

Game Store - YAM

**Joshua Seymour**

**N00181604**

Year 2 2019-20

DL836 BSc (Hons) in Creative Computing

Table of Contents

[1 Introduction 1](#_Toc35949525)

[2 Business Concept 1](#_Toc35949526)

[2.1 Introduction 1](#_Toc35949527)

[2.2 Business model 1](#_Toc35949528)

[2.3 Market Research 1](#_Toc35949529)

[2.4 Other optional points go here 1](#_Toc35949530)

[3 Website Design 2](#_Toc35949531)

[3.1 Introduction 2](#_Toc35949532)

[3.2 Design Research 2](#_Toc35949533)

[3.3 Design Process. 2](#_Toc35949534)

[3.4 Design Framework 3](#_Toc35949535)

[4 Database Design 3](#_Toc35949536)

[4.1 Introduction 3](#_Toc35949537)

[4.2 ERD Diagram 3](#_Toc35949538)

[4.3 Additional information 3](#_Toc35949539)

[5 Implementation 4](#_Toc35949540)

[5.1 Introduction 4](#_Toc35949541)

[5.2 PHP 4](#_Toc35949542)

[5.3 Javascript 4](#_Toc35949543)

[5.4 Bootstrap 4](#_Toc35949544)

[5.5 Cascading Stylesheets 4](#_Toc35949545)

[5.6 Database Design 4](#_Toc35949546)

[6 Testing & Results 5](#_Toc35949547)

[6.1 Introduction 5](#_Toc35949548)

[6.2 Discussion of Functional Testing Results 6](#_Toc35949549)

[7 Project Management 6](#_Toc35949550)

[7.1 Discussion of use of Github for Software Project. 6](#_Toc35949551)

[8 Reflection 6](#_Toc35949552)

[8.1 Assessment of your learning. 6](#_Toc35949553)

[9 References 8](#_Toc35949554)

# Introduction

I learned what I needed for the php part of the project in class and in the online teams classes.

I had to do a lot of research on html buttons. I mainly went to Bootstrap and got parts of cards that I liked.

This is the link to my invissionapp freehand design process, and where my idea’s for what I wanted my website to look like developed. It also has the paper wireframes.

<https://projects.invisionapp.com/freehand/document/HhbRe6cCE>

Installation.

**# 1 Open xampp start Apache and MySQL.**

**# 2 Type in localhost/phpmyadmin in your browser (worked on google).**

**# 3 Click new on the left to make a new database called gamestore\_joshua.**

**# 4 While selected gamestore\_joshua click import and find the sql database file.**

**# 5 Make sure you have CA2 folder in the xampp->htdocs folder.**

**# 6 Type this into your browser to open my website localhost/CA2/Source**

**# 7 This will open the index.php file and show you the home page.**

# Business Concept

## Introduction

I will make a market for games that will be downloadable. The person will need to log in and sign up. The games they buy will always be downloadable online. There will be no physical copy of the game. The demand will be for games that will be cheaper than competitors.

## Business model

There will be no need for an actual disc for games, no physical copy of the game, and the game will always be available to download once you buy it.

Seasonal sales

## Market Research

The target audience is mostly gamers who will buy this, or just someone who’s looking for cheap games to buy.

Many people buy a lot of games they don’t even play during sales just because they are so cheap. This is a way we can clear games we don’t sell as much of.

## Other optional points go here

Approximately $138 billion was spent on games this year.

Competitors would be Steam, Blizzard, Epic Games, Origin, and Valve.

Price range from about $5~$60

# Website Design

## Introduction

I wanted to make a website to sell PC games. My research found that most of the gaming sites were quite dark and blue / purple / black in colour scheme, so that’s why my colour scheme is dark blueish.

## Design Research

I went to the Steam / Blizzard site and took screenshots of parts I liked and then made them in Photoshop. When In Photoshop it was much easier to create the parts I really like in html as I could get the text sizes and box sizes, and the general layout of each card the way I wanted them. That made it much easier to plan out how to make them in html.

