UTD

CS6360.003 Database Design

Prof. Jalal Omer

ONLINE AUCTION DATABASE

Final Report

Name	Student-ID
Dharav Bhatt	DNB210000
Rainam Shah	RJS190010
Neha Ann John	NAJ210000
Rutvik Avaiya	RXA210000
Rahul Gauri	RXG200002

Erik Johnson School of Engineering and Computer Science

12/03/2021

Table of Contents

Section 1 - Introduction …… 3

Section 2 - System Requirement

2.1 System Architecture System ……... 4

2.2 Functional Requirements …… 4

2.3 Non-Functional Requirements …… 6

2.4 Interface Requirements ……... 7

Section 3 - Conceptual Design of the Database

3.1 Entity-Relationship (ER) Model ……. 8

3.2 Data Dictionary ……... 9

3.3 Business Rules ……... 12

Section 4 - Logical Database Schema

4.1 Schema of the Database…….. 13

4.2 SQL Statement for constructing the schema …….. 14

Section 5 - Functional Dependencies

5.1 Functional Dependencies ……... 19

Section 6 - The User Manual of the System

6.1 System Installation Description ……. 21

Section 7 - Additional Queries and/or Views ……... 33

References …… 35

Conclusion and Future Work ……... 36

Appendix …….. 37

Section 1
Introduction

System Description

Our system/program will contain information about each valid member (buyers and sellers) and will be recognized by their unique identification number and is detailed with other information including phone number, Member Location, Member name, E-mail address, and password.

Users participating in our system would be either buyers or sellers. In the database, the buyer's shipping address would be stored and from the seller's end, the bank account number and the routing number will be stored.

The products for sale are listed by the seller and are allocated a unique article the number assigned by the system. Products are also described by the article name, beginning bid price, bidding increment, auction start date, and the auction end date.

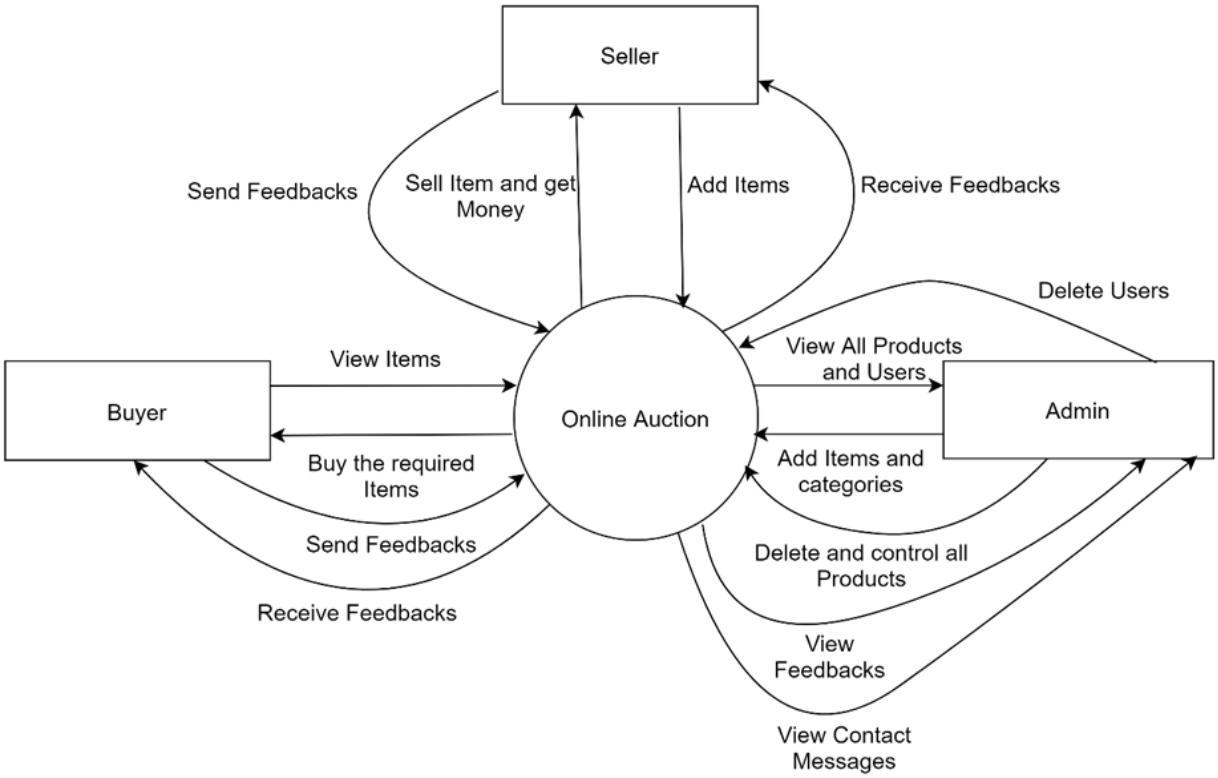
A predefined categorization hierarchy is also used to categorize elements.

Bids are placed on things buyers are interested in. The Bid amount and the Bid occurrence time are being recorded in the database. The purchaser with the utmost bid at the end of the auction is announced the winner and then the agreement between the buyer and seller is set about.

Rating the facility is implemented in our system through which buyer and seller can rate their individual feedback regarding their completed deal and can comment as well.

Section 2
System Requirements

2.1 System Architecture Diagram



2.2 Functional Requirements

Functional Requirements of	Functional Requirements
Login	The system should allow the users to log in only after verifying the username and password.
	The system should restrict unauthorized users from logging in.
	Facilitate the login process by remembering username and password.

	A new user should be able to create a new account.
	Users can Sign Out.
Browser	The system should allow users to bid on all available and desired products.
	The System should allow users to view expired as well as live biddings.
	The System should allow sellers to advertise the product they want to sell.
	The System should allow buyers to view their purchase history.
Administration	Admin should be able to control, view, delete, and search all the products.
	Admin should be able to view all the users and their feedbacks.
	Admin should be able to view messages between buyers and sellers.
	Admin should be able to delete a user's account in case of a violation of website policies.
	Admin should be able to view all the users whether it is buyers or sellers.
	Admin should be able to add categories in the database.

2.3 Non-Functional Requirements

NFR Type	NFR	Description
Time	Latency	Time period between issuing of command and response received for it.
Application Access	Multi-user Bids	Multiple-user bidding for the same item at a given point of time, will be able to put up bids.
Performance	System Performance	Time taken to access the information stored in the database, should not take more than a minute.
		The application should load in less than a minute.
		The application should be able to handle 1000 transactions per hour.
		The database should be able to handle/store a minimum of 5 Gb's of data.
Availability	Operational Hours	Maintenance times/Downtime when upgrading the application.

2.4 Interface Requirements

1. Hardware Requirements:

- RAM: 2 GB or higher
- OS: Windows 8 or higher
- Hard disk space: 10GB
- CPU: i5 core

2. Software Requirements:

- Java (to create the Application program) / JDBC (for connection purposes)
- MySQL (for database creation)

3. User Interface Requirements:

- Admin
- Sign up (for old buyer/sellers and new buyers/sellers)
- Buyer's Page
- Seller's Page

