UofC Buy & Sell

CPSC 471 - Data Base Management Systems

Final Report for E-Commerce Website

L01-T01

Yossri Khalil - 30028691

Md Rashik Hassan - 30048022

Riyad Hossain - 30047717

Abstract

For this project, we have created an e-commerce platform for customers to buy and sell products similar to websites like Kijiji and Rentfaster, where a product can be purchased through a request from the potential buyer to the seller. The request created by the buyer is sent to the seller via an auto-generated email implemented using the *node-mailer* module in *ExpressJS* where the *transporter* variable activates the server to send an email to the seller with the buyers' contact information. If the user wanted to contact the seller before buying the product, they can also contact the seller using the Contact function in the website, which sends a custom mail with the user's information and message. Furthermore, there is also an admin who is able to manipulate products by approving and disapproving products after the user creates a post on the website to sell their product, this is an important functionality to remove potential scams and inappropriate products that may discourage users to use the product.

Introduction

Before websites were invented, buying and selling were all real life transactions between two or more individuals and over the years as the world progressed and technology developed, [2] World Wide Web (WWW) was announced in 1990 which was the first browser created by Tim Berners Lee. A few years later, Jeff Bezos invents Amazon, one of the first e-commerce sites that became successful and on the same year eBay was launched as it became one of the fastest growing e-commerce sites in the late 90s. The process of electronically buying or selling products online was a big hit and has developed rapidly over the past two decades as Amazons now have drones to deliver their packages for them.

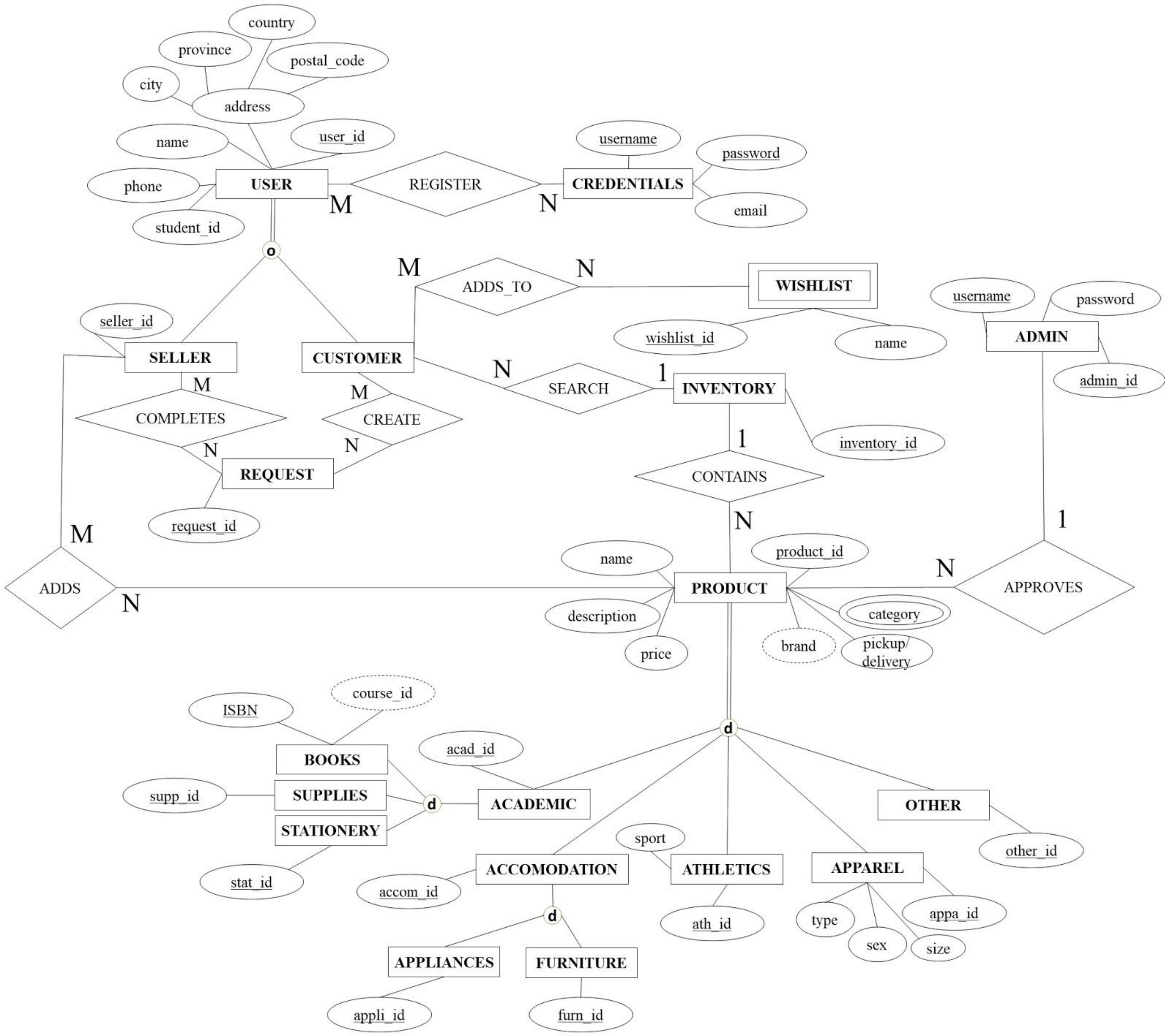
As for our e-commerce website, we wanted to understand the buyer's needs and at the same time help fellow sellers sell their products to those who maybe interested in it, which we have successfully implemented because according to our website, whenever the customer clicks "Buy", an email with their information is automatically sent to the user.

