E-commerce Gaming Website

Wiktor Lesiak

April 2018

Group Project

Submitted in part fulfilment for the degree of

B.Sc. Computing

School of Informatics and Engineering,

Institute of Technology Blanchardstown,

Dublin, Ireland

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Degree of **B.Sc. in Computer Science** in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed:_________________________ Dated: ____/_____/_____

Abstract

This is an e-commerce website which allows users to register and purchase the latest and new releases of games, interact with other users using forum, rate games that are implemented by admin to the website and read the news and updated on the homepage. Once the user is registered all his details are sent to the database with the unique hash and encrypted password and activation link which has been sent to their registered email. Registered users can look through the website but cannot purchase anything until they activate their account using that link that has bed send to their email account.

Only administrators can change and edit the database using the ADMIN CURD option after they are logged in. Admins can edit the store by adding new products(games) edit them or delete them, they are also able to change, add and remove news form homepage.

The technologies I used included html for creating the layout of the website, php to create functionality to the web pages, SQL for storing and querying the database, laragon software to connect to the database and Apache server and run the program on localhost, myPhpAdmin extension used with Laragon to configure MySQL database using the browser.

This website could be adopted by anyone who wants to purchase the games, see the latest new relating to the games, discuss their opinions using the forum, and rate or see the rating of the list of games that are on the website. And thanks to the dynamic design, the website is able to be used on any device available.

Contents

List of Figures

Chapter 1: Introduction

1.1 Main Objective

The main objective is to create basic dynamic functional website with database connectivity, administrator functions shop, rating, and implement forum function. Also, to document the project with System Analysis and Design throughout implementation to a working system with critical analysis and functionality.

The main purpose:

There is general big interest in games all around the internet, so it would be ideal to create website where everything is in one place, and where gamers can spend their time to see other people's opinions on the new released games and the what's new around using the News.

They are many shops, forums, and news websites but none of them are in one place and simple in design, so users of all ages can use them.

I've studied the technologies involved in this project, such as php, html, SQL, in previous module through the years in college and it felt possible to create functional website by using those skills.

1.2 Body of Argument

The Problem:

To create E-commerce Gaming Website featuring User Registration, Admin Privileges, Products Rating, and implemented Forum with dynamic design.

Aims and Objectives:

- To create a fully functional website that allows for database connectivity using php.
- To provide a User registration and login functionality.
- To allow admin account to log in and provide a function that allows them to modify the data in database.
- To Create News page, Rating page, Store with shopping cart that would store all the selected products, and implementation of Forum.
- To implement data from database using php.
- Creating simple and dynamic design allowing users to use it on all devices.

Structure of Dissertation:

Throughout the thesis I will demonstrate document how I used the mentioned technologies and skills to create the website by breaking this report into sections.

Chapter 2 will consist of System analysis and it explains what the application is supposed to do and how it supposed to behave in different situations. I also include the Technical Requirements, Inputs, and outputs of the Use-Case Diagrams.

Chapter 3 deals with the System Design and explains how the application was implemented by describing html and CSS design as well as functional data design considerations

Chapter 4 describes how the technologies mentioned were implemented.

Chapter 5 deals with testing and evaluation and includes Functional Correctness and screen-shots.

Chapter 6 concludes the report

Project Listings, Source Code, a Diary and References are included in Appendices.

1.3 Sources of information

The sources of information are included in the lecture notes from my Ordinary Bachelors Degree various computing books and the Internet.

I used 'PHP and MySQL Web Development' by Luke Welling to improve my knowledge about php and MySQL implementation to the websites. I also used 'Introducing HTML5' by Bruce Lawson and Remy Sharp to aid the html design of the pages.

Also, I used couple of websites to research the technologies used and structure of my website. Include: www.w3schools.com for the design structure and the layout of the site, www.php.net used for ideas regarding using the php implementation.

Chapter 2: Literature Review

2.1 Overview of fields reviewed and sources consulted

To successfully develop a dynamic easily simple designed website there are many technologies that could come into play and the technologies that are crucial. Web development is a huge growing industry with new technologies that are being implemented daily.

There are two main crucial parts to the website development, it’s a front end where the graphical user interface is and the back end where code implementation and database takes place.

Those are the technologies used for dynamic functionality in this project and list of the priorities where 1 is most important:

Priority	Description
1	PHP
2	HTML
3	MySQL
4	CSS
5	JavaScript

The following are reviews that describe role of the technologies used in this project:

2.2 Review of HTML

What’s HTML5

HTML (HyperText Makup Language) is what makes the Web layout work in different positions. Web pages are build using HTML code, which in turn is read and used by Web browsers to construct layout and format text and implement images and video on a page. The web browser interprets HTML code into readable and visible web pages and its readable and visible by all the browsers and computer devices.

HTML is a simple text-based mark-up language, so you can edit and create HTML in any text editor available.

HTML will be used in this project as a base and structure web pages.

2.3 Review of CSS3

What's CSS

CSS (Cascading Style Sheets) which is known as style sheet language is used for adding style to the HTML documents e.g. set colours, layout, changing fonts, etc. CSS makes documents and make is presentable to the web user.

CSS3 will be used in this project for application of styling rules to make a good looking and well-designed website.

2.4 Review of PHP

What's PHP

PHP (HyperText Pre-processor) is a scripting language as opposed to programming language. PHP was designed and developed by Rasmus Lerdorf to write Web scripts. It's a server-side, cross-platform technology.

It was designed to produce dynamic webpages and it's the fastest and easiest to learn rather than the alternatives. It was one of the first sever side scripting languages to be imbedded into a HTML source rather than calling on external files. PHP has been implemented in over 20 million domains all around the world. It plays a role of interpreter as it communicated and processes pages with the files system, mail server and database, it delivers data from and into the web servers.

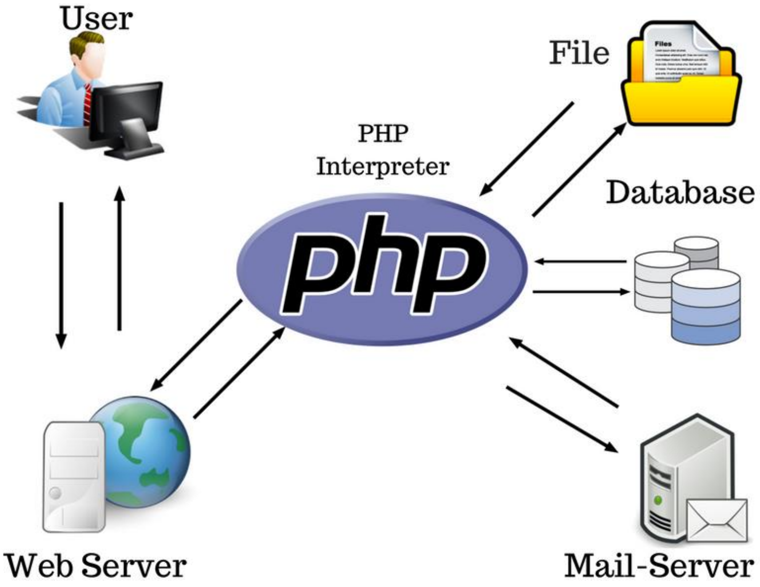


Figure 1 - PHP Diagram

Source: http://www.codekul.com/blog/wp-content/uploads/2018/03/PHP-work.png

PHP will be implemented into the HTML document to create dynamic capability of the website for this project

2.5 Review of SQL

What's SQL

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system, providing multi-user access to numerous databases. It is used a lot in web applications.

What's Database

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements.

Database Usage

Data generated by php can be retrieved from MySQL. It develops the site's content from a static into a dynamic site.

2.6 Review of JavaScript

What's JavaScript

JavaScript is a scripting or programming language that allows you to implement complex things on web pages every time a web page does more than just sit there and display static information for you to look at displaying timely content updates, interactive functions.

2.7 Conclusions of review

After the research of the technologies that are used to create, a fully dynamic and functional website it was decided to use HTML, PHP, CSS, JavaScript, and MySQL as mentioned before. HTML and PHP most crucial and were used to write the code for dynamic user interface and connecting website to the database. MySQL was used for as database management for our stored data. CSS used to present the layout of the pages and to adapt to the presentation to different types of devices, such as large or small screens.

Chapter 3: Method

3.1 Overview of method

This system is an Online E-Commerce Gaming Website with forum implemented. Users can only access the website when they register using their email, once user registers their details are stored in database and email with verification is sent to their email account with activation link. Once logged in and account is activated user can purchase and rate games.

If user forgot their password, they can reset the password using the Forgot Password function on login page. Once they prompt this function they must input the registered email and they receive an email with password change link.

Users can access the forum and create forum account to use fully use the forum.

Admin can configure forum using admin login details which allows them to fully configure users, topics, and categories of the forum they are also able to change roles of the users.

Administrator can log in through their account and modify the data in database by the ADMIN CURD which is only able to be accused by admin.

This website can be used by anyone who is interested in purchasing, rate, and discuss or share their opinion about all the different games.

Chapter 4: System Requirements and Specification

4.1 Introduction

Implementation of web projects requires correct structure and design. This web project has its own development plan which consists of stages known as SDLC (Software Development Life Cycle) phases. Those stages help developers to identify and accurately plan their project, its requirements, and have structured work plan which can allow the work to guide in right direction and successfully acquire the goal of the project.

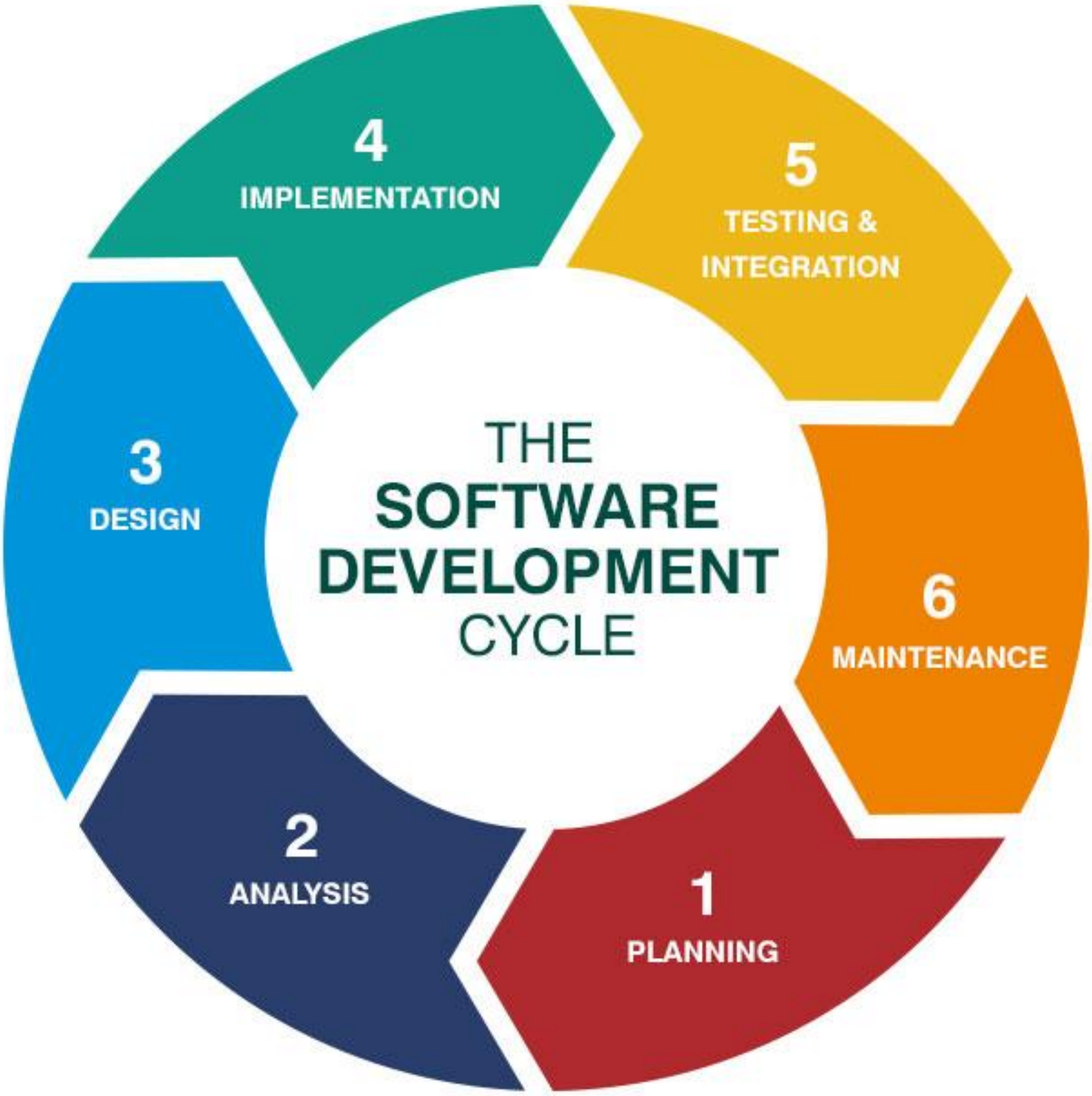


Figure 2 - Software Development Cycle

Source: https://online.husson.edu/software-development-cycle/

4.2 Requirements specification

The functionality which is required form the system is listed below:

- The system should allow user to create an account using their existing mail and generate unique hash for each user.
- System should send activation email with verification link to the users registered email.
- The system should send reset password email when Forgot Password is used.
- The system should allow registered user to view News, Shop, Forum all generated form database
- The system should allow users with activated and verified account purchase games and view their cart.
- The system should provide login function for administrators.
- The system should allow administrators to modify data in database using ADMIN CURD.
- System should contain rating for every game in database with average rating from all registered users.
- Users should be able to create forum account to fully add, edit and delete their own posts forum
- The system should allow forum admin to confirm posted topics by users.

Main Technical Challenges

- Learning and Implement PHP
- Connecting to database using PHP and MySQL
- Working and implementing data in SQL
- Implementing Forum using phpBB
- Simple, compatible, and dynamic functional design

4.3 Use Case Diagrams

Method of collection the requirements for a Web project require Use Case Diagrams. Its job is to gather the needs and system requirements for each user and helps to determinate the goals and functional of the system.

4.3.1 Customer Use Case Diagram

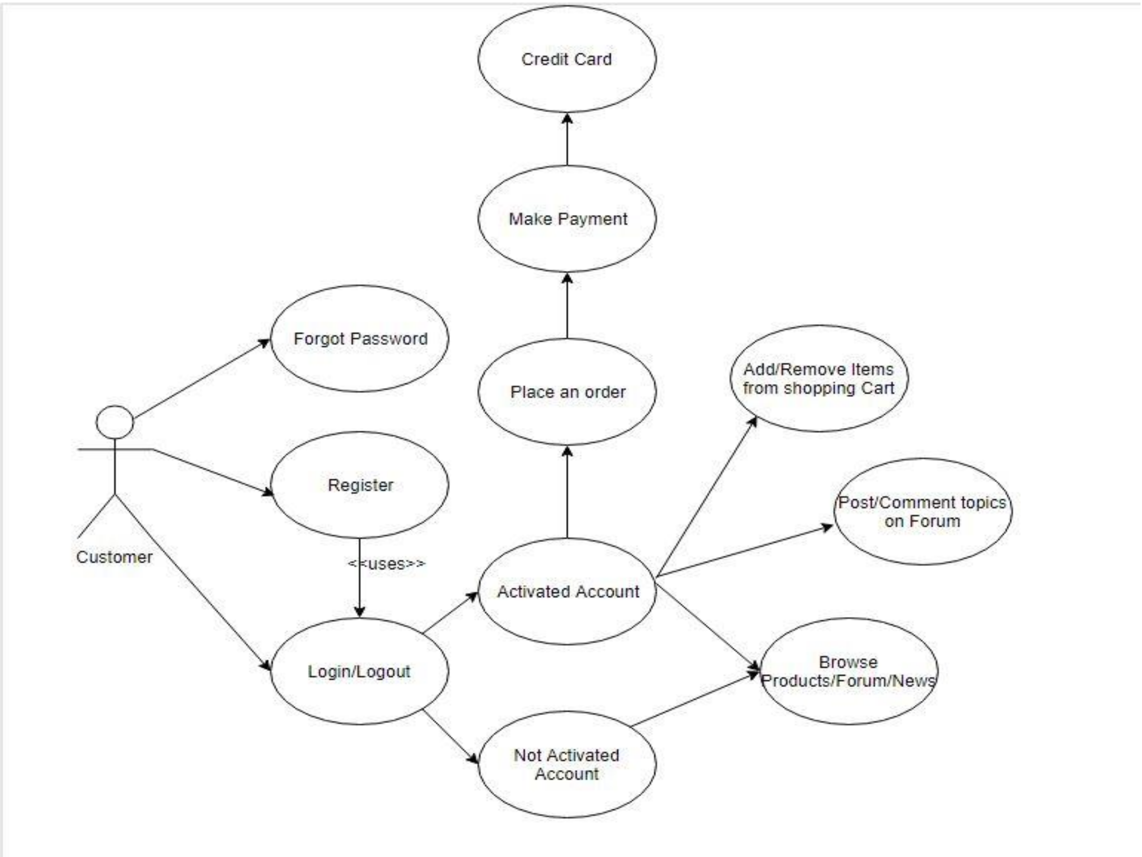


Figure 3 - Customer use case diagram

4.3.2 Admin Use Case Diagram

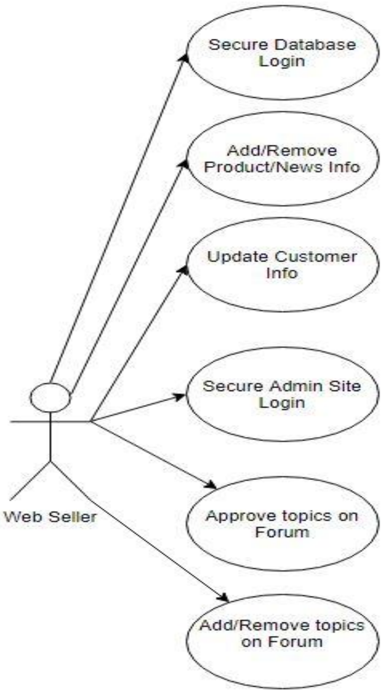


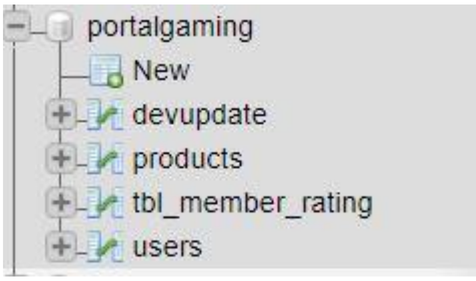
Figure 4 - Admin use case diagram

4.4 Data Requirements

Database consists of four tables: devupdate, products, tbl_member_rating and users

Admin can modify devupdate and products tables

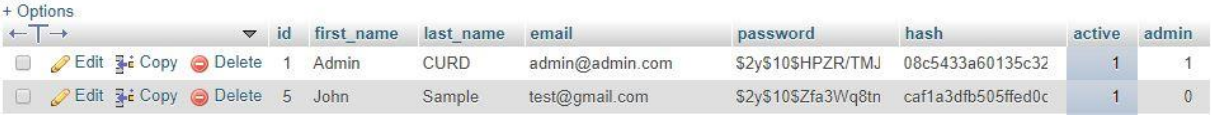
In *Figure 4* is a screen of screen of the phpMyAdmin portalgaming database with four tables.



portalgaming
- New
- devupdate
- products
- tbl_member_rating
- users

Figure 5 - portalgaming database

The users table consists list of users who has created account and account each of the users have their own unique hash and encrypted password where if active is equal to one then user account is verified and active and if admin is equal to one it means that this is admin and can modify the database using the ADMIN CURD.