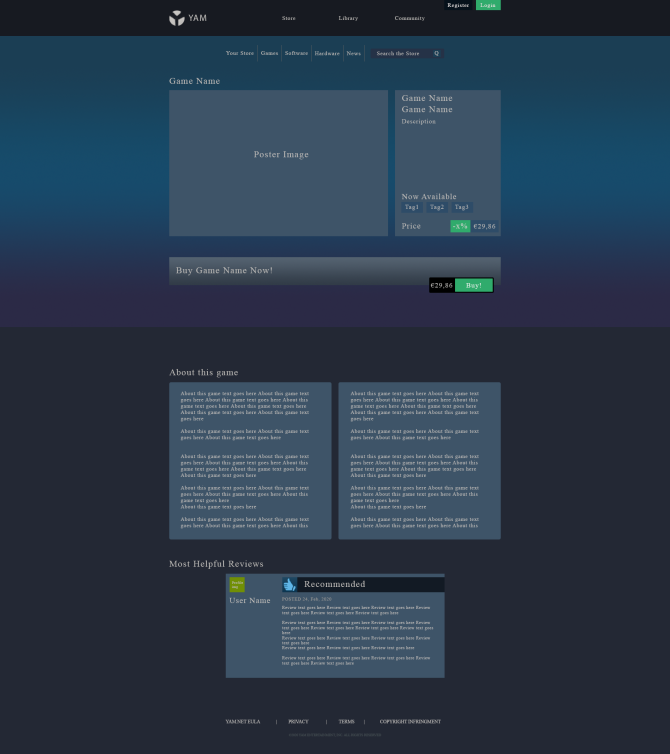
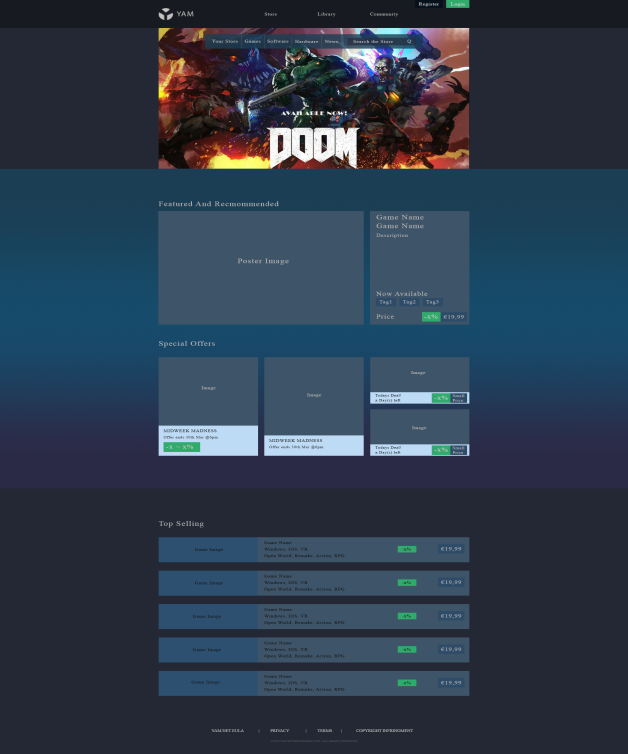
## Design Process.

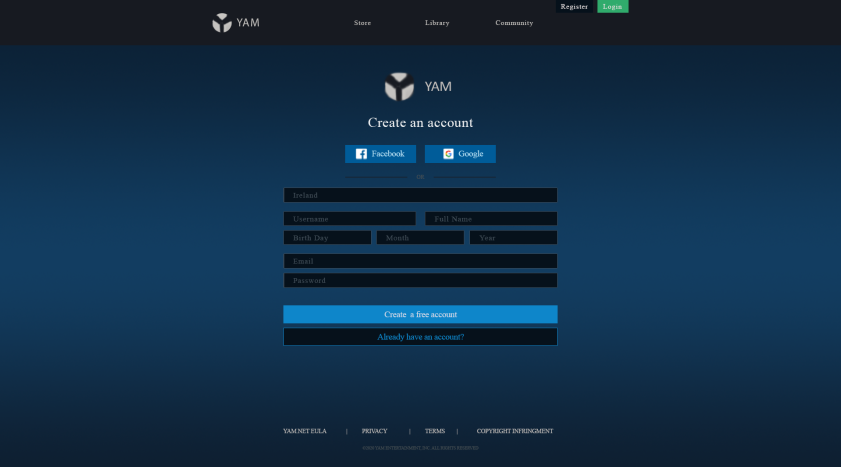
This is my invisionapp moodboard.

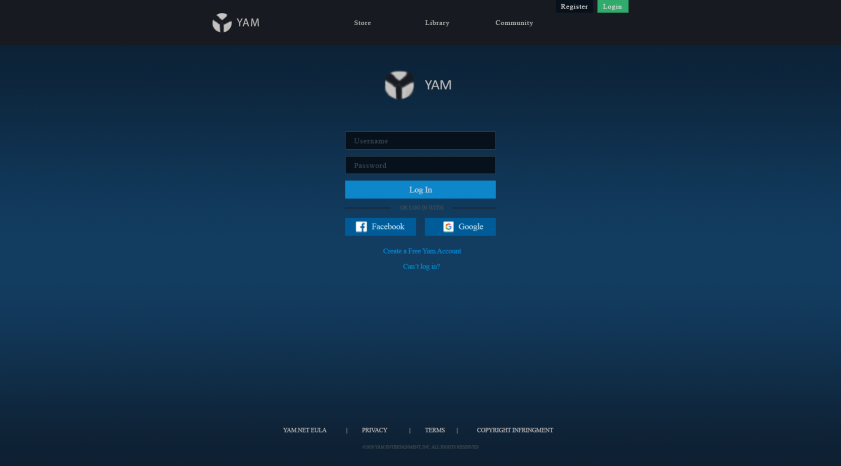
<https://projects.invisionapp.com/freehand/document/HhbRe6cCE>

It shows my wireframe sketches and thought process when I was designing the website.

