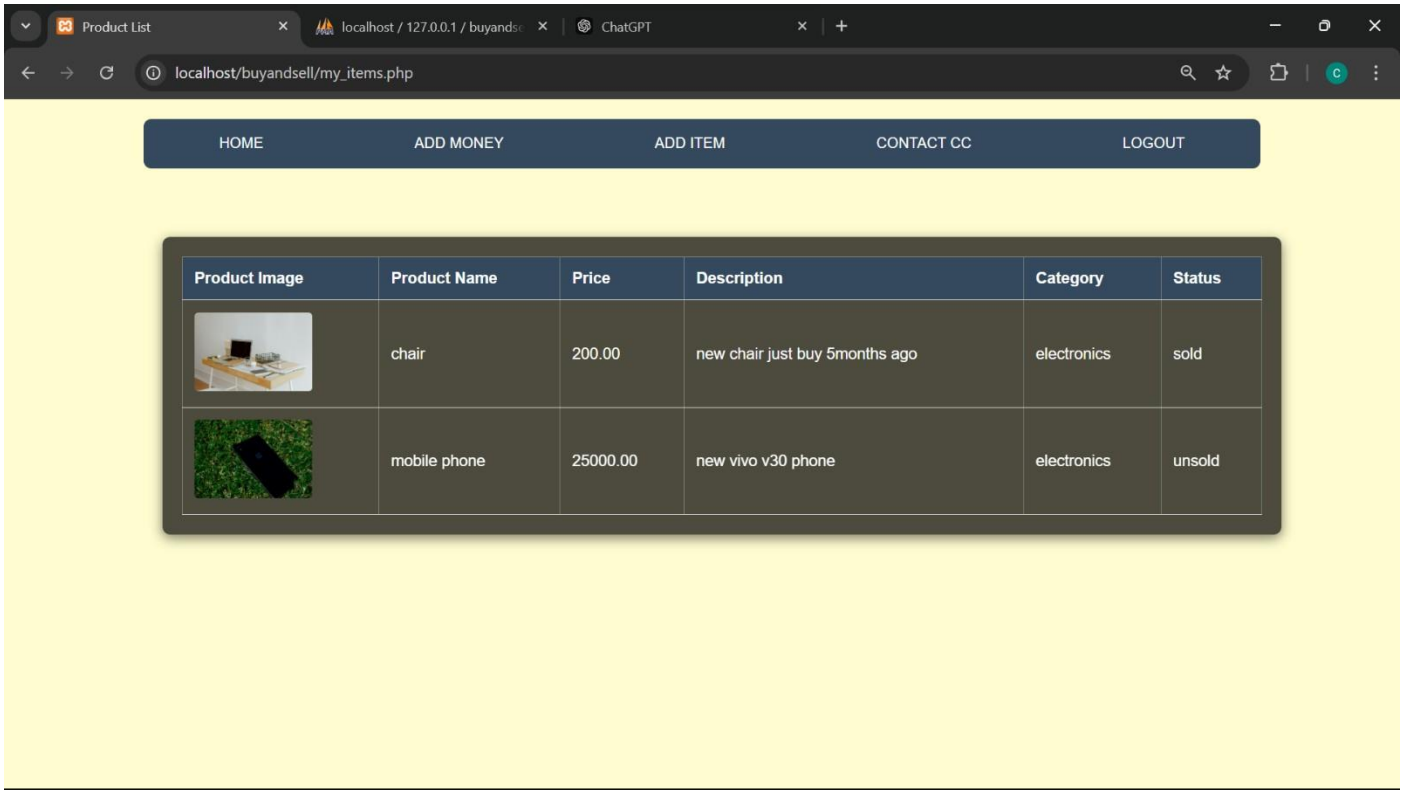
IMPORTANT POINT TO BE NOTICE- here password is sent to the database buyandsell in hashed form

3.c. seller page



After logging in, the seller is redirected to sell.php, where they can list products and upload photos (only .jpg or .jpeg format, max size 2MB). Upon submission, the seller is redirected to my_items.php, displaying all their listed items with details, including the product's status (sold/unsold).