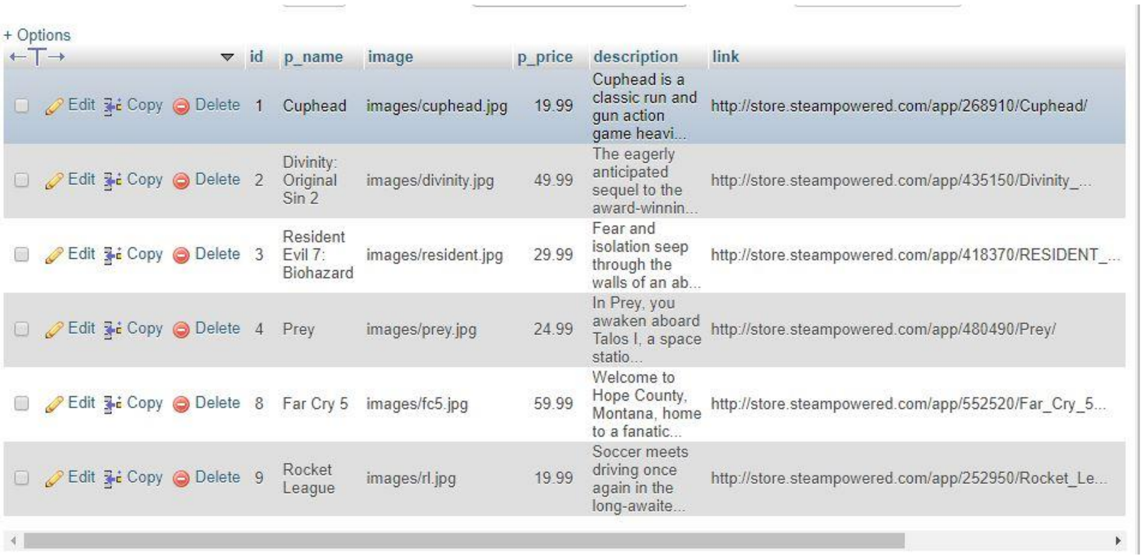


+ Options

←T→	id	first_name	last_name	email	password	hash	active	admin
□ Edit Copy Delete	1	Admin	CURD	admin@admin.com	$2y$10$HPZR/TMJ	08c5433a60135c32	1	1
□ Edit Copy Delete	5	John	Sample	test@gmail.com	$2y$10$Zfa3Wq8tn	caf1a3dfb505ffed0c	1	0

Figure 6- users database table

The products table contains all the products in shop it consists of Name of the game, image, price of the game, description, and link when user want more info.

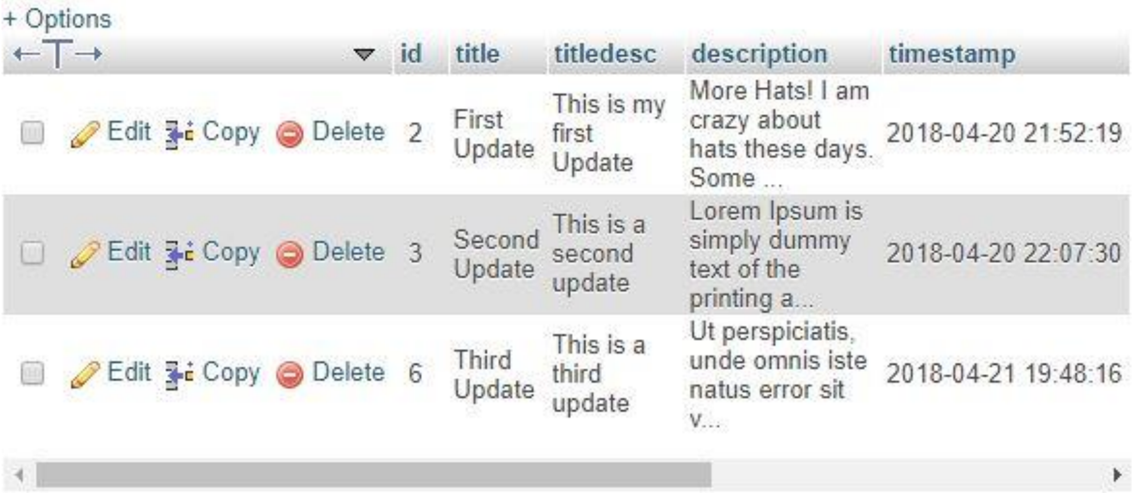


+ Options

←T→	id	p_name	image	p_price	description	link
□ Edit Copy Delete	1	Cuphead	images/cuphead.jpg	19.99	Cuphead is a classic run and gun action game heavi...	http://store.steampowered.com/app/268910/Cuphead/
□ Edit Copy Delete	2	Divinity: Original Sin 2	images/divinity.jpg	49.99	The eagerly anticipated sequel to the award-winnin...	http://store.steampowered.com/app/435150/Divinity_...
□ Edit Copy Delete	3	Resident Evil 7: Biohazard	images/resident.jpg	29.99	Fear and isolation seep through the walls of an ab...	http://store.steampowered.com/app/418370/RESIDENT_...
□ Edit Copy Delete	4	Prey	images/prey.jpg	24.99	In Prey, you awaken aboard Talos I, a space statio...	http://store.steampowered.com/app/480490/Prey/
□ Edit Copy Delete	8	Far Cry 5	images/fc5.jpg	59.99	Welcome to Hope County, Montana, home to a fanatic...	http://store.steampowered.com/app/552520/Far_Cry_5...
□ Edit Copy Delete	9	Rocket League	images/rl.jpg	19.99	Soccer meets driving once again in the long-awaite...	http://store.steampowered.com/app/252950/Rocket_Le...

Figure 7- products database table

The contents of the devupdate table is responsible for News and updates page it consist of Title, Title Description, Description, and the timestamp which shows the time the news has been posted.

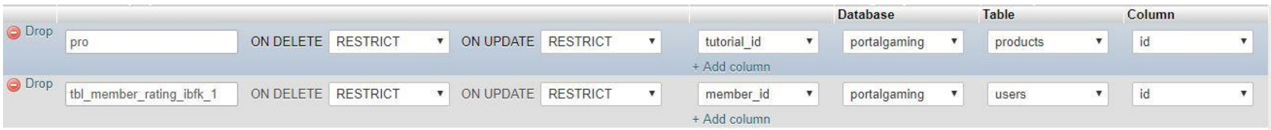


+ Options

←T→	id	title	titledesc	description	timestamp
□ Edit Copy Delete	2	First Update	This is my first Update	More Hats! I am crazy about hats these days. Some ...	2018-04-20 21:52:19
□ Edit Copy Delete	3	Second Update	This is a second update	Lorem Ipsum is simply dummy text of the printing a...	2018-04-20 22:07:30
□ Edit Copy Delete	6	Third Update	This is a third update	Ut perspiciatis, unde omnis iste natus error sit v...	2018-04-21 19:48:16

Figure 8- devupdate database table

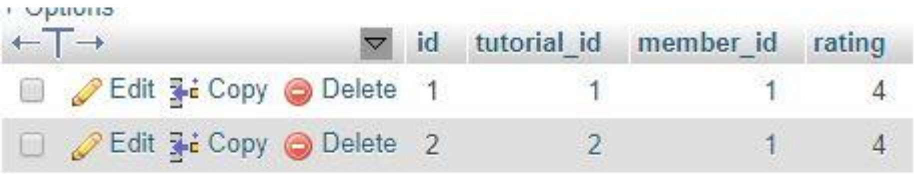
The rating table called tbl_member_rating is connected to the user and procuct tables where tutorial_id column take id from product id and member_id takes id from users table.



						Database	Table	Column
Drop	pro	ON DELETE	RESTRICT	ON UPDATE	RESTRICT	tutorial_id + Add column	portalgaming	products	id
Drop	tbl_member_rating_ibfk_1	ON DELETE	RESTRICT	ON UPDATE	RESTRICT	member_id + Add column	portalgaming	users	id

Figure 9- Database table connections

In *Figure 9* when the tutorial_id is one it takes id one from products table so the game is Cuphead. When member_id is one it take id one form users table so the user is Admin



←T→	id	tutorial_id	member_id	rating
□ Edit Copy Delete	1	1	1	4
□ Edit Copy Delete	2	2	1	4

Figure 10- rating database table

Chapter 5: System Design

5.1 Introduction

The development of this E-Commerce Gaming Website Management follows the System Development Life Cycle as mentioned before.

The main design will be based on Three Tier Architecture which represents dynamic website structure.

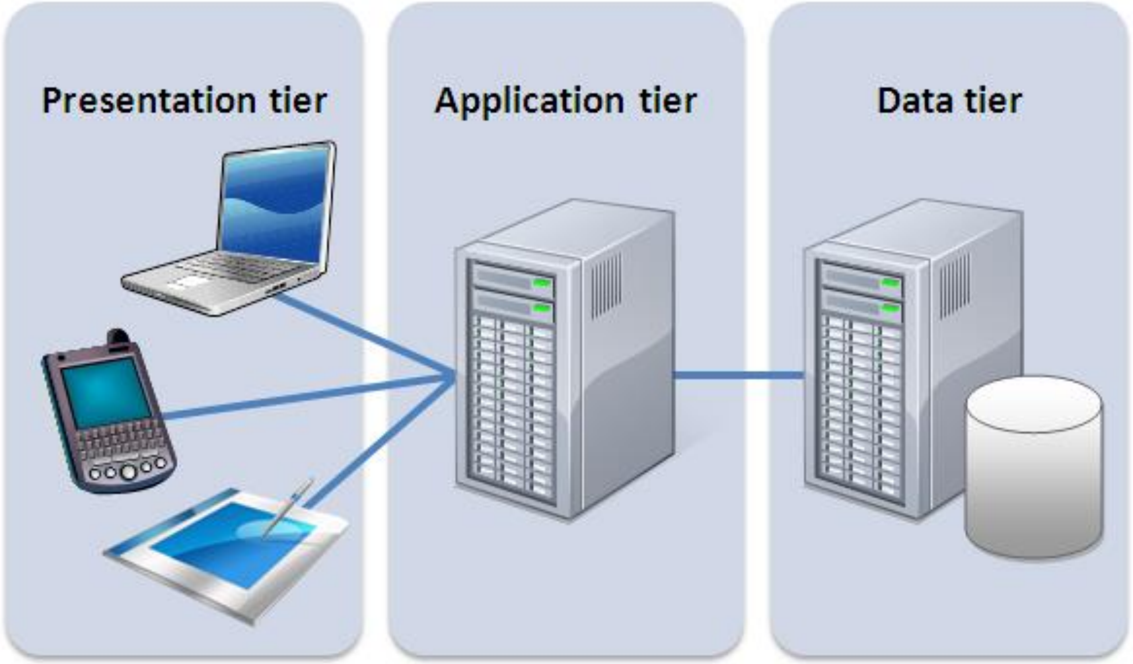


Figure 11 - Three Tier Architecture

Source: https://managementmania.com/en/three-tier-architecture

Data tier is a backbone that contain all the dynamic data on the website it provides Application tier with requested information stored in database, it plays function of data storage for the Application.

Application Tier passes requested actions and sends information's between presentation form and data tier by interpreting requests, location and retrieves file from database.

Presentation tier known as user-end presented graphical display in a browser which is used to to display requested data from the server.

5.2 Concept of the Project

Planning, Requirements Analysis and Design are the steps where questions are part of the development and research and analysis are being presented. In this part of report all the initial ideas behind the project are clarified together with all preliminary program descriptions.

The design of the website goes more into the detail of the project, it shows the look, database design, and feel of the product where the product is represented as prototype. All the detail of required elements and their interactions with each other are specified. Flowcharts and storyboards are often produced in this stage.

5.3 Database Design

Efficient database structure plays major role in the database management system. Database login is best represented with ERD diagram which is displayed below:

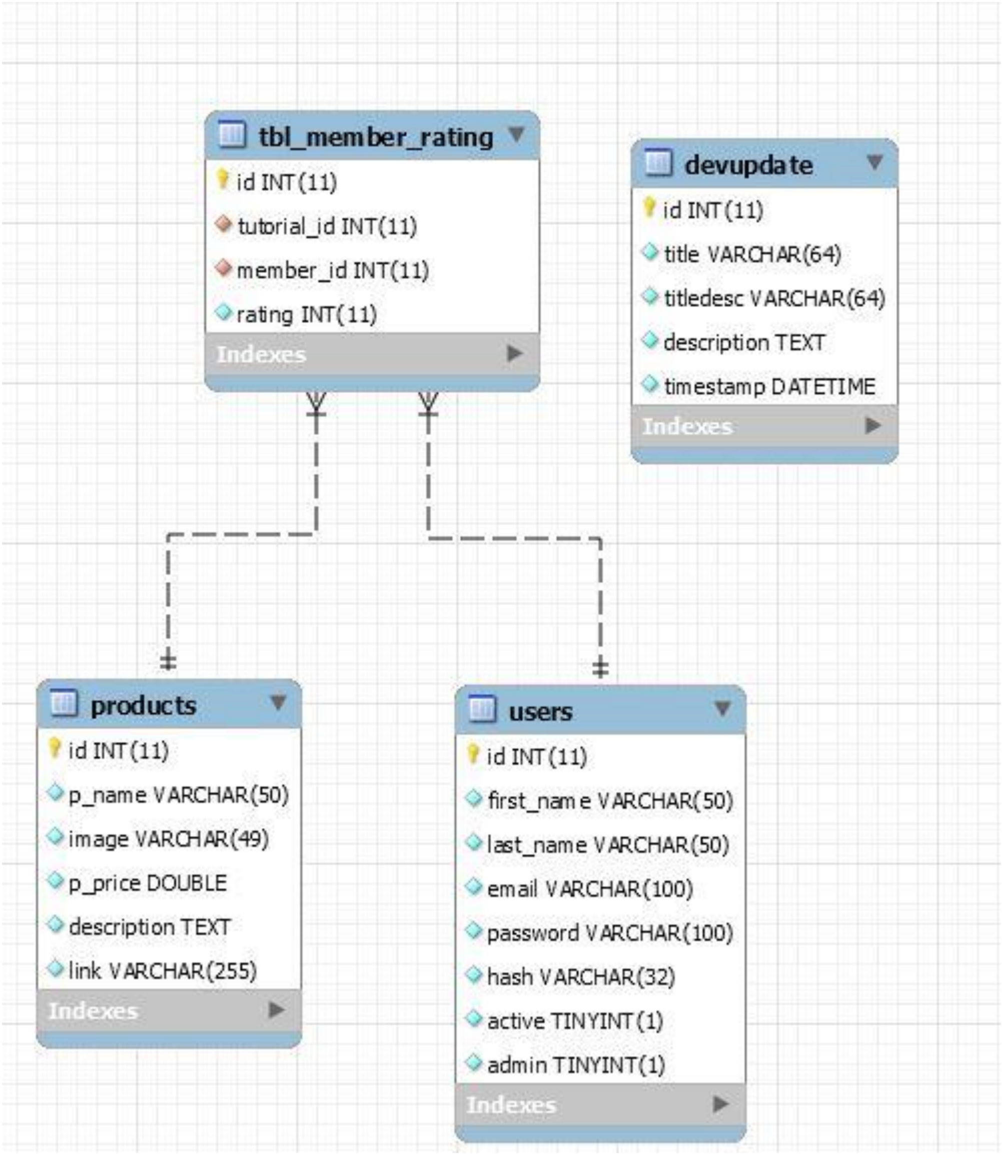


Figure 12- Database ERD Diagram

5.4 Graphical User Interface (GUI) Design

5.4.1 Introduction

GUI allows users to simply interact with the system, the simpler design the easier it allows for interactions between user and system through the web browser.

The GUI Client Side consists of seven main pages described below:

Home – It’s a first page people see after they are logged so it usually pays big role in the website. Home page should be attractive to catch the attention and have simple look for user to know where they are. The homepage will consist of a clear website name, simple design, list of updates and news about the website so members of the website can always know what’s happening around the website.

About – Consist of list of pages the user can go to and describes them individually, about page should tell user what they should expect on the website and what kind of activities they are able to do.

Forum – Leads to a forum website where user is able create account and interact with other user by posting new topics, commenting different topics posted by other users. Member are also allowed to add other members to their friends and send private messages.

Shop – Is the main tool for making money in web pages and it should have simple design. An online store contains large number of products and should be easily modified by admin.

Rating – This is the place where members can rate games. Rating consists of list of games that are stored in database and displayed in store, is contains description and 5-star rating system where members can rate and see average rating of all the games.

Cart – Every e-commerce website should have cart. Whenever members purchase products from store it goes directly to the cart making list of products members is about to purchase before checkout.