Design Analysis

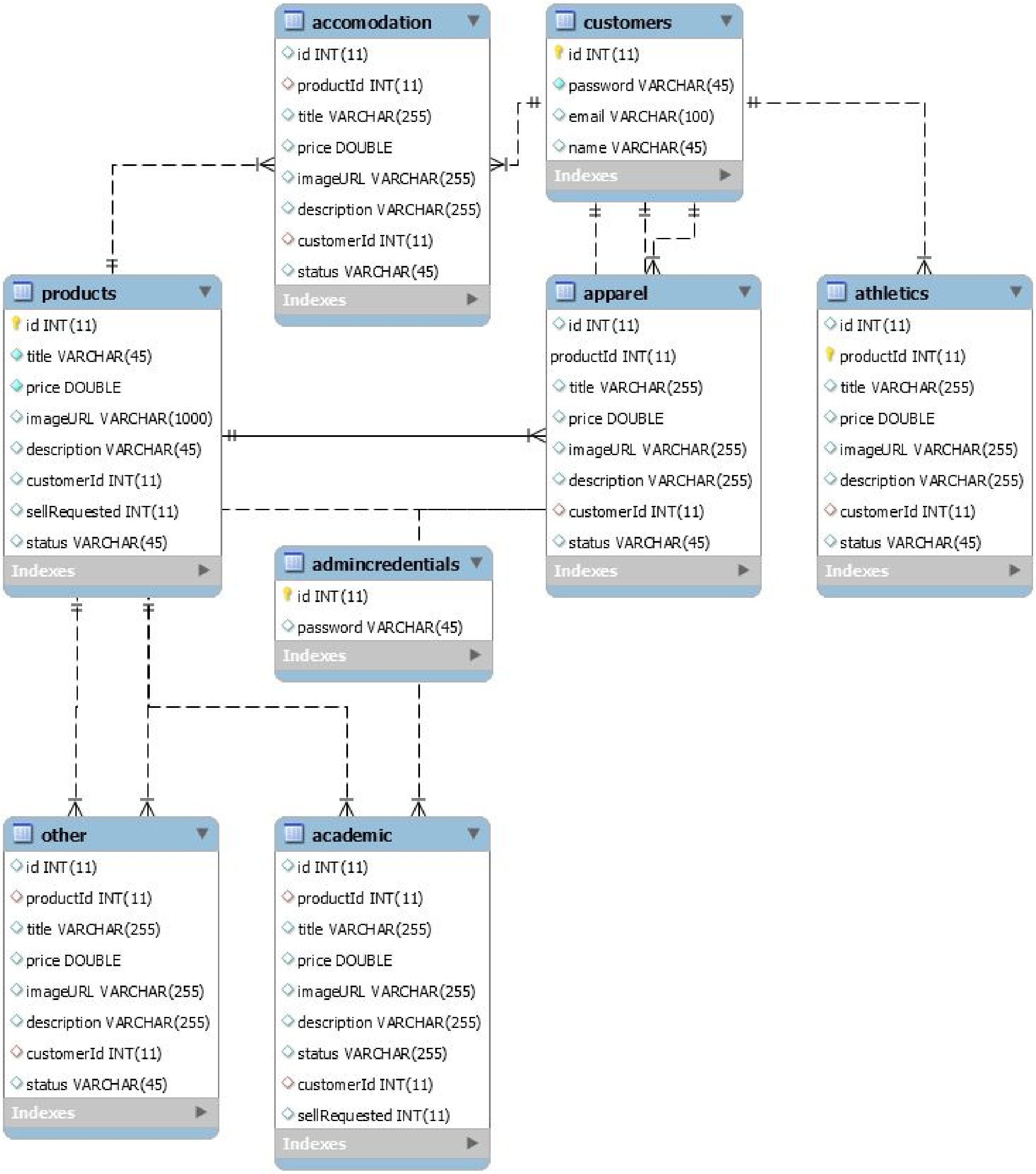
The diagram below is the EERD we created for the database of our program. There are two kinds of users in our system, **customer** and **admin**. According to the EERD, we decided to remove **user** and **seller**. Instead, we made the customer both the buyer and the seller and credentials are a part of the customer table. We implemented all five categories of product but not the subcategories of the categories.