4. Front-end Technology:

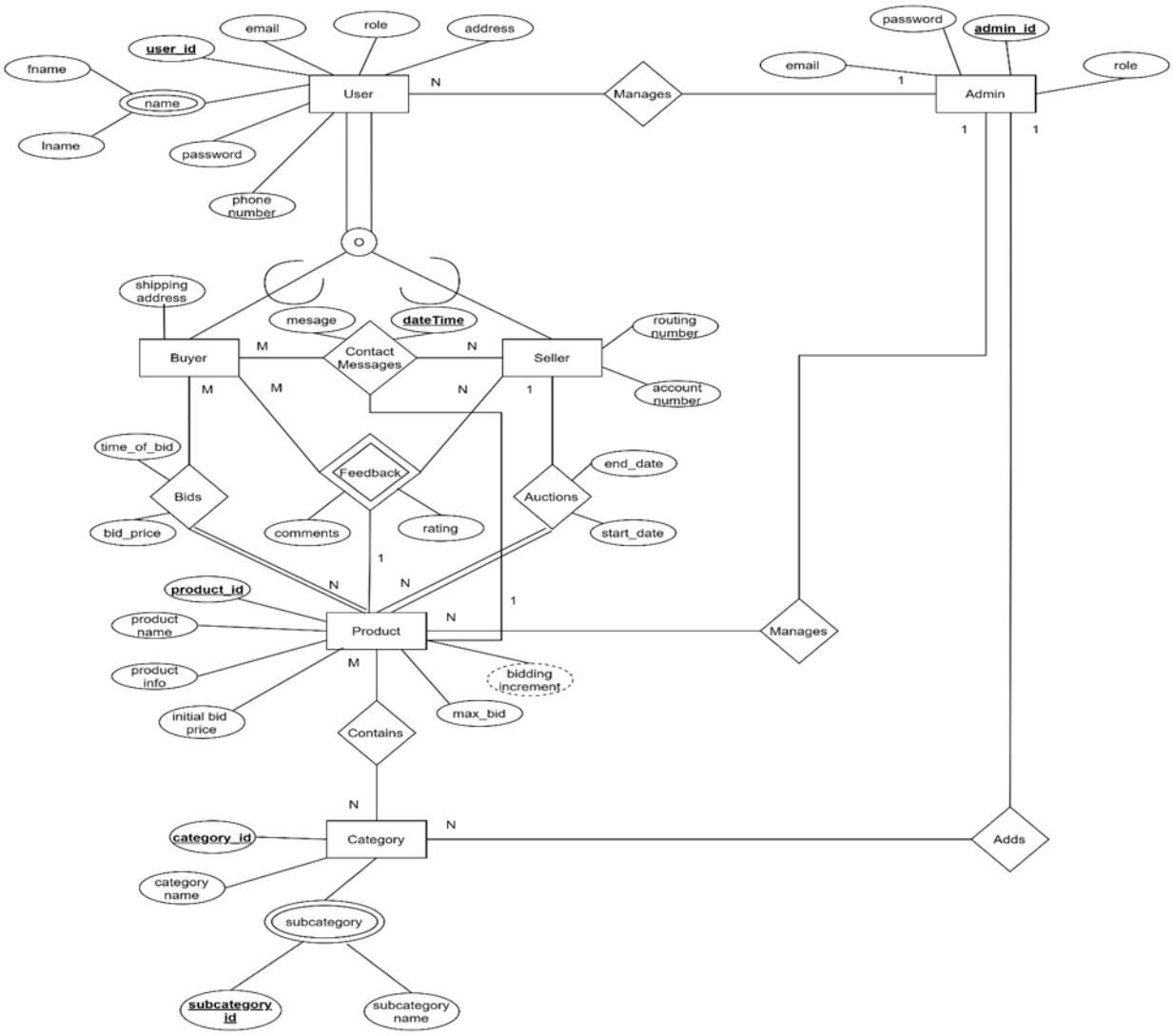
- JSP
- JavaScript
- CSS

5. Back-end Technology:

- Java – Spring Boot
- MySQL Database

Section 3
Conceptual Design of the Database

3.1 ER Diagram



3.2 Data Dictionary

ADMIN		
admin_id	INT	Primary Key of the table.
email	VARCHAR	Email ID of the admin.
Phone_number	VARCHAR	Phone Number of the admin.
password	VARCHAR	Password which will be set by admin to login to the account.

USER		
user_id	INT	Primary Key of the table.
email	VARCHAR	User Email ID.
role	VARCHAR	User role. i.e. either Buyer or Seller.
address	VARCHAR	User Address.
fname	VARCHAR	User first name.
lname	VARCHAR	User last name.
password	VARCHAR	password which will be set by user to login to the account.
phone_number	VARCHAR	User Phone Number.
managed_by	VARCHAR	Admin name. Ref: ADMIN.

SELLER		
seller_id	INT	Primary Key of the table.
user_id	INT	Seller id. Ref: USER.
routing_number	VARCHAR	Seller routing number of payment purpose.
account_number	VARCHAR	Seller account number of payment purpose.

BUYER		
buyer_id	INT	Primary Key of the table.
user_id	INT	Seller id. Ref: USER.
shipping_address	INT	Shipping Address of the user for delivery purpose.

CATEGORY		
category_id	INT	Primary Key of the table.
category_name	VARCHAR	Name of the Category. Eg. Sports.
creaed_by	INT	Id of the Admin who will add this category to system.

SUBCATEGORY		
sub_category_id	INT	Primary Key of the table.
sub_category_name	VARCHAR	Name of the Sub-category. Eg. Tennis-racket.
Category_id	INT	Id of the Category under which this sub-category falls.

PRODUCT		
product_id	INT	Primary Key of the table.
product_name	VARCHAR	Name of the product.
product_info	VARCHAR	Product Information.
initial_bid_price	INT	Price at which the bid starts.
max_bid_price	INT	Price at which the bid stopped.
Image_path	VARCHAR	Image of the product
managed_by	INT	Id of the admin.
sub_category_id	INT	Id of the sub-category.

FEEDBACK		
feedback_id	INT	Primary Key of the table
seller_id	INT	Id of the seller who sold the product.
buyer_id	INT	Id of the buyer who gives the feedback.
product_id	INT	Id of the product for which feedback is given.
feedback_time	DATETIME	Time at which feedback is given.
rating	INT	Rating buyer gave for the product purchased.
comments	VARCHAR	Comment buyer gave for the product purchased.

AUCTION		
auction_id	INT	Primary Key of the table
seller_id	INT	Id of the seller who sold the product.
product_id	INT	Id of the product purchased.
auction_date	DATETIME	Date on which auction took place.
expiration_date	DATETIME	Date on which auction will expire.

BIDS		
bids_id	INT	Primary Key of the table
buyer_id	INT	Id of the buyer who gives the feedback.
product_id	INT	Id of the product purchased.
bid_time	DATETIME	Time at which an individual made a bid.
bid_price	INT	Price of the bid an individual submitted for a given product.

CONTACTMESSAGES		
contactmessages_id	INT	Primary Key of the table
seller_id	INT	Id of the seller who sold the product.
buyer_id	INT	Id of the buyer who purchased the product.
product_id	INT	Id of the product for which buyer is contacting the seller.
message_time	DATETIME	Time at which an individual made a bid.
messages	VARCHAR	Text communication that took place between buyer and seller.

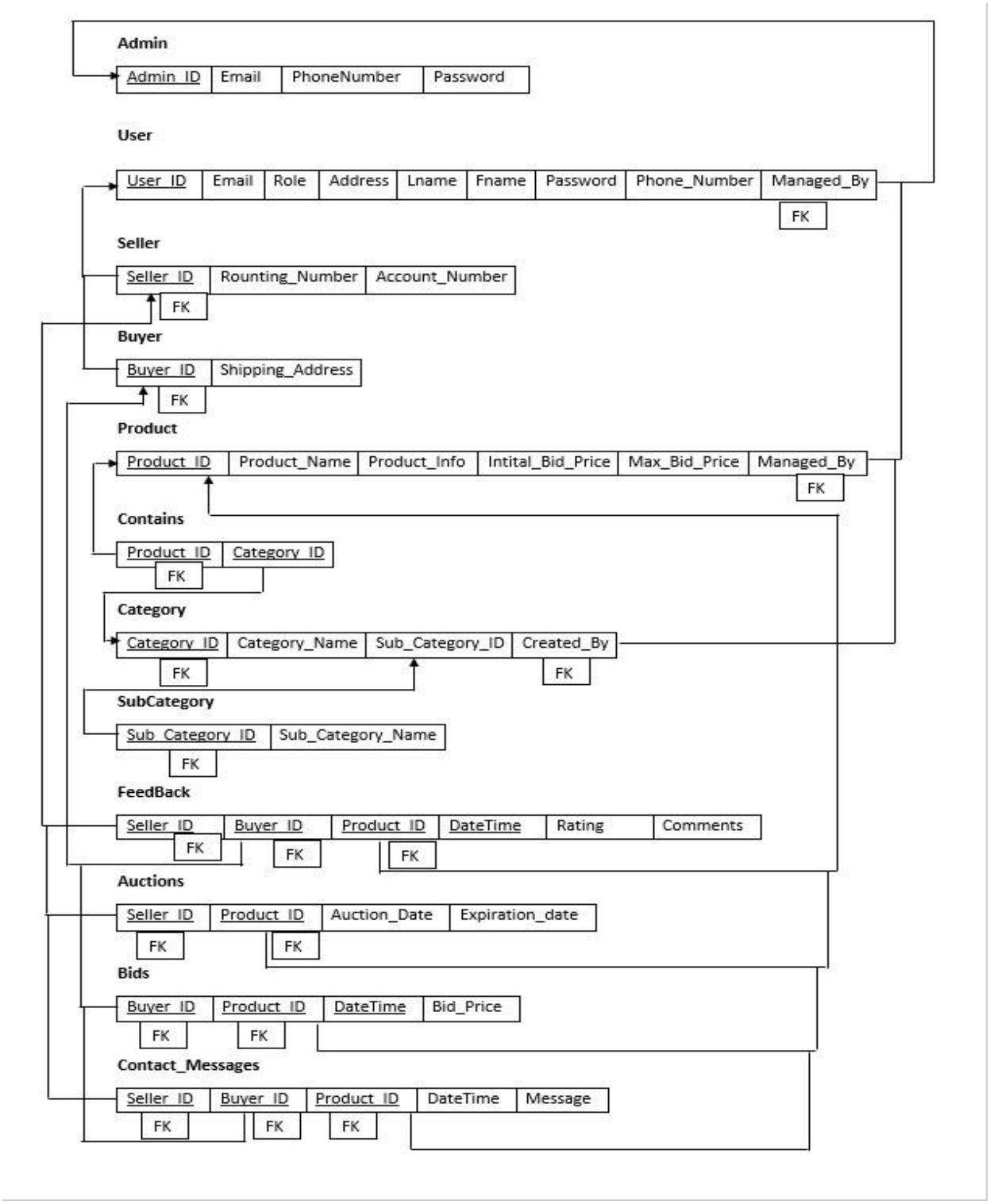
3.3 Business Rules

Below is the list of business rules we have taken in our project:

- Admin can view all contact messages/feedbacks, manages data from buyers and sellers.
- User can either be buyer or seller or it can be both.
- Buyer can bid on multiple products.
- Seller can auction for one or more products.
- A product might fall under multiple categories.
- Buyers and sellers can send feedback after completion of the transaction.

