

CS 6360.003 Database Design  
Database Project  
Instructor: Dr.Jalal Omer

Team Members:

Adhikari Utsav, Devictor Antoine Bruno  
Marie, Ganesan Nadhiya, Xie Mike,  
Zhang Eric Weida

12/03/2021

# Table Of Contents

<b>Table Of Contents</b>	<b>2</b>
<b>Introduction</b>	<b>4</b>
<b>System Requirements</b>	<b>6</b>
Context diagram (system architecture diagram)	6
Interface requirements	7
Functional requirements	7
Non-Functional requirements	8
<b>Conceptual Design of the Database</b>	<b>9</b>
Data Dictionary	10
Entities	10
Relations	10
Business Rules	11
<b>Logical Database Schema</b>	<b>12</b>
SQL Statements	13
# Create and use DB	13
# Define Tables	13
# Trigger to mark seller's items as inactive if not active	15
Expected DB Operations	15
Estimated Data Volumes	16
<b>Functional Dependencies and Database Normalization</b>	<b>17</b>
<b>The Database System</b>	<b>18</b>
<b>Suggestions on Database Tuning (optional)</b>	<b>19</b>
<b>Additional Queries and Views</b>	<b>20</b>
# Views	20
# Select Queries	21
<b>User application interface</b>	<b>23</b>
Screenshots of the User Interface:	27
<b>Conclusion and future work</b>	<b>40</b>
<b>References</b>	<b>41</b>
<b>Appendix</b>	<b>42</b>



## Introduction

### Description of the Project

- The goal of the project was to create an online auction database system for users to buy or sell items.
- Members of the site are uniquely identified by the system and will be described by an e-mail address, name, password, home address, and phone number
  - Buyers have a shipping address
  - Sellers have a bank account number and routing number
- Items are described by various attributes and are categorized in a fixed hierarchy.
  - Items are identified by a unique item id, along with an item name, description, starting bid price, bid increment, start date, and end date.
- The online auction site must allow sellers to place items for sale, and buyers to place bids for items.
  - The winner is declared when the bid ends and is determined by the buyer who entered the highest bid. And the transaction between the buyer and seller will be done outside the site.
  - Buyer and Seller will send feedback after the transaction.

### Section organization

This report is organized into 11 sections. After this Introductory section, Section 2 is the System Requirements, which is an analysis of the requirements for the project. It contains information about how the users may interact with the online auction and both functional and non-functional requirements of the database.

Section 3 is the Conceptual Design of the Database, which describes the conceptual design of the database, and contains the EER model of the database. As well as the data dictionary and business rules.

Section 4 is the Logical Database Schema, which describes the logical design of the database. It shows the schema of the database, as well as the SQL to build the schema and the expected database operations and estimated data volumes.

Section 5 is the Functional Dependencies and Database Normalization, which shows the functional dependencies and what their normalization would be.

Section 6 is the Database System, which shows how to install and use our system. “Screen dumps” will be shown for using our system step by step.

Section 7, Suggestions on Database Tuning (Optional), which gives suggestions on tuning our database??? Optimize database how?

Section 8, Additional Queries and Views, which shows the queries and views we have implemented and its execution results

Section 9, User Application Interface, describes how we built the user interface, and how the different types of users will use our system. It will also show the different functions in our system that will help our users and how they are implemented in SQL.

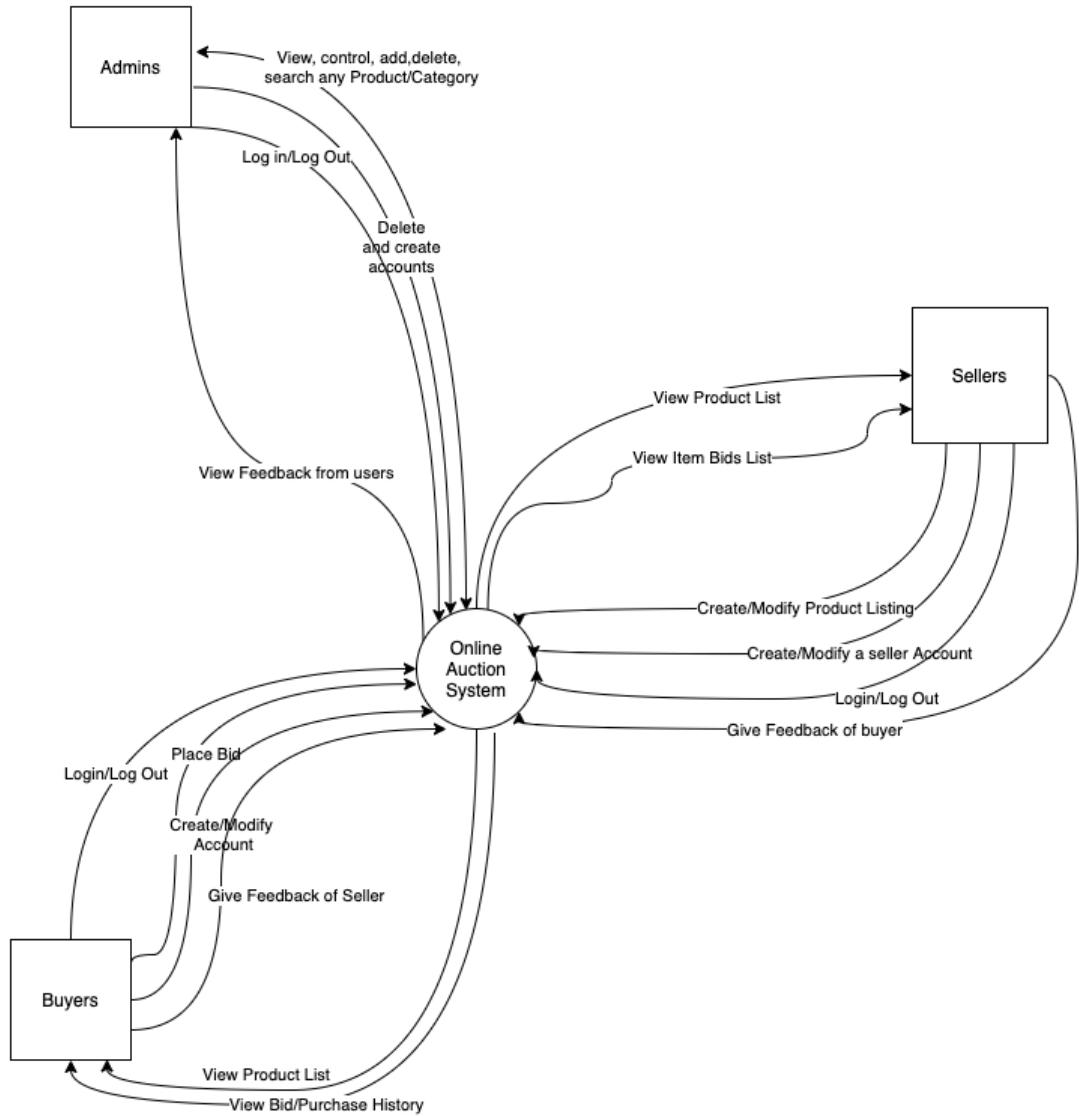
Section 10, Conclusion and Future Work, describes our conclusion with the project and possible future work.