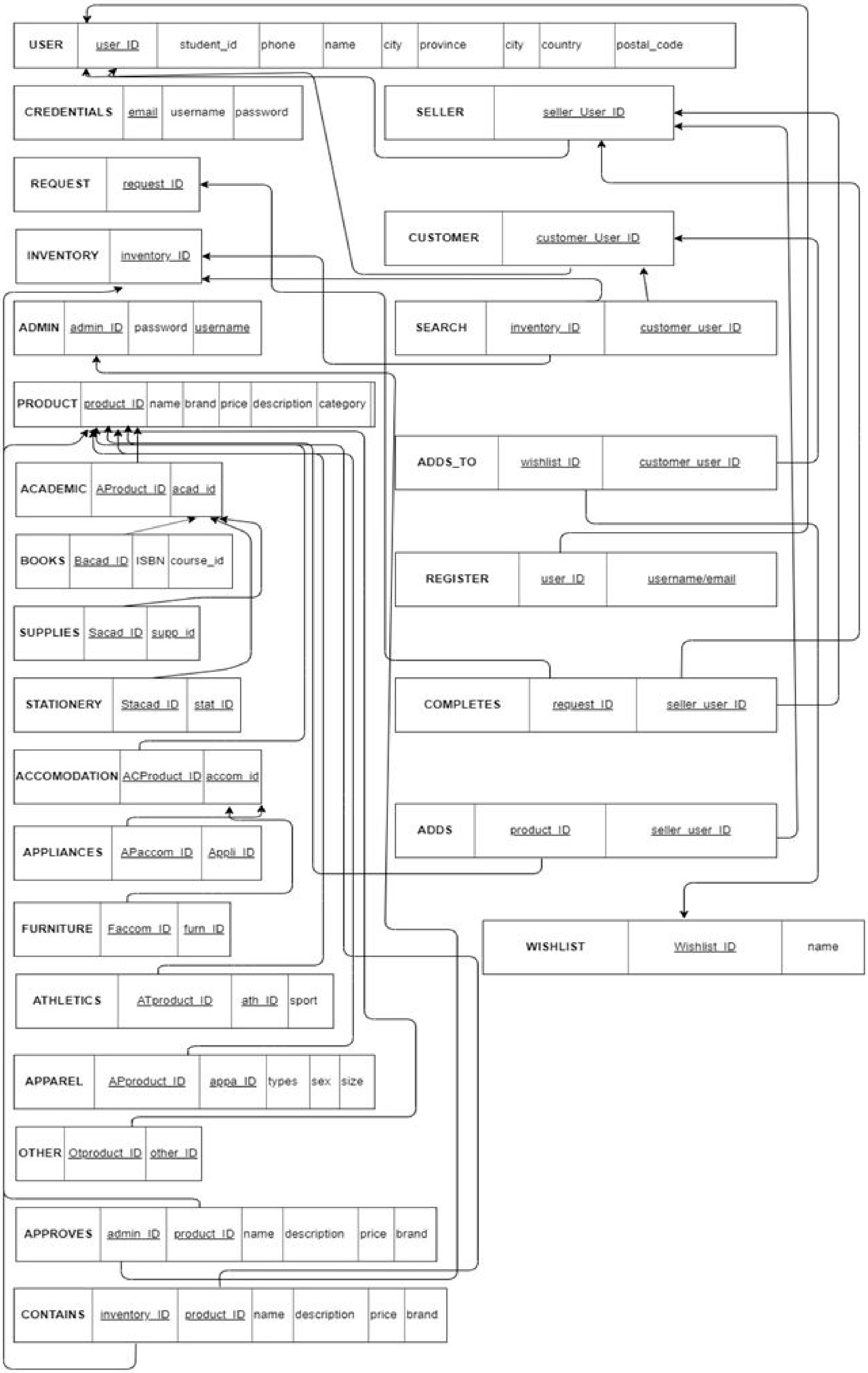
As for the relational model, we have removed a lot of the attributes and some tables. Wishlist was not implemented. Compared to the previous EERD, there aren't that many changes in the relational model since our ERD and tables are less complex.