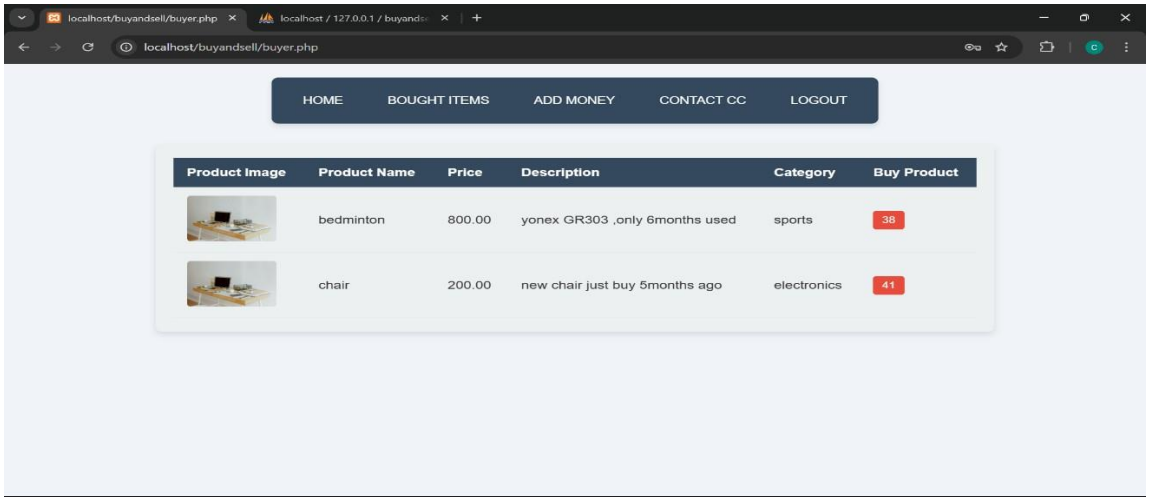
3.d.my_items



The seller can view all their listed items in a table, including the product status indicating whether items are sold or unsold.

Navigation links (HOME, ADD MONEY, ADD ITEMS, LOGOUT) are available for easy access to key functionalities.

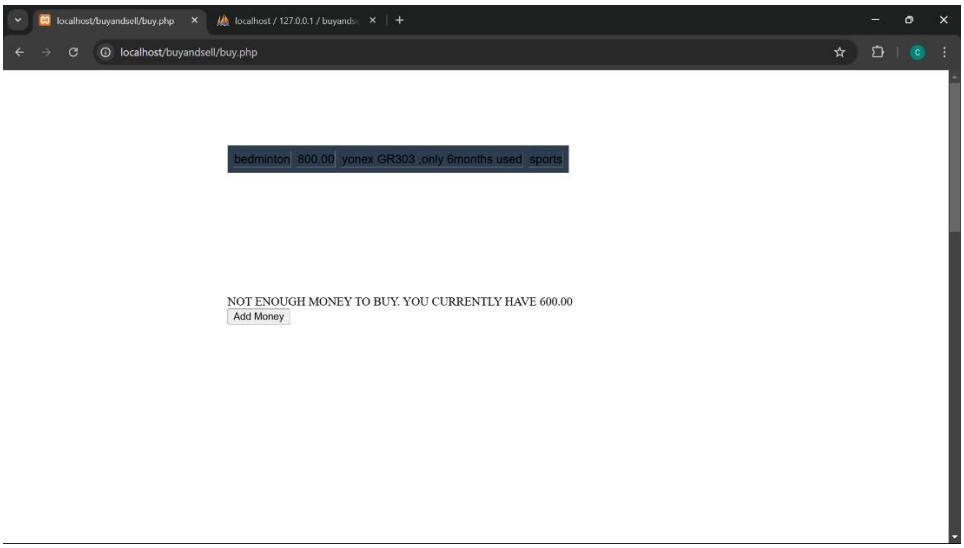
3.e. buyer



Upon logging in, the buyer is shown only those items that are unsold and not listed by themselves.

