Our Final project report for ISC 329
The Tool Shop.

By shaun godfrey and Vitaliy Shydlonok

https://isc329toolshop.000webhostapp.com/index.php

The project:

- Our project is a web-based Tool shop that allows its users to create an account and to search for or buy tools from our online store. Our Users will be able to search through our database for specific items based on their category, price, group, and terms in the name or description of the item they are looking for. Our Users will have a shopping cart with the items that they have chosen to purchase and through their account will also be able to view any order details. Lastly, our database will keep information from our user's account and details on the price and quantity of any item in our store.

What the project accomplished:

- The database incorporates the customer's name, email, card info, password, billing address, purchase history, and a current list of items in their shopping cart. The information saved for each item in our database is the name, description, price, quantity, review, and rating. The users will also be able to view the details of an item, add and remove items to their shopping cart, purchase items, and rate any items that they have purchased. Lastly, all orders will have details of the users, their purchase totals, the date of the order, and a record of where it was delivered too.

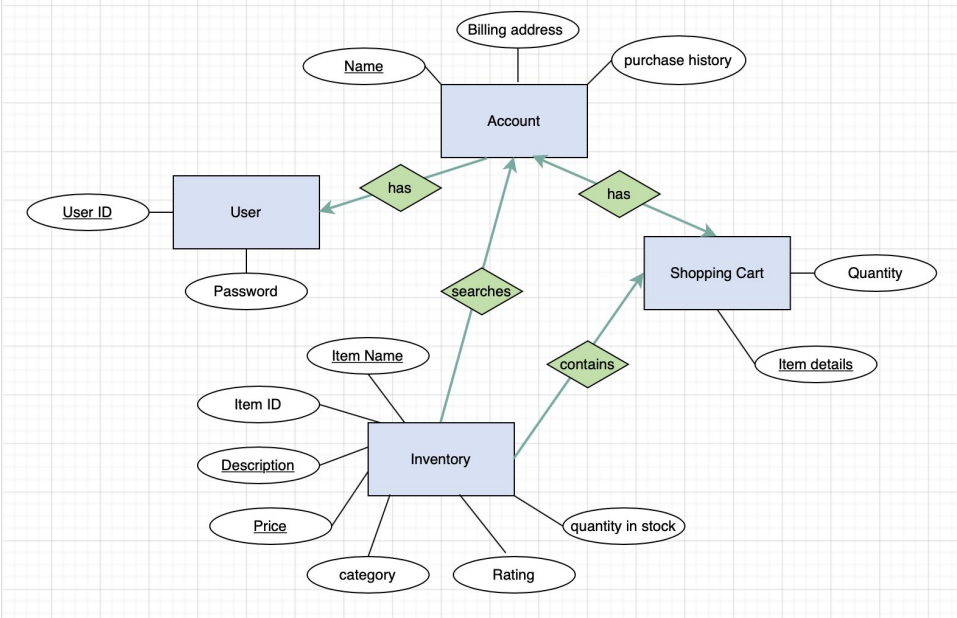
What the project Design:

- When it came to the design of our database we started with the ER diagram, which helped to lay out the basic entity sets and their attributes that we needed. This helped also to define their relationships with each other and which