Previous EERD



Database EERD (ecommerce.sql)





USER | user_ID | student_id | phone | name | city | province | city | country | postal_code

CREDENTIALS | email | username | password

REQUEST | request_ID

INVENTORY | inventory_ID

ADMIN | admin_ID | password | username

PRODUCT | product_ID | name | brand | price | description | category

ACADEMIC | AProduct_ID | acad_id

BOOKS | Bacad_ID | ISBN | course_id

SUPPLIES | Sacad_ID | supp_id

STATIONERY | Stacad_ID | stat_ID

ACCOMODATION | ACProduct_ID | accom_id

APPLIANCES | APaccom_ID | Appli_ID

FURNITURE | Faccom_ID | furn_ID

ATHLETICS | ATproduct_ID | ath_ID | sport

APPAREL | APproduct_ID | appa_ID | types | sex | size

OTHER | Otproduct_ID | other_ID

APPROVES | admin_ID | product_ID | name | description | price | brand

CONTAINS | inventory_ID | product_ID | name | description | price | brand

SELLER | seller_User_ID

CUSTOMER | customer_User_ID

SEARCH | inventory_ID | customer_user_ID

ADDS_TO | wishlist_ID | customer_user_ID

REGISTER | user_ID | username/email

COMPLETES | request_ID | seller_user_ID

ADDS | product_ID | seller_user_ID

WISHLIST | Wishlist_ID | name

Relational Model

Implementation

Here are the list of all the queries of all the functions used to communicate with the database. SQL Injection was implemented.