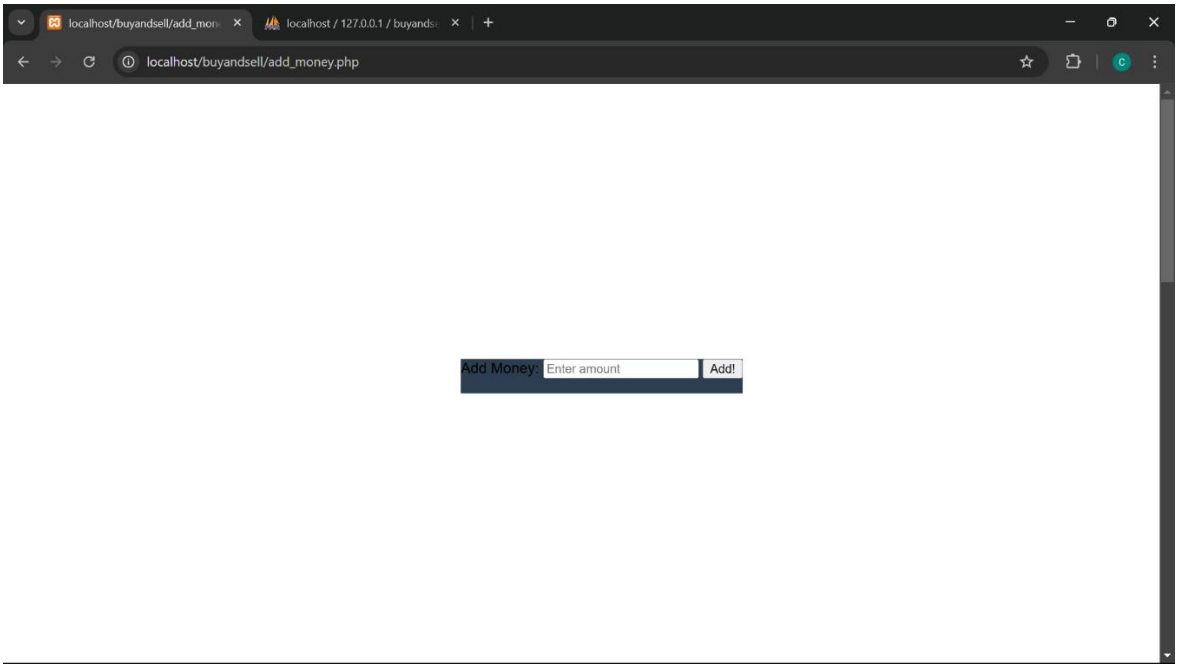
After selecting an item to purchase, the buyer is redirected to buy.php to proceed with the buying process.