Section 11, References, which lists what technologies we used.

Section 12, Appendix, which gives the zip file for source and docs.

## System Requirements

Context diagram (system architecture diagram)



## Interface requirements

- The application will be structured as a wizard, either in a CLI or GUI interface. If CLI, options will be selected by inputting a number associated with each option. If GUI, options will be selected through radio buttons.

- If input is invalid (e.g. creating an account with an existing username) an error message will be displayed, and the previous prompt will be displayed again.
- The wizard will always contain an option to cancel or “back out” of an option sub-menu. For example, cancelling a bid to go back to the options menu or logging out to go back to the login menu.
- Important, irreversible actions will be protected by a confirmation question.
- Users will only be presented with options available to them in any menu, depending on account type. The three account types are buyer, seller, and admin.

## Functional requirements

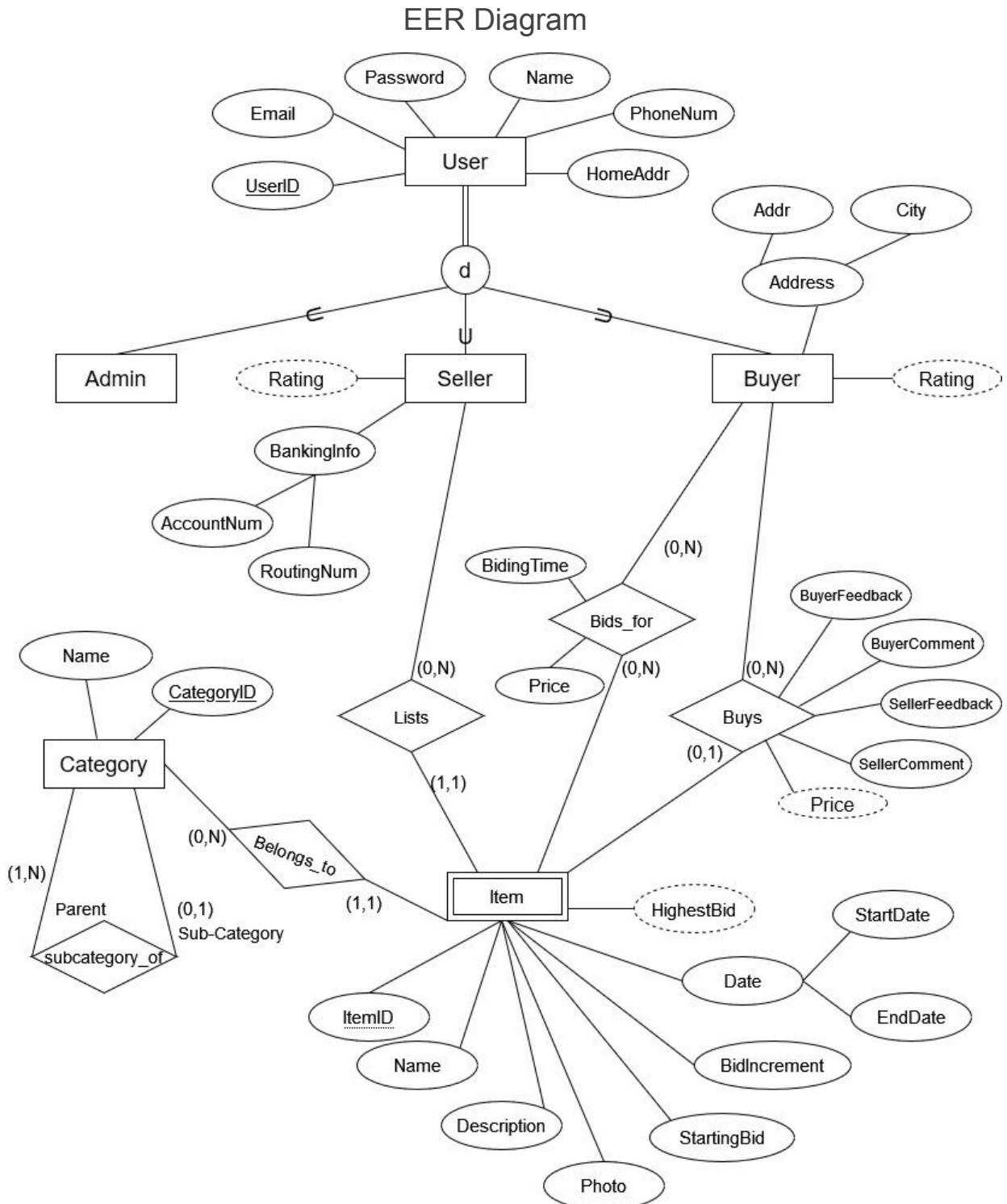
1. Accounts
  - a. System will allow users to login and logout of their accounts
  - b. System will verify the username and password
  - c. System will not allow users to log in with invalid username or password
  - d. System will allow users to create accounts.
  - e. System will assign a unique member number to each user.
  - f. Usernames must be unique across all account types (buyer, seller, admin).
2. Browsing
  - a. System will allow users to search for products that are available for auction and shall display the result.
  - b. System will allow users to search by item name and item classification.
  - c. System will be able to provide all information about items, including details and bids.
3. Buyers
  - a. Buyers must make an account and sign in before bidding.
  - b. Buyers must supply a username, password, home address, phone number, and shipping/mailing address.
  - c. Buyers can view all current and expired bids.
  - d. Buyers can view their purchase history (i.e. won bids).
  - e. Buyers can bid on products after an item auction starts.
  - f. Buyers can only make bids that are at least the current bid plus the seller's bid increment.
  - g. The system will record bid time and price.
  - h. Feedback of the seller will be sent to the system when a transaction is completed, including a rating of the seller and a comment.
  - i. Buyers can modify their account information.
4. Sellers
  - a. Sellers must make an account and sign in before selling.
  - b. Sellers must supply a username, password, home address, phone number, bank account number, and routing number.
  - c. The bidder with the highest bid price is declared the winner and the transaction between buyer and seller may then proceed.
  - d. Sellers can list items to be auctioned.

- e. Sellers will provide item title, description, starting bid price, bidding increment, classification, start date of auction, and end date of auction.
  - f. Items have a unique item number assigned by the system.
  - g. Sellers can view all items they have listed, as well as the bids for those items.
  - h. Feedback of the buyer will be sent to the system when a transaction is completed, including a rating of the buyer and a comment.
  - i. Sellers can modify their account information.
5. Administrators
- a. Admins can view all the products.
  - b. Admins can control, delete and search any product.
  - c. Admins can view the feedback sent from users.
  - d. Admins can add and modify categories in the database.
  - e. Admins can view all users.
  - f. Admins can delete users.
  - g. Admins can view the contact messages.
  - h. Admins can create buyer, seller, and other admin accounts.