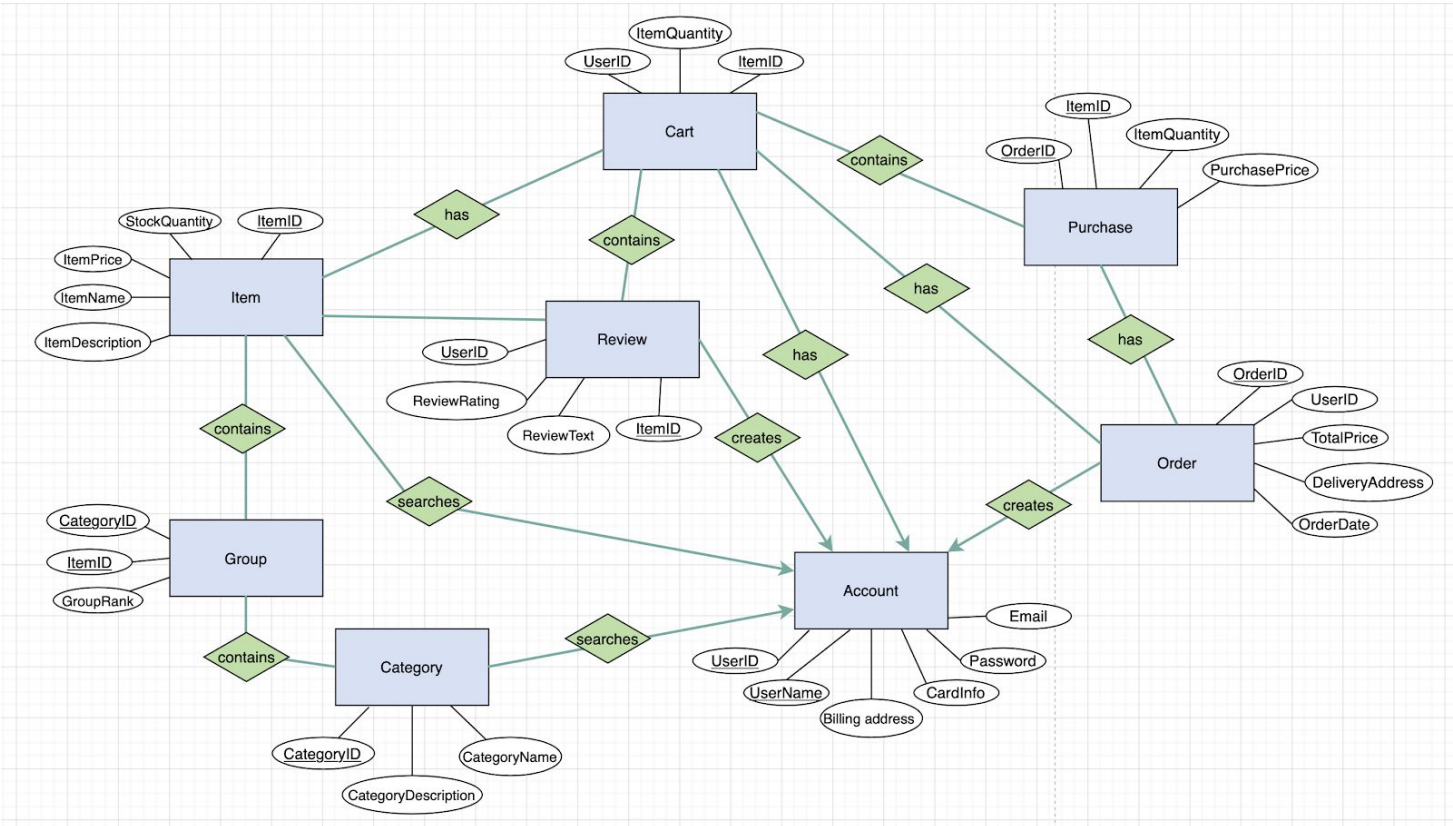
attributes within this design would be our primary keys. We then designed the relational schema and realized that we needed more attributes for each of our entity sets, and some more entities as well. We thought we had everything figured out until we did the normalization process. During this process, we realized that we only had UserID as the main identifier for a User and that two accounts could have the same username. So, we fixed this by adding a UserName as a primary key to the ACCOUNT. We also added the attributes Email and CardInfo to the ACCOUNT to help keep track of each user. We then removed the attributes CartID from the CART as we realized we could just reference ItemID and the UserID instead to give us their shopping cart. We then removed the attribute PurchaseID from PURCHASE as we figured out that we just use OrderID and ItemID instead to get their purchase information. We then removed the attributes CategoryID and ItemRating from ITEM as we felt this was no longer needed in that entity set and that CategoryID could have multiple values. We also removed ReviewID from REVIEW as we could just use UserID and ItemID to retrieve a user's reviews. Finally, we added in the entity GROUP with attributes CategoryID, ItemID, and GroupRank, as we felt this relation was needed as each ITEM can have multiple values for CategoryID. This design process we took helped us to put the database together and after all these steps were taken our database came together with no hiccups or surprises.