3.f.buy.php

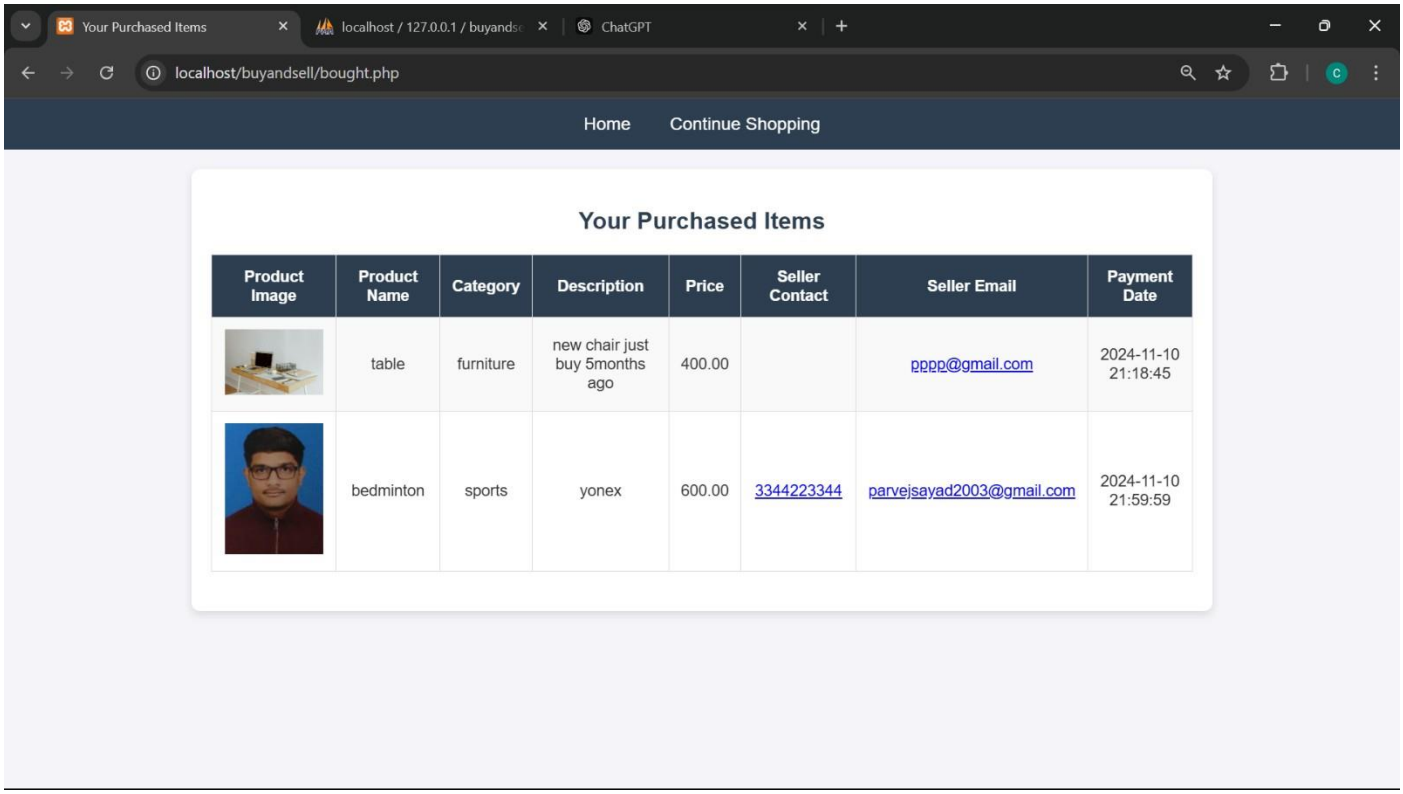


If the buyer has sufficient funds in their wallet, they can proceed to buy the product, and the payment will be transferred to the seller's wallet while updating the buyer's wallet balance.

If the buyer doesn't have enough funds, they can add money to their wallet by clicking on the "Add Money" link before completing the purchase.