1. Add Product:
   'INSERT INTO products(title, price, imageURL, description, customerId) VALUES (?,?,?,?,?)'
   a. Add to Accommodation: 'SELECT * FROM accomodation WHERE status <> "pending"'
   b. Add to Apparel: 'SELECT * FROM apparel WHERE status <> "pending"'
   c. Add to Academic: 'SELECT * FROM academicWHERE status <> "pending"'
   d. Add to Athletics: 'SELECT * FROM athletics WHERE status <> "pending"'
   e. Add to Others: 'SELECT * FROM others WHERE status <> "pending"'

2. Display All Products In Homepage (not pending)
   SELECT * FROM products WHERE status <> "pending"

3. Customer Login
   SELECT * FROM customers WHERE id = ? AND password = ?

4. Register
   'INSERT INTO customers(id, password, email, name) VALUES (?,?,?,?)'

5. Display Customer Homepage Products
   SELECT * FROM products WHERE customerId <> ? AND status <> "pending"

6. Filter All Products
   SELECT * FROM products WHERE status <> "pending"

7. Admin Login

```
'SELECT * FROM admincredentials WHERE id = ? AND password = ?'
```

8. Show pending products

```
'SELECT * FROM products WHERE status="pending"'
```

9. Remove pending products

```
SELECT * FROM products WHERE status="pending"
```

10. Approve pending products

```
UPDATE products SET status="approved" WHERE id = ?
```

11. Search

```
'SELECT * FROM products where title LIKE "%?%"  AND status = "approved" '
```

12. Search For Admin

```
'SELECT * FROM products where title LIKE ?  AND status = "approved" '
```

13. Filter Search(Just for Apparel)

```
'SELECT * FROM apparel '
```

14. Sign Up for a customer

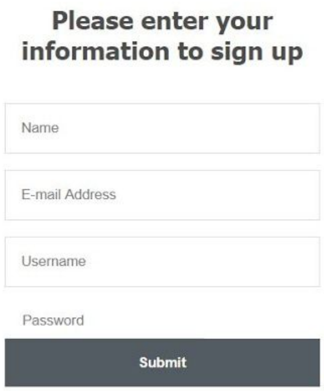
```
'INSERT INTO customers( id, password, email, name) VALUES (?,?,?,?)'
```

User Manual

1. User Controls

a) Register

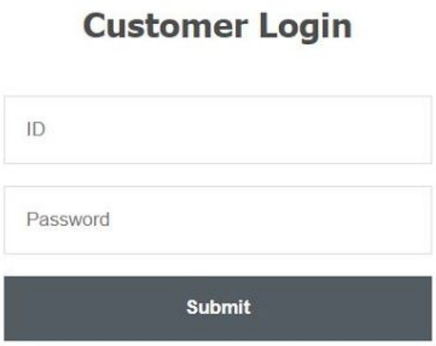
New users can click on the “Sign Up” button at the home screen of the website to open the Register form to create an account.



Please enter your information to sign up

b) Customer Login

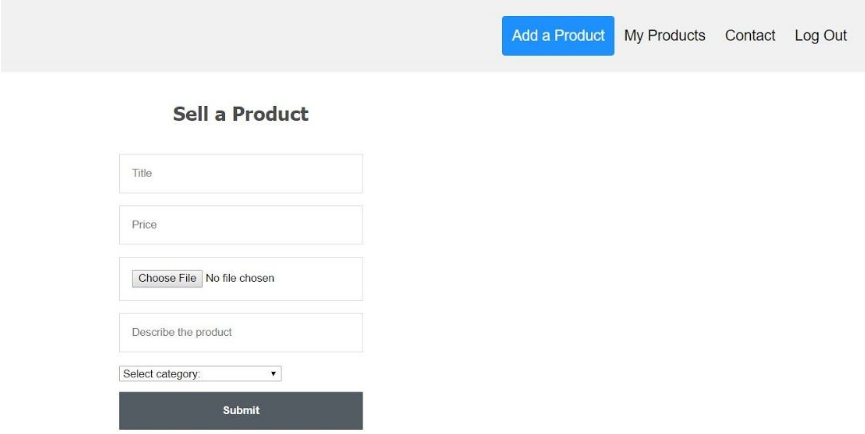
New users are redirected to the login page right after creating a new account. They can enter their credentials correctly to sign in right after. Existing users can click on the “Customer Login” page to login to their account.