Section 4
Logical Database Schema

4.1 Schema of the Database



4.2 SQL statements for constructing the Schema

```
CREATE DATABASE online_auction;
USE online_auction;
CREATE TABLE ADMIN (
admin_id        VARCHAR (5) NOT NULL,
email           VARCHAR (25),
phone_number        VARCHAR (10) NOT NULL,
passcode        VARCHAR (10) NOT NULL,
PRIMARY KEY (admin_id)
);

CREATE TABLE USER (
user_id         VARCHAR (5)         NOT NULL,
email           VARCHAR (25)        NOT NULL,
role            VARCHAR (15),
address         VARCHAR (30),
fname           VARCHAR (15)        NOT NULL,
lname           VARCHAR (15)        NOT NULL,
passcode        VARCHAR (10)        NOT NULL,
phone_number        VARCHAR (10)        NOT NULL,
managed_by    VARCHAR (5),
PRIMARY KEY (user_id),
FOREIGN KEY (managed_by) REFERENCES ADMIN (admin_id)
);
```

```
CREATE TABLE SELLER (
seller_id        VARCHAR (5)        NOT NULL,
routing_Number       VARCHAR (9)        NOT NULL,
account_Number       VARCHAR (12)       NOT NULL,
PRIMARY KEY (seller_id),
FOREIGN KEY (seller_id) REFERENCES USER (user_id)
);

CREATE TABLE BUYER (
buyer_id             VARCHAR (5)        NOT NULL,
shipping_Address     VARCHAR (30)       NOT NULL,
PRIMARY KEY (buyer_id),
FOREIGN KEY (buyer_id) REFERENCES USER (user_id)
);

CREATE TABLE CATEGORY (
category_id    VARCHAR(5)         NOT NULL,
category_name        VARCHAR(20)        NOT NULL,
created_by     VARCHAR(5)         NOT NULL,
PRIMARY KEY (category_id),
FOREIGN KEY (created_by) REFERENCES ADMIN(admin_id)
);

CREATE TABLE SUBCATEGORY (
sub_category_id      VARCHAR(5)         NOT NULL,
sub_category_name    VARCHAR(15)        NOT NULL,
```

```
category_id             VARCHAR(5),
PRIMARY KEY (sub_category_id)
);

CREATE TABLE PRODUCT (
product_id      INT(5) NOT NULL,
product_name VARCHAR(20)        NOT NULL,
product_info    VARCHAR(500),
initial_bid_price INT           NOT NULL,
max_bid_price           INT             NOT NULL,
managed_by    INT(5) NOT NULL,
sub_category_id         INT(5) NOT NULL,
Image_path     VARCHAR(500),
PRIMARY KEY (product_id),
FOREIGN KEY (managed_by) REFERENCES ADMIN(admin_id),
FOREIGN KEY (sub_category_id) REFERENCES SUBCATEGORY(sub_category_id)
);

CREATE TABLE FEEDBACK(
seller_id       VARCHAR(5) NOT NULL,
buyer_id        VARCHAR(5) NOT NULL,
product_id      VARCHAR(5) NOT NULL,
feedback_time           DATETIME ,
rating          INT,
comments        VARCHAR(250),
```

```
PRIMARY KEY (seller_id, buyer_id, product_id),
FOREIGN KEY (seller_id) REFERENCES SELLER(seller_id),
FOREIGN KEY (buyer_id) REFERENCES BUYER(buyer_id),
FOREIGN KEY (product_id) REFERENCES PRODUCT(product_id)
);

CREATE TABLE AUCTIONS (
auction_id      VARCHAR(5) NOT NULL,
seller_id       VARCHAR(5) NOT NULL,
product_id      VARCHAR(5) NOT NULL,
auction_date    DATETIME NOT NULL,
expiration_date DATETIME,
PRIMARY KEY (seller_id,product_id, auction_date),
FOREIGN KEY (seller_id) REFERENCES SELLER(seller_id),
FOREIGN KEY (product_id) REFERENCES PRODUCT(product_id)
);

CREATE TABLE BIDS(
buyer_id        VARCHAR(5) NOT NULL,
product_id      VARCHAR(5) NOT NULL,
bid_time        DATETIME,
bid_price       INT,
PRIMARY KEY (buyer_id, product_id),
FOREIGN KEY (buyer_id) REFERENCES BUYER(buyer_id),
FOREIGN KEY (product_id) REFERENCES PRODUCT(product_id));
```

```
CREATE TABLE CONTACTMESSAGES(
seller_id VARCHAR(5) NOT NULL,
buyer_id VARCHAR(5) NOT NULL,
product_id VARCHAR(5) NOT NULL,
message_time DATETIME NOT NULL,
messages VARCHAR(250),
PRIMARY KEY (seller_id, buyer_id, product_id, message_time),
FOREIGN KEY (seller_id) REFERENCES SELLER(seller_id),
FOREIGN KEY (buyer_id) REFERENCES BUYER(buyer_id),
FOREIGN KEY (product_id) REFERENCES PRODUCT(product_id)
);
```

Section 5
Functional Dependencies

5.1 Functional Dependencies

Admin Relation:-

- Admin_id → {Email, Phone, Number, Password}
- Email → {Admin_id, Phone_number, Password}
- Phone_numbert → {Admin_id, Email, Password}

User Relation:-

- User_id → {Email, Role, Address, Lname, Fname, Phone_number}
- Email, Phone_number, Role → {User_id, Address, Fname, Lname}
- Email, Password → {User_id, Fname, Lname, Role}

Seller Relation:-

- Seller_id → {Routing_number. Account_number}

Buyer Relation:-

- Buyer_id → {Shipping_address}

Bid Relation:-

- BuyerID,ProductID →{Date, Time, Bid_Price}
- BuyerID, Date, Time → {ProductID, Bid_Price}
- ProductID, Bid_Price → {BuyerID, Date, Time}

Auction Relation:-

- SellerID, ProductID → {Auction_Date, Expiration_Date}

Product Relation:-

- Product_id → {Product_name, Prduct_info, Initial_bid_price, Max_Bid_price, Manage_by}
- Product_id, Product_name → {Product_info, Initial_bid_price, Max_bid_price}
- Product_name → {Product_info, Initial_bid_price, Max_bid_price}

Category Relation:-

Category_id → {Category_name, Sub_category_id, created_by}
Category_id, Sub_category_id → {category_name}
Sub_category_id → {Category_name}

SubCategory:-

- Sub_category_id → Sub_category_name

Feedback Relation:-

- Seller_id, Buyer_id, Product_id → {DateTime, Rating, Comments}
- Product_id, DateTime → {Seller_id, Buyer_id, Rating, Comments}
- Product_id → {Rating, Comments}

Contact_Messages:-

- Seller_id, Buyer_id, Product_id, DateTime → {Message}
- Buyer_id, Product_id, DateTime → {Seller_id, Message}
- Seller_id, Buyer_id, DateTime → {Product_id, Message}
- Seller_id, Product_id, DateTime → {Buyer_id, Message}

Section 6
The User Guide of the System

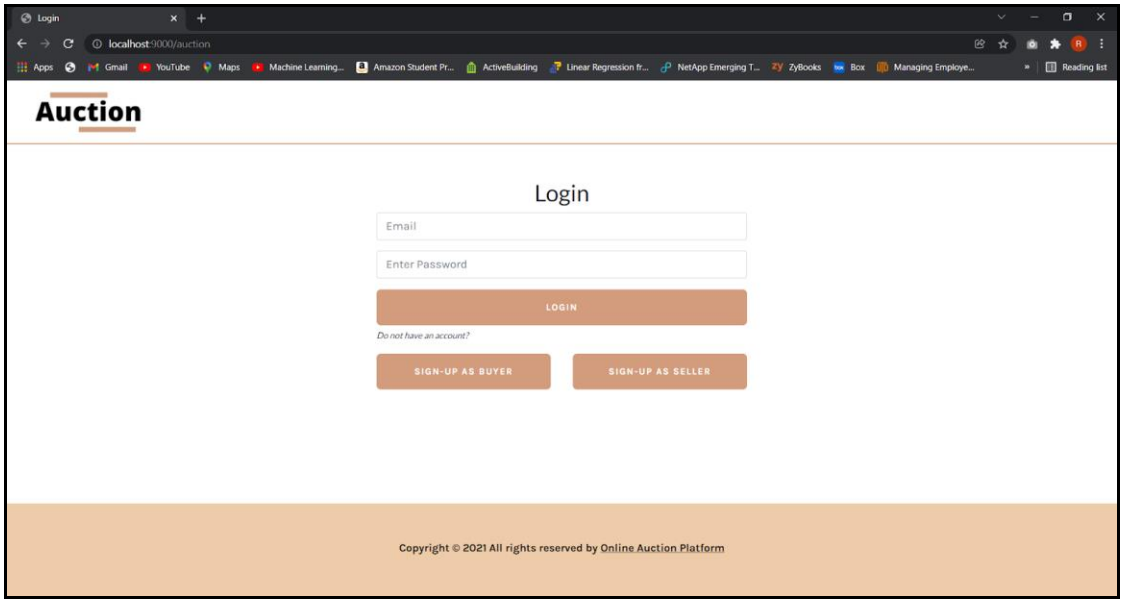
To run this application, you can type the following as a URL on Google Chrome:
Localhost:9000/auction