## Non-Functional requirements

- The system will be optimized to minimize response time.
- The administrators will have features available to them to support long-term maintenance.
- The system will have features to promote account security.
- The system should scale well when the number of users and bids increase.

## Conceptual Design of the Database



## Data Dictionary

### Entities

User: Represents user. Superclass of all user types of the system: admins, sellers, and buyers.

Attributes: *UserID* (assigned by system), *Email*, *Password*, *Name*, *PhoneNum*,  
*HomeAddr*

Key: *UserID*

Participation constraints: total participation in disjoint derived classes

Admin: Represents admin users with elevated permissions. Admins cannot participate in bidding.

Key: *Username* (inherited from User)

Seller: Represents seller user. Sellers can only list items, and cannot purchase items.

Attributes: *BankingInfo* (composite of *AccountNum* and *RoutingNum*), *Rating* (derived from *buys.SellerFeedback*)

Key: *UserID* (inherited from User)

Buyer: Represents buyer user. Buyers can only bid/purchase items, and cannot list items.

Attributes: *Address* (shipping address, composite of *Addr* (street address) and *City*),  
*Rating* (derived from *buys.BuyerFeedback*)

Key: *UserID* (inherited from User)

Category: Represents categories that may be assigned to items.

Attributes: *Name*, *SubCategoryID*

Key: *CategoryID*

Item: Weak entity. Represents item listing.

Attributes: *ItemID*, *Name*, *Description*, *Photo*, *StartingBid*, *BidIncrement*, *Date* (composite of *StartDate* and *EndDate*), *HighestBid* (derived from *bids\_for.Price*)

Key: *ItemID*, but identified by the *lists* relation.

### Relations

Lists: Defines item listings by seller. Identifying relationship for Item

Cardinality: one-to-many

Participation: partial participation by Seller, total participation by Item

Bids\_for: Defines bids which buyers place on items

Cardinality: many-to-many.

Participation: partial participation by Buyer, partial participation by Item

Assumption: buyers may place multiple bids on the same item.

Buys: Defines final purchases of items by buyers

Cardinality: one-to-many

Participation: partial participation by Buyer, partial participation by Item

Belongs\_to: Defines category which an item belongs to

Cardinality: one-to-many

Participation: partial participation by Category, total participation by Item

Subcategory\_of: Defines hierarchy between categories and subcategories. Recursive relation.

Cardinality: one-to-many

Participation: partial participation on both sides.

## Business Rules

### Domain Constraints

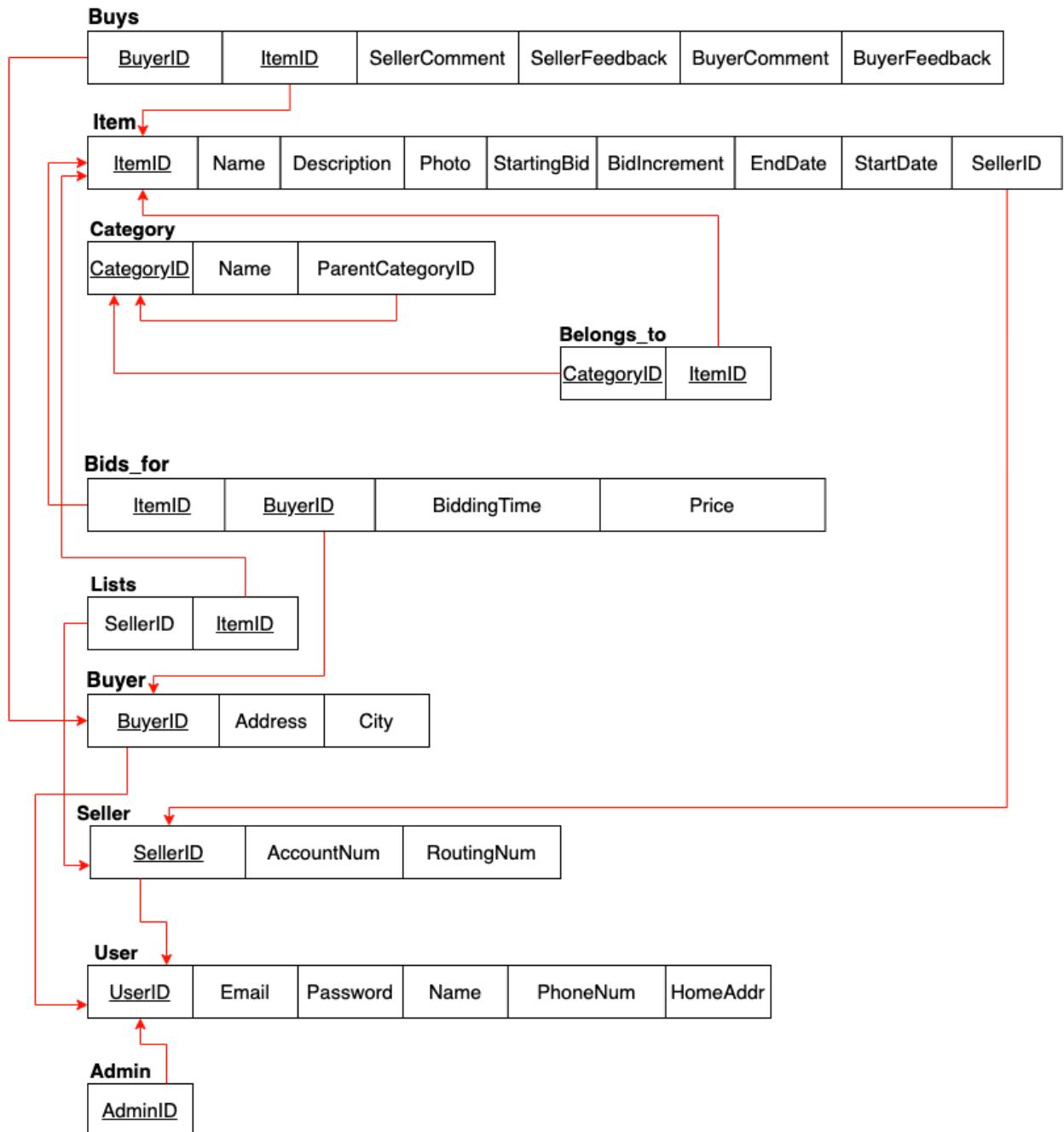
- Alphanumeric username, item name and category name
- Password minimum length is 8 and maximum of 20
- Account Number must be 5-17 digits
- Routing Number must be 9 digits
- Price is in USD
- Feedback (Buyer/ Seller) is an integer from 0 to 10 (inclusive)

### Business Rules

- Administrators are capable of removing users and items which break rules.
- Bids prices for an item must increase by some bid increment with each successive bid.
- End date must be greater than start date for listings.
- User ID is a member number generated by the system.
- The CategoryID and SubCategoryID recursive relation cannot create cycles.