Customer Login

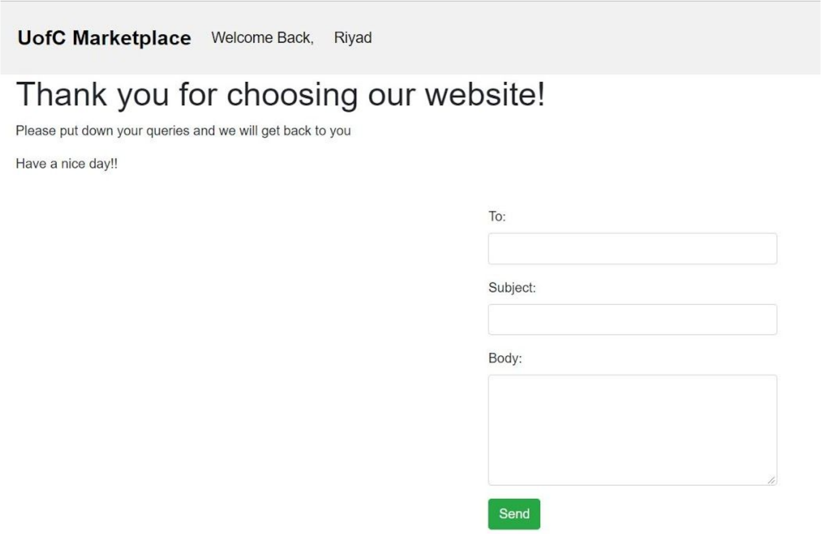
c) Add Product

Users can click on the “Add Product” button to create a post for the product they wish to sell on the website for other viewers to see.



d) Contact Sellers

Interested buyers will be able to reach out to seller by clicking “Contact”. They are redirected to this page where the user has to option to send a custom mail to another person with their e-mail address.



Thank you for choosing our website!

Please put down your queries and we will get back to you

Have a nice day!!

e) Delete Products

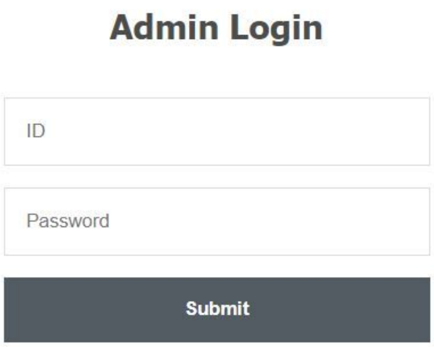
Users can click on the remove button when inside “My Products” to delete products from their list.