6.1 The User Manual of the System

1. **User Login:**

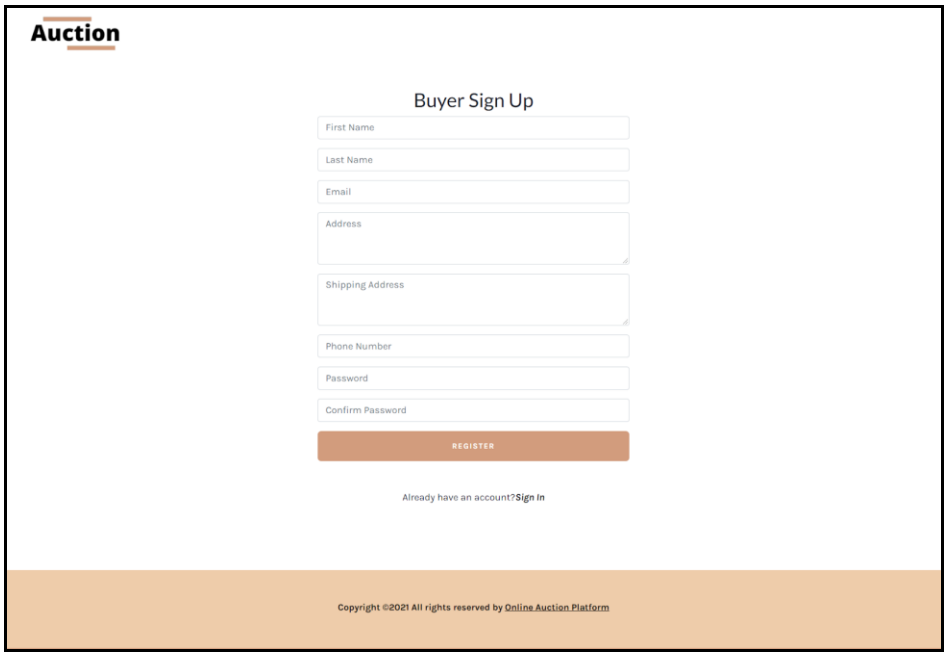
This page has 2 functionalities.

- An existing user can log in using his/her credentials.
- A new user can log in as either a Buyer or Seller.

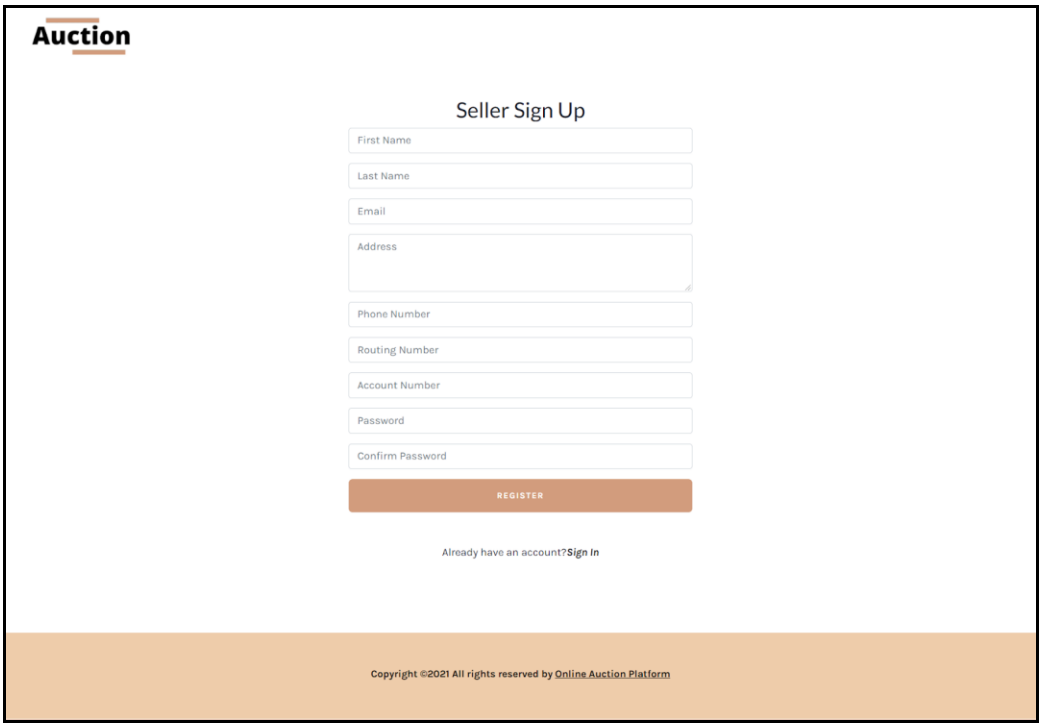


2. **Buyer Sign Up Page:**

Once a new user decides to register as a Buyer, he/she will be redirected to the buyer sign-up page.

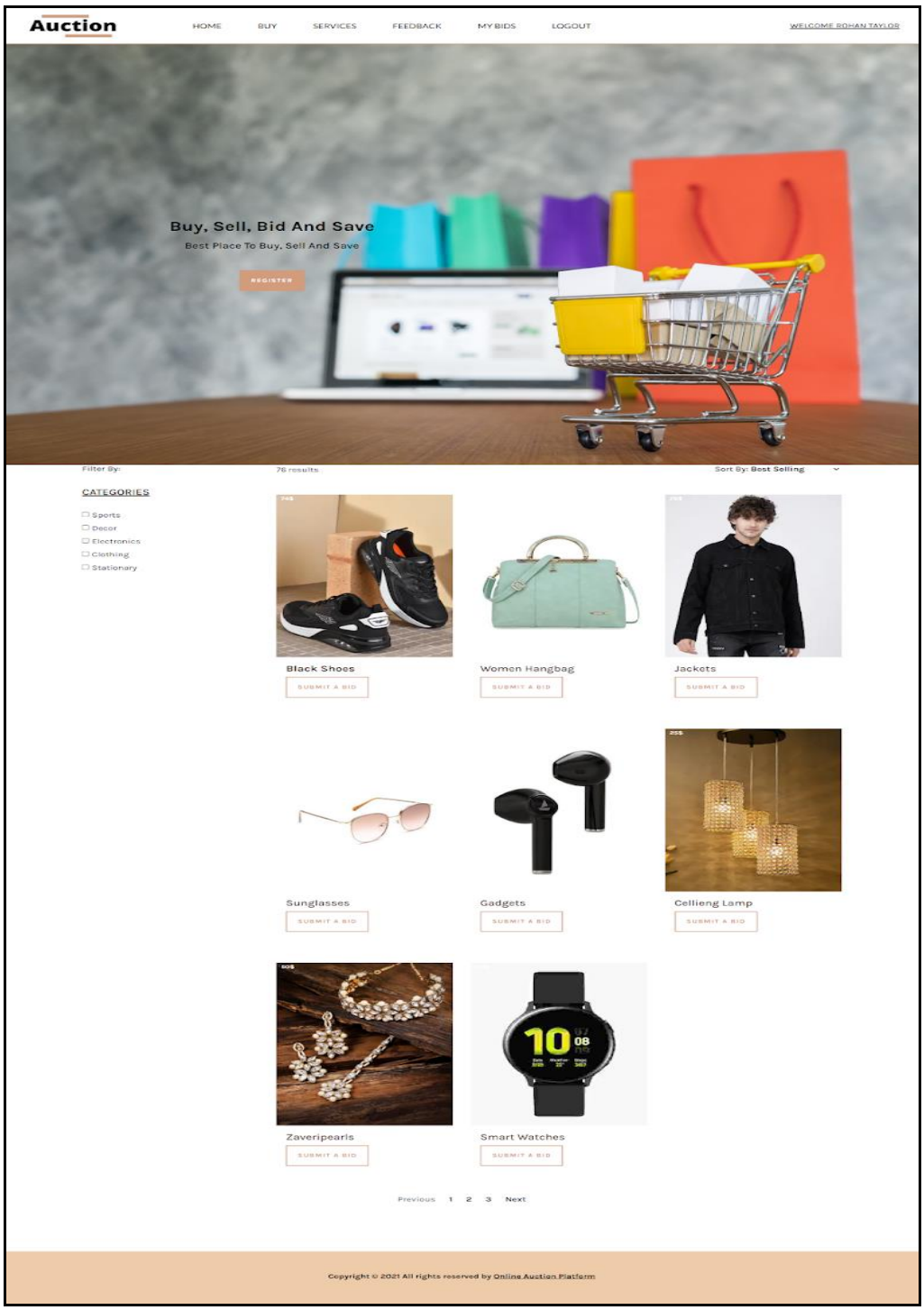


3. **Seller Sign Up Page:** If a new user decides to register as a Seller, he/she will be redirected to the seller sign-up page.