## Logical Database Schema

### Relational Database Schema



## SQL Statements

```
# Create and use DB
```

```
DROP DATABASE IF EXISTS ONLINE_AUCTION;  
CREATE DATABASE ONLINE_AUCTION;  
USE ONLINE_AUCTION;
```

```
# Define Tables
```

```
CREATE TABLE IF NOT EXISTS User(  
    UserID INT NOT NULL AUTO_INCREMENT,  
    Name VARCHAR(255) NOT NULL,  
    Email VARCHAR(100) NOT NULL,  
    Passwd VARCHAR(20) NOT NULL,  
    PhoneNum CHAR(10) NOT NULL,  
    PRIMARY KEY (UserID),  
    HomeAddr VARCHAR(255),  
    IsActive BOOL NOT NULL DEFAULT TRUE  
)
```

```
CREATE TABLE IF NOT EXISTS Admin(  
    AdminID INT NOT NULL,  
    PRIMARY KEY (AdminID),  
    FOREIGN KEY (AdminID) REFERENCES User(UserID)  
)
```

```
CREATE TABLE IF NOT EXISTS Seller(  
    SellerID INT NOT NULL,  
    AccountNum VARCHAR(17) NOT NULL,  
    RoutingNum CHAR(9) NOT NULL,  
    PRIMARY KEY (SellerID),  
    FOREIGN KEY (SellerID) REFERENCES User(UserID)  
)
```

```
CREATE TABLE IF NOT EXISTS Buyer(  
    BuyerID INT NOT NULL,  
    Address VARCHAR(255) NOT NULL,  
    City VARCHAR(255) NOT NULL,  
    PRIMARY KEY (BuyerID),  
    FOREIGN KEY (BuyerID) REFERENCES User(UserID)  
)
```

```
CREATE TABLE IF NOT EXISTS Item(
    ItemID INT NOT NULL AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL,
    Description VARCHAR(255),
    Photo LONGBLOB,
    StartingBid FLOAT NOT NULL,
    BidIncrement FLOAT NOT NULL,
    StartDate TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    EndDate TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    SellerID INT NOT NULL,
    IsActive BOOL NOT NULL DEFAULT TRUE,
    PRIMARY KEY (ItemID),
    FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)
);
```

```
CREATE TABLE IF NOT EXISTS Buys(
    BuyerID INT NOT NULL,
    ItemID INT NOT NULL,
    SellerComment VARCHAR(255),
    SellerFeedback INT,
    BuyerComment VARCHAR(255),
    BuyerFeedback INT,
    PRIMARY KEY (BuyerID, ItemID),
    FOREIGN KEY (BuyerID) REFERENCES Buyer(BuyerID),
    FOREIGN KEY (ItemID) REFERENCES Item(ItemID),
    CHECK (SellerFeedback >= 0 AND SellerFeedback < 10),
    CHECK (BuyerFeedback >= 0 AND BuyerFeedback < 10)
);
```

```
CREATE TABLE IF NOT EXISTS Category(
    CategoryID INT AUTO_INCREMENT NOT NULL,
    Name VARCHAR(255) NOT NULL,
    ParentCategoryID INT NULL,
    PRIMARY KEY (CategoryID)
);
```

```
ALTER TABLE Category ADD FOREIGN KEY (ParentCategoryID) REFERENCES
Category(CategoryID);
```

```
CREATE TABLE IF NOT EXISTS Belongs_to(
    CategoryID INT AUTO_INCREMENT NOT NULL,
    ItemID INT NOT NULL,
    PRIMARY KEY (CategoryID, ItemID),
    FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID),
```

```

    FOREIGN KEY (ItemID) REFERENCES Item(ItemID)
);

CREATE TABLE IF NOT EXISTS Bids_for(
    ItemID INT NOT NULL,
    BuyerID INT NOT NULL,
    BiddingTime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    Price FLOAT NOT NULL,
    PRIMARY KEY (ItemID, BuyerID, BiddingTime),
    FOREIGN KEY (ItemID) REFERENCES Item(ItemID),
    FOREIGN KEY (BuyerID) REFERENCES Buyer(BuyerID)
);

# Trigger to mark seller's items as inactive if not active
CREATE TRIGGER DeactivateSellerItems
AFTER UPDATE ON User
FOR EACH ROW
UPDATE Item, Seller, User SET Item.IsActive=False
WHERE Item.SellerID=Seller.SellerID
    AND Seller.SellerID = User.UserID
    AND User.IsActive=False;

```

## Expected DB Operations

- User:
  - Check if a user with a given userID password combination exists
  - Add a new user to the database
  - Deactivate a User
- Admin
  - Check if a user is an admin
- Seller
  - Check if a user is a seller
  - Get a Seller's routing info
  - If a deactivated user is a seller, deactivate their items that are still active
- Buyer
  - Check if a user is a buyer
  - Get a Buyer's shipping address
  - If a deactivated user is a buyer, remove their active bids
- Item
  - List all existing items
  - List all active items
  - Add an item
  - Get attributes of an item

- Update ending-date for an item
- Buys
  - Create a new Buys tuple
  - Get attributes of a given Buys tuple
- Category
  - Create a new category
  - List all Categories
- Belongs\_to
  - Create a new Belongs\_to tuple
  - List all items in a given category
- Bids\_for
  - Create a new Bids\_for tuple
  - Get the highest bid on an item

### Estimated Data Volumes

- User:
  - Hundreds
- Admin
  - Several
- Seller
  - Hundreds
- Buyer
  - Hundreds
- Item
  - Hundreds
- Buys
  - Hundreds
- Category
  - Dozens
- Belongs\_to
  - Hundreds
- Bids\_for
  - Hundreds

## Functional Dependencies and Database Normalization

### Functional Dependencies

Buys

    ItemID, BuyerID  $\rightarrow$  BuyerComment, BuyerFeedback, SellerComment, SellerFeedBack

Item

    ItemID  $\rightarrow$  Name, Description, Photo, StartingBid, BidIncrement, EndDate, StartDate, SellerID, IsActive