Xbox One

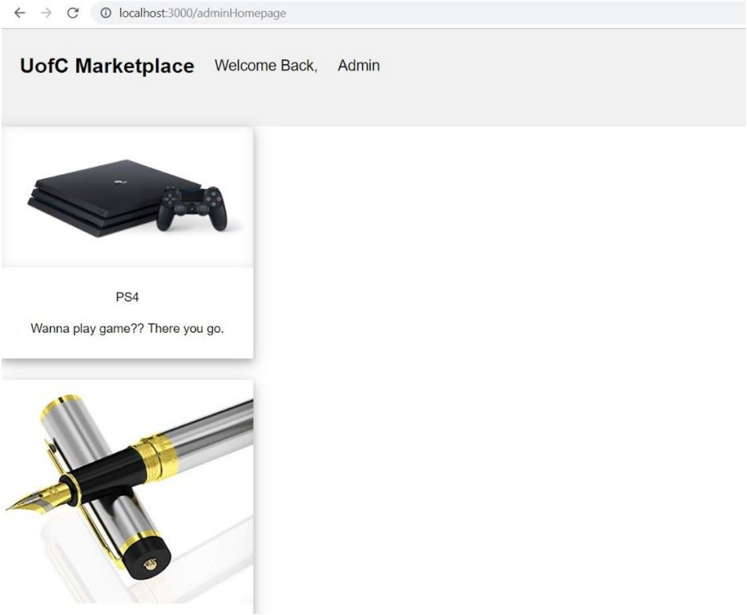
Brand new still intact

2. Admin Controls

a) Login: similar to the login process of the normal customer. the user is redirected for a login page, where they input their credentials.



A query then runs in the background to check that these credentials exist in our adminCredentials table. If the input is correct, then the user is then prompted access to the admin webpage.

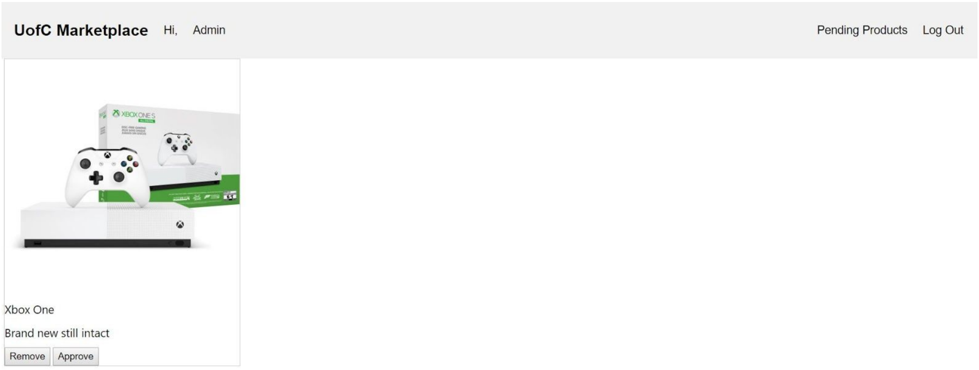

localhost:3000/adminHomepage

They have full access over our system:

- checking all listings
- deleting products
- approving on products that customer added to our system.

b) **Approve/Disapprove Product:**

After the admin logs in, they can then approve of products added by customers or disapprove on them. When a customer adds a product it automatically gets a ‘pending status’ and is not shown in the inventory until the admin approves the product. Approving items posted is done using a special query, to change the pending status flag to ‘active status’ flag. When an item is disapproved, an automatic email is sent for the customer explaining that the item was disapproved. The item is then deleted from our system, using the “Delete” query.



Xbox One

Brand new still intact

c) **Logout:**
Session ends when the user clicks the “Log Out” button. Works for both the customer and admin.

Conclusion

Within a short span of time and huge course load of this semester, we have successfully been able to finish the project close to 90 percent. A marketplace website built for the students of UofC will pave the way for the students to easy and safe access to the materials they need. And also in the long run saving their time and money. A successful website built over one of the most powerful web designing framework express.js really carries a heavy load and helped to redirect and render the HTML files pretty smoothly which results in a flawless transaction between the students.

References

[1] Elmasri. R. Navathe S. B. Fundamentals of Database Systems. 7th Edition. Pearson.(2015).

[2] Wikipedia. (2019). E-commerce. https://en.wikipedia.org/wiki/E-commerce.

[3] Wikipedia. (2019).

[4] w3schools.com.(2019).HTML Tutorial.https://www.w3schools.com/html/.
[5] w3schools.com.(2019).Node.js Tutorial.https://www.w3schools.com/nodejs/.
[6] w3schools.com.(2019).SQL Tutorial.https://www.w3schools.com/sql/.
[7] npm.(2019).NPM.https://www.npmjs.com/.