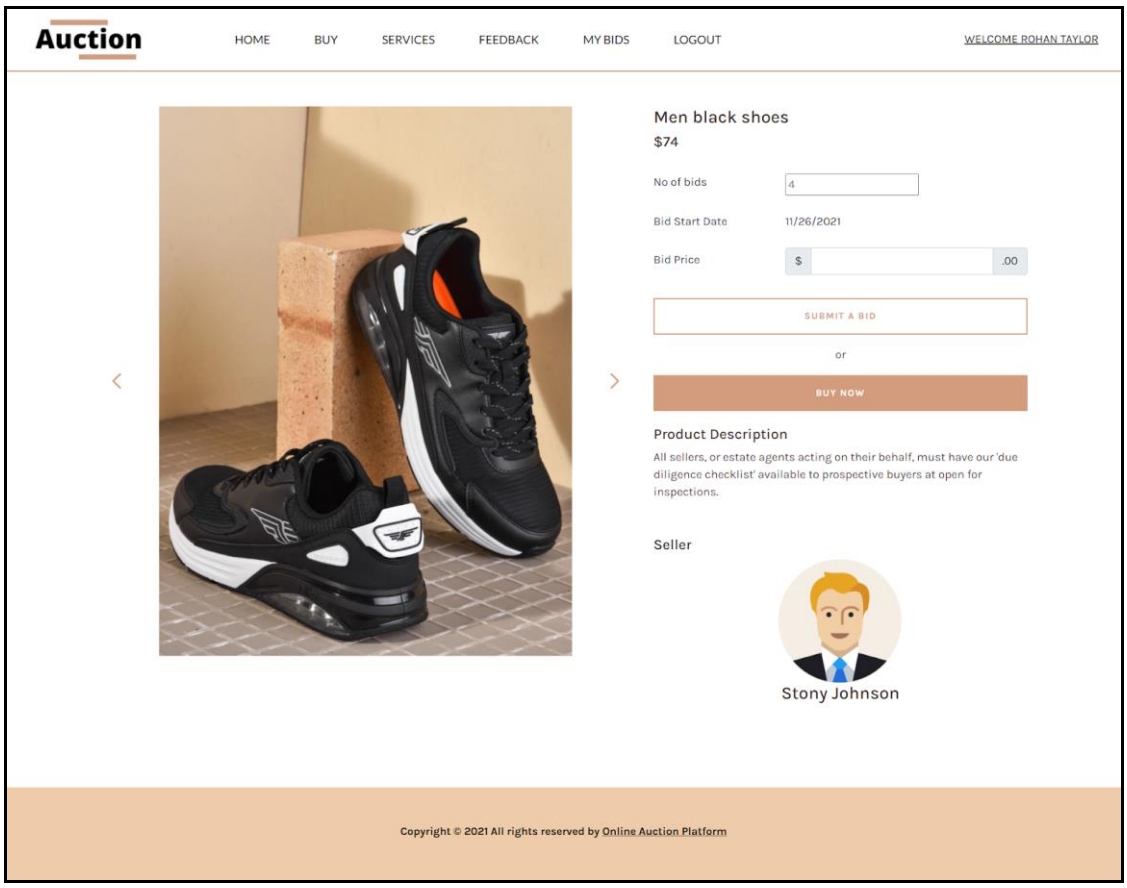


4. **Buyer Home Page:** Once the user has logged in as a Buyer, he /she will be redirected to the buyer home page where the user will be able to view:
    - Recommended products.
    - The tab on the top of the page allows the buyer to navigate between different pages.

5. **Buy Product Page:** Now if the buyer wants to buy any product, he/she can:
   - Either click on the products on the home page and place bids on them (as per the user's liking).
   - Or the user can click on the BUY option available at the top of the page to view all products being auctioned. This will take the user to the buy product page.

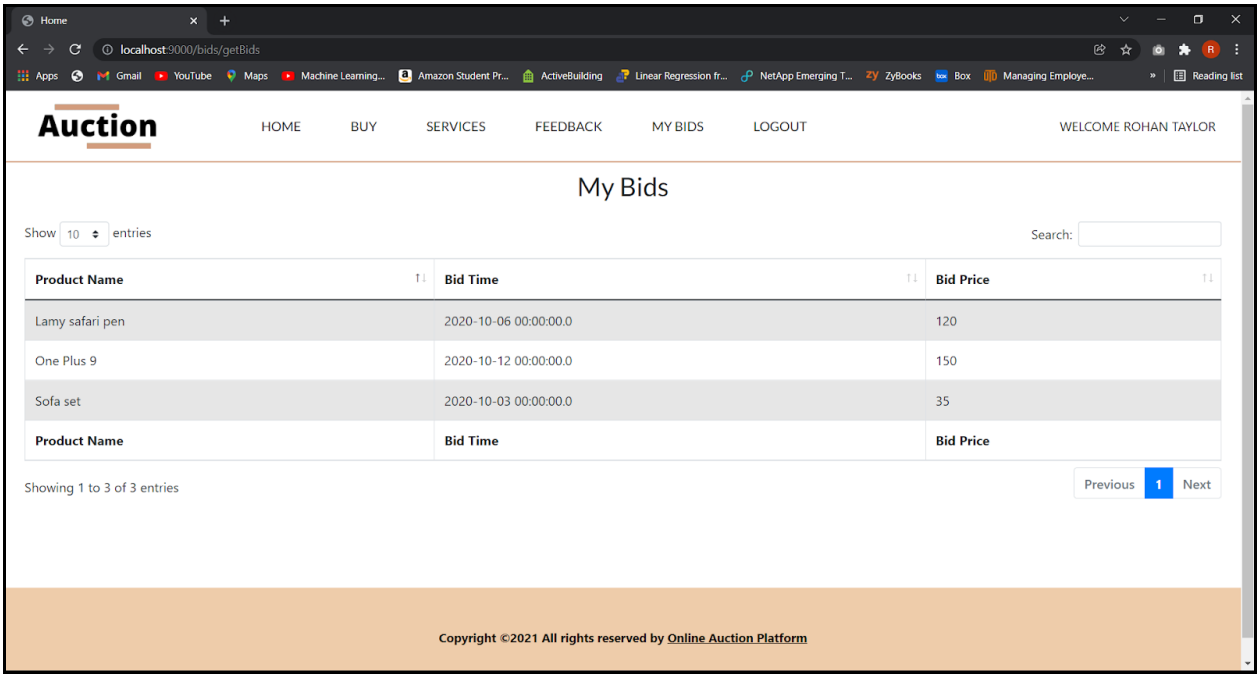


6. **Product Bid Page:**
   - Once the buyer decides to buy a product, he/she can click on the product of his/her choice from the buy product page.
   - Once the user clicks on the product, he/she can view the product details and place a bid (as per the user's liking).