Category

    CategoryID  $\rightarrow$  Name, ParentCategoryID

Belongs\_to

    ItemID, CategoryID  $\rightarrow$  {}

Bids\_For

    ItemID, BuyerID, BiddingTime  $\rightarrow$  Price

Buyer

    BuyerID  $\rightarrow$  Address, City

Seller

    SellerID  $\rightarrow$  AccountNum, RoutingNum

User

    UserID  $\rightarrow$  Email, Password, Name, PhoneNum, HomeAddr, IsActive

Admin

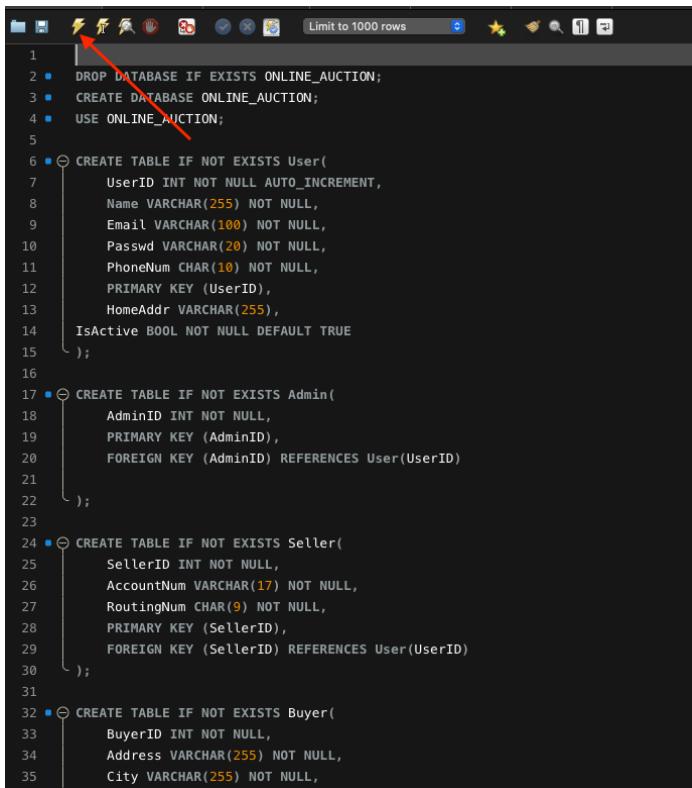
    AdminID  $\rightarrow$  {}

The database is in BCNF.

## The Database System

### How to Install and Invoke the Database System

1. Install MySQL
2. Run schema.sql to initialize the database, preferably using MySQLWorkbench



```

1
2 • DROP DATABASE IF EXISTS ONLINE_AUCTION;
3 • CREATE DATABASE ONLINE_AUCTION;
4 • USE ONLINE_AUCTION;
5
6 • CREATE TABLE IF NOT EXISTS User(
7     UserID INT NOT NULL AUTO_INCREMENT,
8     Name VARCHAR(255) NOT NULL,
9     Email VARCHAR(100) NOT NULL,
10    Passwd VARCHAR(20) NOT NULL,
11    PhoneNum CHAR(10) NOT NULL,
12    PRIMARY KEY (UserID),
13    HomeAddr VARCHAR(255),
14    IsActive BOOL NOT NULL DEFAULT TRUE
15 );
16
17 • CREATE TABLE IF NOT EXISTS Admin(
18     AdminID INT NOT NULL,
19     PRIMARY KEY (AdminID),
20     FOREIGN KEY (AdminID) REFERENCES User(UserID)
21 );
22
23
24 • CREATE TABLE IF NOT EXISTS Seller(
25     SellerID INT NOT NULL,
26     AccountNum VARCHAR(17) NOT NULL,
27     RoutingNum CHAR(9) NOT NULL,
28     PRIMARY KEY (SellerID),
29     FOREIGN KEY (SellerID) REFERENCES User(UserID)
30 );
31
32 • CREATE TABLE IF NOT EXISTS Buyer(
33     BuyerID INT NOT NULL,
34     Address VARCHAR(255) NOT NULL,
35     City VARCHAR(255) NOT NULL,

```

3. Install Python 3 and Pip
4. Run `pip install -r requirements.txt` to install dependencies
5. Create a configuration file called `config.json`, based off of `config.example.json` (included in the source code) and populate its values with your database's setup. The database should be `ONLINE_AUCTION`.
6. Run `python main.py`
7. Navigate to `http://localhost:5000`

## Suggestions on Database Tuning (optional)

### Suggestions

We can index on

- ItemID (Since it is searched frequently.)
- Name categories (on User, Item, and Category)
- User Address (for searching for nearby products OR location based searching)

because they may be searched for.

## Additional Queries and Views

### Queries and Views

# Views

```
CREATE VIEW `V_Buyers` AS
SELECT Buyer.BuyerID, User.Name, User.Email, Buyer.Address, Buyer.City,
AVG(Buys.BuyerFeedback)
FROM Buyer, User, Buys
WHERE Buyer.BuyerID = User.UserID and Buyer.BuyerID = Buys.BuyerID and User.IsActive
GROUP BY Buyer.BuyerID;
```

	BuyerID	Name	Email	Address	City	AVG(Buys.BuyerFeedback)
▶	5	Devan	devan@gmail.com	4321 James St	Richardson, TX	3.0000

```
CREATE VIEW `V_Sellers` AS
SELECT Seller.SellerID, User.Name, User.Email, Seller.AccountNum, Seller.RoutingNum,
AVG(Buys.SellerFeedback) AS Feedback
FROM Seller, User, Buys, Item
WHERE Item.ItemID = Buys.ItemID and Seller.SellerID = Item.SellerID and Seller.SellerID =
User.UserID
GROUP BY Seller.SellerID;
```

	SellerID	Name	Email	AccountNum	RoutingNum	Feedback
▶	2	Adam	adam@gmail.com	1111111111111111	111111111	7.0000

```
CREATE VIEW `V_Buyers Rating` AS
SELECT Buyer.BuyerID, User.Name, AVG(Buys.BuyerFeedback)
FROM Buyer, Buys, User
WHERE Buyer.BuyerID = Buys.BuyerID and Buyer.BuyerID = User.UserID and User.IsActive
GROUP BY Buyer.BuyerID;
```

	BuyerID	Name	AVG(Buys.BuyerFeedback)
▶	5	Devan	3.0000

```
CREATE VIEW `V_Highest Bids` AS
SELECT Item.ItemID, Item.Name, MAX(Bids_for.Price) AS HighestBid
FROM Item, Bids_for
WHERE Item.ItemID = Bids_for.ItemID and Item.IsActive
GROUP BY Item.ItemID;
```

	ItemID	Name	HighestBid
▶	1	Iphone X	30
	4	Notebook	7

-- Product that are on sale currently, sorted by close end dates

```
CREATE VIEW `V_Current Product close end dates` AS
SELECT Item.ItemID, Item.Name, Item.Description, Item.Photo, Item.StartingBid,
Item.BidIncrement,
    Item.StartDate, Item.EndDate, MAX(Bids_for.Price)
FROM Item, Bids_for
WHERE Item.ItemID = Bids_for.ItemID and Item.EndDate > CURRENT_DATE() and
Item.IsActive
Group by Item.ItemID
ORDER BY Item.EndDate;
```

	ItemID	Name	Description	Photo	StartingBid	BidIncrement	StartDate	EndDate	MAX(Bids_for.Price)
▶	1	Iphone X	Modern Phone	NULL	12	1	2021-10-20 00:00:00	2021-12-30 00:00:00	30
	4	Notebook	College ruled Notebook	NULL	2	1	2021-10-20 00:00:00	2021-12-30 00:00:00	7