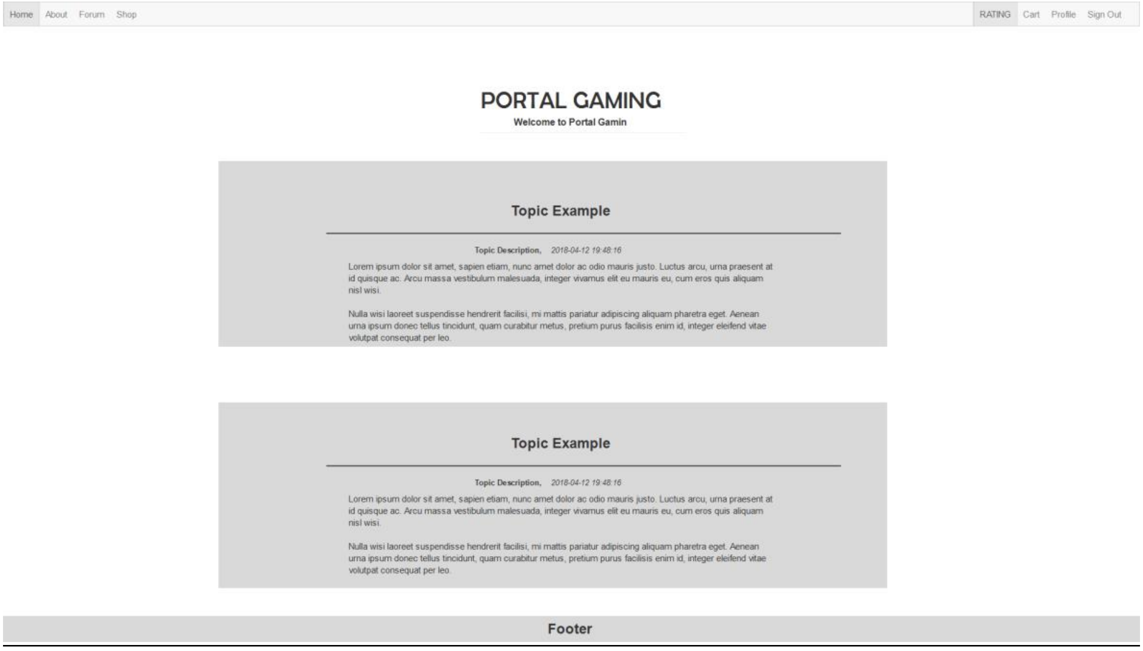
Profile - It displays user info like their e-mail, name and surname and allows user to log out.

Administration Site – After administrator login in with his login details link to management page will be displayed in the navigation bar. Products and news/updates will be managed.

5.4.2 Design of pages

This section will consist of wireframe image for of the pages mentioned in *5.4.1*. Wireframe shows the structural framework for each of the page and make design planning process easier.

<u>Home Page</u>



Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.

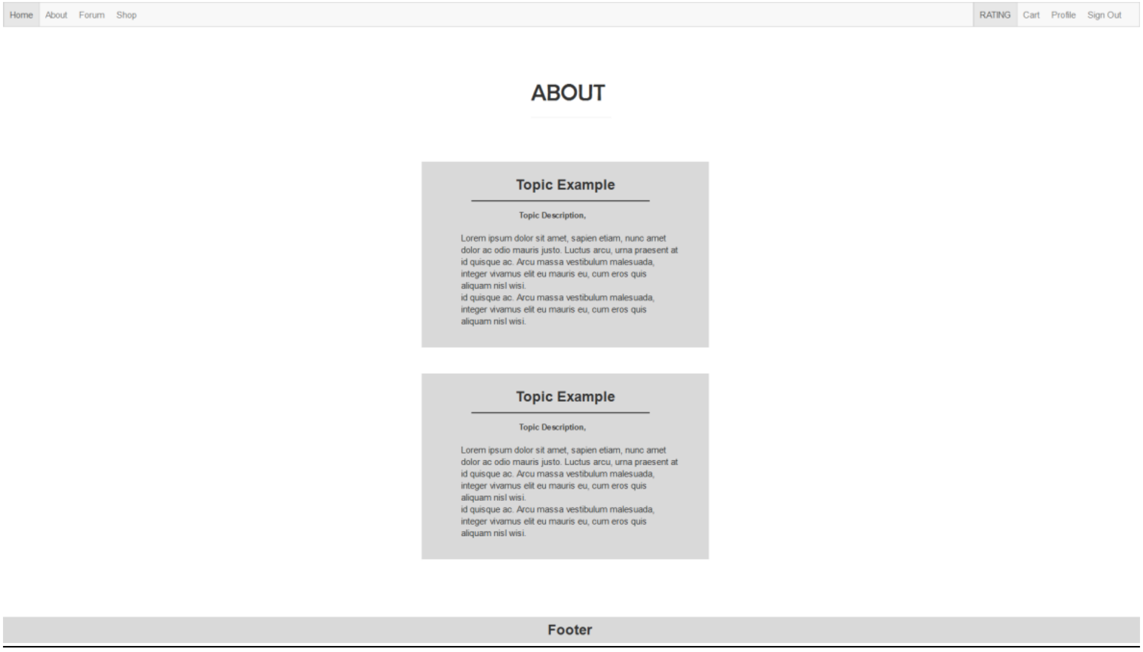
Nulla wisi laoreet suspendisse hendrerit facilisi, mi mattis pariatur adipiscing aliquam pharetra eget. Aenean urna ipsum donec tellus tincidunt, quam curabitur metus, pretium purus facilisis enim id, integer eleifend vitae volutpat consequat per leo.

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.

Nulla wisi laoreet suspendisse hendrerit facilisi, mi mattis pariatur adipiscing aliquam pharetra eget. Aenean urna ipsum donec tellus tincidunt, quam curabitur metus, pretium purus facilisis enim id, integer eleifend vitae volutpat consequat per leo.

Figure 13- Home Page Wireframe

<u>About</u>



Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.
id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.
id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.

Figure 14 - About Page Wireframe

Forum

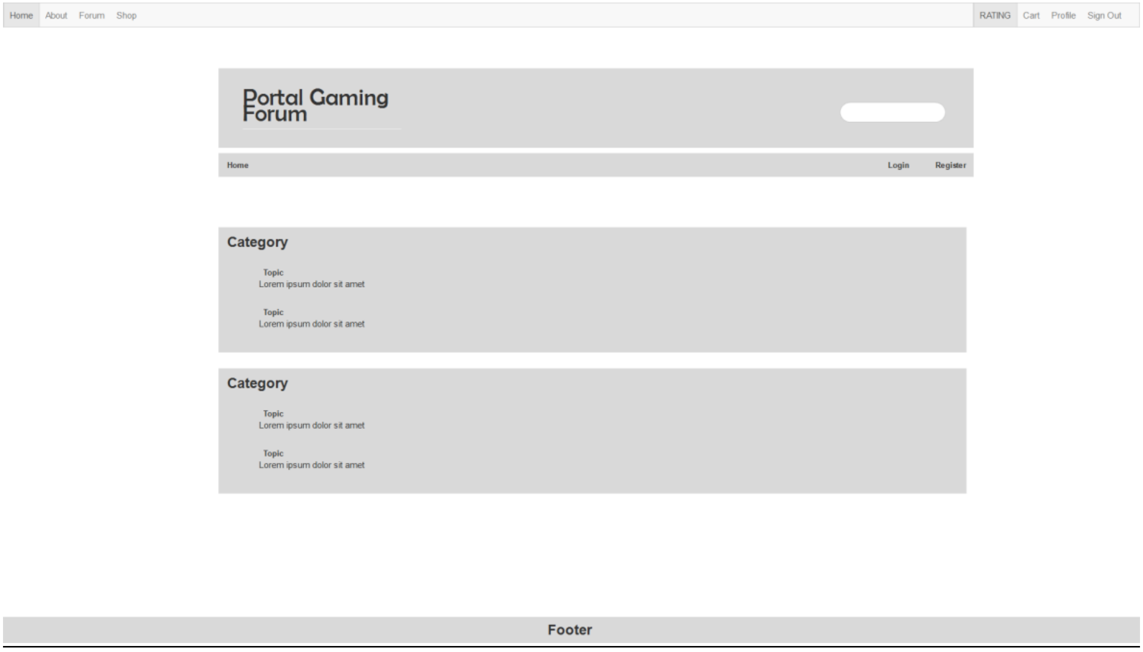


Figure 15 - Forum Page Wireframe

Shop

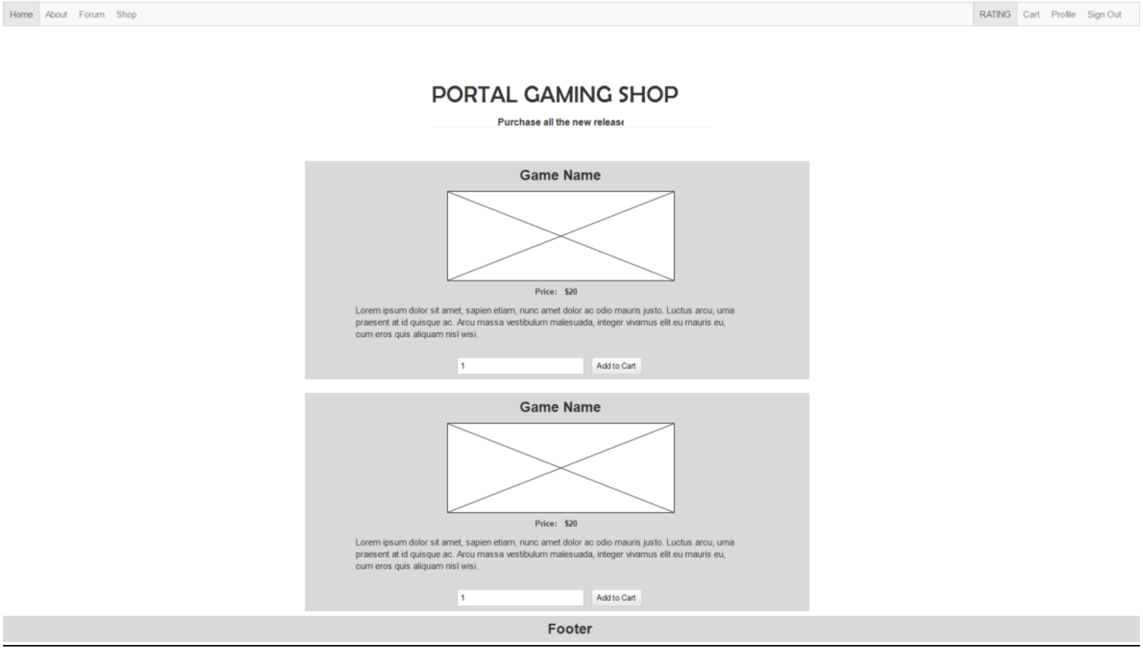
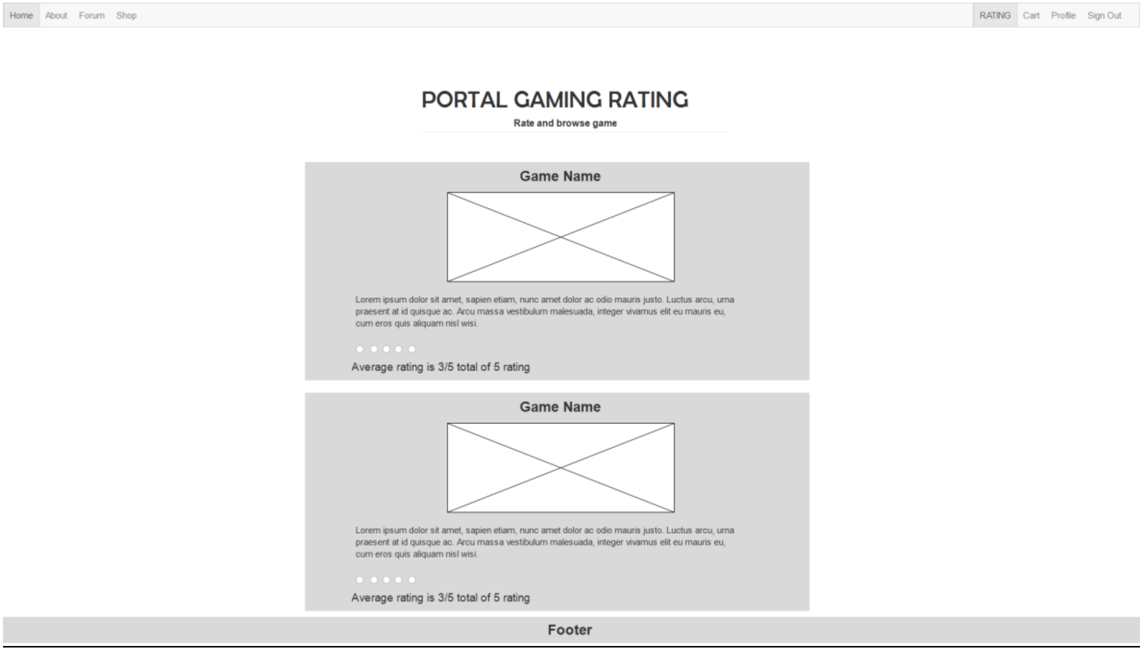

Home
About
Forum
Shop
RATING
Cart
Profile
Sign Out
PORTAL GAMING SHOP
Purchase all the new releas
Game Name
Price: $20
Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.
1
Add to Cart
Game Name
Price: $20
Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.
1
Add to Cart
Footer

Figure 16 - Shop Page Wireframe

Rating



Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.

○ ○ ○ ○ ○

Average rating is 3/5 total of 5 rating

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisl wisi.

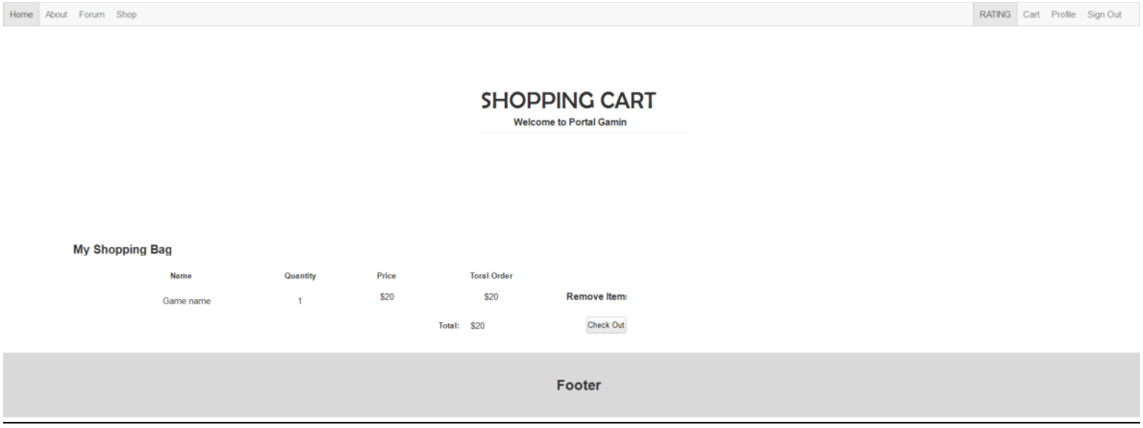
○ ○ ○ ○ ○

Average rating is 3/5 total of 5 rating

Figure 17 - Rating Page Wireframe

Cart

User will be able to browse the product and select them form list, add them to the shopping list, move to check out, at the checkout, user can also remove items form list of products.



My Shopping Bag

Name	Quantity	Price	Toral Order	
Game name	1	$20	$20	Remove Item:
			Total: $20	Check Out

Figure 18 - Cart Page Wireframe

Profile

Profile provides simple information about user.

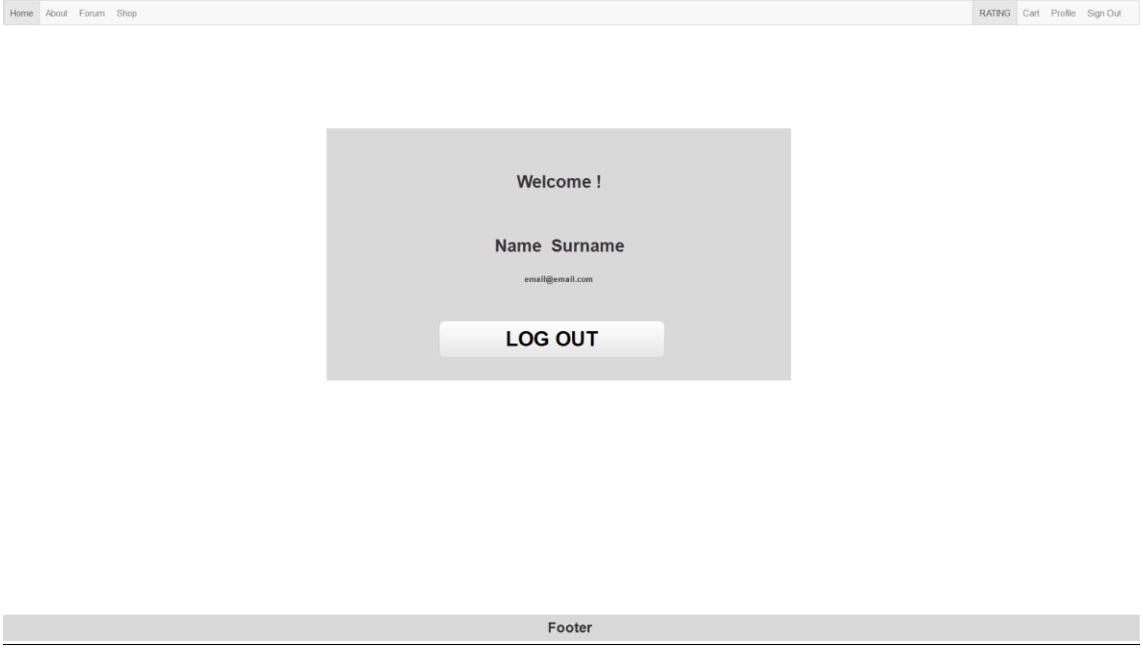
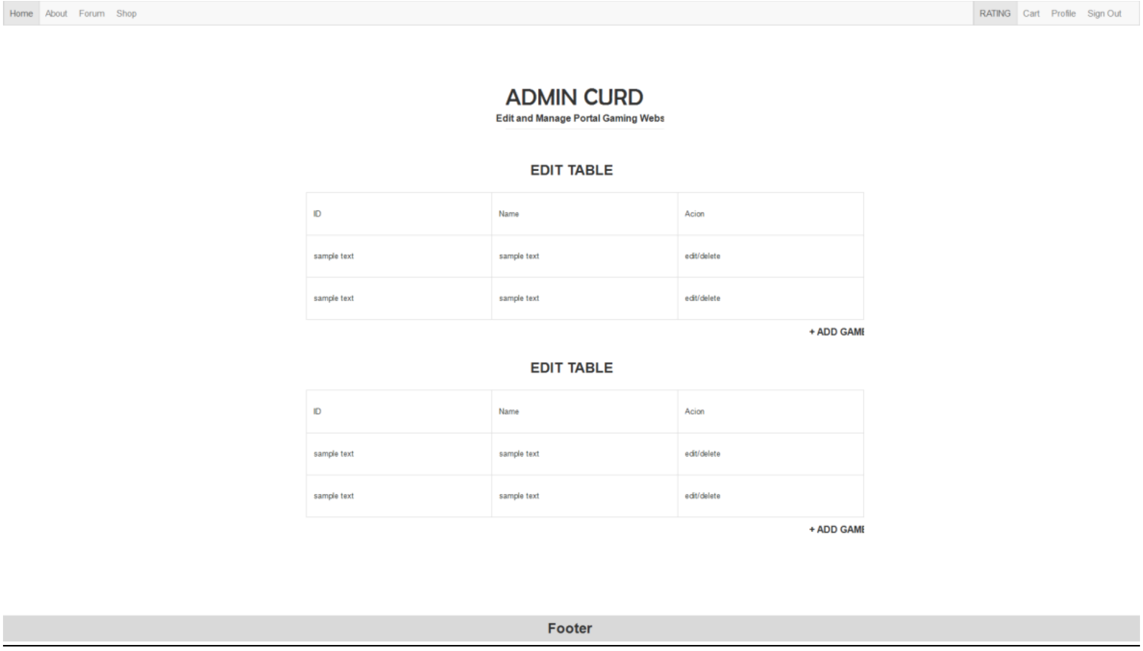