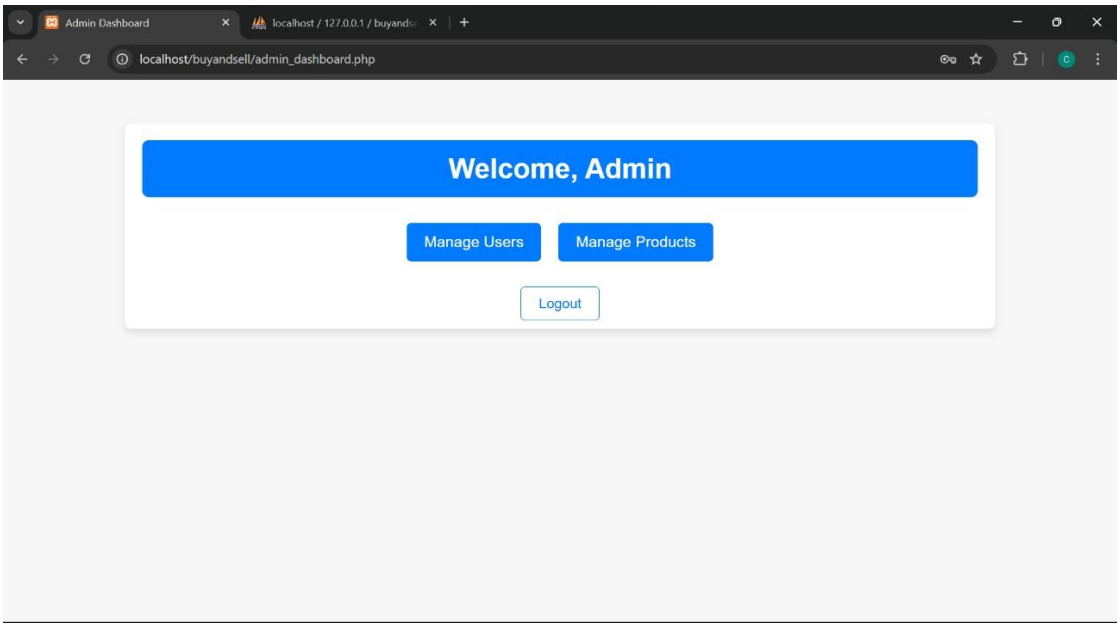
3.g. bought items.



The buyer can view their purchase history, including a list of all items purchased along with the payment date for each product.

If the buyer has any queries regarding a product, they can directly contact the seller for assistance.

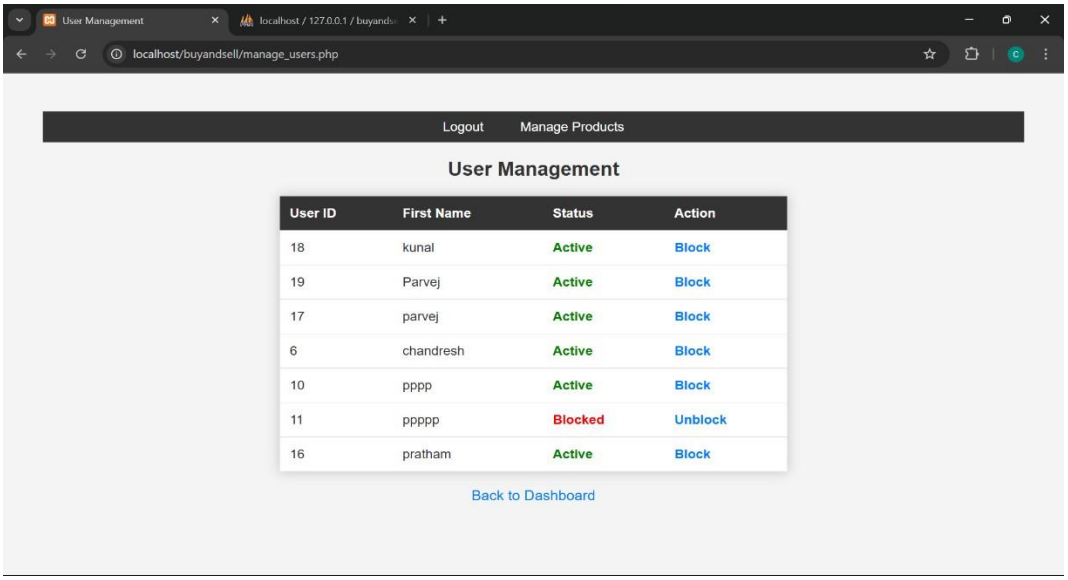
3.h. Admin dashboard



The admin can manage both users and products.