# Select Queries

```
# Query for number of Bids for all items
SELECT I.Name, COUNT(*)
FROM Item AS I, Bids_for AS Bf
WHERE I.ItemID = Bf.ItemID
GROUP BY I.ItemID;
```

	Name	COUNT(*)
▶	Iphone X	9
	Notebook	6

# Query for listing all bids for Item id 0 sorted by bid price (highest to lowest)

```
SELECT Price, BiddingTime, BuyerID
FROM Bids_for
WHERE ItemID = 1
ORDER BY Price DESC;
```

	Price	BiddingTime	BuyerID
▶	30	2021-12-03 15:40:58	5
	25	2021-10-28 00:00:00	5
	22	2021-10-27 00:00:00	7
	21	2021-10-26 00:00:00	6
	19	2021-10-25 00:00:00	8
	18	2021-10-24 00:00:00	5
	17	2021-10-23 00:00:00	7
	13	2021-10-22 00:00:00	6
	12	2021-10-21 00:00:00	5

```
# Query for all bids from Buyer id 0 sorted by BiddingTime (newest to oldest)
```

```
SELECT BiddingTime, Price, ItemID
```

```
FROM Bids_for
```

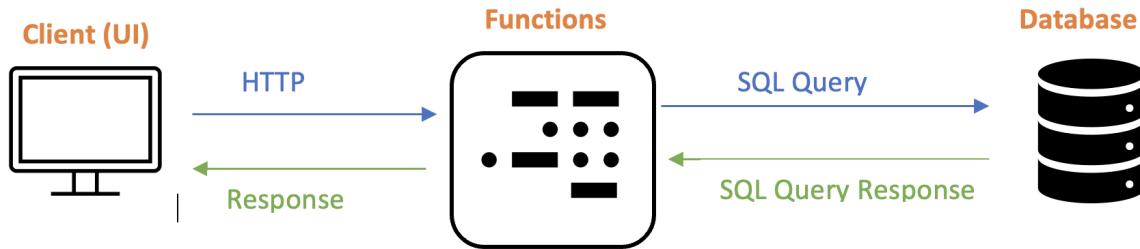
```
WHERE BuyerID = 5
```

```
ORDER BY BiddingTime ASC;
```

	BiddingTime	Price	ItemID
▶	2021-10-21 00:00:00	12	1
	2021-10-23 00:00:00	4	4
	2021-10-24 00:00:00	18	1
	2021-10-28 00:00:00	25	1
	2021-12-03 15:40:58	30	1

## User application interface

### Functions and Queries



List Of Functions (Web-Server) and their respective queries:

1. **login(<string>username, <string> password)**

**Route :** '/login'

**Methods Accepted :** 'GET', 'POST'

**Function :** If 'GET' returns the login page, else verifies the login credential and returns success if true, else login page.

**Query Executed :**

```
SELECT Passwd, Name FROM User where UserID = {request.args.get("id")};
```

Get Buyer ID:

```
SELECT * FROM Buyer where BuyerID = {request.args.get("id")};
```

Get Seller ID:

```
SELECT * FROM Seller where SellerID = {request.args.get("id")};
```

Get Admin ID:

```
SELECT * FROM Admin where AdminID = {request.args.get("id")};
```

2. **logout()**

**Route :** '/logout'

**Function :** It logs the user out from the session, if logged in.

**Query:**

```
session.clear()
```

3. **signup()**

**Route :** '/sign\_up'

**Methods Accepted :** 'GET', 'POST'

**Function :** If 'GET' returns the sign up page, else verifies the details and returns success if true, else sign up page.

**Query:**

```
INSERT INTO User (UserID, Name, Email, Passwd, PhoneNum,  
HomeAddr) VALUES ({new_user_vals});  
If buyer:  
    INSERT INTO Buyer (BuyerID, Address, City) VALUES  
    ({new_buyer_vals});  
  
elif user is seller:  
    INSERT INTO Seller (SellerID, AccountNum, RoutingNum) VALUES  
    ({new_seller_vals});  
  
elif user is admin:  
    INSERT INTO Admin (AdminID) VALUES ({new_id});
```

**4. productDetail(<int> itemId)**

**Route :** '/product/<int>:itemId'

**Methods Accepted :** 'GET', 'POST'

**Function :** If 'GET' returns the product details page, else verifies the details of bidding and stores in the database. If successful, it returns product details and confirmation message, else returns an error message.

**Query:**

I. Get item Details

```
SELECT * FROM ONLINE_AUCTION.Item WHERE ItemID = { itemId };
```

II. Get Seller Details

```
SELECT * FROM ONLINE_AUCTION.V_Sellers WHERE SellerID = { s_id };
```

III. Get all the bids from Bids\_for table.

```
SELECT * FROM ONLINE_AUCTION.Bids_for WHERE ItemID = { itemId };
```

IV. Get the Highest Bid from the View Table.

```
SELECT HighestBid FROM ONLINE_AUCTION.V_Highest_Bids WHERE ItemID = { itemId };
```

V. If the highest Bid is there, get all the details of the highest Bid.

```
SELECT * FROM V_Highest_Bid_Details WHERE ItemID = { itemId };
```

VI. Store Bids detail in Buys Table.

```
INSERT INTO Bids_for (ItemID, BuyerID, BiddingTime, Price) VALUES ({new_bid_vals});
```

**5. feedback(<int>SellerID, <int>BuyerID, <int>ItemID)**

**Route :** '/feedback/<int>SellerID/<int>BuyerID/<int>ItemID/'

**Methods Accepted :** 'POST'

**Function :** Validates the input fields and inserts a new feedback in Buys table if successful else returns an error.

**Query:**

```
INSERT INTO Buys (ItemID, BuyerID, SellerComment,  
SellerFeedback, BuyerComment, BuyerFeedback) VALUES  
({new_feedback_vals});
```

**6. view\_profile(<int>userID)**

**Route :** '/profile/<int>userID'

**Methods Accepted :** 'GET'

**Function :** Gets user details for supplied userID if it exists.

**Query:**

```
SELECT Name, Email, PhoneNum FROM User where UserID =  
{userId}';
```

**7. edit\_profile(<int>userID)**

**Route :** '/edit\_profile/<int>userID'

**Methods Accepted :** 'POST'

**Function :** Verifies Details that are entered and update the User and Buyer/Seller table if required.

**Query:**

```
Update User SET {update_user_vals} WHERE UserID = {userId};
```

**8. search\_Category(<string>query, <string>category = all)**

**Route :** '/search/<string>query, <string>category'

**Methods Accepted :** 'POST'

**Function :** It searches for input query of items in all the categories or just a specific category that is selected by the user. It then returns all the items that are there in the database matching these criteria.