Figure 19 - Profile Page Wireframe

Admin



ID	Name	Acion
sample text	sample text	edit/delete
sample text	sample text	edit/delete

ID	Name	Acion
sample text	sample text	edit/delete
sample text	sample text	edit/delete

Figure 20 - Admin Page Wireframe

5.4.3 Navigation Design

Website will have simple style horizontal navigation bar that will be located at the top of the site with will allow user to navigate user across different pages of the site, as shown in diagram below:

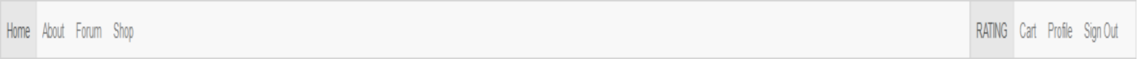


Figure 21- Navigation Bar Wireframe

Footer will contain about, contact link, copyright notice, and button what will allow user to go to the top of the page.

5.5 Dynamic Content Design

5.5.1 Introduction

PHP language will be used to achieve dynamic content design. It's what makes website responsive and provide functionality and interactivity. Allows user to use action like processing form, ordering online, and sending user request to the database.

5.5.2 Forms

PHP will allow the forms to function as it will allow them to send and receive information form database.

<u>Register Form</u>

New users can register by clicking on the register tab that will show then registration form page. Where all the fields need to be completed for database to receive user info.

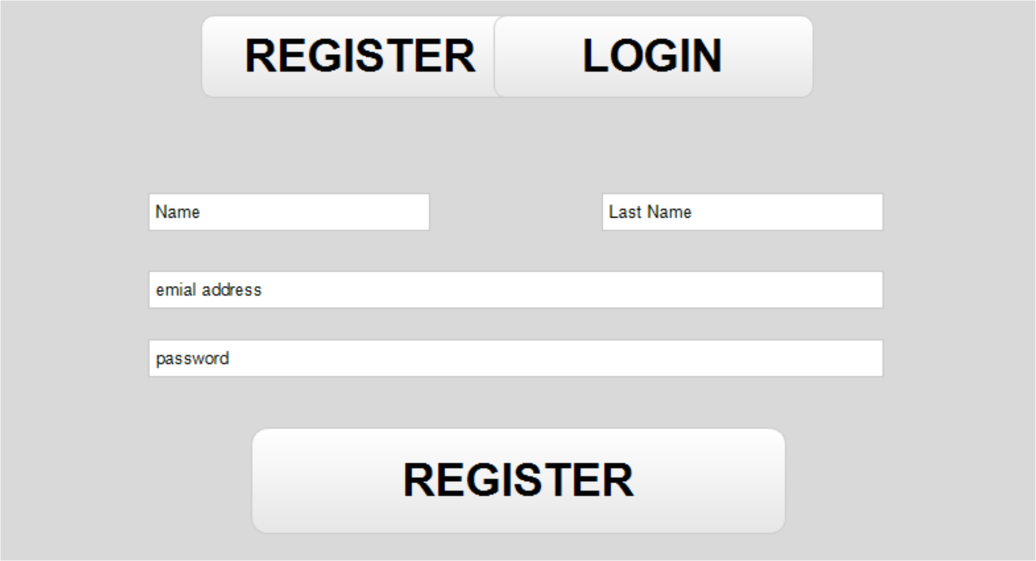


Figure 22- Register Form Wireframe

Login Form

User will be able to login with his details he provided in the registration form by entering his email and password.

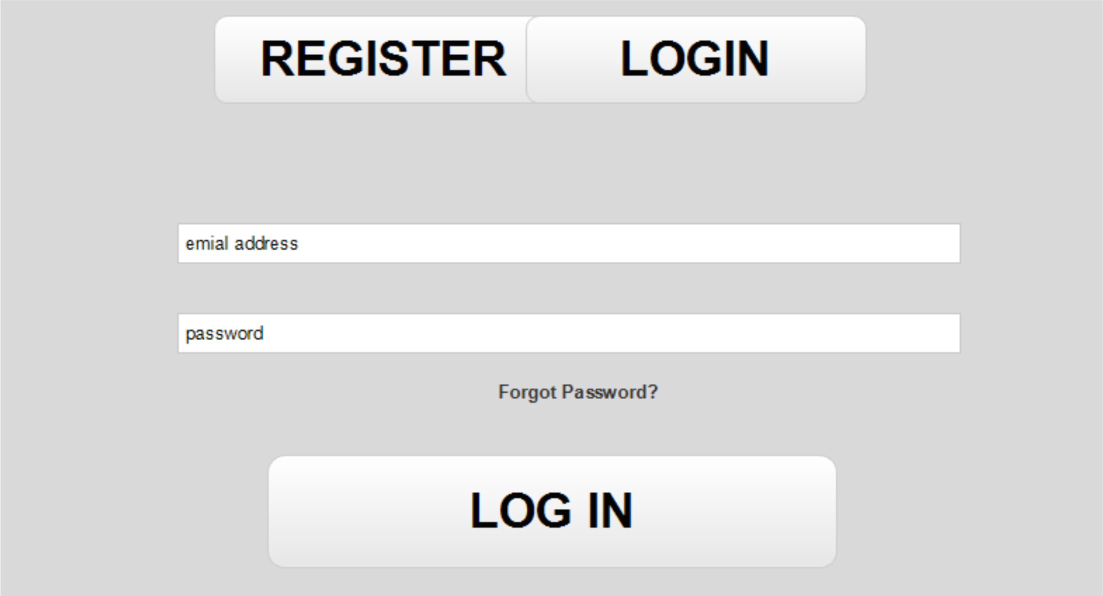
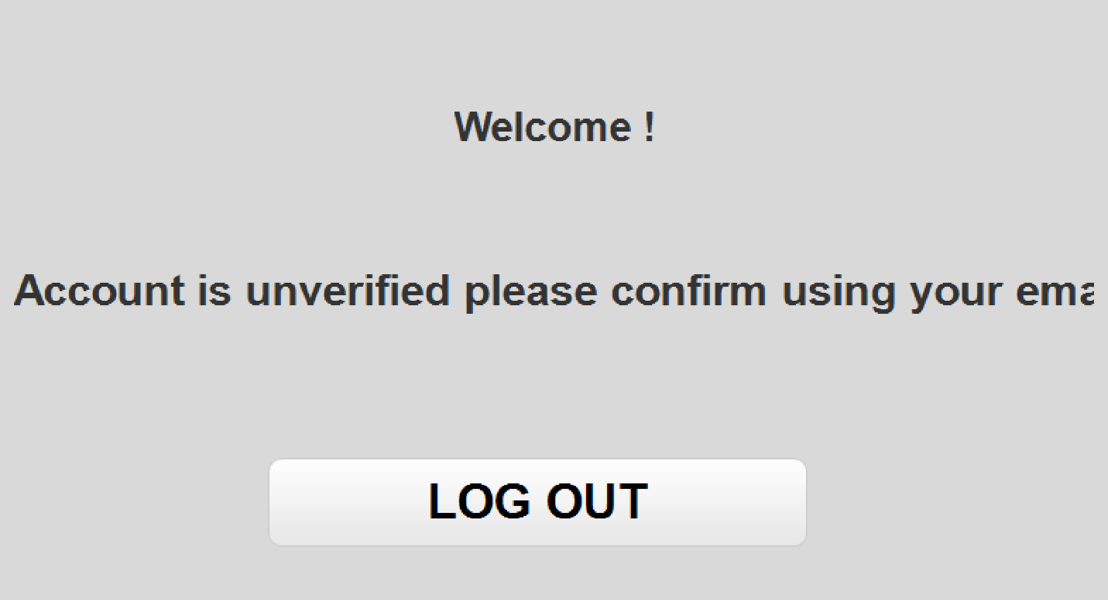


Figure 23 - Login Form Wireframe

Activation Notice

After they log in they will be displayed with notice to activate their account using the email they provided when they registered they account.


Welcome !

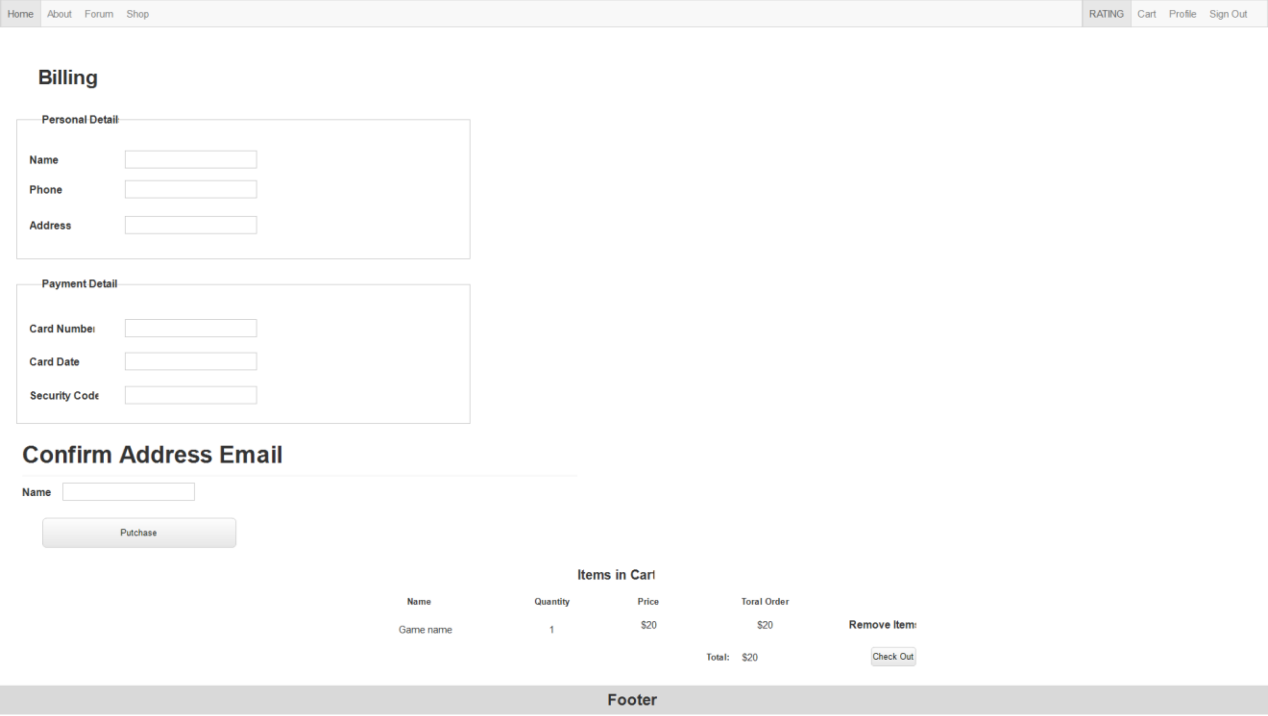
Account is unverified please confirm using your ema

Figure 24 - Activation Notice Wireframe

Billing form

Members who purchase products form the store in order to complete transaction they will be displayed the billing form where they have to complete all the displayed credit card information

After they successfully complete the form they will receive receipt via email confirming that they purchased they items.



Items in Cart

Name	Quantity	Price	Toral Order	
Game name	1	$20	$20	Remove Item:
			Total: $20	Check Out

Figure 25 - Billing Page Wireframe

5.5.3 Administrator Site Functionality

After admin logs in to the website with his login details they will have extra tab "ADMIN CURD" that will lead them to the list of products and news that they are able modify and to add new products, edit existing products and delete them. They will also be able to create new updates on the news page like creating new update, editing existing updates and deleting them.

Chapter 6: Implementation of Prototype

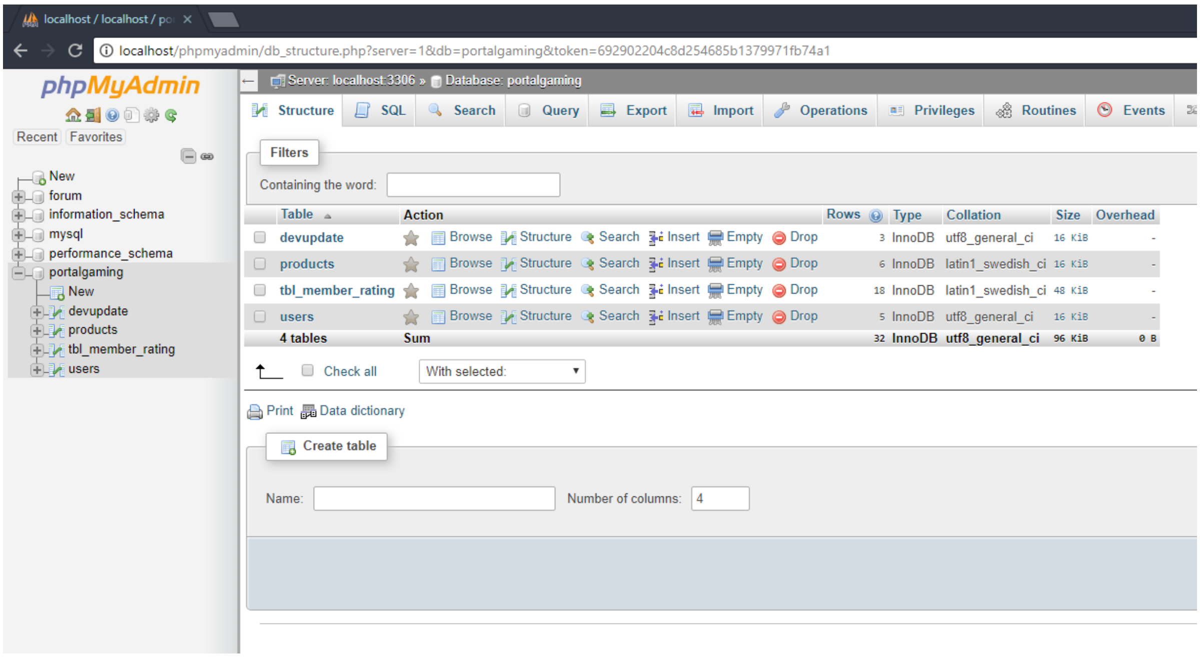
6.1 Introduction

This Chapter will show and explain how the design was implemented and display different development stages and the final appearance. It shows the structure of the database and the dynamic form of the website.

6.2 Database Implementation

Most crucial and one of the first implementations to the website is the database, I made “portalgaming”, all the database the tables were successfully created using “phpMyAdmin”.