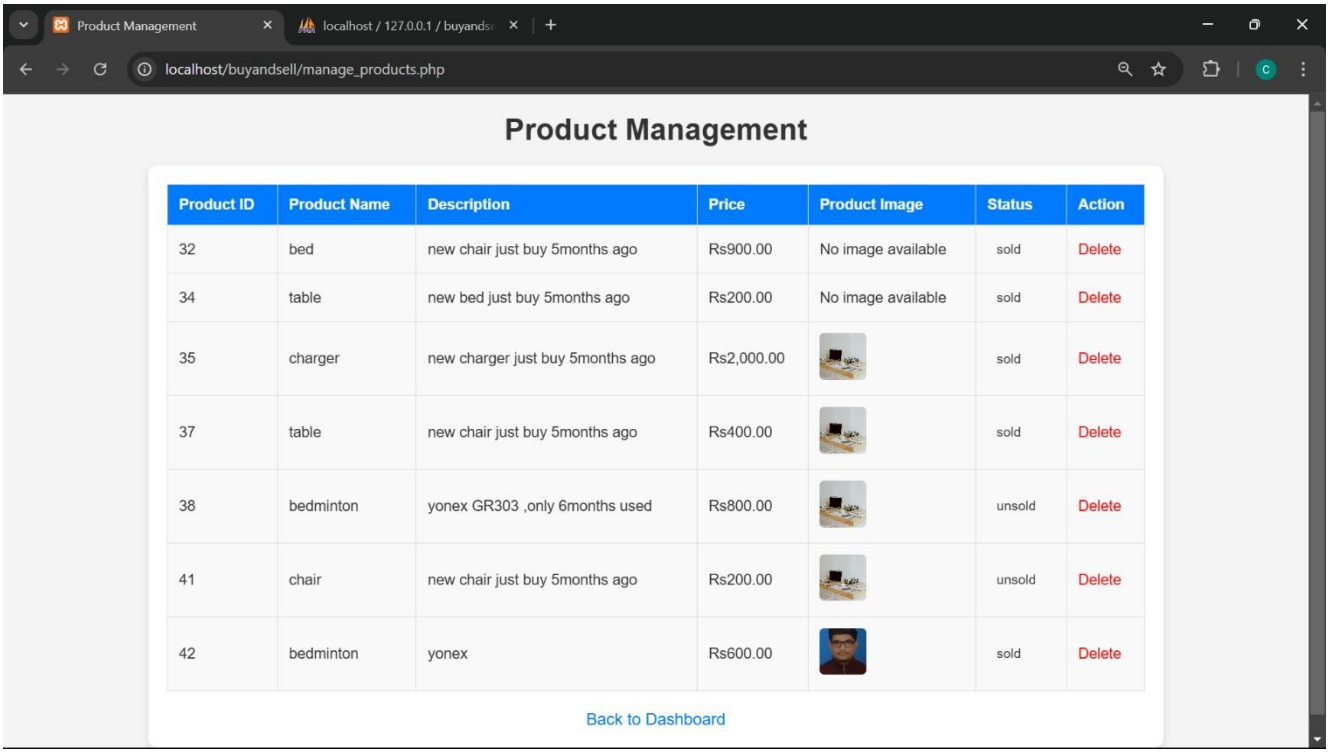
The admin has the ability to block or unblock users and delete products as needed.