**Query:**

I. To search in particular Categories.

```
SELECT Item.ItemID, Item.Name, Item.Description,  
Item.Photo, Item.StartingBid, Item.BidIncrement,  
Item.StartDate, Item.EndDate, Item.SellerID  
FROM Item, Belongs_to, Category  
WHERE Category.Name = "{categoryName}" and  
Category.CategoryID = Belongs_to.CategoryID and  
Belongs_to.ItemID = Item.ItemID and Item.EndDate >  
CURRENT_DATE() and Item.IsActive  
Group by Item.ItemID  
ORDER BY Item.EndDate;
```

II. Search in General

```

SELECT Item.ItemID, Item.Name, Item.Description,
Item.Photo, Item.StartingBid, Item.BidIncrement,
Item.StartDate, Item.EndDate, Item.SellerID
FROM Item
WHERE (lower(Item.name) LIKE lower("%{query_item}%") OR
lower(Item.description) LIKE lower("%{query_item}%")) and
Item.EndDate > CURRENT_DATE() and Item.IsActive
Group by Item.ItemID
ORDER BY Item.EndDate;

```

## 9. sellForm()

**Route :** '/sell/'

**Methods Accepted :** 'GET', 'POST'

**Function :** Returns sell form if 'GET' else, validates and saves the supplied input in the database.

**Query:**

```

INSERT INTO Item (ItemID, Name, Description, Photo,
StartingBid, BidIncrement, StartDate, EndDate, SellerID) VALUES
({new_item_vals});

```

## 10. home()

**Route :** '/'

**Methods Accepted :** 'GET'

**Function :** Returns home page.

## 11. adminHome()

**Route :** '/adminHome'

**Methods Accepted :** 'GET'

**Function :** Returns all users and items.

**Query:**

I. Get all the Users.

```
SELECT * FROM ONLINE_AUCTION.User;
```

II. Get all the Items

```
SELECT * FROM ONLINE_AUCTION.Item;
```

## 12. adminDeleteUserPage(<int>userID)

**Route :** '/adminDeleteUserPage/<int>userID/'

**Methods Accepted :** 'POST'

**Function :** Checks if the userID exists in User table and updates this user as inactive.

**Query:**

```
UPDATE ONLINE_AUCTION.User SET IsActive = 0 WHERE UserID={  
    user_id };
```

### 13. adminDeleteItemPage(<int>itemID)

**Route** : '/adminDeleteItemPage/<int>itemID/'

**Methods Accepted** : 'POST'

**Function** : Checks if the itemID exists in the Item table and updates this item as inactive.

**Query:**

```
UPDATE ONLINE_AUCTION.Item SET IsActive = 0 WHERE ItemID={  
    item_id };
```

## Screenshots of the User Interface

Home Page:

Online Auction Home Login Sign Up

**Holiday Offer**  
15% discount  
on all products

**Buyer**  
A Buyer account can bid for items in the auction.  
[Create a Buyer Account](#)

**Seller**  
A Seller account can put items up for auction.  
[Create a Seller Account](#)

**Admin**  
Admins help manage the website and users.  
[Create an Admin Account](#)

**ONLINE AUCTION**  
We are the leading company for online bidding of used and new items.

**OFFICE ADDRESS**  
800 W Campbell Road,  
Richardson, TX 75080

**CONTACT**  
✉ support@onlineauction.com  
📞 +01 478 888 9999

© 2021 Copyright: Online Auction

- This is the home page when someone accesses the site without logging in.
- Each account creation link leads to the account creation screen.

## Login Page:

Online Auction Home Login Sign Up

# Welcome!

Member ID:

Password:

#### ONLINE AUCTION

We are the leading company for online bidding of used and new items.

#### OFFICE ADDRESS

800 W Campbell Road,  
Richardson, TX 75080

#### CONTACT

 support@onlineauction.com

 + 01 478 888 9999

© 2021 Copyright: Online Auction

- Users can put in their id and password
- All Users log in from the same login page
- Application detects what kind of user they are after they login

Feedback Form to fill with comment and Rating for Seller/Buyer.



Sold By Adam

Start Date 2021-10-12 00:00:00

End Date 2021-11-01 00:00:00

Starting Bid \$ 8.0

Bid \$ 2.0

Increment

Please Enter Following Feedback

Feedback For Seller : 3

Enter Feedback

Rating

10

Submit Feedback

Conditions for Feedback to show:

1. Bidding EndDate should be less than current date.
2. Current user who is logged in should be either Seller OR buyer with highest bid who haven't put any comment or feedback till now.

## Account Creation:

Auction Home Login Sign Up

### Register your account below

Name:

Test User

Phone Number (10 digits, no dash):

0123456789

Home Address:

Example Address

Email:

email@example.com

Password (8-20 characters, must have letter and number):

.....

Account type:

Buyer

Seller

Shipping Address:

Example Address

City:

Example City

Sign Up

ONLINE AUCTION

OFFICE ADDRESS

CONTACT

- Users add their name, phone number, home address, email, password, and account types
- Buyer accounts add their shipping address and city
- Seller accounts add their bank account number and routing number
- When making an account, Users decide if their account will be buying items or selling items.

Account type:

Buyer

Seller

Account Number (5-17 digits):

.....

Routing Number (9 digits):

.....

Sign Up

## Registration Confirmation Page

Online Auction

All

Search an item

Search

Home

Sell

Profile

Logout Welcome Test User

# Registration completed!

Your member ID is 9. Please use this to log in from now on.

#### ONLINE AUCTION

We are the leading company for online bidding of used and new items.

#### OFFICE ADDRESS

800 W Campbell Road,  
Richardson, TX 75080

#### CONTACT

 support@onlineauction.com

 +01 478 888 9999

© 2021 Copyright: Online Auction

- Users will use their member ID to log in in the future. The ID is generated by the database to ensure that everyone has a unique ID.

An Item fill up form Page to sell an item.

Online Auction All Search an item Search Home Sell Profile Logout Welcome Test User

## Enter your item below to sell

Item Name:

Description

Bid Increment

Photo

Starting Bid

Start Date

 mm/dd/yyyy 

End Date

 mm/dd/yyyy 

### ONLINE AUCTION

We are the leading company for online bidding of used and new items.

### OFFICE ADDRESS

800 W Campbell Road,  
Richardson, TX 75080

### CONTACT

 support@onlineauction.com  
 +01 478 888 9999

© 2021 Copyright: Online Auction

A seller would provide Item above details when he/she is listing an item for auction.

- A seller is expected to fill all the details in the form.
- For image it would be a url path from where image is to be rendered.