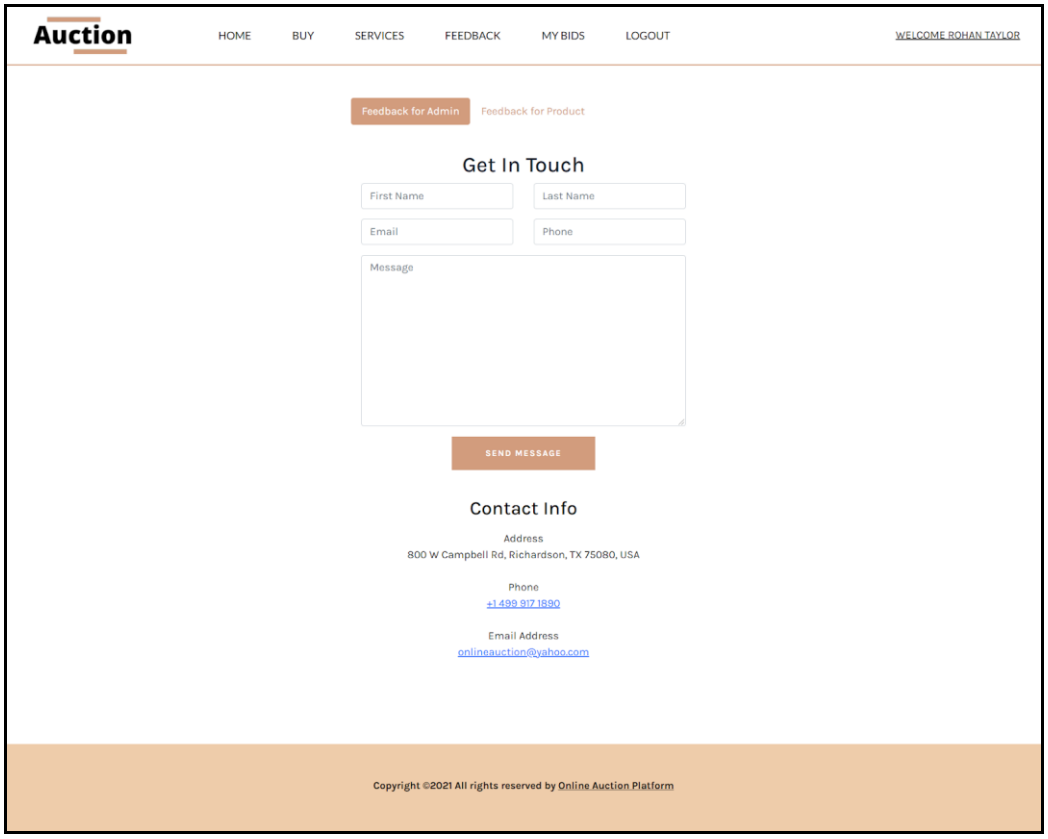


7. **Service Page:** Clicking on the SERVICES option available at the top of the page will open the services page. Here the user/buyer can get an idea of the services the website has to offer.

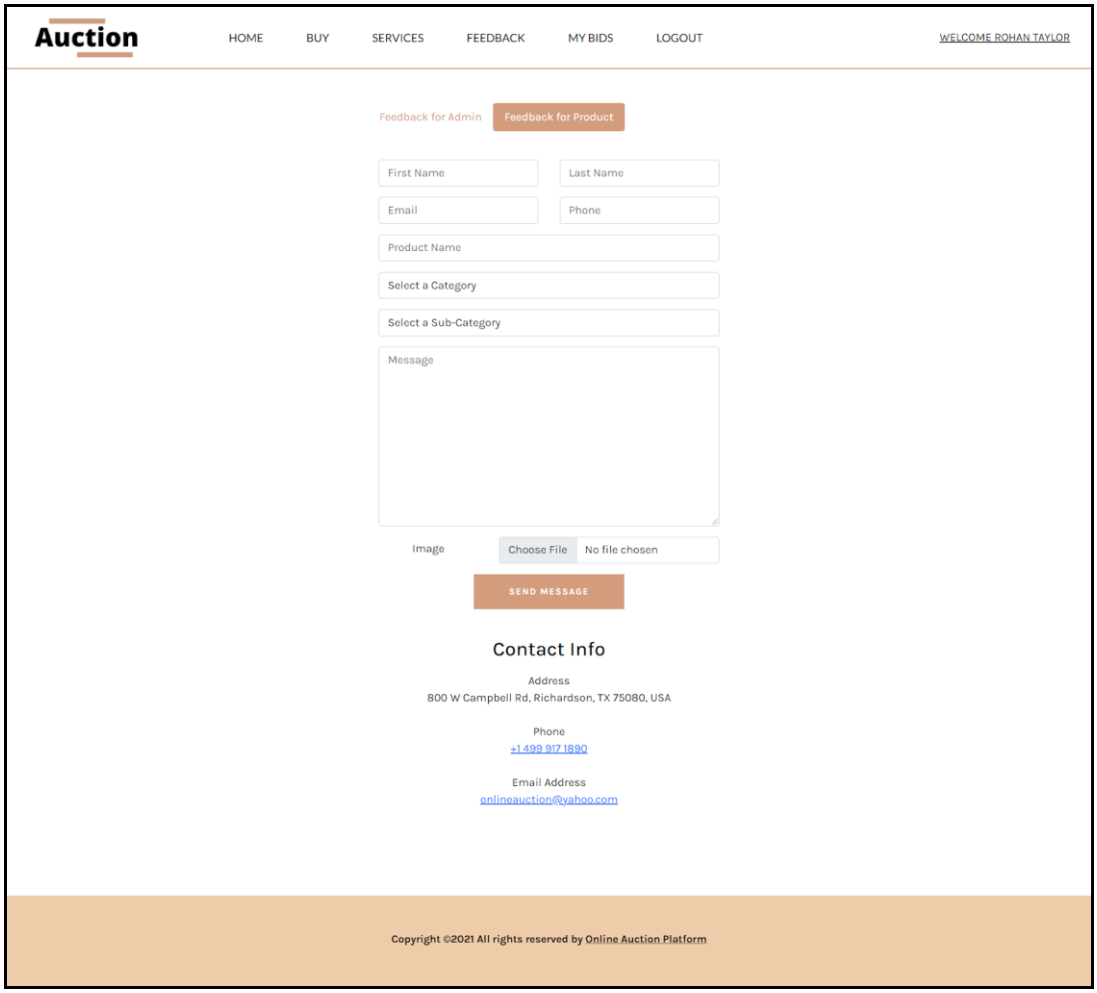
8. **Bid Transactions for Buyer Page:** Here the buyer can view all his/her previous bids.



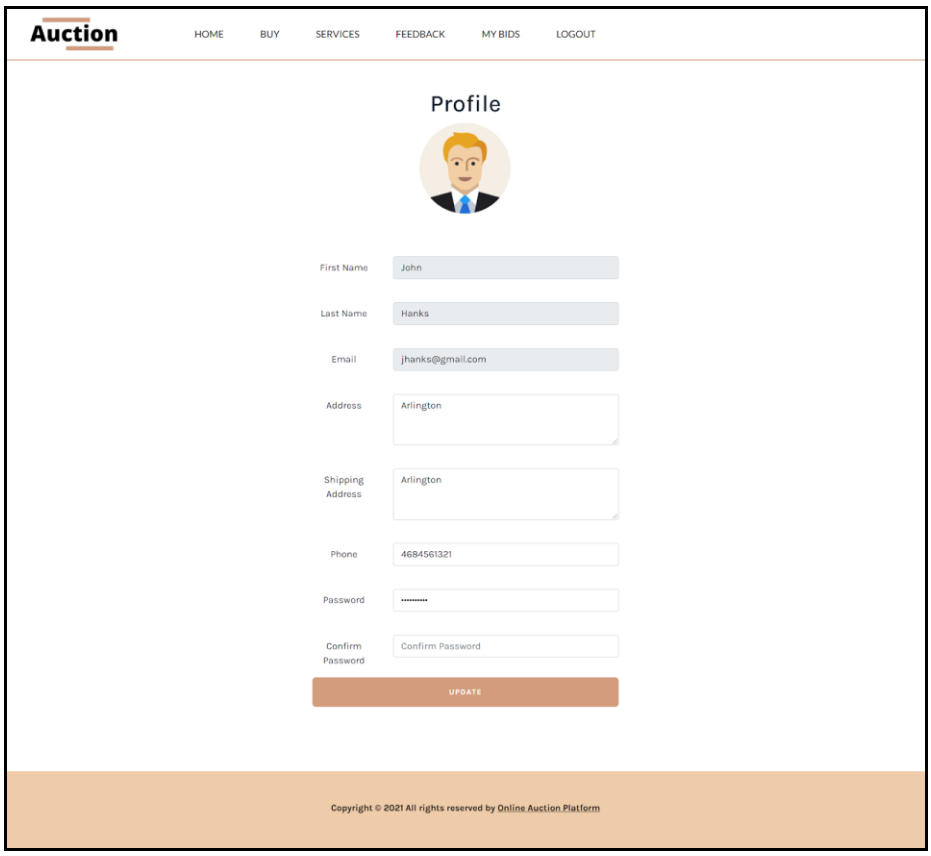
9. **Feedback to Admin Page:** On this page, the buyer can send concerns regarding any issues the user is facing on the website, feedbacks related to products, etc. to the Admin. The buyer can do so by clicking on the FEEDBACK option available at the top of the page.