Images of our design:

- Initial ER diagram



- Final ER diagram after the normalization process



- Final relational schema

Bold: Primary Key
Underline: Foreign Key

UserID	**UserName**	Email	Password	CardInfo	BillingAddress

ACCOUNT

ItemID	**UserID**	ItemQuantity

CART

OrderID	**ItemID**	ItemQuantity	PurchasePrice

PURCHASE

OrderID	UserID	TotalPrice	DeliveryAddress	OrderDate

ORDER

ItemID	ItemName	ItemDescription	ItemPrice	StockQuantity

ITEM

CategoryID	CategoryName	CategoryDescription

CATEGORY

CategoryID	**ItemID**	GroupRank

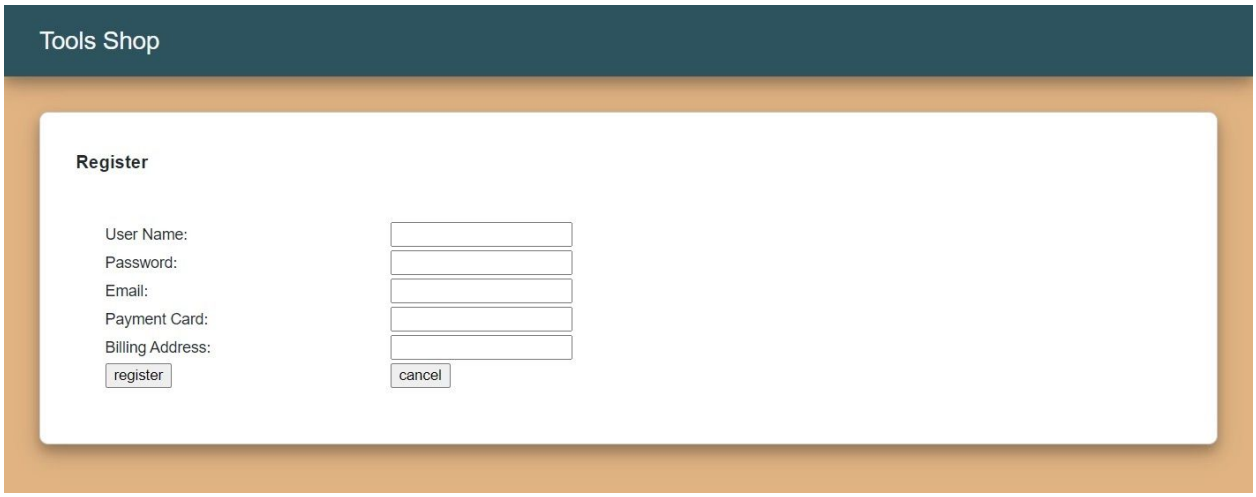
GROUP

UserID	**ItemID**	ReviewRating	ReviewText

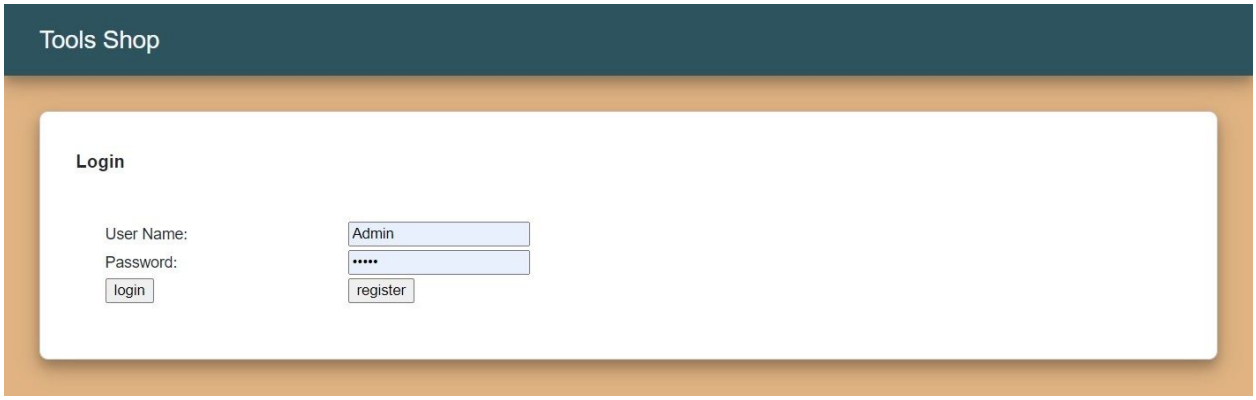
REVIEW

Screenshots of our Database:

Registration:



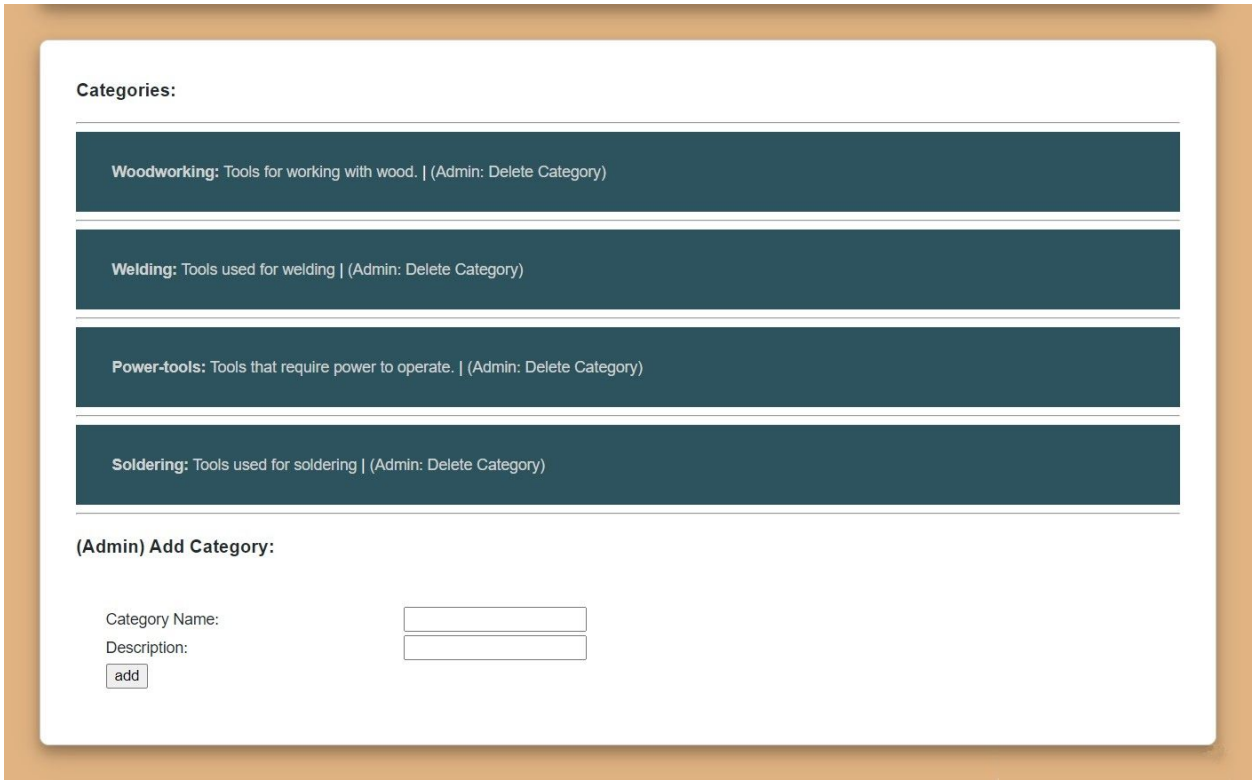
Login:



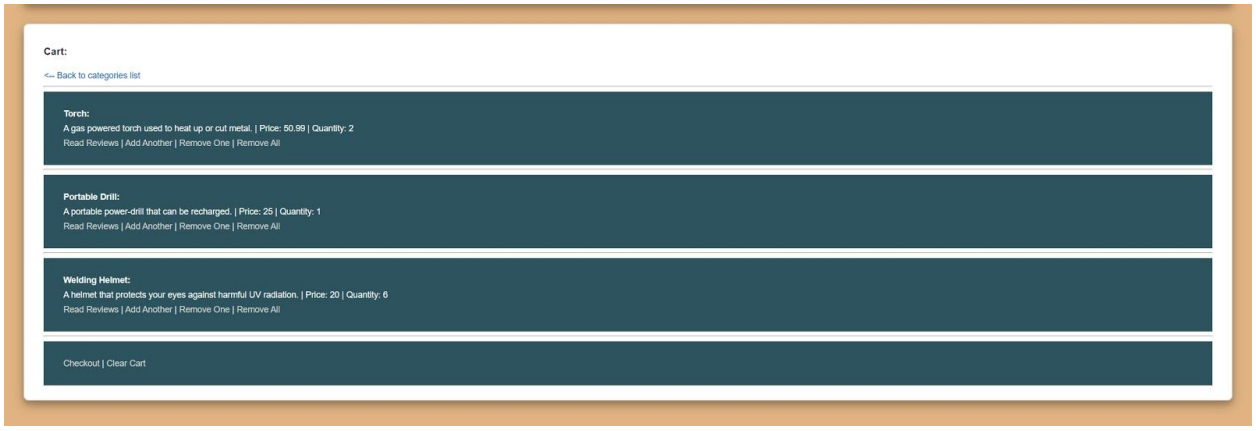
Account:

Items List:

Categories:



Shopping Cart:



Orders:

Adding a Review

Item Reviews:

Creating the database in SQL:

```
create table account
(
	UserID int auto_increment,
	UserName char(255) not null,
	Email char(255) not null,
	Password char(255) not null,
	CardInfo text null,
	BillingAddress text null,
	primary key (UserID, UserName)
);

create table item
(
	ItemID int auto_increment
		primary key,
	ItemName char(255) default 'Item Name' not null,
	ItemDescription text not null,
	ItemPrice double(16,2) not null,
	StockQuantity int default 0 not null
);

create table cart
(
	ItemID int not null,
	UserID int not null,
	ItemQuantity int default 1 not null,
	primary key (ItemID, UserID),
	foreign key (UserID) references account (UserID),
	foreign key (ItemID) references item (ItemID)
);

create table category
(
	CategoryID int auto_increment
		primary key,
	CategoryName char(255) default 'Default Category' not null,
	CategoryDescription text not null
);

create table group_table
(
	CategoryID int not null,
	ItemID int not null,
```

```
      GroupRank int not null,
      primary key (CategoryID, ItemID),
      foreign key (CategoryID) references category (CategoryID),
      foreign key (ItemID) references item (ItemID)
);

create table order_table
(
      OrderID int auto_increment
            primary key,
      UserID int not null,
      TotalPrice double(16,2) not null,
      DeliveryAddress text not null,
      OrderDate date not null,
      foreign key (UserID) references account (UserID)
);

create table purchase
(
      OrderID int not null,
      ItemID int not null,
      ItemQuantity int default 1 not null,
      PurchasePrice double(16,2) not null,
      primary key (OrderID, ItemID),
      foreign key (ItemID) references item (ItemID),
      foreign key (OrderID) references order_table (OrderID)
);

create table review
(
      UserID int not null,
      ItemID int not null,
      ReviewRating int default 5 not null,
      ReviewText text not null,
      primary key (UserID, ItemID),
      foreign key (UserID) references account (UserID),
      foreign key (ItemID) references item (ItemID)
);
```