Structure of each table is displayed below in the attached screenshots.



localhost/phpmyadmin/db_structure.php?server=1&db=portalgaming&token=692902204c8d254685b1379971fb74a1

Containing the word:

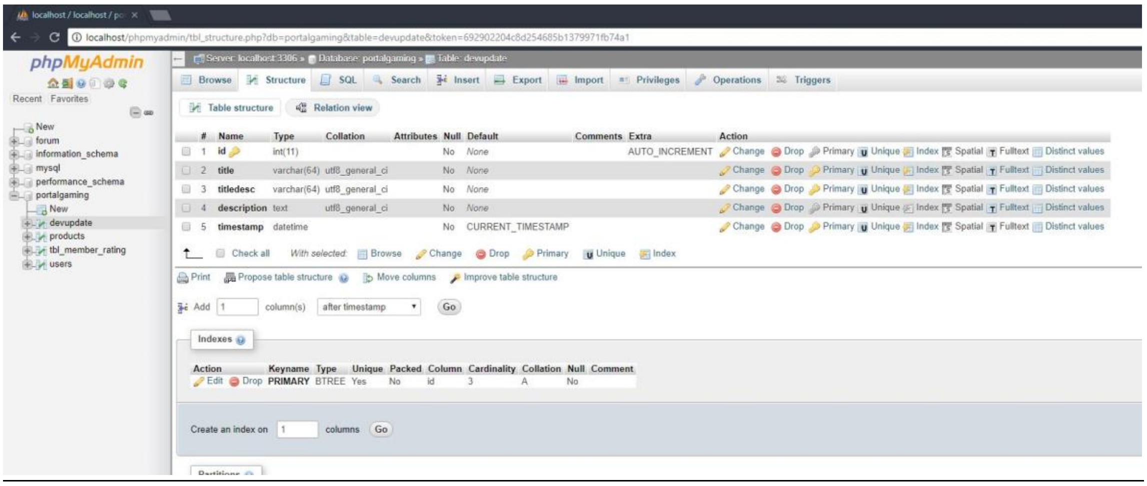
Table	Action	Rows	Type	Collation	Size	Overhead
devupdate	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	16 KiB	-
products	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16 KiB	-
tbl_member_rating	Browse Structure Search Insert Empty Drop	18	InnoDB	latin1_swedish_ci	48 KiB	-
users	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_general_ci	16 KiB	-
4 tables	Sum	32	InnoDB	utf8_general_ci	96 KiB	0 B

Name: Number of columns: 4

Figure 26 - Portalgaming Database

“devupdate” Table

Table that is responsible for the news and updates on home page. It contains information about Title, Title Description, description and the time the update was posted.


localhost / localhost / po
localhost/phpmyadmin/tbl_structure.php?db=portalgaming&table=devupdate&token=692902204c8d254685b1379971fb74a1
phpMyAdmin
Recent Favorites
New
forum
information_schema
mysql
performance_schema
portalgaming
New
devupdate
products
tbl_member_rating
users
Server: localhost:3306 » Database: portalgaming » Table: devupdate
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Table structure Relation view
Check all With selected: Browse Change Drop Primary Unique Index
Print Propose table structure Move columns Improve table structure
Add 1 column(s) after timestamp Go
Indexes
Create an index on 1 columns Go
Partitions

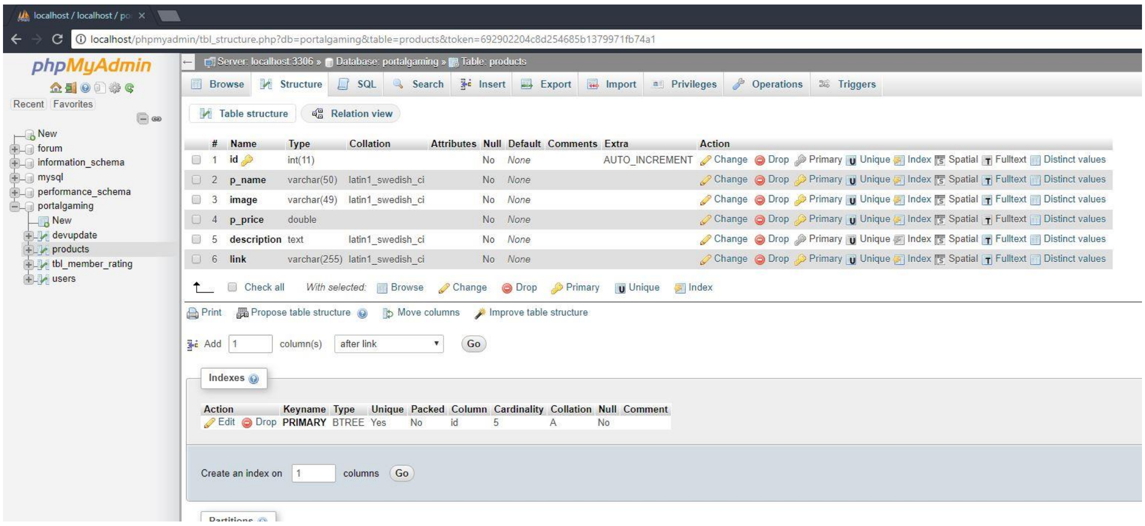
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	2	**title**	varchar(64)	utf8_general_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	3	**titledesc**	varchar(64)	utf8_general_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	4	**description**	text	utf8_general_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	5	**timestamp**	datetime			No	CURRENT_TIMESTAMP			Change Drop Primary Unique Index Spatial Fulltext Distinct values

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	3	A	No	

Figure 27 - devupdate Table Structure

“products” Table

This table is responsible for the products that are located in the store page. It contains data for product name, image, price of the product, description, and link.


localhost / localhost / po
localhost/phpmyadmin/tbl_structure.php?db=portalgaming&table=products&token=692902204c8d254685b1379971fb74a1
phpMyAdmin
Recent Favorites
New
forum
information_schema
mysql
performance_schema
portalgaming
New
devupdate
products
tbl_member_rating
users
Server: localhost:3306 » Database: portalgaming » Table: products
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Table structure Relation view
Check all With selected: Browse Change Drop Primary Unique Index
Print Propose table structure Move columns Improve table structure
Add 1 column(s) after link Go
Indexes
Create an index on 1 columns Go
Partitions

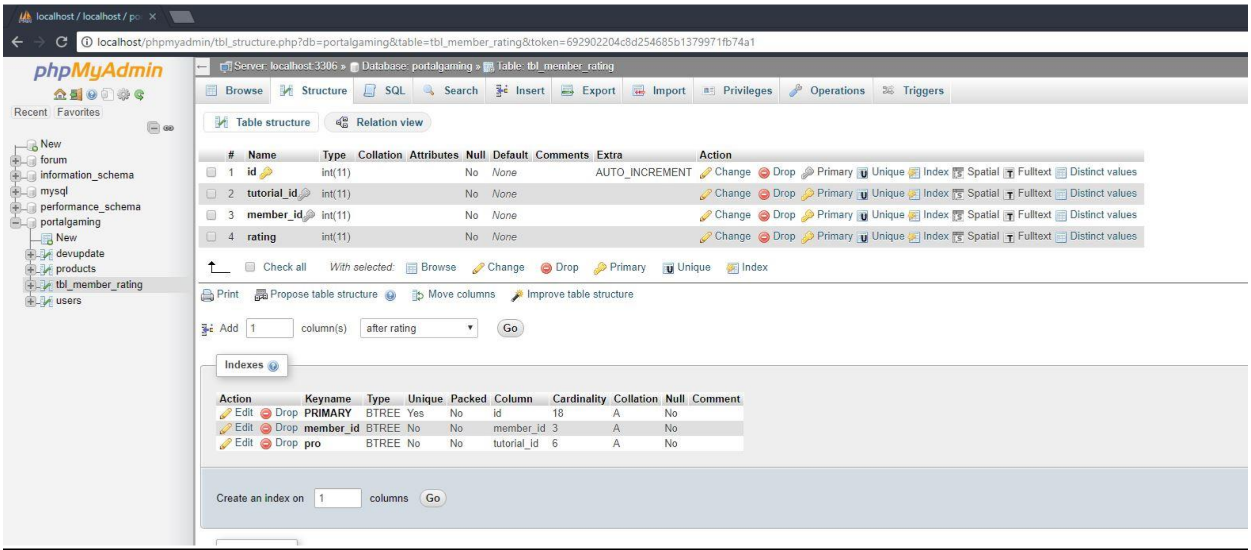
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	2	**p_name**	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	3	**image**	varchar(49)	latin1_swedish_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	4	**p_price**	double			No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	5	**description**	text	latin1_swedish_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
□	6	**link**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	5	A	No	

Figure 28 - products Table Structure

“tbl_memmber_rating” Table

This table is responsible for the rating page. Its stores tutotial_id that stores games, member_id that sores users and rating.



localhost/phpmyadmin/tbl_structure.php?db=portalgaming&table=tbl_member_rating&token=692902204c8d254685b1379971fb74a1

Server: localhost:3306 » Database: portalgaming » Table: tbl_member_rating

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	tutorial_id	int(11)			No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	member_id	int(11)			No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	rating	int(11)			No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values

Add 1 column(s) after rating

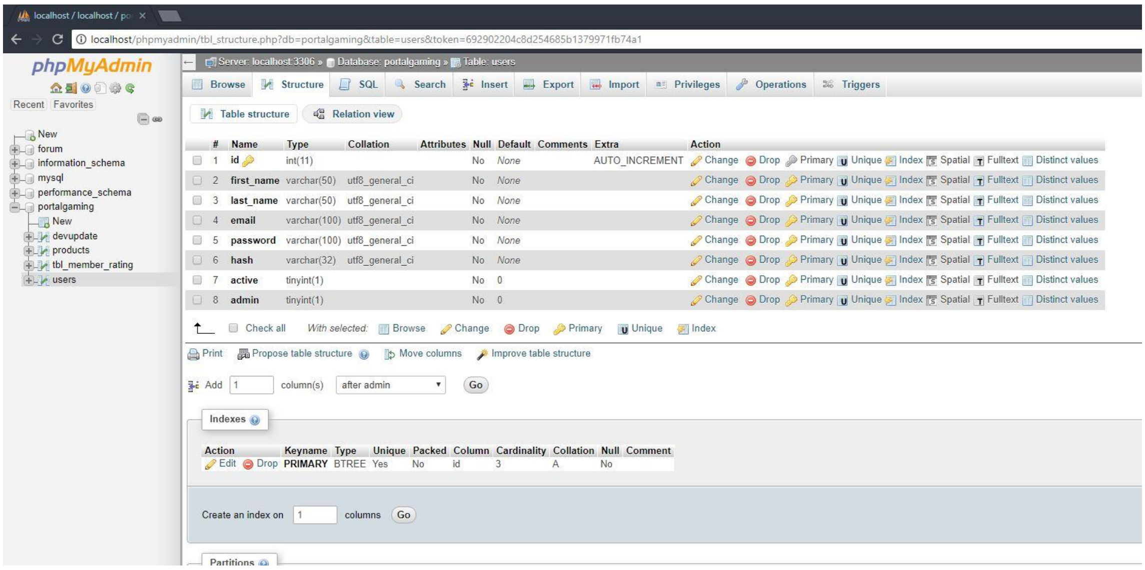
Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	18	A	No	
Edit Drop	member_id	BTREE	No	No	member_id	3	A	No	
Edit Drop	pro	BTREE	No	No	tutorial_id	6	A	No	

Create an index on 1 columns

Figure 29 - rating Table Structure

“users” Table

This table sores user data which allows user to create account and login to the website. It also allows for admin login.



localhost/phpmyadmin/tbl_structure.php?db=portalgaming&table=users&token=692902204c8d254685b1379971fb74a1

Server: localhost:3306 » Database: portalgaming » Table: users

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	first_name	varchar(50)	utf8_general_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	last_name	varchar(50)	utf8_general_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	email	varchar(100)	utf8_general_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	password	varchar(100)	utf8_general_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
6	hash	varchar(32)	utf8_general_ci		No	*None*			Change Drop Primary Unique Index Spatial Fulltext Distinct values
7	active	tinyint(1)			No	0			Change Drop Primary Unique Index Spatial Fulltext Distinct values
8	admin	tinyint(1)			No	0			Change Drop Primary Unique Index Spatial Fulltext Distinct values

Add 1 column(s) after admin

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	3	A	No	

Create an index on 1 columns

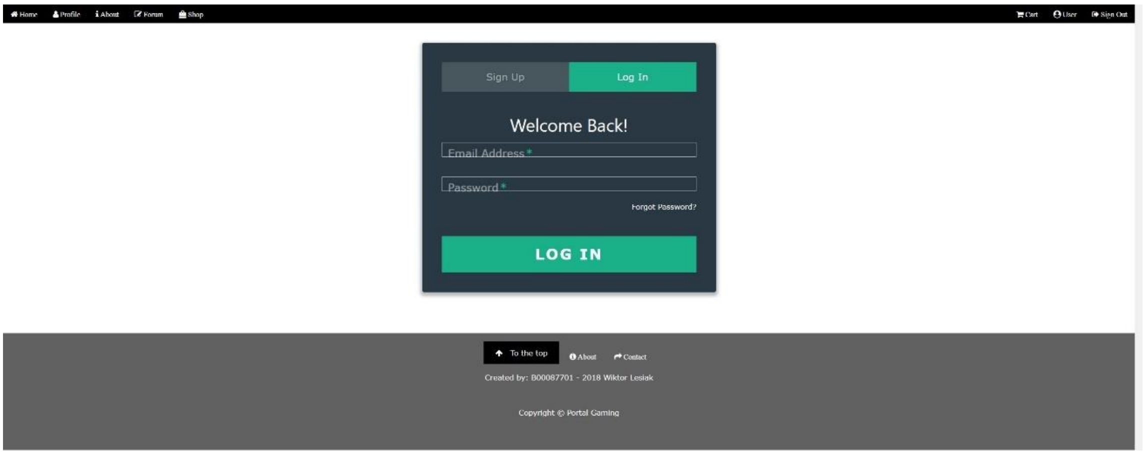
Figure 30 - users Table Structure

6.3 Graphical User Interface

These results are attained for Graphical User Interface using HTML, CSS and PHP where static content was written with HTML and CSS each of the pages are made with three sections achieved with html and CSS: header, footer, and middle section it allows to only change middle section

6.4 Forms

6.4.1 Login Page

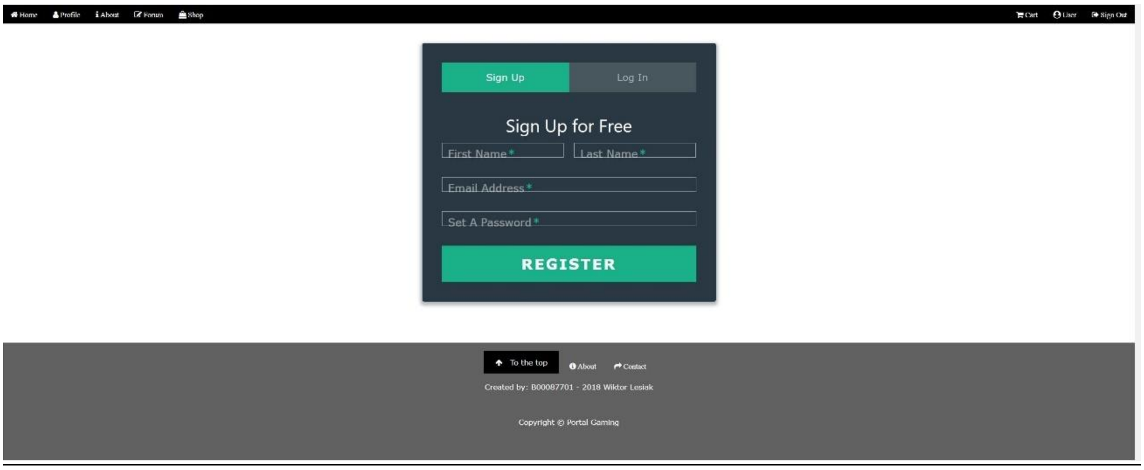


Created by: B00087701 - 2018 Wiktor Lesiak

Copyright © Portal Gaming

Figure 31 - Login page GUI

6.4.2 Register Page

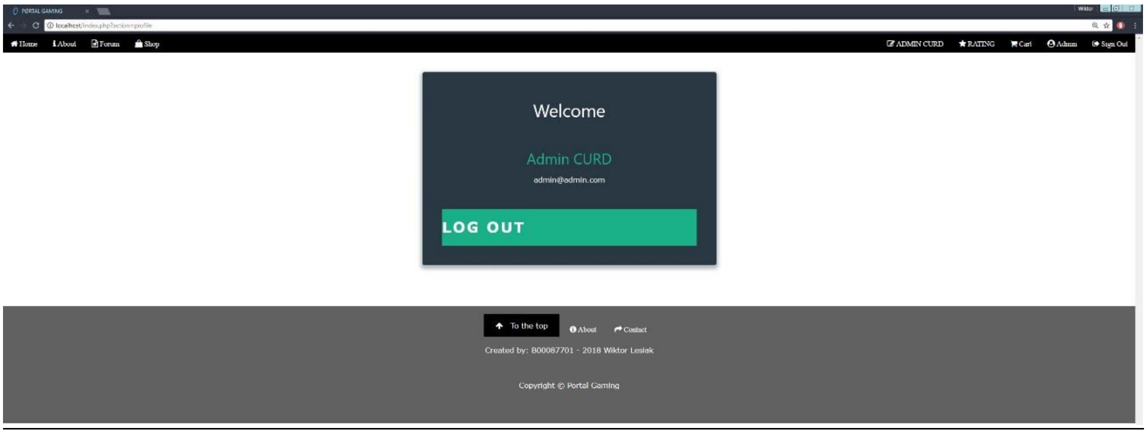


Created by: B00087701 - 2018 Wiktor Lesiak

Copyright © Portal Gaming

Figure 32 - Register GUI

6.4.3 Profile



Welcome

Admin CURD

admin@admin.com

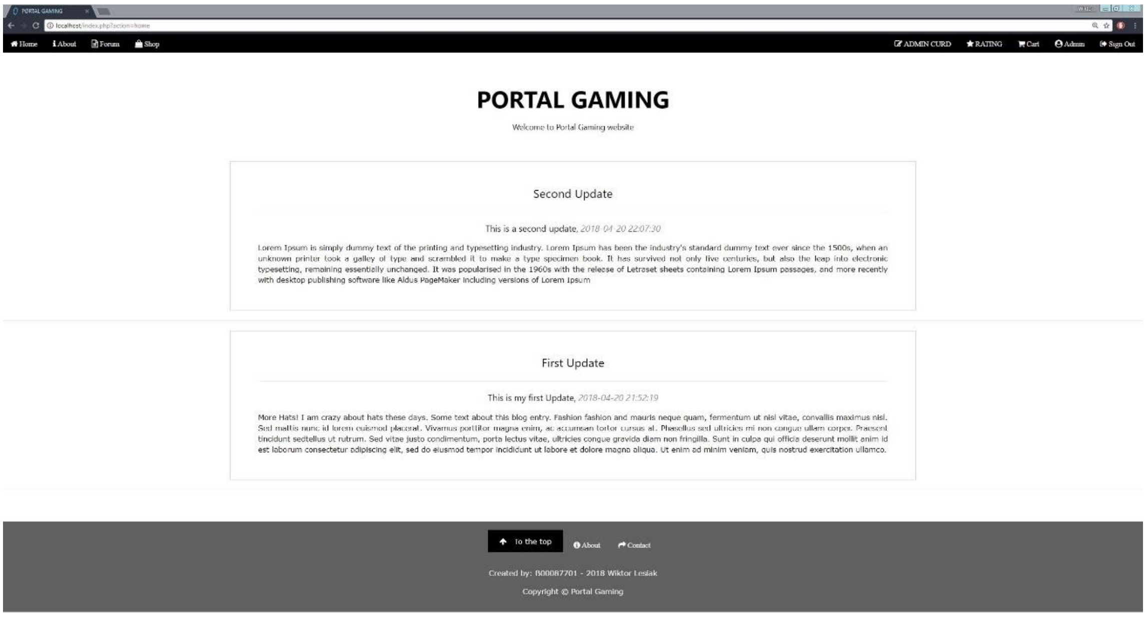
Created by: B00087701 - 2018 Wiktor Lesiak