3.i. manage users by admin

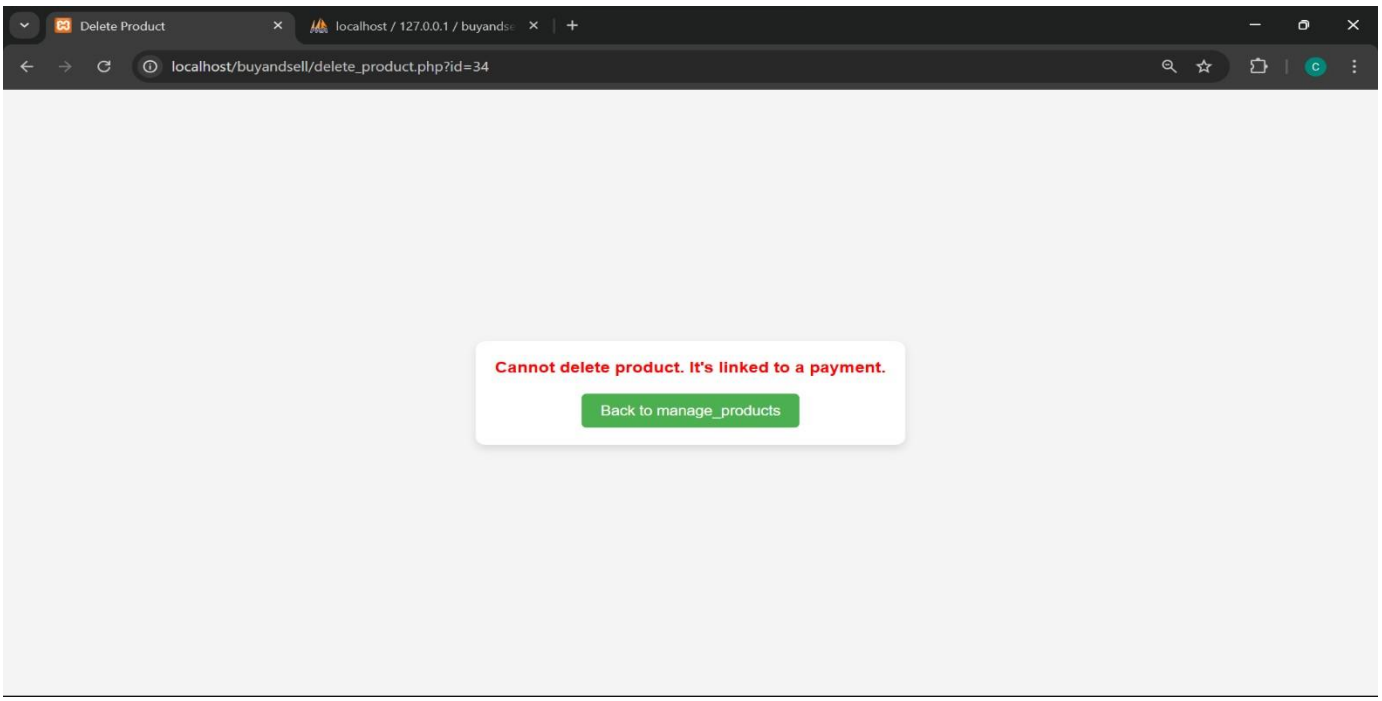


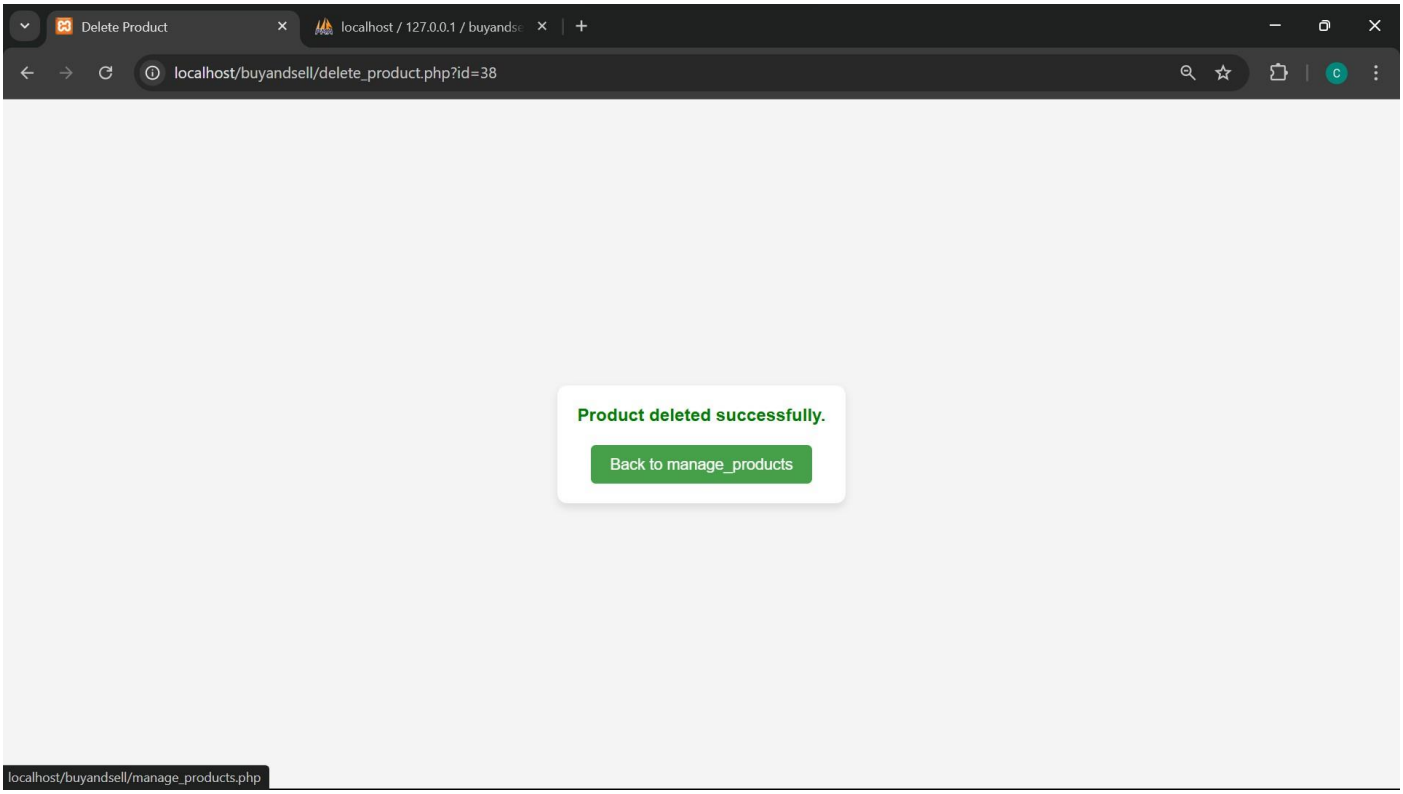
Admin can block and unblock a user by clicking on the block and unblock. there will be a massage “user blocked successfully’’ and “user unblocked successfully”.

3.j. manage products.



The admin can only delete unsold products. If a product has been sold, a payment date is generated for that product, and it is linked to the payment table via the product ID as a foreign key. Due to the **"ON DELETE RESTRICT"** constraint, the admin cannot delete products that have been sold.




Delete Product
localhost / 127.0.0.1 / buyands
localhost/buyandsell/delete_product.php?id=38
Product deleted successfully.
Back to manage_products
localhost/buyandsell/manage_products.php

4. Future Enhancements

While the current platform offers essential functionality for users to buy and sell goods, several improvements can be made to further enhance its usefulness:

1. **Product and Seller Ratings**:
    - A **product and seller rating system** could be added, where buyers can leave feedback on products and sellers, enhancing transparency and trust within the community.
2. **Payment Gateway Integration**:
    - Introducing a secure **QR code payment system** to allow buyers to scan a code for seamless transactions would simplify the buying process and ensure secure payments between buyers and sellers.

5. Conclusion

The IITG Buy-and-Sell platform provides a reliable and scalable solution for the IITG community to buy and sell products in a secure and easy-to-use environment. With its user-friendly interface, efficient backend, and structured database, the platform aims to create a vibrant marketplace for second-hand goods. As we continue to enhance its features and integrate advanced functionalities, this platform has the potential to become an integral part of the IITG campus ecosystem.

Thank You!