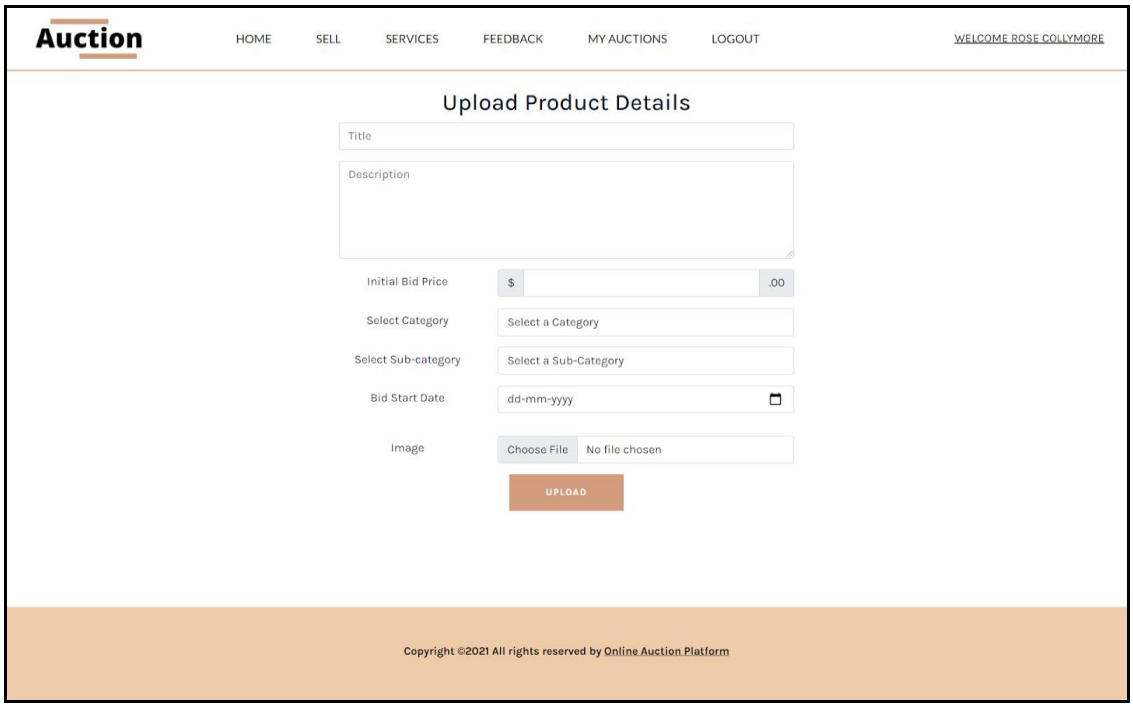
10. **Feedback to Seller Page:** If the buyer has queries, concerns, or reviews regarding a certain product, the buyer can send feedback to the seller by clicking on the feedback to seller option under feedback.



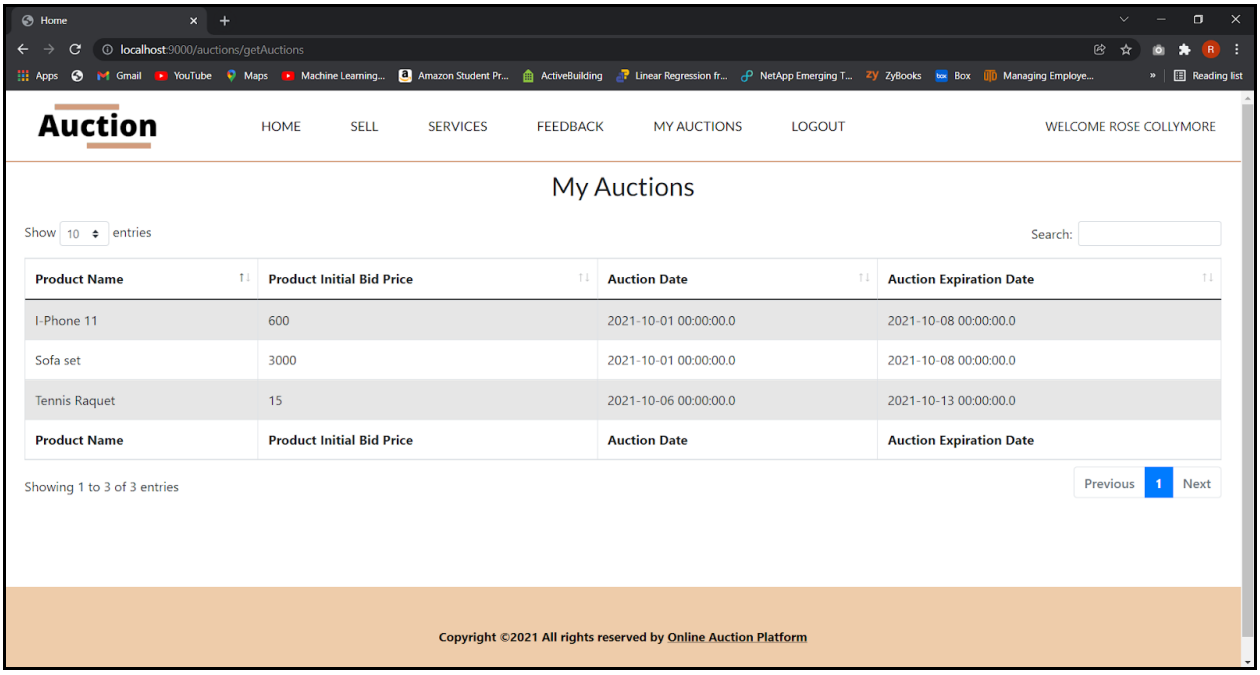
11. **Buyer Update Profile Page:** If the buyer wants to change any information on his/her profile he/she can do so by clicking on the top right corner of the page (welcome prompt).



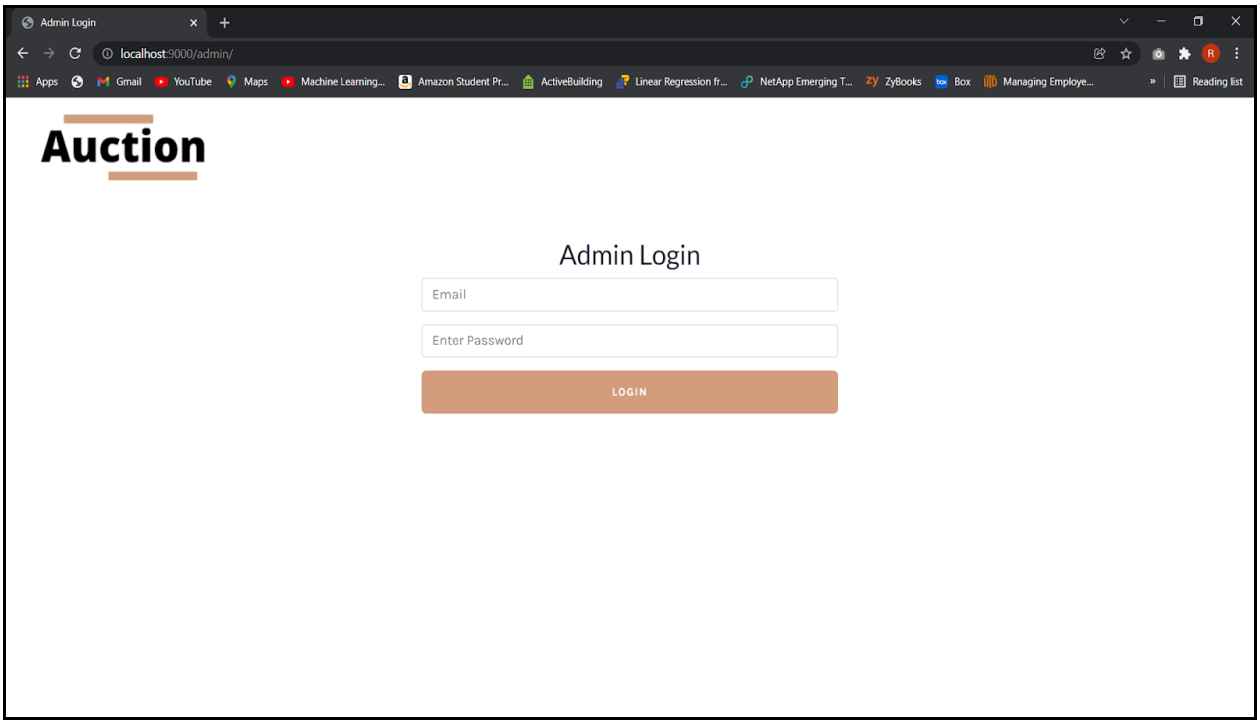
12. **Sell Product page for Seller:** Once the user logs in as a seller, he/she can sell a product by simply clicking on the SELL option available at the top of the seller page. The seller can:
    - Upload images of the product
    - Description of the product
    - Starting Bid price
    - Start date of the bidding