Copyright © Portal Gaming

Figure 33 - User Profile GUI

6.5 Display Pages

6.5.1 Homepage



PORTAL GAMING

Welcome to Portal Gaming website

Second Update

This is a second update, 2018-04-20 22:07:30

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum

First Update

This is my first Update, 2018-04-20 21:52:19

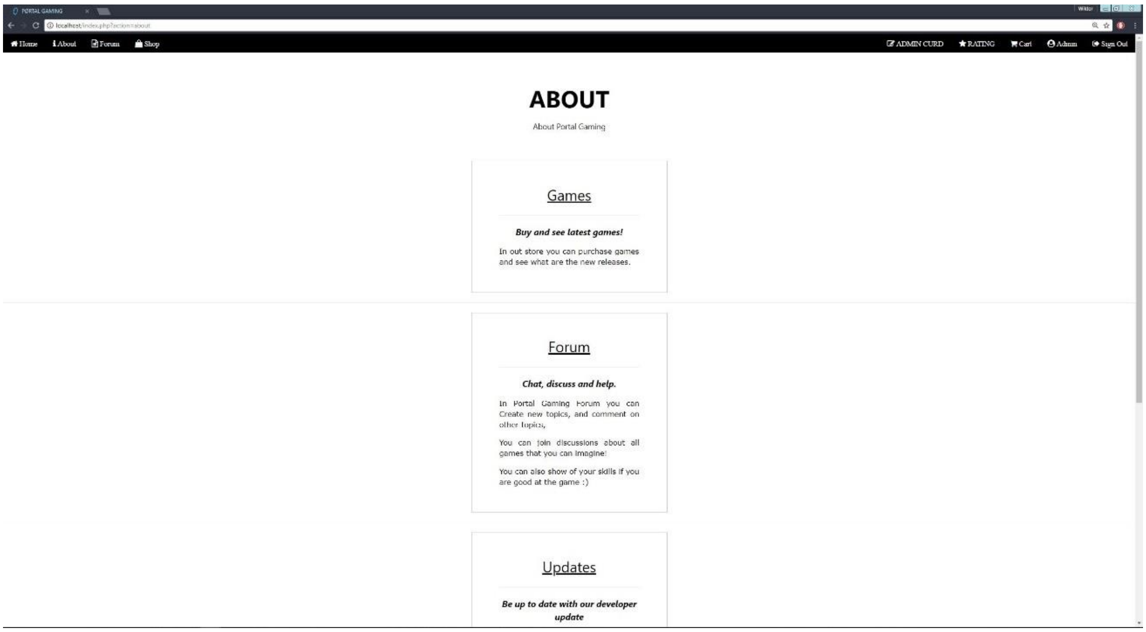
More Hats! I am crazy about hats these days. Some text about this blog entry. Fashion fashion and mauris neque quam, fermentum ut nisi vitae, convallis maximus nisl. Sed mattis nunc id lorem euismod placerat. Vivamus porttitor magna enim, ac accumsan tortor cursus at. Phasellus sed ultricies mi non congue ullam corper. Praesent tincidunt sedtellus ut rutrum. Sed vitae justo condimentum, porta lectus vitae, ultricies congue gravida diam non fringilla. Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

Created by: B00087701 - 2018 Wiktor Lesiak

Copyright © Portal Gaming

Figure 34 – Homepage GUI

6.5.2 About



ABOUT

About Portal Gaming

Games

Buy and see latest games!

In out store you can purchase games and see what are the new releases.

Forum

Chat, discuss and help.

In Portal Gaming Forum you can Create new topics, and comment on other topics,

You can join discussions about all games that you can imagine!

You can also show of your skills if you are good at the game :)

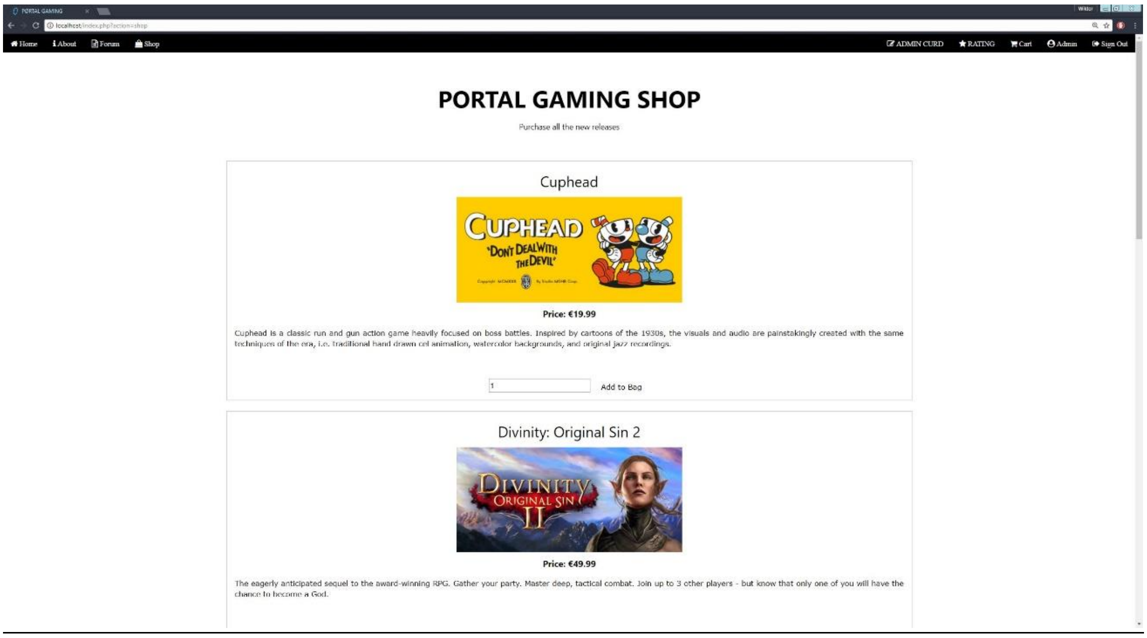
Updates

Be up to date with our developer update

Figure 35 - About GUI

6.6 Shop

Users will be able to purchase games from the displayed list of games and finish the purchase in the cart with the checkout button.



PORTAL GAMING SHOP

Purchase all the new releases

Cuphead

Price: €19.99

Cuphead is a classic run and gun action game heavily focused on boss battles. Inspired by cartoons of the 1930s, the visuals and audio are painstakingly created with the same techniques of the era, i.e. traditional hand drawn cel animation, watercolor backgrounds, and original jazz recordings.

1

Divinity: Original Sin 2

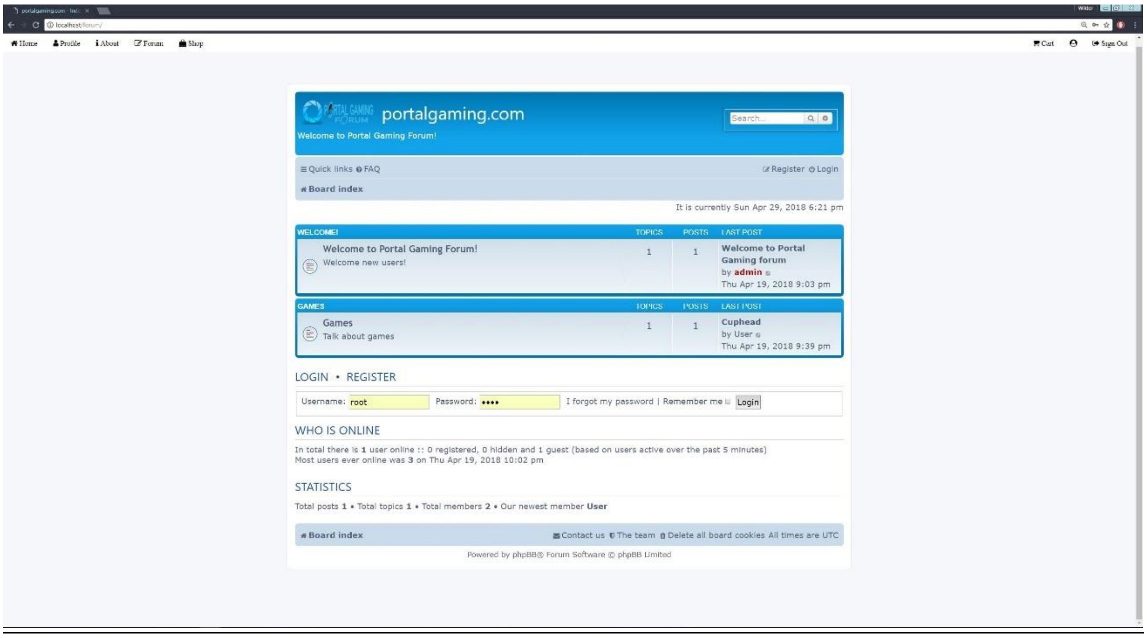
Price: €49.99

The eagerly anticipated sequel to the award-winning RPG. Gather your party. Master deep, tactical combat. Join up to 3 other players - but know that only one of you will have the chance to become a God.

Figure 36 - Shop GUI

6.7 Forum

Forum is implemented using phpBB (PHP Bulletin Board), it’s an open-source forum which include features like, private message system, search engine, user preferences, moderation and administration.



portalgaming.com

Welcome to Portal Gaming Forum!

It is currently Sun Apr 29, 2018 6:21 pm

WELCOME!	TOPICS	POSTS	LAST POST
Welcome to Portal Gaming Forum! Welcome new users!	1	1	Welcome to Portal Gaming forum by admin Thu Apr 19, 2018 9:03 pm

GAMES	TOPICS	POSTS	LAST POST
Games Talk about games	1	1	Cuphead by User Thu Apr 19, 2018 9:39 pm

LOGIN • REGISTER

Username: root Password: •••• I forgot my password | Remember me □ Login

WHO IS ONLINE

In total there is **1** user online :: 0 registered, 0 hidden and 1 guest (based on users active over the past 5 minutes)
Most users ever online was **3** on Thu Apr 19, 2018 10:02 pm

STATISTICS

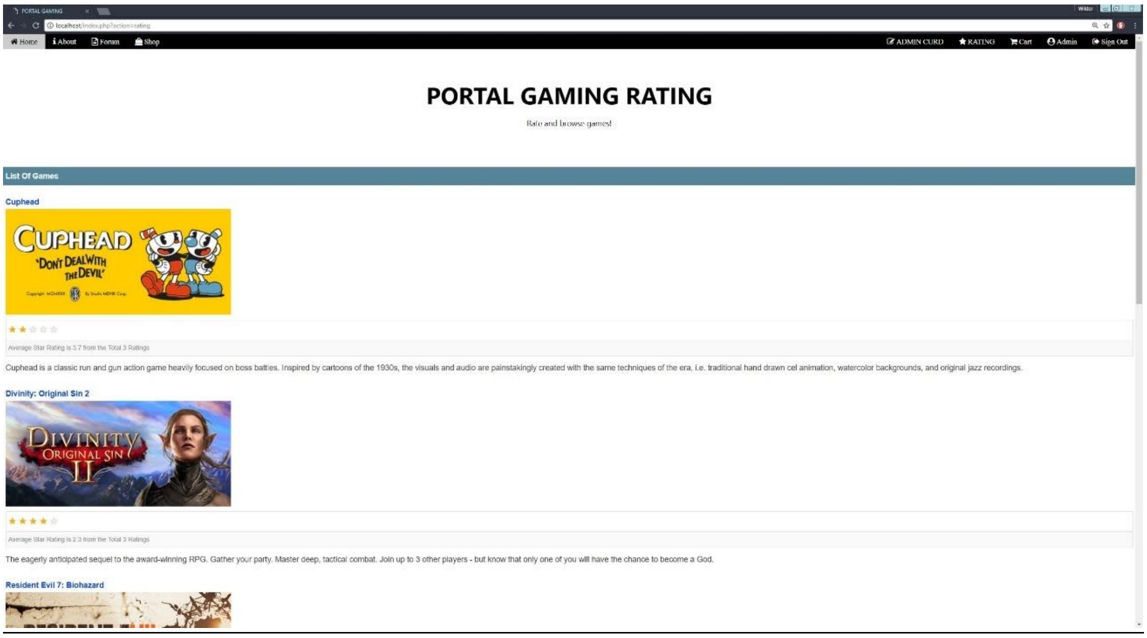
Total posts **1** • Total topics **1** • Total members **2** • Our newest member **User**

Powered by phpBB® Forum Software © phpBB Limited

Figure 37 - Forum GUI

6.8 Rating

Rating will display list of games with picture, description and average rating. Only logged in users can rate the games.



PORTAL GAMING RATING

Rate and browse games!

List Of Games

Cuphead

★★☆☆☆

Average Star Rating is 3.7 from the Total 3 Ratings

Cuphead is a classic run and gun action game heavily focused on boss battles. Inspired by cartoons of the 1930s, the visuals and audio are painstakingly created with the same techniques of the era, i.e. traditional hand drawn cel animation, watercolor backgrounds, and original jazz recordings.

Divinity: Original Sin 2

★★★★☆

Average Star Rating is 2.3 from the Total 3 Ratings

The eagerly anticipated sequel to the award-winning RPG. Gather your party. Master deep, tactical combat. Join up to 3 other players - but know that only one of you will have the chance to become a God.

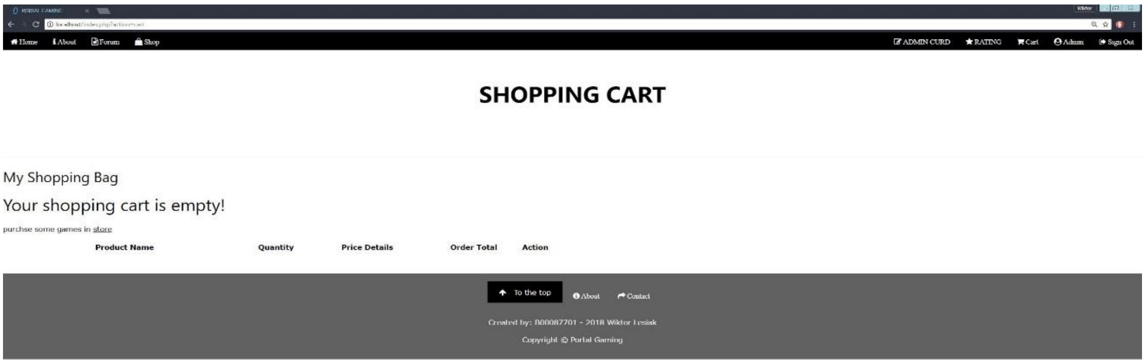
Resident Evil 7: Biohazard

Figure 38 - Rating GUI

6.9 Shopping Cart

Shopping cart will display message “Your shopping cart is empty!” if no items are put in the cart, but then they are items in the cart they will display all the information, total price of the purchase and the checkout button .

6.9.1 Shopping Cart empty



SHOPPING CART

My Shopping Bag

Your shopping cart is empty!

purchse some games in store

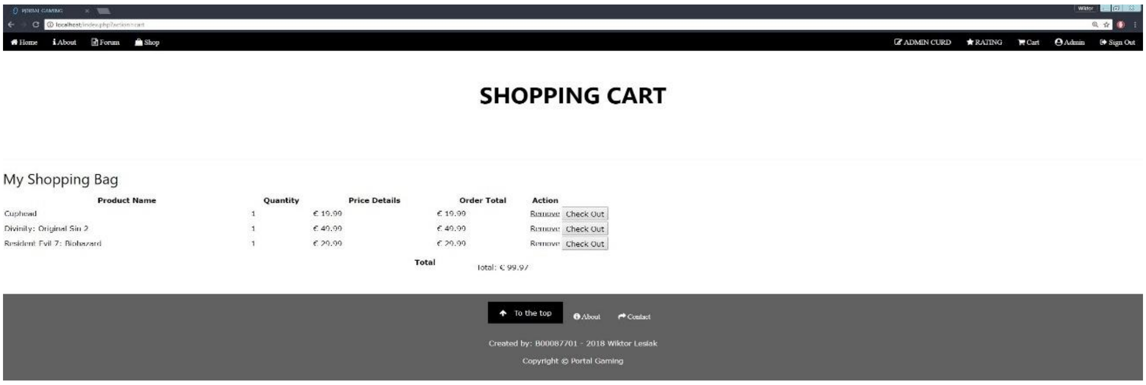
Product Name	Quantity	Price Details	Order Total	Action

Created by: B00087701 - 2018 Wiktor Lesiak

Copyright © Portal Gaming

Figure 39 - Shopping Cart Empty

6.9.2 Shopping Cart with items



SHOPPING CART

My Shopping Bag

Product Name	Quantity	Price Details	Order Total	Action
Cuphead	1	€ 19.99	€ 19.99	Remove Check Out
Divinity: Original Sin 2	1	€ 40.99	€ 40.99	Remove Check Out
Resident Evil 7: Biohazard	1	€ 29.99	€ 29.99	Remove Check Out
		Total	Total: € 99.97	

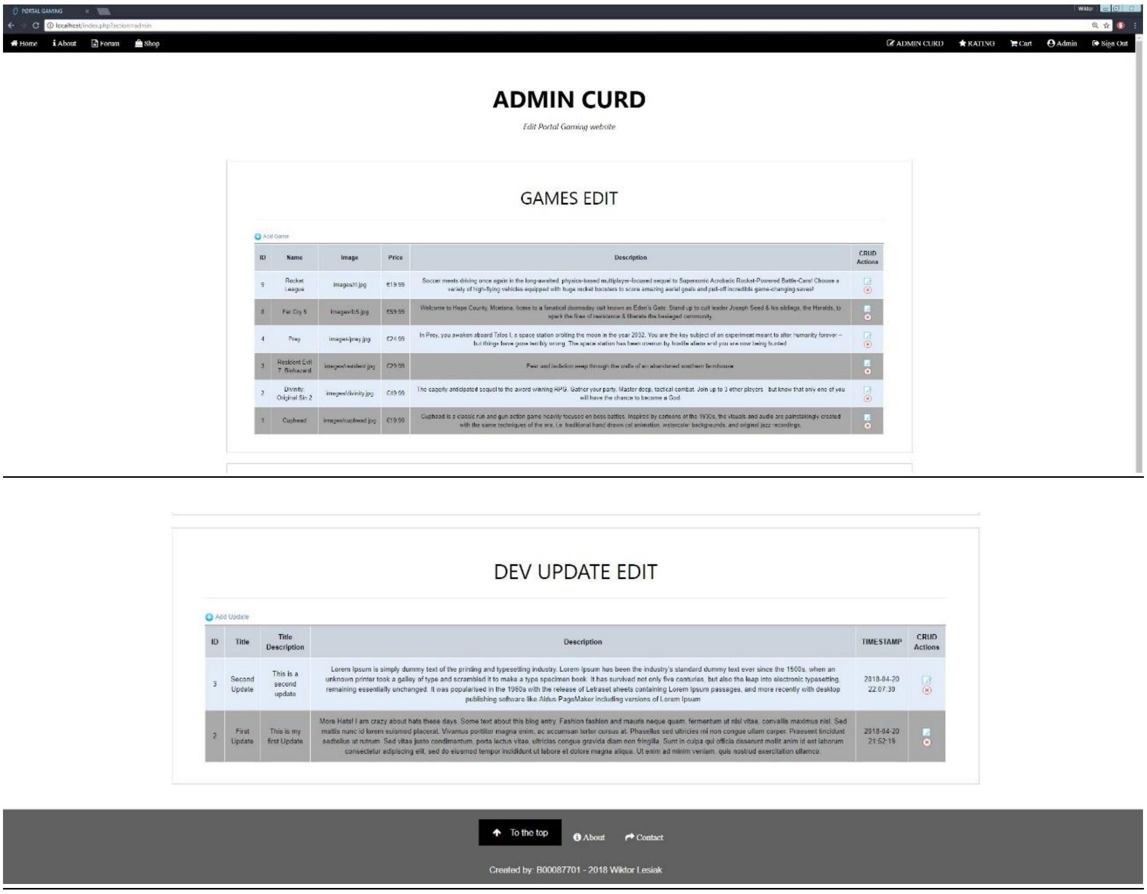
Created by: B00087701 - 2018 Wiktor Lesiak

Copyright © Portal Gaming

Figure 40 - Shopping Cart with items

6.10 Administration site page

Admin can log in to the system with his username and password. The ADMIN CURD tab in navigation bar, once admin opens the admin curd the games edit, and dev update edit control panels are displayed. Admin can view all the products available on the website, add, edit and delete all products that are stored in database.