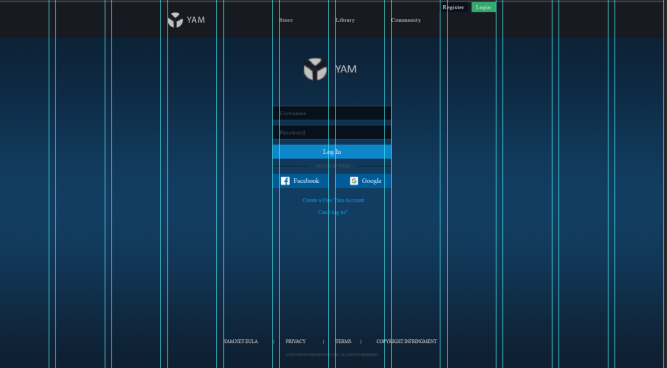
Here are my photoshoped wireframes. The psd files and my wireframe sketches are in my UI folder.

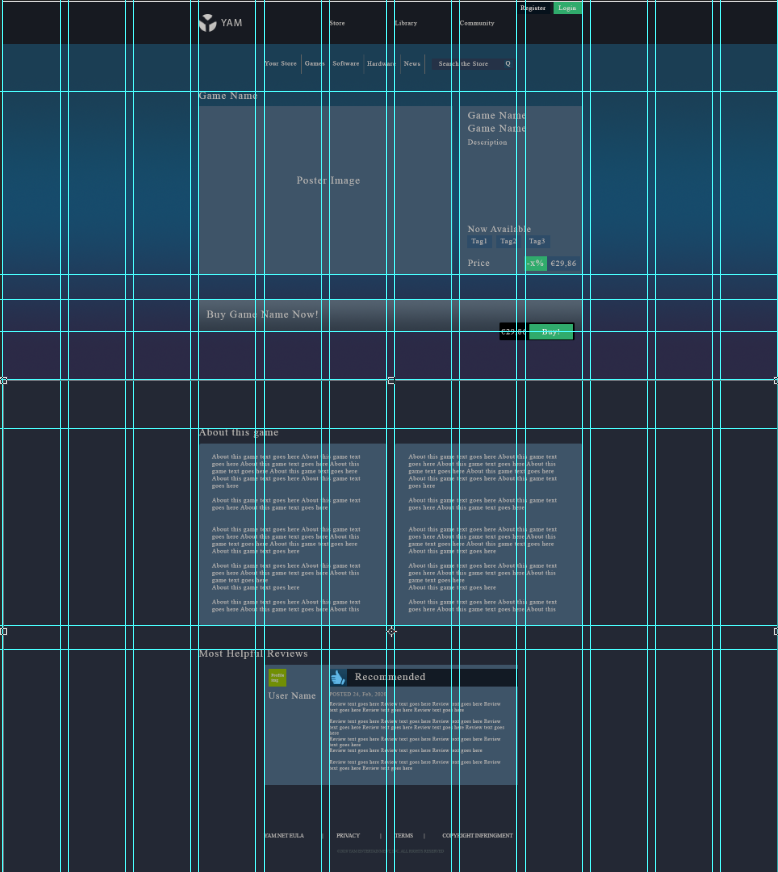
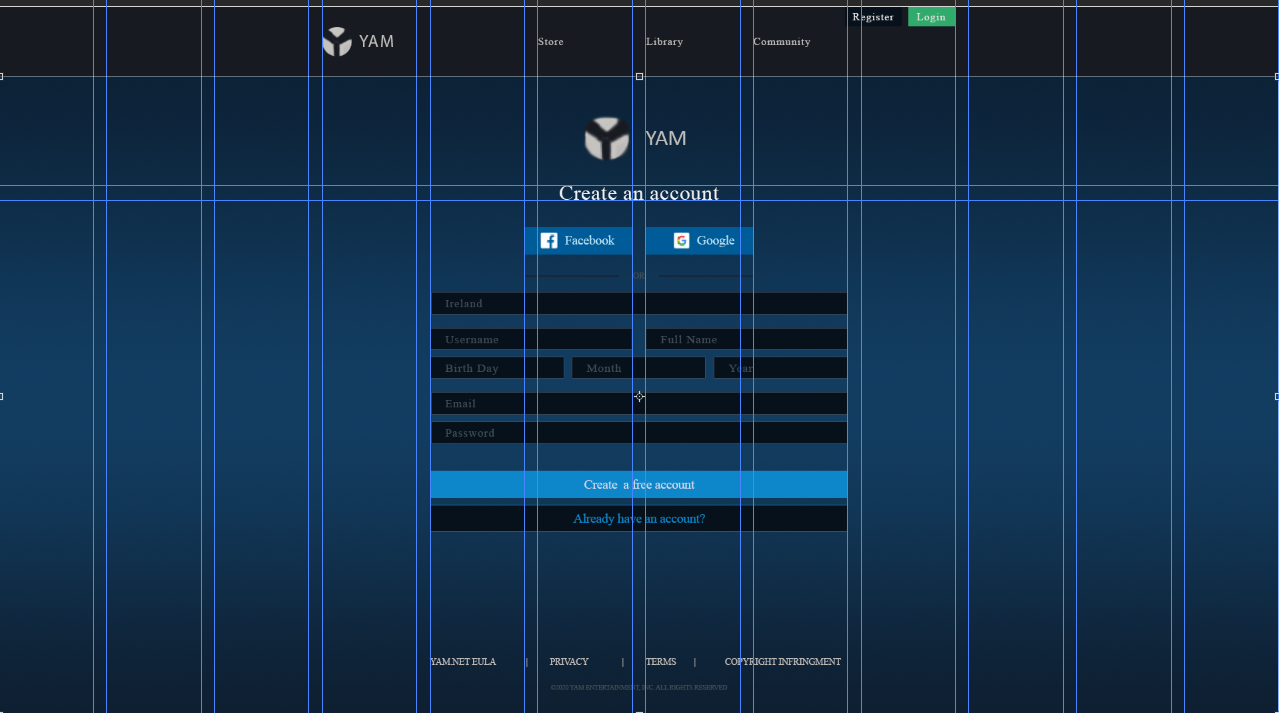
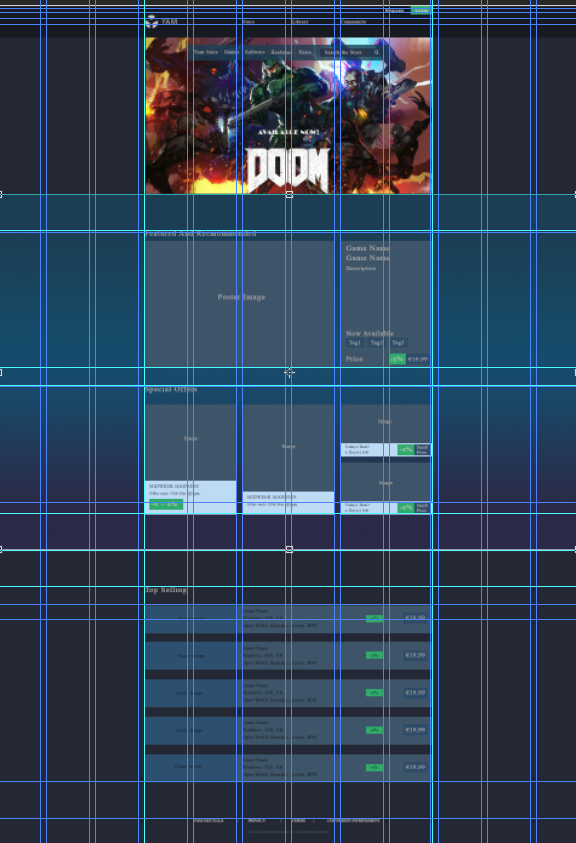