SQL queries implemented:

```
DELETE FROM cart WHERE ItemID = ? AND UserID = (SELECT UserID FROM account
WHERE UserName = ?)

DELETE FROM cart WHERE UserID = (SELECT UserID FROM account WHERE UserName =
?)

DELETE FROM category WHERE CategoryID = ?

DELETE FROM item WHERE ItemID = ?

DELETE FROM review WHERE ItemID = ? AND UserID = ?

INSERT INTO account (UserName,Password,Email,CardInfo,BillingAddress) VALUES
(?,?,?,?,?)

INSERT INTO cart (ItemID,UserID,ItemQuantity) VALUES (?,(SELECT UserID FROM
account WHERE UserName = ?),1)

INSERT INTO category (CategoryName,CategoryDescription) VALUES (?,?)

INSERT INTO group_table (CategoryID,ItemID,GroupRank) VALUES (?,( SELECT
LAST_INSERT_ID() ),1)

INSERT INTO item (ItemName,ItemDescription,ItemPrice,StockQuantity) VALUES
(?,?,?,?)

INSERT INTO order_table (UserID,TotalPrice,DeliveryAddress,OrderDate) VALUES
(?,?,?,?)

INSERT INTO purchase (OrderID,ItemID,ItemQuantity,PurchasePrice) VALUES
(?,?,?,?)

INSERT INTO review (UserID,ItemID,ReviewRating,ReviewText) VALUES ((SELECT
UserID FROM account WHERE UserName = ?),?,?,?)

SELECT * FROM account WHERE Username = ?

SELECT * FROM account WHERE Username = ? and Password = ?

SELECT * FROM cart WHERE ItemID = ? AND UserID = (SELECT UserID FROM account
WHERE UserName = ?)
```

```
SELECT * FROM cart WHERE UserID = (SELECT UserID FROM account WHERE UserName =
?)
SELECT * FROM category

SELECT * FROM item WHERE ItemID IN (SELECT ItemID FROM group_table WHERE
CategoryID = ?)

SELECT * FROM item WHERE ItemID = ?

SELECT * FROM order_table WHERE UserID = (SELECT UserID FROM account WHERE
UserName = ?)

SELECT * FROM purchase WHERE OrderID = ?

SELECT * FROM review WHERE ItemID = ? AND UserID = (SELECT UserID FROM account
WHERE UserName = ?)

SELECT account.UserID as UserID, UserName, ReviewRating, ReviewText FROM
review,account WHERE ItemID = ? AND review.UserID = account.UserID

SELECT CategoryName FROM category WHERE CategoryID = ?

SELECT ItemName FROM item WHERE ItemID = ?
SELECT ItemPrice FROM item WHERE ItemID = ?

SELECT SUM(ReviewRating) as ReviewTotal FROM review WHERE ItemID = ? AND
UserID = (SELECT UserID FROM account WHERE UserName = ?)

SELECT UserID,SUM(ItemTotal) as TotalPrice FROM (SELECT
UserID,ItemPrice*ItemQuantity as ItemTotal FROM (SELECT account.UserID as
UserID,ItemPrice,ItemQuantity FROM item,cart,account WHERE item.ItemID =
cart.ItemID and cart.UserID = account.UserID and account.UserName = ?) AS
PriceQuantity) AS TotalTable

UPDATE cart SET ItemQuantity = ItemQuantity + 1 WHERE ItemID = ? AND UserID =
(SELECT UserID FROM account WHERE UserName = ?)

UPDATE cart SET ItemQuantity = ItemQuantity - 1 WHERE ItemID = ? AND UserID =
(SELECT UserID FROM account WHERE UserName = ?)

UPDATE item SET StockQuantity = StockQuantity - ? WHERE ItemID = ?

UPDATE review SET ReviewRating = ?, ReviewText = ? WHERE ItemID = ? AND UserID
= (SELECT UserID FROM account WHERE UserName = ?)
```

Final division of labor:

- Our project was divided up pretty fairly as there were only two of us designing and implementing this database. Shaun did the ER diagrams and Vitaliy did the relational schemas. We both worked on the final report / presentation and built the database together.

Any other strengths:

- We feel our project was notably uncomplicated and concise, and this was perfect for a two member group. We did not have to change much from our initial design and only when we encountered the normalization process did we realize a few things we either needed or did not need. Other than that, our project was adequately composed for the bottom-up. Throughout all the process we had a clear understanding of what we needed and how we needed to get it all to work.