## Viewing Profile

Online Auction All Search an item Search Home Profile Logout Welcome Devan

### Devan's Profile [Edit](#)

<b>Member ID</b>	5
<b>Account Type</b>	Buyer
<b>Email Address</b>	devan@gmail.com
<b>Phone Number</b>	1234567894
<b>Completed Transactions</b>	1
<b>Average Rating</b>	3.00

### Comments

[Iphone X](#) Bad stuff

### Bids

<a href="#">Iphone 12</a>	2021-11-30 22:22:43	\$20.0
<a href="#">Iphone 12</a>	2021-11-30 22:22:37	\$11.0
<a href="#">Iphone 12</a>	2021-11-30 21:58:44	\$11.0
<a href="#">Google Pixel</a>	2021-11-30 11:23:41	\$8.0
<a href="#">Google Pixel</a>	2021-11-30 11:23:36	\$8.0
<a href="#">Iphone X</a>	2021-11-30 00:27:30	\$30.0
<a href="#">Iphone X</a>	2021-10-28 00:00:00	\$25.0
<a href="#">Iphone X</a>	2021-10-24 00:00:00	\$18.0
<a href="#">Iphone X</a>	2021-10-21 00:00:00	\$12.0

**ONLINE AUCTION**  
We are the leading company for online bidding of used and new items.

**OFFICE ADDRESS**  
800 W Campbell Road,  
Richardson, TX 75080

**CONTACT**  
 support@onlineauction.com  
 +01 478 888 9999

© 2021 Copyright: Online Auction

- Buyer profile shown above. Includes rating, comments, and bid history.
- Edit button only shows up if a user is viewing their own profile or is an admin
- For sellers, Bids is replaced by Items Listed

## Editing profile

Online Auction All Search an item Search Home Sell Profile Logout Welcome Test User

### Editing Test User's Profile

Phone Number (10 digits, no dash):

0123456789

Home Address:

Example Address

Email:

email@example.com

Account Number (5-17 digits):

11111111111111112

Routing Number (9 digits):

123456789

Confirm changes with your password:

#### ONLINE AUCTION

We are the leading company for online bidding of used and new items.

#### OFFICE ADDRESS

800 W Campbell Road,  
Richardson, TX 75080

#### CONTACT

 support@onlineauction.com

 +01 478 888 9999

© 2021 Copyright: Online Auction

- This page allows users to edit their profile (this is a buyer profile).
- Users cannot change their account type.
- Changes cannot be finalized without the user entering their password

Table that shows all the categories and all the products with details of the item and link to view detail of the respective product.

Online Auction

All

Search an item

Search

Home

Sell

Profile

Logout

Welcome Test User

## Item List of All Category

ID	Name	Description	Starting Bid	Bid Increment	Start Date	End Date
1	Iphone X	Modern Phone	\$12.00	\$1.00	2021-10-20 00:00:00	2021-10-30 00:00:00
4	Calculator	New Modern Casio Calculator	\$8.00	\$2.00	2021-10-12 00:00:00	2021-11-01 00:00:00
3	Iphone 12	Modern IOS Phone	\$11.00	\$3.00	2021-11-20 00:00:00	2021-12-20 00:00:00
2	Google Pixel	Modern Android Phone	\$8.00	\$2.00	2021-12-21 00:00:00	2021-12-27 00:00:00

### ONLINE AUCTION

We are the leading company for online bidding of used and new items.

### OFFICE ADDRESS

800 W Campbell Road,  
Richardson, TX 75080

### CONTACT

 support@onlineauction.com  
 +01 478 888 9999

© 2021 Copyright: Online Auction

Category based searching for the items.

Online Auction  All  Electronics  Phone  Good Stuff

Search an item  Home Profile Logout Welcome Devan

## Items List of Electronics Category

ID	Name	Description	Starting Bid	Bid Increment	Start Date	End Date	Seller ID	Bidding Link
3	Iphone 12	Modern IOS Phone	\$11.00	\$3.00	2021-11-20 00:00:00	2021-12-20 00:00:00	4	<a href="#">Product Detail</a>
2	Google Pixel	Modern Android Phone	\$8.00	\$2.00	2021-12-21 00:00:00	2021-12-27 00:00:00	3	<a href="#">Product Detail</a>

### ONLINE AUCTION

We are the leading company for online bidding of used and new items.

### OFFICE ADDRESS

800 W Campbell Road,  
Richardson, TX 75080

### CONTACT

 support@onlineauction.com  
 +01 478 888 9999

© 2021 Copyright: Online Auction

## Product Description

Online Auction All Search an item Search Home Profile Logout Welcome Devan



### Iphone X

<b>Item ID</b>	1
<b>Description</b>	Modern Phone
<b>Sold By</b>	<a href="#">Adam</a>
<b>Start Date</b>	2021-10-20 00:00:00
<b>End Date</b>	2021-10-30 00:00:00
<b>Starting Bid</b>	\$12.00
<b>Bid</b>	\$1.00
<b>Increment</b>	

ONLINE AUCTION  
We are the leading company for online bidding of used and new items.

OFFICE ADDRESS  
800 W Campbell Road,  
Richardson, TX 75080

CONTACT  
 support@onlineauction.com  
 +01 478 888 9999

© 2021 Copyright: Online Auction

- Product description of a sold product and its rating in the end.

## Item Page for Active Item:

Online Auction All Search an item Search Home Profile Logout Welcome Devan



### Iphone 12

**Item ID** 3

**Description** Modern IOS Phone

**Sold By** [Yennifer](#)

**Start Date** 2021-11-20 00:00:00

**End Date** 2021-12-20 00:00:00

**Starting Bid** \$11.00

**Bid Increment** \$3.00

[Highest Bid](#) [All Bids](#)

**Highest Bid** : \$20.00

Bid Value : \$

[Bid Now](#)

#### ONLINE AUCTION

We are the leading company for online bidding of used and new items.

#### OFFICE ADDRESS

800 W Campbell Road,  
Richardson, TX 75080

#### CONTACT

[support@onlineauction.com](mailto:support@onlineauction.com)

+01 478 888 9999

© 2021 Copyright: Online Auction

- Picture of an iphone that was uploaded by the seller. The seller sets the starting bid price and bid increments.
- Buyers can see all the previous bids and place their own bids

## Conclusion and future work

### Conclusion

We were able to create a complex application with a relational database in the back. This was a great experience for learning how to use databases in real-life applications, as well as how to work in a team.

### Future Work

Future work might include making the application more secure and user-proof. For example, salting and hashing passwords when storing them, and also performing more user checks on each website rather than trusting them based on the fact that they found the URL and they have a valid session cookie.

## References

Fundamentals of Database Systems - Covers database topics.

Fundamentals of Database Systems, 7th Edition, R. Elmasri and S. B. Navathe,  
AddisonWesley.ISBN-13: 978-0133970777

[https://amirsmvt.github.io/Database/Static\\_files/Fundamental\\_of\\_Database\\_Systems.pdf](https://amirsmvt.github.io/Database/Static_files/Fundamental_of_Database_Systems.pdf)

MySQL documentation and downloads - To download the MySQL Installer, Workbench and Server

<https://dev.mysql.com/doc/>

SQLAlchemy - Performs database access for python

<https://docs.sqlalchemy.org/en/14/>

Flask documentation - Covers the Installation and connection to databases

<https://flask.palletsprojects.com/en/2.0.x/>

Jinja2 documentation - Covers how to create templates for python

<https://jinja2docs.readthedocs.io/en/stable/>

HTML/CSS bootstrap - HTML and CSS

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

# 12

## Appendix

/doc and /project are in the zip file.