## Design Framework





# Database Design

## Introduction

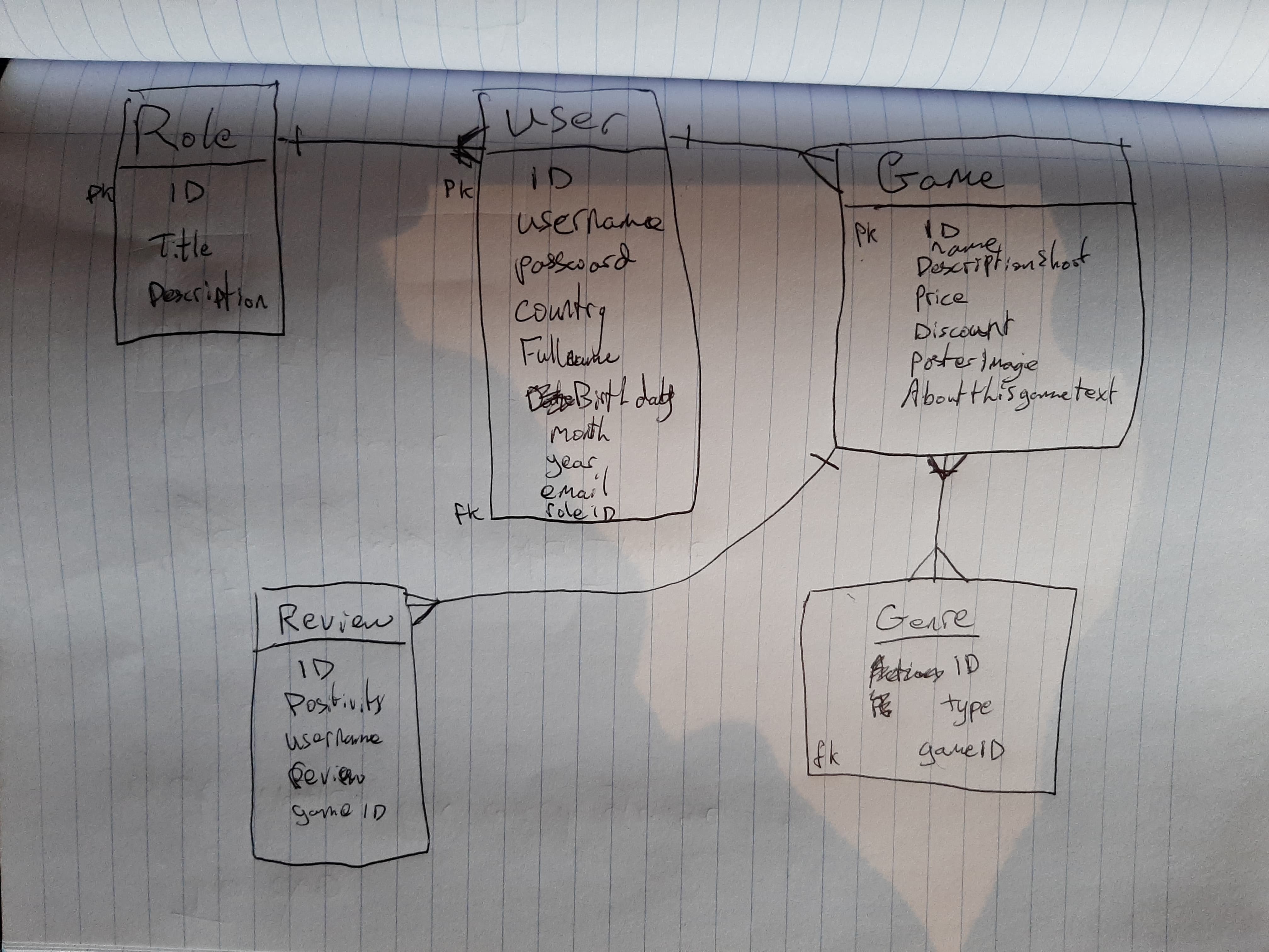
My main tables will be Role, User, Game, Review and Genre. I don’t actually know how to do the many to many so I will probably take out genre or not have enough time to add it in.

Role has id(pk), title and description. User has id(pk), username, password, country, full name, birth day, month, year, email, role id(fk). A role is given to a user, eg manager or user or admin. One user can have one role. One role can have many users. Game has id(pk), name, descriptionShort, price, discount, posterImage, and aboutThisGame. Review has id(pk), positivity, username, review, gameId(fk). The review foreign key relates to the game id. One game can have many reviews. One review can only have one game. (Genre will probably not be in this project but, in Genre one game can have many genres and one genre can have many games.)

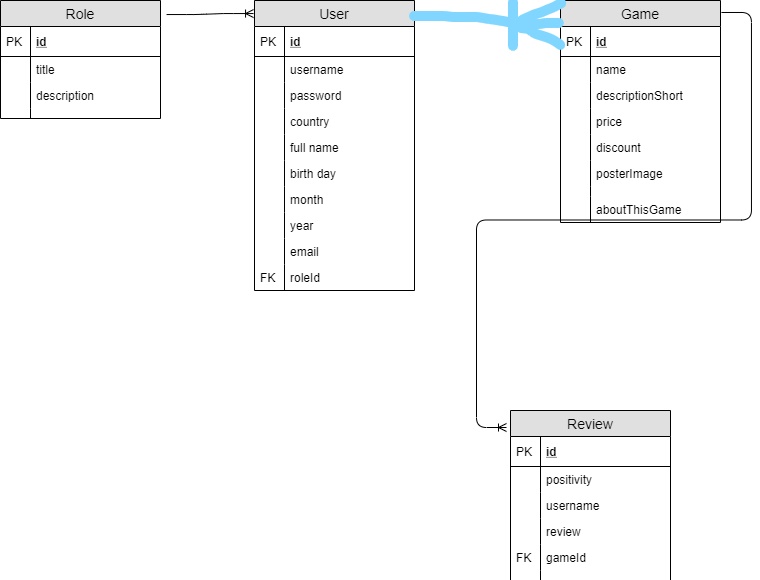
## ERD Diagram

I described the relationship in the previous heading.

