Full Stack Website Projet: GamePortal Jingyu Wang

Index:

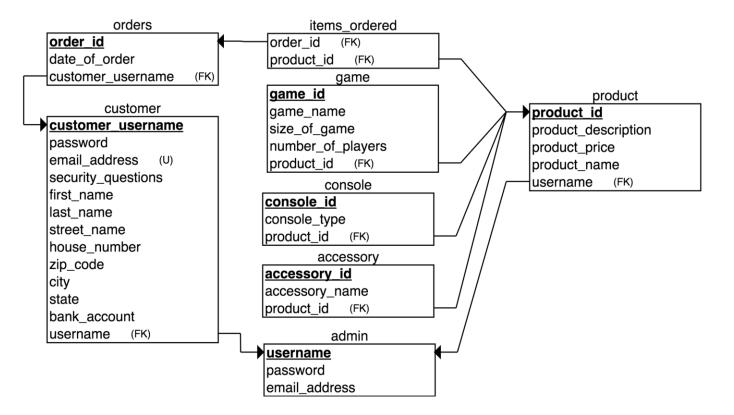
- Description
- Relational Model
- ERD
- Index Selection
- SQL for creating and populating
- How we anticipated and how this drove our design
- Screenshots of the interface (all of them)
- Implementing Constraints
- Ensuring Integrity
- Error Checking
- Source Files

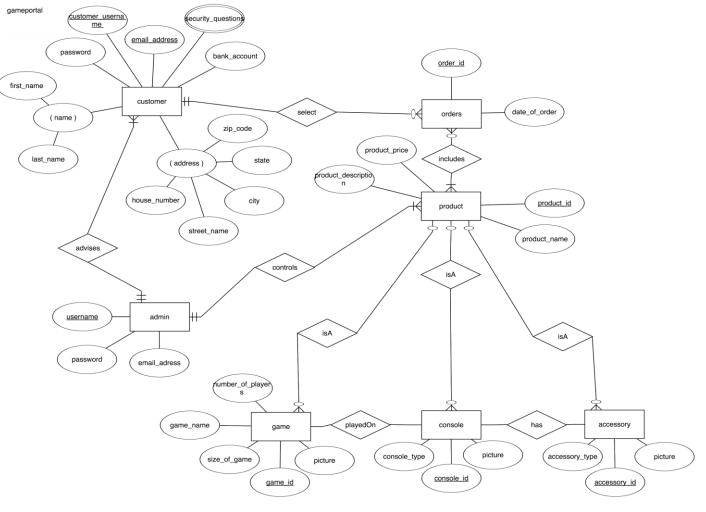
The database we have come to build, is a game store database. It is made for the gamer in us. With a wide variety of games, for a various amounts of consoles those games apply to, such as XBOX, PC or PlayStation, and a copious amount of genres that these games fall into, whether it be a simulation game, sport game, or even a strategic game; these games may apply to all the gamers, everywhere. This database is designed for a customer to sign up with his or her's unique username, unique email address followed by a password and other attributes. Admin is one who oversees all the actions being taken. Example being, an administrator may be able to add or remove a game, and also add or remove a customer. Once a customer purchases a game, they will automatically be charged the price of the product they desired to have – game, accessory, or console. The user is required to sign up for an account. The

information the administrators need is the customer's name, both first and last, an address, which provides a street name, number, city, state and zip code, a unique email address is also needed as well as a unique account username. Once user is signed up, he/she may log in and will not have to sign up any more due to the customer's information is stored in the game store database. This database is designed by 8 tables: customer, orders, items_ordered, product, admin, game, console, and accessory.

Below, you will see both the Relational Model and ERD, respectively. The Relational Model is as followed:

gameportal





The ERD is as followed:

When it came to the Index Selection, we know that the index selection is a method in which several useful traits are selected simultaneously. We used index selection in the search part, where we created an index selection by using the name of the product to make it more convenient and efficient for the user for his or hers needs. This is an example of index selection queries that we have used in the project:

```
$index = "CREATE INDEX g_name ON game (game_name)";
$index2 = "CREATE INDEX c_type ON console (console_type)";
$index3 = "CREATE INDEX a_name ON accessory (accessory_name)";
```

For making the SQL for creating and populating the tables, below you will see a an example of our code pasted so you will be able to see how the code was designed as well as the links to the file with all of the code:

-- Table structure for table `accessory`--

CREATE TABLE 'accessory' (

'accessory_name' varchar(100) NOT NULL,

'image_name' varchar(100) NOT NULL,

- 'image' longblob NOT NULL,
- 'product_id' varchar(100) NOT NULL,
- `accessory_id` int(11) NOT NULL
-) ENGINE=InnoDB DEFAULT CHARSET=latin1;

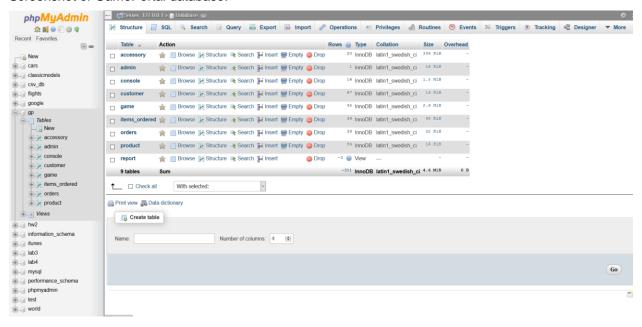
Here are the links to our database to view all of our codes:

https://github.com/hamalis/DB Project

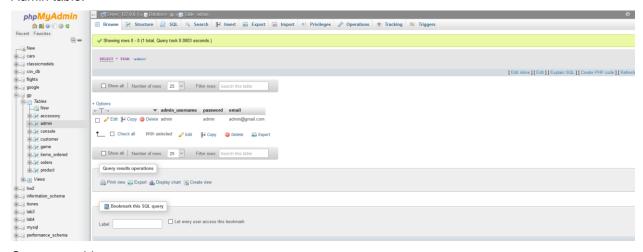
https://github.com/sjuDB/hello-git-hamalis

Below you will find screen shots of our database for your benefit:

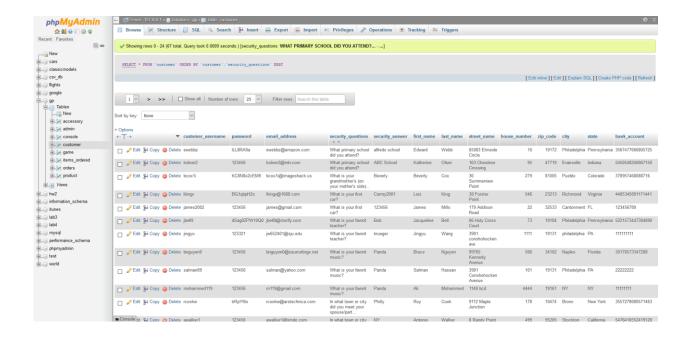
Screenshot of GamePortal database:



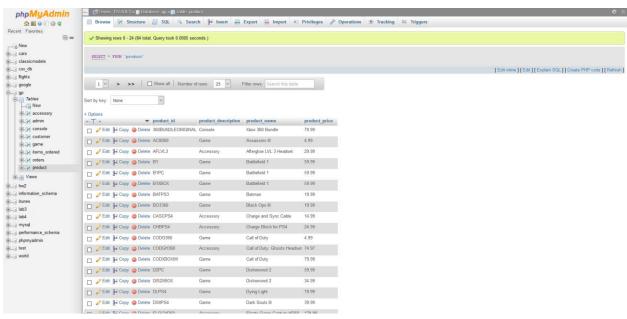
Admin table:



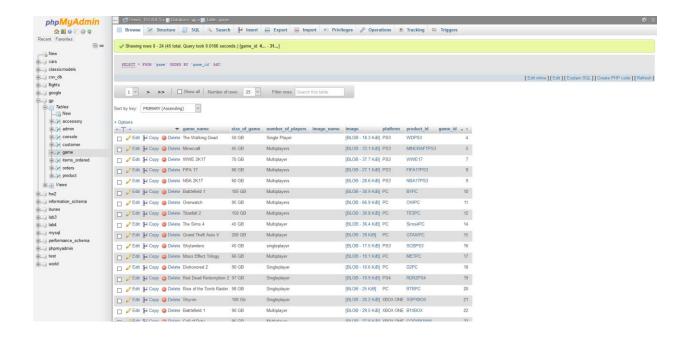
Customer table:



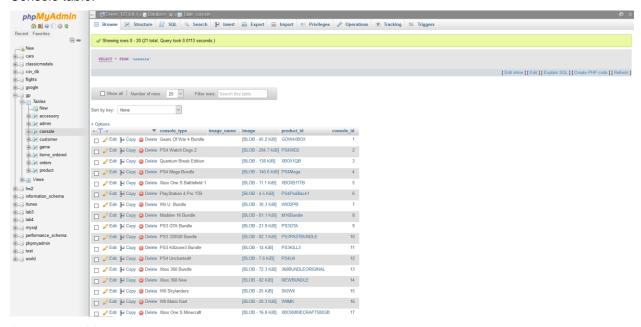
Product table:



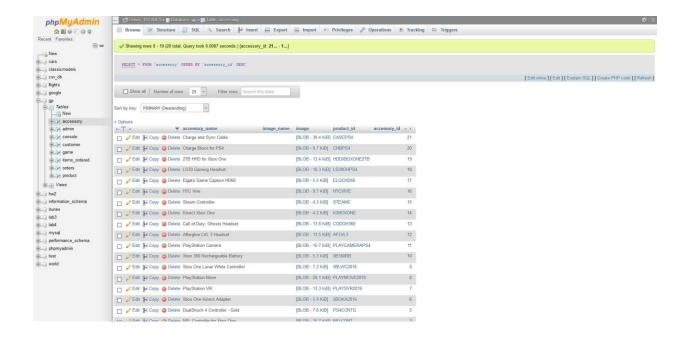
Game table:



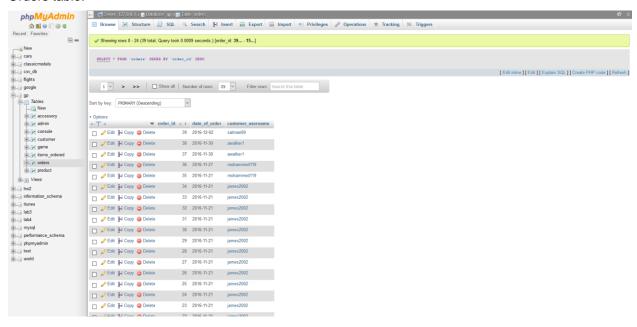
Console table:



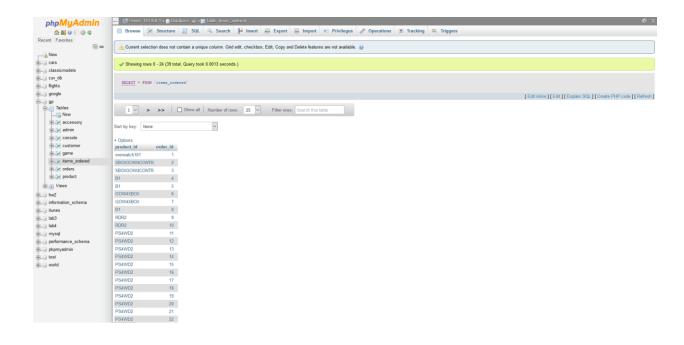
Accessory table:



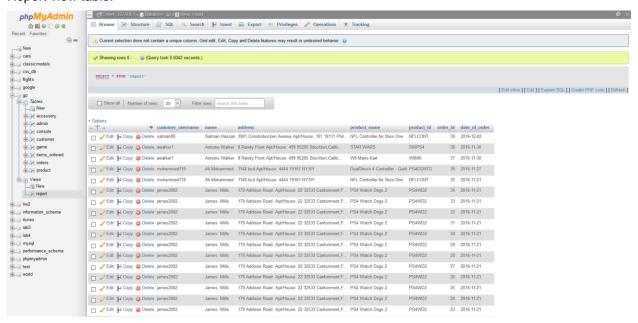
Orders table:



Items ordered table:

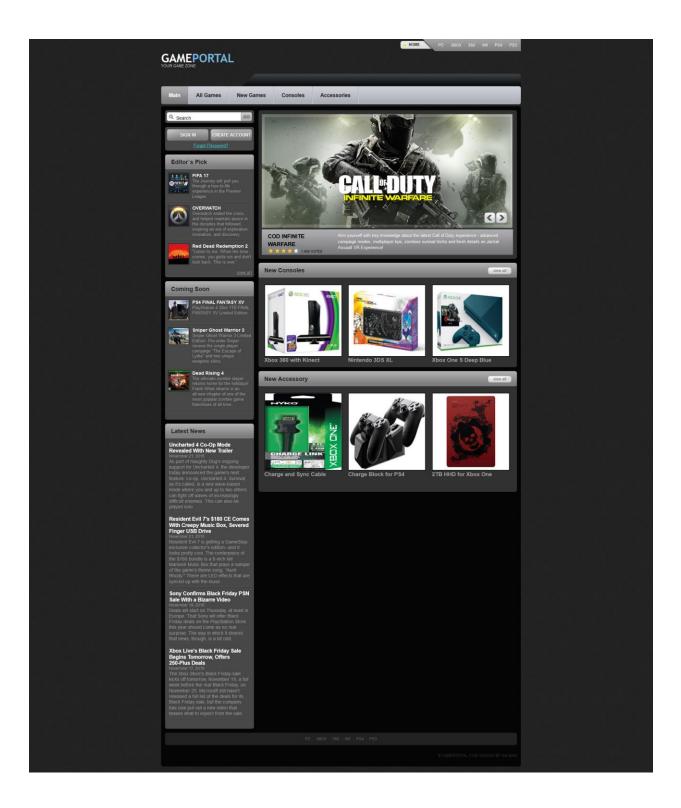


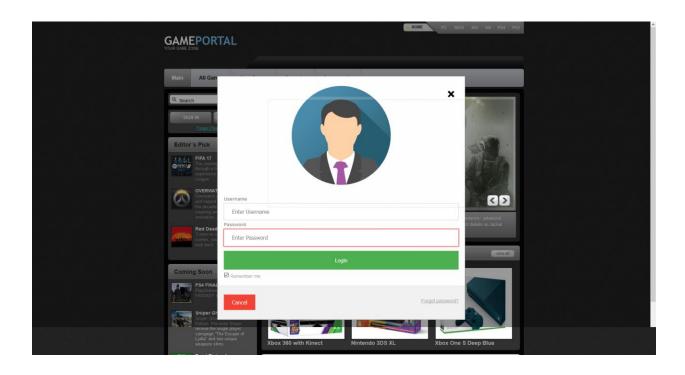
Report view table:



In the beginning, we anticipated for the application to be used as if it were an online game store, where a user can buy a game whenever he/she wants to. This drove us to expand our initial idea from just games to consoles, and also, accessories. We have designed the most convenient and quickest way to be a gamer. We deliver a product once the user purchases the item. We are super convenient, whereas, a user does not have to go anywhere to get whatever they need to in order to improve their gaming experience, but only has to go online to satisfy his or her needs for their entertainment purposes. We have initially wanted an online game store to be used to just play PC games and from there, we wanted to

increase our demographic by selling other console games, such as XBOX, PlayStation and Nintendo. This thought process expanded to us developing a database for gamers that we can sell not only games but also, other products that will help a user play a game; such as, a power wire for the console, a controller, a headset and other accessories.
Below you will see the interface of our database. HomePage:





and the second s	HOME PC 20,000 240 WII PSA PSS	
GAMEPORTA	AL.	
YOUR GAME ZONE		
	COMMUNITY FORIM WORD GREATS FEATURES COMMEANDS SPORTS TECH	
	s New Games Consoles Accessories	
New Member S	Sign Up	
First Name	First Name	
Last Name	Last Name	
Username	Username	
Email	Enail	
Password		
Address		
Street Name	Street	
House/Aprtment Number	House/Apt Number	
Zip Code	ZIp Code	
City	City	
State	select a state	
Security Question	select a question v	
Answer	Answer	
Bank Acount	Account Number	
	Cancel Submit	
	COMMANY FOLIAN MODO DERSE FIRITINES SOMMANOS SPORS TECH PC 380X 580 WH PM PB	
11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		

Signup form:

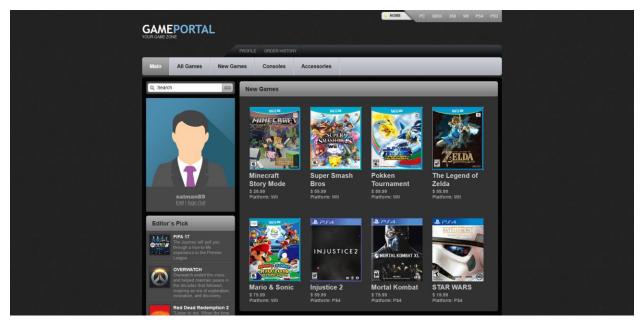
User Logged in page:





Page that shows all games for all types:

Page that shows new games only:



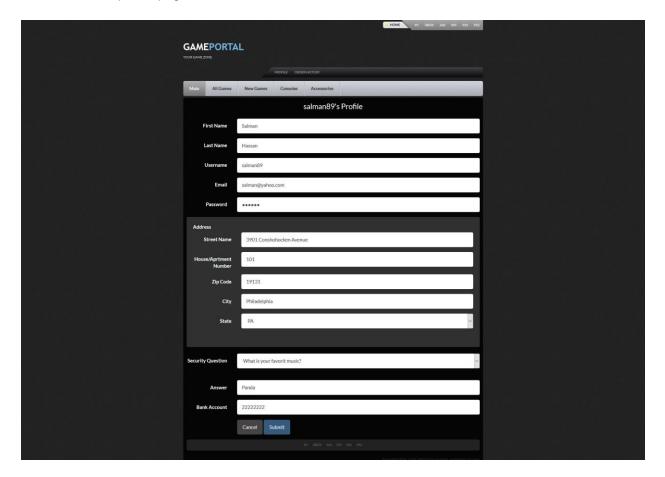
Page that shows all consoles:



Page that shows all accessories:



Edit customer profile page:



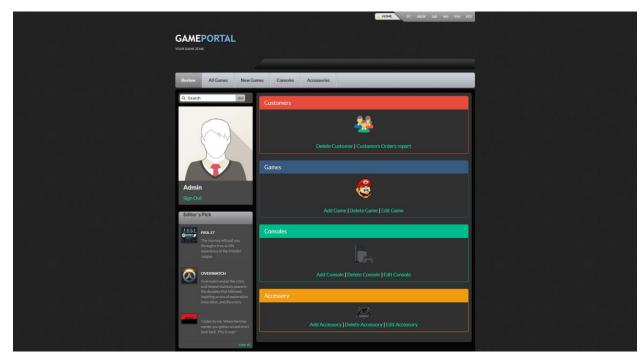
Customer orders history:



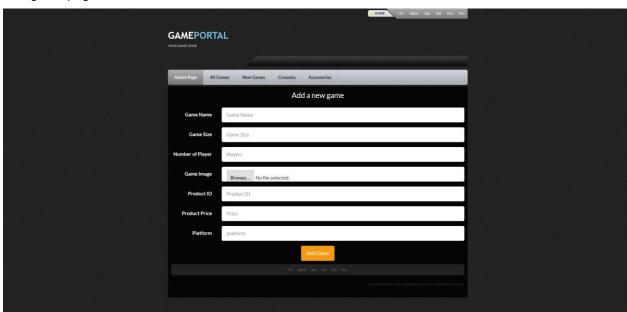
Customer confirm order page:



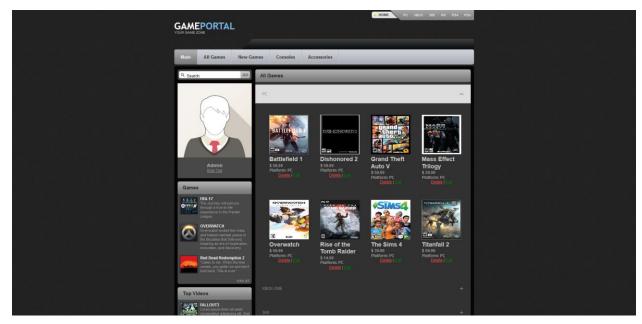
Admin homepage:



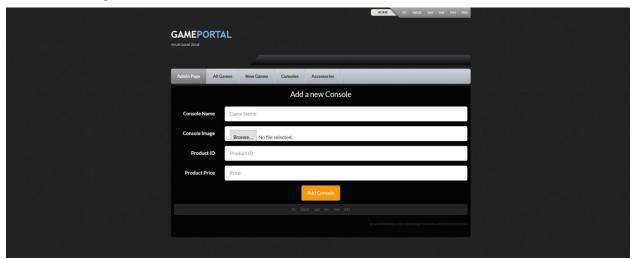
Add game page:



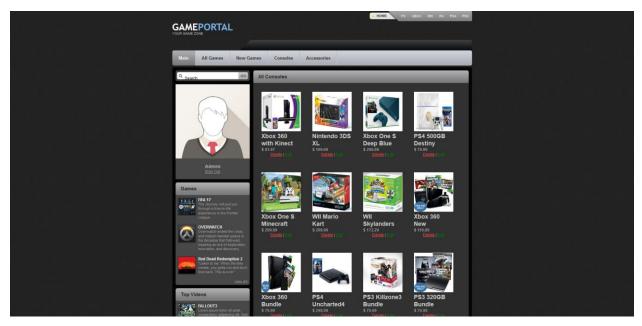
Edit/Delete games page:



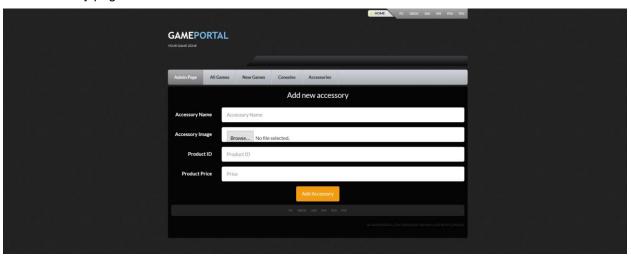
Add Console Page:



Edit/Delete consoles page:

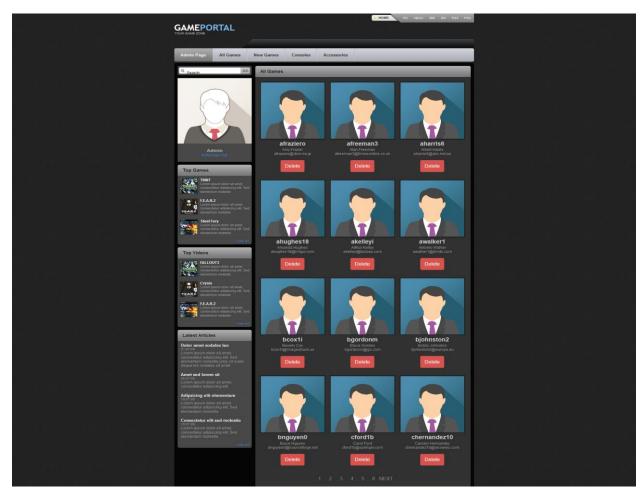


Add accessory page:



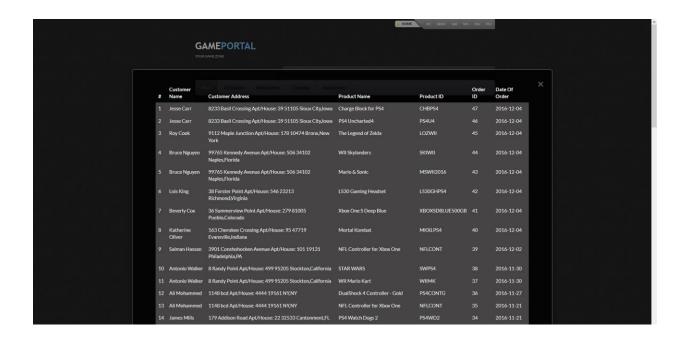
Edit/Delete accessories page:





Delete customers accounts page:

Customers order report for admin user:



Constraint is used to define rules to allow or restrict what values can be stored in columns. The purpose of inducing constraints is to enforce the integrity of a database. For constraints, we use different constraints to create our tables and foreign keys too. This is an example of how we created the tables:

```
CREATE TABLE `accessory` (
  `accessory_name` varchar(100) NOT NULL,
  `image_name` varchar(100) NOT NULL,
  `image` longblob NOT NULL,
  `product_id` varchar(100) NOT NULL,
  `accessory_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Other constraints we have implemented for foreign keys as the following:

-- Constraints for table `accessory`

ALTER TABLE `accessory`

ADD CONSTRAINT `accessory_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `product` (`product_id`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Constraints for table `console`

ALTER TABLE `console`

ADD CONSTRAINT `console_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `product` (`product_id`) ON DELETE CASCADE ON UPDATE CASCADE;

--

```
-- Constraints for table `game`

-- ALTER TABLE `game`

ADD CONSTRAINT `game_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `product` (`product_id`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Constraints for table `items_ordered`

-- ALTER TABLE `items_ordered`

ADD CONSTRAINT `items_ordered_ibfk_3` FOREIGN KEY (`product_id`) REFERENCES `product` (`product_id`),

ADD CONSTRAINT `items_ordered_ibfk_4` FOREIGN KEY (`order_id`) REFERENCES `orders` (`order_id`);

-- Constraints for table `orders`

-- ALTER TABLE `orders`

ADD CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`customer_username`) REFERENCES `customer` (`customer_username`);
```

Ensure integrity it has been one of our main priority in our project. Data integrity is a fundamental component of information security. So, we kept the information of our customers private, only the customers have the ability to edit and see their informations. As a part of integrity inside the website for instance, customers cannot submit their information at signup form unless he/she filled up the complete form with a security question. Also, when the customer order a product will be asked to confirm his/her information (shipping address, last four digits of his/her bank account) before confirm the order.

For error checking we use a javascript to check for the data entered from the users when they create a new account. For example, if the user try to enter incorrect information when he/she register for a new account it will show an error message like email not correct, password is invalid or need 16 digits only for bank account, picture below illustrates the error messages type