ADMIN CURD

Edit Portal Gaming website

GAMES EDIT

ID	Name	Image	Price	Description	CRUD Actions
5	Rocket League	images/rl.jpg	€19.99	Soccer meets driving once again in the long-awaited, physics-based multiplayer-focused sequel to Supersonic Acrobatic Rocket-Powered Battle-Cars! Choose a variety of high-flying vehicles equipped with huge rocket boosters to score amazing aerial goals and pull-off incredible game-changing saves!	
8	Far Cry 5	images/fc5.jpg	€59.99	Welcome to Hope County, Montana, home to a fanatical doomsday cult known as Eden's Gate. Stand up to cult leader Joseph Seed & his siblings, the Heralds, to spark the fires of resistance & liberate the besieged community.	
4	Prey	images/prey.jpg	€24.99	In Prey, you awaken aboard Talos I, a space station orbiting the moon in the year 2032. You are the key subject of an experiment meant to alter humanity forever – but things have gone terribly wrong. The space station has been overrun by hostile aliens and you are now being hunted.	
3	Resident Evil 7: Biohazard	images/resident.jpg	€29.99	Fear and isolation seep through the walls of an abandoned southern farmhouse.	
2	Divinity: Original Sin 2	images/divinity.jpg	€49.99	The eagerly anticipated sequel to the award winning RPG. Gather your party. Master deep, tactical combat. Join up to 3 other players - but know that only one of you will have the chance to become a God.	
1	Cuphead	images/cuphead.jpg	€19.99	Cuphead is a classic run and gun action game heavily focused on boss battles. Inspired by cartoons of the 1930s, the visuals and audio are painstakingly created with the same techniques of the era, i.e. traditional hand drawn cel animation, watercolor backgrounds, and original jazz recordings.	

DEV UPDATE EDIT

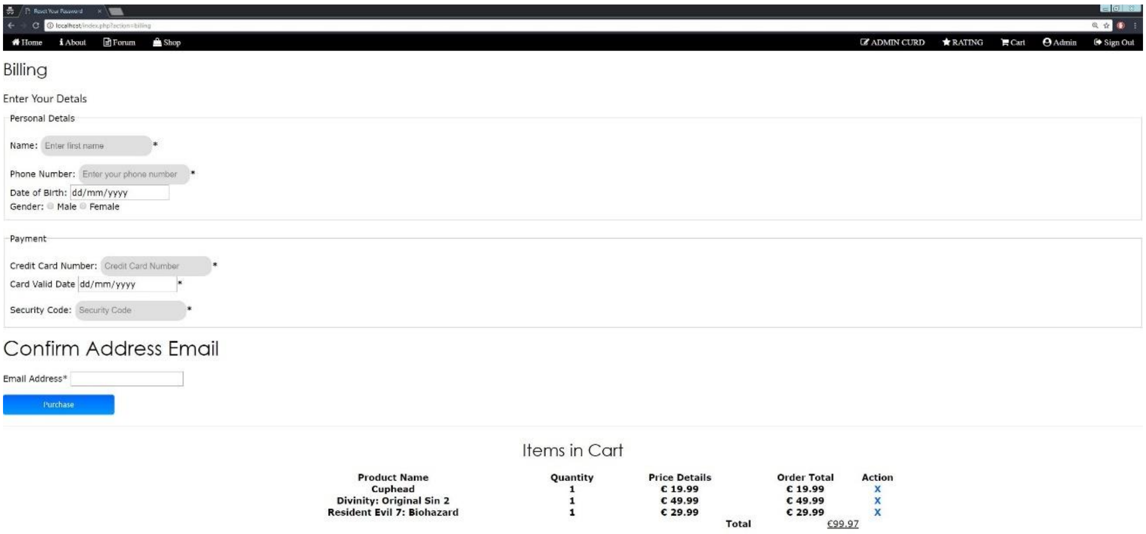
ID	Title	Title Description	Description	TIMESTAMP	CRUD Actions
3	Second Update	This is a second update	Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum	2018-04-20 22:07:30	
2	First Update	This is my first Update	More Hats! I am crazy about hats these days. Some text about this blog entry. Fashion fashion and mauris neque quam, fermentum ut nisl vitae, convallis maximus nisl. Sed mattis nunc id lorem euismod placerat. Vivamus porttitor magna enim, ac accumsan tortor cursus at. Phasellus sed ultricies mi non congue ullam corper. Praesent tincidunt sedtellus ut rutrum. Sed vitae justo condimentum, porta lectus vitae, ultricies congue gravida diam non fringilla. Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.	2018-04-20 21:52:19	

Created by: B00087701 - 2018 Wiktor Lesiak

Figure 41 - ADMIN GUI

6.11 Billing

In this project I will simulate the cart verification. The customer will be asked to input his card details which don’t have to be valid. To complete the checkout user must type email they logged in with, if the email is wrong they will get an error, if email is correct they will receive receipt to their email.



Billing

Enter Your Detals

Personal Detals

Name: Enter first name *

Phone Number: Enter your phone number *

Date of Birth: dd/mm/yyyy

Gender: ○ Male ○ Female

Payment

Credit Card Number: Credit Card Number *

Card Valid Date dd/mm/yyyy *

Security Code: Security Code *

Confirm Address Email

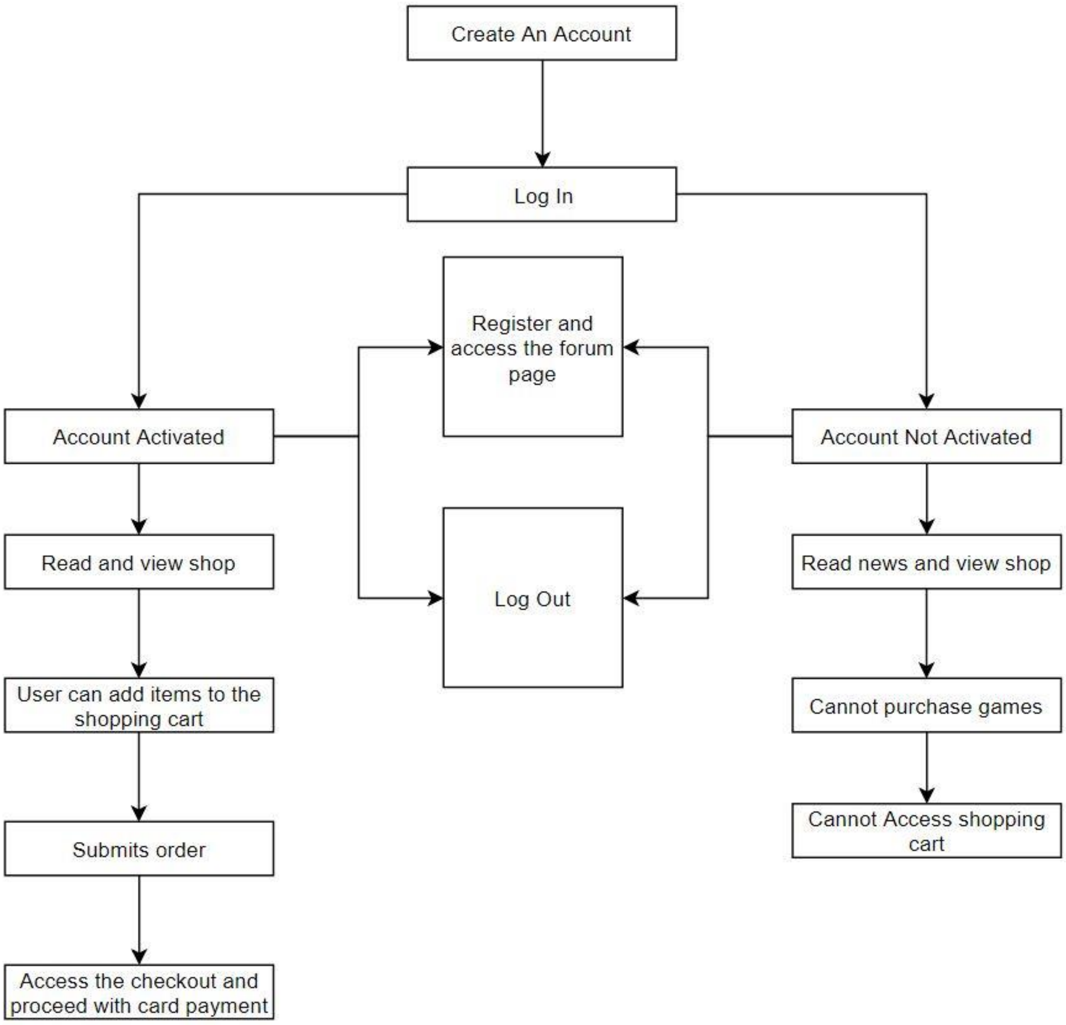
Email Address*

Items in Cart

Product Name	Quantity	Price Details		Order Total	Action
Cuphead	1	€ 19.99		€ 19.99	X
Divinity: Original Sin 2	1	€ 49.99		€ 49.99	X
Resident Evil 7: Biohazard	1	€ 29.99		€ 29.99	X
			Total	€99.97	

Chapter 7: Testing and evaluation

Testing of this website was successful and performed on both sides according to the plan and there was not issues with the website. I’ve created the diagram to test the website functionalities that will allow to detect errors easier in the future, it describes how user gather the required information and how does each stage of the website work with give a simple explanation of the processing of the website.


Create An Account
Log In
Account Activated
Register and access the forum page
Log Out
Account Not Activated
Read and view shop
User can add items to the shopping cart
Submits order
Access the checkout and proceed with card payment
Read news and view shop
Cannot purchase games
Cannot Access shopping cart

Chapter 8: Conclusions and Further Work

8.1 Achieved results

This e-commerce website project was undertaking to successfully design to implement a fully functional and dynamic website. In previous chapters I displayed the results and planning of this project and how the desired goal was achieved. I managed to create the online shop, with news and implemented phpbb forum and administration management page will all the different functionalities. Looking back through different stages and problems during the implementation process I was pleased with the gained results.

8.2 Future Work

The e-commerce website can be implemented with large number of products that can be added anytime, but the shop website can benefit from implementation of online cat system, trading system, wish list product, recommended products, and lots of different marketing tools.

Its always possible to improve a website starting with design and modifying the functionality of the website There are plenty improvements to the shop site like be able for users to review the product after they purchase it and be able to decide if they want to recommend the product.

8.3 What would be done differently

While the tests were successful my attention was brought to the system shopping cart.

Key function to the shop website is cart and some more functions could be implemented to improve it with icon on navigation bar icon showing how many items are in the cart.

Appendix A: Project Planning

8.4 Project Timeline

Week	Task	Description
1	Research	Introduction and research
2	Requirements	Requirements analysis
3	Design	Design architecture analysis
4	Development	Develop static pages using html
5	Style	Design and style using css
6	Functions	Create dynamic pages Using PHP
7	Development	Creating user and admin logins
8	Development	Additional pages and login functions
9	Testing	System functionality testing
10	Finalizing	Writing report and finishing project

.

Biblography

Luke Welling (2008) PHP and MySQL Web Development, 4th Edition Learning Web development and MySQL design p 8- 40

Jon Duckett (2011) HTML & CSS Design and Bold Websites Indianapolis, Indiana: john Wiley & Sons. p1-230.

Marie Brennan I.T.B (2015) Web Development 2 Lecture notes, 2015

Alan Forbes(2015) Joy of PHP: A Beginner's Guide to Programming Interactive Web Applications with PHP and MySQL. p. 10 - 150

Vikram Vaswani(2008) PHP: A Beginner's Guide

Eric Freeman, Elisabeth Freeman(2005) Head first HTML with CSS & XHTML p. 20 – 150

Zoe Mickley Gillenwater (2013) PUTTING FLEXBOX into PRACTICE. Charlotte, NC: Blend Conference p.1-60

Joel Murach (2012) Murach's MySQL. USA: Mike Murach & Associates Inc.p1-240

Michael Peacock (2010) PHP 5. Ecommerce development BIRMINGHAM – MUMBAI: Packet publishing p.1-280

Jennifer Niederst Robbins (August 2012) Learning Web Design A Beginners Guide to HTML CSS JavaScript and Web Graphics. 4h edition. Canada O'Reilly Media p1-600

Russel Dyer J. T. Dyer (2005) MySQL in a Nutshell, 2nd Edition A Desktop Quick Reference p1-79

Program listings

8.5 Summary of included program listings

This appendix comprises the following program listings:

/Public

- index.php

  /CSS

  - public/css/admin.css
  - public/css/css.html
  - public/css/rating.css
  - public/css/style.css
  - public/css/main.css

  /js

  - public/js/jquery-3.2.1.min.js

  /forum

  Implemented phpbb

/src

- src/main_controller.php

/templates

- templates/_footer.php
- templates/__header.php
- templates/__nav.php
- templates/_about.php
- templates/_billing.php
- templates/_cart.php
- templates/_db.php
- templates/_error.php
- templates/_errorActivate.php
- templates/_forgot.php

- templates/_function.php
- templates/_home.php
- templates/_index.php
- templates/_login.php
- templates/_logout.php
- templates/_profile.php
- templates/_register.php
- templates/_reset.php
- templates/_reset_password.php
- templates/_shop.php
- templates/_success.php
- templates/_verify.php

/admin

  - templates/admin/add_update.php
  - templates/admin/edit_update.php
  - templates/admin/add_user.php
  - templates/admin/edit_user.php
  - templates/admin/admin.php
  - templates/admin/db.php
  - templates/admin/delete_user.php
  - templates/admin/delete_update.php

/admin

  - templates/rating/add_rating.php
  - templates/ rating /DBController.php
  - templates/ rating /Rate.php
  - templates/ rating /rating.php

8.6 [1]Source Codes

8.6.1 The following is a commented listing of the public/index.php source code

```
<?php


/*takes all information from main_controller*/

require_once __DIR__ . '/../src/main_controller.php';


$action = filter_input(INPUT_GET, 'action');


switch ($action) {


    case 'home':

        homepage_action();

        break;

    case 'profile':

        profile_action();

        break;

    case 'about':

        about_action();

        break;

    case 'forum':

        forum_action();

        break;

    case 'shop':

        shop_action();

        break;
```

[1] FORMAT | PARAGRAPH | Line and Page breaks - page break before … keeps it all neat

```
case 'add':
    add_action();
    break;
case 'cart':
    cart_action();
    break;
case 'logout':
    logout_action();
    break;
case 'error':
    error_action();
    break;
case 'errorActivate':
    errorActivate_action();
    break;
case 'success':
    success_action();
    break;
case 'login':
    login_action();
    break;
case 'delete':
    remove_action();
    break;
case 'verify':
    verify_action();
    break;
case 'admin':
    admin_action();
```

```
        break;
    case 'add_user':
        add_user_action();
        break;
    case 'edit_user':
        edit_user_action();
        break;
    case 'delete_user':
        delete_user_action();
        break;
    case 'add_update':
        add_update_action();
        break;
    case 'edit_update':
        edit_update_action();
        break;
    case 'delete_update':
        delete_update_action();
        break;
    case 'forgot':
        forgot_action();
        break;
    case 'reset':
        reset_action();
        break;
    case 'reset_password':
        reset_password_action();
        break;
    case 'rating':
        rating_action();
        break;
    case 'add_rating':
        add_rating_action();
        break;
    case 'billing':
        billing_action();
        break;

/*in case wrong web redirect to index*/
    case 'main':
    default:
        index_action();

}
```