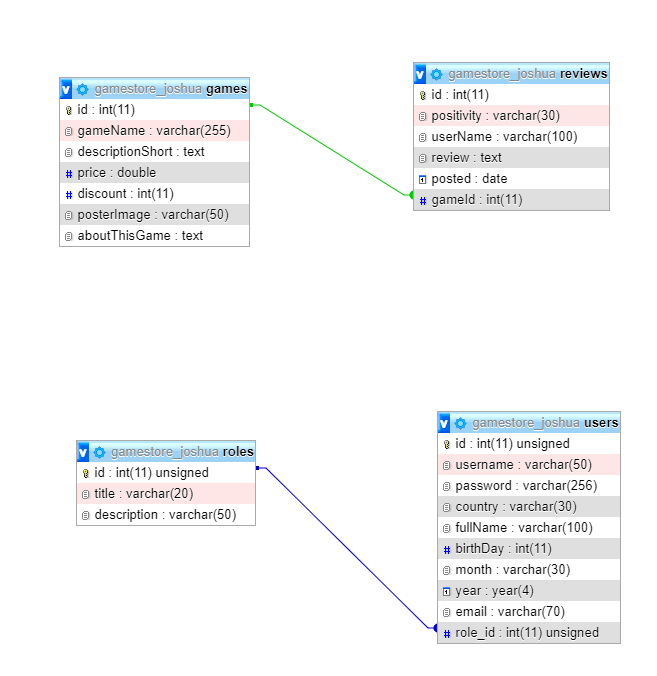
First ERD:



Final ERD:



ERD view From the designer in the database:

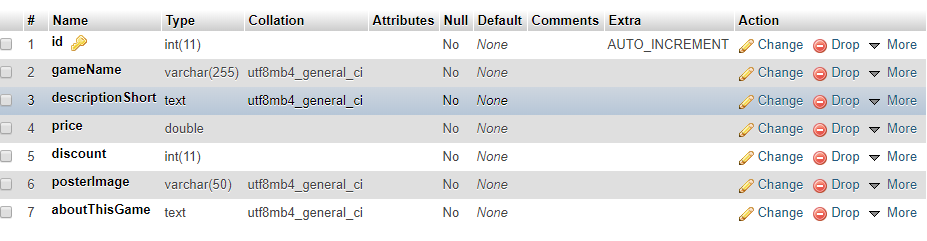


## Additional information

I had a bug where I couldn’t make a foreign key and it kept giving me an error whenever I tried to make it, so I ended up having to remake the table for gameId.

## Table structure

Games table

id for the primary key (int, auto increment so its unique)

gameName to store the name of any game.

descriptionShort to store any short relevant info about the game.

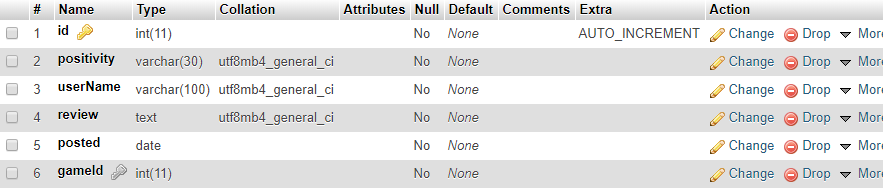
price to store the price as a double so it shows the .00 after the number.

discount to store the discount %.

poster image to store the actual image that will be displayed.

aboutThisGame stores the text for the description.

Reviews Table.



Id for the primary key for unique identifier.

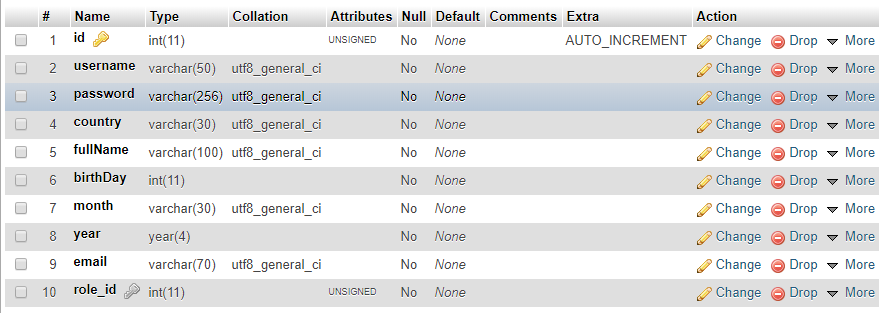
positivity was originally a enum but I couldn’t get that to work with php so I just changed it to a varchar and had a drop down list for the values it could be.

userName to store the username of the person who wrote the review.

posted is to store the date the review was posted.

gameId to store the game id of the game the review is related to. (fk)

Users table



id for pk.

username to store the username.

password to store the users password.

country to store their country.

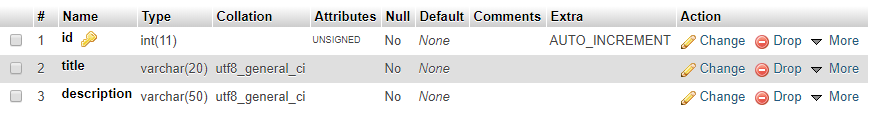
fullName to store their full name

birthday, month, and year to get their DOB.

email to store their email.

role\_id to store the role that this user relates to.