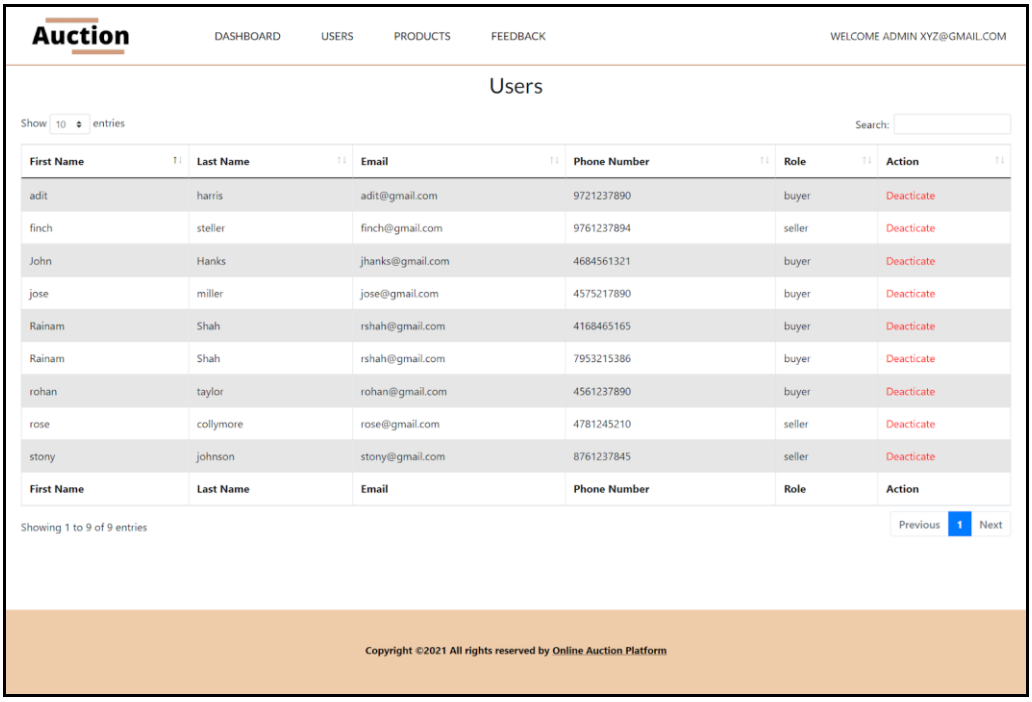
13. **Seller Auction Transaction Page:** On this page, the seller can view all of his/her auctioned products. The seller can view this by clicking on the My Auctions option available at the top of the seller page.



14. **Admin Login:** The Admin will have a different login to the website. Here the Admin can use his/her credentials to log in to the website.



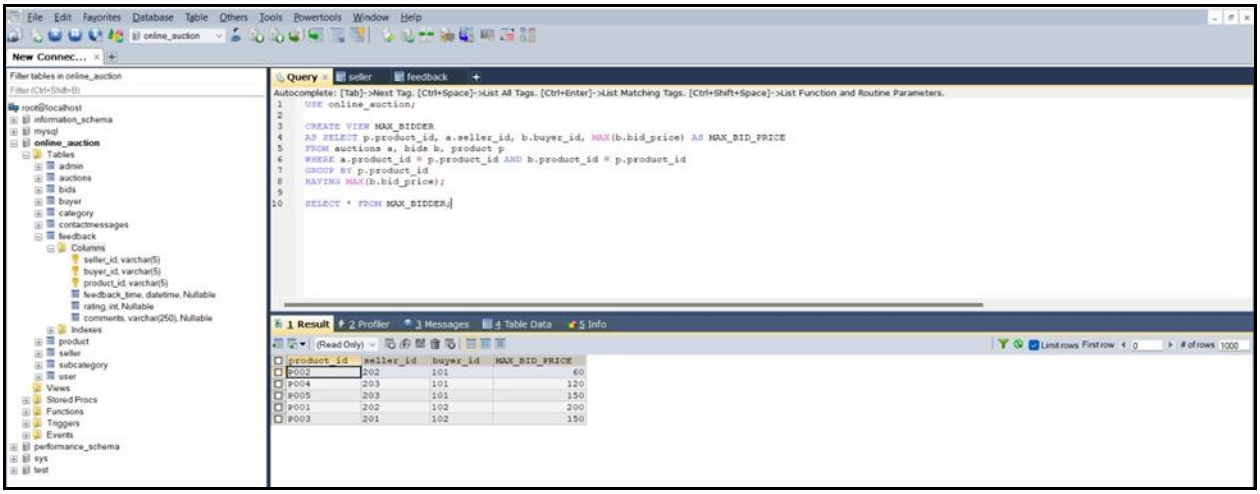
15. **All Users Page for Admin:** Here the Admin can view all the user details (buyers and sellers) who have registered with the auction website. He/she can do so by clicking on the USER'S option available at the top of the admin page.



Section 7
Additional Queries and/or Views

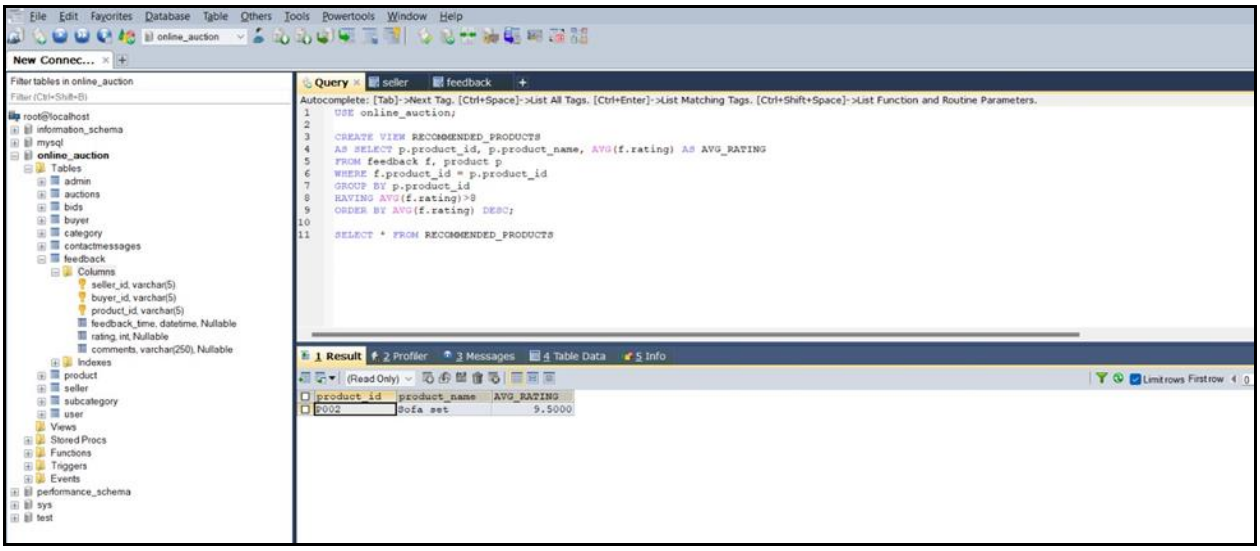
VIEW 1:

```
CREATE VIEW MAX_BIDDER

AS SELECT p.product_id, a.seller_id, b.buyer_id, MAX(b.bid_price) AS MAX_BID_PRICE

FROM auctions a, bids b, product p

WHERE a.product_id = p.product_id AND b.product_id = p.product_id

GROUP BY p.product_id

HAVING MAX(b.bid_price);
```



VIEW 2:

```
CREATE VIEW RECOMMENDED_PRODUCTS
AS SELECT p.product_id, p.product_name, AVG(f.rating) AS AVG_RATING
FROM feedback f, product p
WHERE f.product_id = p.product_id
GROUP BY p.product_id
HAVING AVG(f.rating)>8
ORDER BY AVG(f.rating) DESC;
```



References

1. Fundamentals of Database Systems, 7th edition by Emlasri and Navathe

2. https://spring.io/guides/gs/accessing-data-jpa/

3. https://www.baeldung.com/spring-data-jpa-query

4. https://www.javaguides.net/2018/09/spring-mvc-using-spring-boot2-jsp-jpa-hibernate5-mysql-example.html

Conclusion and Future Work

Conclusion

In conclusion, the Online Auction Database will allow users to buy products from different sellers registered on the website by placing bids on the product. In the end, the user with the highest bid will be able to buy the product from the seller.

During the development of this project, we have learned how to use the Spring tool suite, which provides a ready-to-use environment to implement, debug, run and deploy JAVA applications. And we connected the application to the database using JDBC. In process of doing so, we were able to understand how the connection happens.

We understood the business logic involved in building an Online Auction Database. With this project, we also had the opportunity to see in practice the modelling techniques (use cases, user stories, EER diagrams, Database schema, Database constraints, Functional dependencies) learned during the Database design course.

Future Work

The online auction portal works very well in all of its functionality. However, some future works can be done on the existing system:

- Add an SSL security system. Since a registered seller can post new auctions, the buyer places bids, the users send messages, etc., username and password are sensible data. So it could be useful to protect these data from being intercepted by a third party.
- Add a chat room to the portal. It would be nice for a user to enter a chat room to talk with other users about auctions or any other topic. This chat can be realized using the Java Applet technology.
- Add more attractive graphics to the web pages of the portal. The site is very easy to browse, also for new users, because the pages are simple and clear. However, the graphics of the site is also much simple, so it could be the case to improve it to attract more users.
- Add a credit card payment system. It would be nice for users to make payments using their own credit cards to exchange money with the help of the website.