Roles Table



id to store primary key.

title to store the title they have.

description to explain what the title can do.

# Implementation

## Introduction

First I did research for my design.

Then I created my design in Photoshop.

Then I started to code it up in html. I only had a small bug I couldn’t figure out, on the index page / home page. My special offer’s card’s text boxes kept overflowing out of the div.

When I made my database, I found a small error where I couldn’t make a foreign key, so I had to remake the table.

Finally I added the php. This took the longest time and was the thing I spent the most time on as I’m not very good at php but towards the end I was starting to get used to it.

## PHP

• Backend Functionality - View All, Create, Update, Delete Items

The $game = Game::all(); gets all the info about games and puts it into the $game variable. Then I use that to output each of the columns eg. $game->gameName outputs the game name.

For my create, I can both create Reviews and create Games. Basically I have my main form page where you input in all of the information, and it sends it to the games\_store.php where it goes through the validation rules and if any info you put into the form is wrong this will come back with an error when you press submit and you will be prompted to put in the right info. If it has no problems with the information you submitted it will create the new game in game\_store.php and then bring you back to games.php

For Update, the edit form is directed to my update form which validates the information and filters it. If there’s an error it returns the error and prompts you to enter the right info. Once the right info is in the form, it updates the info that was passed in using the id of the game that was used to have the info in the form that wasn’t updated.

For Delete, it gets the id then validates the get(the id) and filters it. If it’s an invalid id then it throws invalid game id / invalid review id. On success it runs the delete

• Backend Login/Register Functionality

Login

The posted data gets put through the validator and filter rules. If the validated data is false it gets the errors array, checks if there’s a user with the username that was entered. If not -> username not valid, does the same with the password except it hashes it so that it can’t be read. and then depending on what role the user has it sends them to a home page admin / manager / user.

Register

The data is put into a post as sanitized\_data. The info is validated and filtered. If data types are wrong it returns errors. If success it puts the validated data into the variables username goes into the username variable etc. checks if the password is the same as the confirmation password. Then finds if the username is taken or not, if it is ->username already registered. Then it saves the data to the User class in the database (With the password hashed). Then it goes to the user home

• Frontend Functionality - Show all items

My view all is my games.php and my reviews.php I use a foreach loop to go through each of the games / reviews with table heads and table rows display each of their columns eg. Game name, short description etc

• Add an item in the backend and show it visible in the frontend

The php retrieves data from the database so if it’s added there or with a create(the create adds it to the database) the php will show it in the front end if there’s a list to show all.

• Optional (ordering system)

For the button sorting , I used get element by class name which gets the element that has the class of ‘btn-sort’ and put that into an array. Then each element in the array is passed through a function which assess if each item will be at the top and the one below and below etc. (the order its shows them)

For the search, I get element by ID this is the id I gave to my table and I get the tag name ‘tbody’ (the table body), there’s a variable for whatever you type into the search filed and if it detects input it runs the function which filters each of the rows if you input something similar to a row it will show the row. If it’s not related to anything it shows nothing.

• Optional (many to many relationship)

## Javascript

I implemented the prompt to ask are you sure you want to delete this on both games.php and reviews.php

I also implemented the search and button sort on both of those pages with javascript.

You can search for a game or a review.

## Bootstrap

I made the index.php reactive lg and sm, Bootstrap was also very helpful for buttons and form layout parts that I thought would look nice with my system.

## Cascading Stylesheets

I used css to size my texts and space everything out correctly.

I used two main css variables one for my font family and one for the color of the font. I started with this because I was going to change my font family at a later stage, but I was trying to change it and it just won’t change so I’m stuck with a font I don’t really like, but it’s the same for new css class I try to make they aren’t working(solved). I just had to reset my cache and have fixed the problem. I was able to switch to the font I wanted because of css variables so it was very helpful.

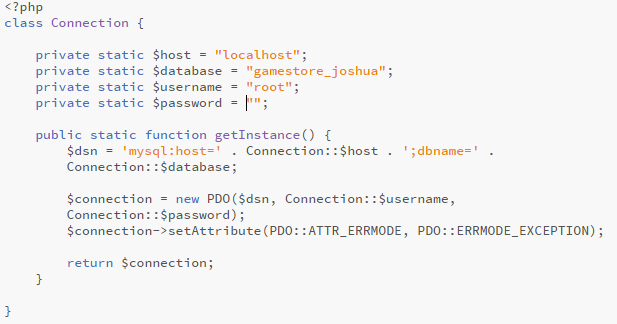
## Database Design

Describe your Database implementation using the following pointers:

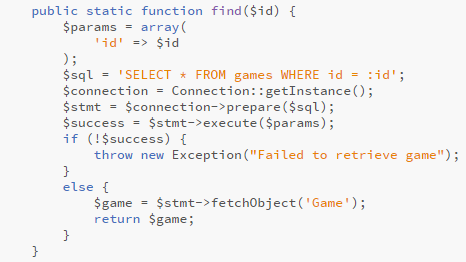
• Referential integrity

• DB connection using PHP and describe the following operations:

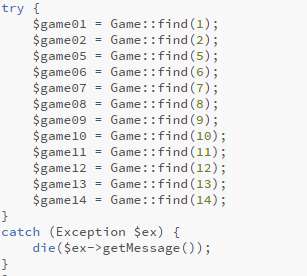
Using the connection class which has the database name the host name the username and the password then uses the getinstance() function to connect.



* A SELECT query with at least one WHERE condition

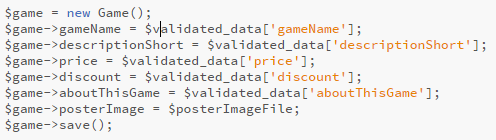


With this select I can get all the data from a specific game. In my index.php I use this to get data about specific games by using their game id as shown in the picture below. The find function takes in a parameter (the id) and uses the select \* from games where id = :id <- being the id that was taken in. if it succeeds then it fetches the object and all the info from the database and returns that data to $game.



* An INSERT command

The games\_create.php form takes in user input to create a game, then sends the data to the games\_store.php where it is validated and inserted into the database



This is creating a new game and filling the data fields with the validated data. The save() function at the end puts the values into an array with database values = the inputted values. If there is no id assigned to the object it’s creating then it lets the primary key auto increment to whatever id it should be and it inserts all the values using INSERT INTO games() values().



* An UPDATE command with at least one WHERE condition

There is an update in the above picture. The edit form sends the data(including the id) to the update form on submit it goes through the update validation and then it is saved (save()) the save function skips the first if because the id is not null and uses the update games sql which updates the id of whatever id was passed in.

# Testing & Results

## Introduction

I have a html problem on my index page at the special offer’s card’s text, the text overflows out of the div. (Unsolved)

I have a bug with my create reviews, whenever you click on a form field the screen shows and un-shows the sidebar to scroll like 5 times which makes the page look jittery when you click on anything. (Unsolved)

I tested each of the forms by entering invalid data into the fields to see if they would still work.

All of them work like I want them except create review where my game id is the problem and won’t submit unless I take game id out of the form completely (just a problem with my php the foreign key).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | Testing login form | Username: Joshua Seymour | success | invalid Username, has to be alphanumeric | JoshuaSeymour without the space will work. |
| 2 | Testing Edit games form | Price: five | Enter a valid number | Enter a valid number | Validation works well. |
| 3 | Testing Create Game Form | Image: image.png | success | Error this image already exists | Can’t enter the same image more than once. |

## Discussion of Functional Testing Results

Describe the results from the tests. Address any functionality where unexpected behavior could not be debugged.

My problem with html on the index page where the text box color overflows out of the div (special offers) I tested moving all the div’s around but nothing I tried worked like I wanted it to.

Test no.2 worked out well I did this for each form.

# Project Management

## Discussion of use of Github for Software Project.

I think it was very nice being able to use Github for this project, I used the git gui, It was very easy to use once I learned how to. It’s nice being able to get my project from the cloud where ever I want. For the to do list, I didn’t use it, I prefer using a pen a paper to tick off tasks I need to do, but I think it’s a good idea.

# Reflection

## Assessment of your learning.

I think I have learned a lot of php during this project, I missed a lot of louise’s classes so it was quite hard to understand most of the php at first because I had little to no understanding of php but now I feel like I have a much better understanding than before the project. I didn’t learn as much html but I learned how to use bootstrap to get pieces of cards that I would like for forms etc. I learned a lot more about css in this project and as I’ve always been quite bad with it and not understood but was forced to use a lot of it this project and I am very comfortable with it now eg. Used css variables.

List which part of the project would need further development and itemize where you feel you have not satisfactorily completed the continuous assessment.

I feel like I could have done a lot more with the php part of this project if I had been in class more it probably would have been a lot easier. I struggle with motivating myself to get into class even though I like the classes so it’s my fault for not being in, but it would be nice if there was a recap or something online.

How could the project could be developed further?

I could add Genre’s and make it a many to many relationship with games. I could also add a stored games that is games you have bought. It’s stored in library. Add to cart system. Add a working Buy game with download of fake file. Add a fake money system to buy games with.

# References

The Department of Technology and Psychology in IADT uses APA referencing style.

Use alphabetical order for your references.

This site gives details about how to cite websites using APA:

https://www.wikihow.com/Cite-a-Website-in-APA

The following is a useful site for creating citations for APA for websites.

<http://www.citationmachine.net/apa/cite-a-website>

You can also use the Referencing tab within Microsoft Word to enter reference information manually. Word then creates an APA style reference.