8.6.2 The following is a commented listing of the public/css/admin.css source code

```
body {
    font-size: 19px;
}
table{
    width: 50%;
    margin: 30px auto;
    border-collapse: collapse;
    text-align: left;
}
tr {
    border-bottom: 1px solid #cbcbcb;
}
th, td{
    border: none;
    height: 30px;
    padding: 2px;
}
tr:hover {
    background: #F5F5F5;
}

form {
    width: 45%;
    margin: 50px auto;
    text-align: left;
    padding: 20px;
    border: 1px solid #bbbbbb;
    border-radius: 5px;
```

```
}


.input-group {
   margin: 10px 0px 10px 0px;
}
.input-group label {
   display: block;
   text-align: left;
   margin: 3px;
}
.input-group input {
   height: 30px;
   width: 93%;
   padding: 5px 10px;
   font-size: 16px;
   border-radius: 5px;
   border: 1px solid gray;
}
.btn {
   padding: 10px;
   font-size: 15px;
   color: white;
   background: #5F9EA0;
   border: none;
   border-radius: 5px;
}
.edit_btn {
   text-decoration: none;
   padding: 2px 5px;
```

```
    background: #2E8B57;

    color: white;

    border-radius: 3px;

}


.del_btn {

    text-decoration: none;

    padding: 2px 5px;

    color: white;

    border-radius: 3px;

    background: #800000;

}

.msg {

    margin: 30px auto;

    padding: 10px;

    border-radius: 5px;

    color: #3c763d;

    background: #dff0d8;

    border: 1px solid #3c763d;

    width: 50%;

    text-align: center;

}
```

8.6.3 The following is a commented listing of the public/css/css.html source code

```
  <link href='http://fonts.googleapis.com/css?family=Titillium+Web:400,300,600' rel='stylesheet' type='text/css'>

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">

  <link rel="stylesheet" href="css/style.css">

  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Oswald">

  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open Sans">

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

8.6.4 The following is a commented listing of the public/css/rating.css source code

```
body {
	width: 100%;
	font-family: Arial;
}
.demo-table {
	width: 100%;
	border-spacing: initial;
	margin: 20px 0px;
	word-break: break-word;
	table-layout: auto;
	line-height: 1.8em;
	color: #333;
}
.demo-table th {
	background: #568498;
	padding: 5px;
	text-align: left;
	color: #FFF;
}
.demo-table td {
	background-color: #ffffff;
	padding: 20px 5px 5px;
}
.demo-table td div.feed_title {
	text-decoration: none;
	color: #003fff;
	font-weight: bold;
}
```

```
}

.demo-table ul {

        margin: 0;

        padding: 5px;

}

.demo-table li {

        cursor: pointer;

        list-style-type: none;

        display: inline-block;

        color: #F0F0F0;

        text-shadow: 0 0 1px #666666;

        font-size: 20px;

}

.demo-table .highlight, .demo-table .selected {

        color: #F4B30A;

        text-shadow: 0 0 1px #F48F0A;

}

.star-rating-box {

        border: #E4E4E4 1px solid;

        margin: 10px 0px;

}

.star-rating-count {

        border-top: #E4E4E4 1px solid;

        font-size: 0.8em;

        padding: 2px 5px;

        background: #f9f9f9;

        color: #828181;

}
```

8.6.5 The following is a commented listing of the public/css/style.css source code

```
h1{
   text-align: center;
}
.thumbnail {
   float: left;
   margin-right: 20px;
   text-align: center;
}
*, *:before, *:after {
   box-sizing: border-box;
}
html {
   overflow-y: scroll;
}
body {
   background: #ffffff; /* #c1bdba */
   font-family: 'Titillium Web', sans-serif;
}
a {
   text-decoration: none;
   color: #1ab188;
   -webkit-transition: .5s ease;
   transition: .5s ease;
}
a:hover {
   color: #179b77;
}

.form {
   background: rgba(19, 35, 47, 0.9);
   padding: 40px;
   max-width: 600px;
   margin: 40px auto;
   border-radius: 4px;
   box-shadow: 0 4px 10px 4px rgba(19, 35, 47, 0.3);
}

.tab-group {
   list-style: none;
   padding: 0;
   margin: 0 0 40px 0;
}
.tab-group:after {
   content: "";
   display: table;
   clear: both;
}
.tab-group li a {
   display: block;
   text-decoration: none;
   padding: 15px;
```

```
    background: rgba(160, 179, 176, 0.25);
    color: #a0b3b0;
    font-size: 20px;
    float: left;
    width: 50%;
    text-align: center;
    cursor: pointer;
    -webkit-transition: .5s ease;
    transition: .5s ease;
}
.tab-group li a:hover {
    background: #179b77;
    color: #ffffff;
}
.tab-group .active a {
    background: #1ab188;
    color: #ffffff;
}

.tab-content > div:last-child {
    display: none;
}

h1 {
    text-align: center;
    color: #ffffff;
    font-weight: 300;
    margin: 0 0 40px;
}

h2 {
    text-align: center;
    color: #1ab188;
    font-weight: 1000;
    margin: 0;
}

span {
    color: #1ab188;
    font-weight: bold;
}

p {
    text-align: center;
    color: #ffffff;
    margin: 0px 0px 50px 0px;
}

div.info {
    color: pink;
    display: box;
    text-align: center;
    padding: 5px;
    margin-top: -20px;
    margin-bottom: 15px;
    border: 1px solid red;
    background: #66131c;
}

label {
    position: absolute;
    -webkit-transform: translateY(6px);
    transform: translateY(6px);
    left: 13px;
    color: rgba(255, 255, 255, 0.5);
    -webkit-transition: all 0.25s ease;
    transition: all 0.25s ease;
    -webkit-backface-visibility: hidden;
    pointer-events: none;
    font-size: 22px;
}
label .req {
    margin: 2px;
    color: #1ab188;
}

label.active {
    -webkit-transform: translateY(50px);
    transform: translateY(50px);
    left: 2px;
    font-size: 14px;
}
label.active .req {
    opacity: 0;
}

label.highlight {
    color: #ffffff;
}
input, textarea {
    font-size: 22px;
    display: block;
    width: 100%;
    height: 100%;
    padding: 5px 10px;
    background: none;
    background-image: none;
    border: 1px solid #a0b3b0;
    color: #ffffff;
    border-radius: 0;
    -webkit-transition: border-color .25s ease, box-shadow .25s ease;
    transition: border-color .25s ease, box-shadow .25s ease;
}
input:focus, textarea:focus {
    outline: 0;
    border-color: #1ab188;
}

textarea {
    border: 2px solid #a0b3b0;
    resize: vertical;
}
```

```
.field-wrap {
   position: relative;
   margin-bottom: 40px;
}
.top-row:after {
   content: "";
   display: table;
   clear: both;
}
.top-row > div {
   float: left;
   width: 48%;
   margin-right: 4%;
}
.top-row > div:last-child {
   margin: 0;
}
.button {
   border: 0;
   outline: none;
   border-radius: 0;
   padding: 15px 0;
   font-size: 2rem;
   font-weight: 600;
   text-transform: uppercase;
   letter-spacing: .1em;
   background: #1ab188;
   color: #ffffff;
   -webkit-transition: all 0.5s ease;
   transition: all 0.5s ease;
   -webkit-appearance: none;
}
.button:hover, .button:focus {
   background: #179b77;
}
.button-block {
   display: block;
   width: 100%;
}

.forgot {
   margin-top: -20px;
   text-align: right;
}
```

8.6.6 The following is a commented listing of the public/css/main.css source code

```
{
        margin: 0px;
        padding: 0px;
}

h1 a:hover {
        text-decoration: none;
}

h1, h2, h3 {
        font-family: Century Gothic;
        font-weight: normal;
        margin-bottom: 15px;
}

div.pane {
        width: 1000px;
        margin: 0px auto;
}

div.header {
        padding-top: 15px;
        padding-bottom: 15px;
}

div.loginpane {
        width: 230px;
        display: block;
        position: relative;
        float: left;
        padding: 15px;
}

div.forumdesc {
        width: 700px;
        display: block;
        position: relative;
        float: left;
        padding: 15px;
}

div.content {
        background-color: #EFEFEF;
        margin-top: 15px;
        width: 100%;
        display: block;
        position: relative;
        float: left;
        padding: 15px;

}

div.header h1 {
        margin-left: 15px;
        font-family: Century Gothic;
        font-weight: normal;
}

a {
        text-decoration: none;
        color: #0066FF;
}

p{
        font-family: Calibri;
}

input[type="submit"], button {
        border-radius: 4px;
        color: #EFEFEF;
        width: 200px;
        padding: 7px;
        font-family: Calibri;
        background-image: linear-gradient(to bottom, #0066FF, #0099FF);
        margin-top: 15px;
        border: none;
}

input[type="text"],[type="password"] {
        border-radius: 15px;
        width: 200px;
        padding: 7px;
        font-family: Arial;
        margin-top: 15px;
        border: none;
        background-color: #DFDFDF;
}

input[type="text"]:focus {
        outline-width: 0;
        outline-style: none;
}

input[type="password"]:focus {
        outline-width: 0;
        outline-style: none;
}

input[type="submit"]:focus {
        outline-width: 0;
        outline-style: none;
}

button:focus {
        outline-width: 0;
        outline-style: none;
}

table.category-table {
        width: 100%;
        margin-bottom: 35px;
        font-family: Calibri;
}

table.category-table td.main-category {
        font-size: 16pt;
        font-weight: bold;
}

table.category-table th {
        font-size: 12pt;
        font-weight: bold;
        text-align: left;
}

table.topic-table {
        width: 100%;
        font-family: Calibri;
}

table.topic-table th {
        vertical-align: top;
        text-align: left;
        font-size: 12pt;
        background-color: #8E8E8E;
        padding: 4px;
}

table.topic-table td {
        font-size: 12pt;
        padding: 4px;
}

input#topic {
        border-radius: 15px;
        width: 350px;
        padding: 7px;
        font-family: Arial;
        margin-top: 15px;
        border: none;
        background-color: #DFDFDF;
}

textarea {
        width: 350px;
        height: 100px;
        margin-top: 15px;
        margin-bottom: 15px;
}

a:hover {
        text-decoration: underline;
}

table.reply-table {
        font-family: Calibri;
```

```
	width: 100%;
}

table.reply-table th {
	vertical-align: top;
	text-align: left;
	font-size: 12pt;
}

table.reply-table td {
	padding-left: 15px;
	vertical-align: top;
	margin-bottom: 5px;
}
```

8.6.7 The following is a commented listing of the src/main_controller.php source code

```
<?php
/**
 * Main Controller
 */
function index_action()
{
    require_once __DIR__ . '/../templates/index.php';
}
function homepage_action()
{

    require_once __DIR__ . '/../templates/home.php';
}
function profile_action()
{
    $pageTitle = "profile page";
    require_once __DIR__ . '/../templates/profile.php';
}

function about_action()
{
    $pageTitle = "about page";
    require_once __DIR__ . '/../templates/about.php';
}
function forum_action()
{
    $pageTitle = "forum page";
    require_once __DIR__ . '/../public/forum/index.php';
}
function shop_action()
{
    $pageTitle = "shop page";
    require_once __DIR__ . '/../templates/function.php';
}
function add_action()
{
    $pageTitle = "add";
    require_once __DIR__ . '/../templates/shop.php';
}
function cart_action()
{
    $pageTitle = "cart";
    require_once __DIR__ . '/../templates/cart.php';
}
function logout_action()
{
    $pageTitle = "cart";
    require_once __DIR__ . '/../templates/logout.php';
}
function error_action()
{
    $pageTitle = "error";
    require_once __DIR__ . '/../templates/error.php';
}
function errorActivate_action()
{
    $pageTitle = "errorActivate";
    require_once __DIR__ . '/../templates/errorActivate.php';
}
function verify_action()
{
    $pageTitle = "verify";
    require_once __DIR__ . '/../templates/verify.php';
}
function success_action()
{
    $pageTitle = "success";
    require_once __DIR__ . '/../templates/success.php';
}
function forgot_action()
{
    $pageTitle = "forgot";
    require_once __DIR__ . '/../templates/forgot.php';
}
function login_action()
{
    require_once __DIR__ . '/../templates/login.php';
}
function remove_action()
{
    require_once __DIR__ . '/../templates/shop.php';
}
function topics_action()
{
    require_once __DIR__ . '/../templates/topics.php';
```

```
}

function search_action()

{

    $pageTitle = "search page";

    require_once __DIR__ . '/../templates/search.php';

}

function admin_action()

{

    require_once __DIR__ . '/../templates/admin/admin.php';

}

function add_user_action()

{

    require_once __DIR__ . '/../templates/admin/add_user.php';

}



function delete_user_action()

{

    require_once __DIR__ . '/../templates/admin/delete_user.php';

}

function edit_user_action()

{

    require_once __DIR__ . '/../templates/admin/edit_user.php';

}

function add_update_action()

{

    require_once __DIR__ . '/../templates/admin/add_update.php';

}

function delete_update_action()
```

```
{
    require_once __DIR__ . '/../templates/admin/delete_update.php';
}
function edit_update_action()
{
    require_once __DIR__ . '/../templates/admin/edit_update.php';
}
function reset_action()
{
    require_once __DIR__ . '/../templates/reset.php';
}
function reset_password_action()
{
    require_once __DIR__ . '/../templates/reset_password.php';
}
function rating_action()
{
    require_once __DIR__ . '/../templates/rating/rating.php';
}
function add_rating_action()
{
    require_once __DIR__ . '/../templates/rating/add_rating.php';
}
function billing_action()
{
    require_once __DIR__ . '/../templates/billing.php';
}
?>
```

8.6.8 The following is a commented listing of the templates/_footer.php source code

```
<br>

<br>

<center>


 <footer class="w3-container w3-dark-grey" style="padding:17px">

    <a href="#" class="w3-button w3-black w3-padding-large w3-margin-bottom"><i class="fa fa-arrow-up w3-margin-right"></i>To the top</a>

    <a href="index.php?action=about" class="w3-bar-item w3-button"><i class="fa fa-info-circle"> About</i></a>

    <a href="mailto:portalgamingofficial@gmail.com" class="w3-bar-item w3-button"><i class="fa fa-mail-forward"> Contact</i></a>

    <p>Created by: B00087701 - 2018  Wiktor Lesiak </p>

    <p>Copyright &copy; Portal Gaming</p>

</center>

</footer>
```

8.6.9 The following is a commented listing of the templates/_header.php source code

```
<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <title>PØRTAL GAMING</title>

    <style>

</style>

    <link rel="icon" href="images/icon.png">
```

8.6.10 The following is a commented listing of the templates/_nav.php source code

```
<?php
/* Displays user information and some useful messages */
session_start();


// Check if user is logged in using the session variable
if ( $_SESSION['logged_in'] != 1 ) {
   $_SESSION['message'] = "You must log in before viewing your profile page!";
   header("location: ?action=error");
}
else {
   $id = (isset($_GET['id']));
   // Makes it easier to read
   $first_name = $_SESSION['first_name'];
   $last_name = $_SESSION['last_name'];
   $email = $_SESSION['email'];
   $active = $_SESSION['active'];
   $active = $_SESSION['admin'];
}
?>
<!--stylesheets and fonts from w3schools-->
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Oswald">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open Sans">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">



<div class="w3-bar w3-black w3-hide-small">
```

```
    <a href="index.php?action=home" class="w3-bar-item w3-button"><i class="fa fa-home"> Home</i></a>
    <a href="index.php?action=about" class="w3-bar-item w3-button"><i class="fa fa-info"> About</i></a>
    <a href="/forum" class="w3-bar-item w3-button"><i class="fa fa-file-text-o"> Forum</i></a>
    <a href="index.php?action=shop" class="w3-bar-item w3-button"><i class="fa fa-shopping-bag"> Shop</i></a>


    <a href="index.php?action=logout" class="w3-bar-item w3-button w3-right"><i class="fa fa-sign-out">  Sign Out</i></a>
    <a href="index.php?action=profile" class="w3-bar-item w3-button w3-right"><i class="fa fa-user-circle">  <?=$first_name?></i></a>
    <a href="index.php?action=cart" class="w3-bar-item w3-button w3-right"><i class="fa fa-shopping-cart">  Cart</i></a>
    <a href="index.php?action=rating" class="w3-bar-item w3-button w3-right"><i class="fa fa-star">  RATING</i></a>


    <?php
    //if admin is logged in it displays ADMIN CURD on navigation bar
    if ( $_SESSION['admin'] == 1 ) {
        ?> <a href="index.php?action=admin" class="w3-bar-item w3-button w3-right"><i class="fa fa-edit">  ADMIN CURD</i></a>
    <?php
    }
    ?>
</div>
```

8.6.11 The following is a commented listing of the templates/about.php source code

```
<?php
//import header
require_once __DIR__ . '/../templates/_header.php';

?>


<?php
//import navigation bar
require_once __DIR__ . '/../templates/_nav.php';
?>
  <!-- Header -->
  <header class="w3-container w3-center w3-padding-48 w3-white">
    <h1 class="w3-xxxlarge"><b>ABOUT</b></h1>
    <h6>About Portal Gaming</h6>
  </header>


    <!--******************GAMES******************-->
    <div class="w3-content" style="max-width:400px">


      <!-- Grid -->
      <div class="w3-row w3-padding w3-border">

        <!-- Blog entries -->
        <div class="w3-col 29 s12">

          <!-- Blog entry -->
          <div class="w3-container w3-white w3-margin w3-padding-large">
            <div class="w3-center">
              <h2> <a href="index.php?action=shop">Games</a> </h2>
              <hr>
              <h5><strong><i>Buy and see latest games!</strong></i></h5>
            </div>

            <div class="w3-justify">
              <p>In out store you can purchase games and see what are the new
releases.</p>

            </div>
          </div>

        </div>

        <!-- END About/Intro Menu -->
      </div>

      <!-- END GRID -->
    </div>

    <!-- END w3-content -->
    <hr>
  <!--******************FORUM******************-->
  <div class="w3-content" style="max-width:400px">


    <!-- Grid -->
    <div class="w3-row w3-padding w3-border">

      <!-- Blog entries -->
      <div class="w3-col 29 s12">

        <!-- Blog entry -->
        <div class="w3-container w3-white w3-margin w3-padding-large">
          <div class="w3-center">
            <h2><a href="/forum">Forum </a></h2>
            <hr>
            <h5><i><strong>Chat, discuss and help.</strong></i></h5>
          </div>

          <div class="w3-justify">
          <p>In Portal Gaming Forum you can Create new topics, and comment on other
topics,</p>
            <p>You can join discussions about all games that you can imagine!</p>
            <p> You can also show of your skills if you are good at the game :)</p>

          </div>
        </div>

      </div>

      <!-- END About/Intro Menu -->
    </div>

    <!-- END GRID -->
  </div>

  <!-- END w3-content -->
  <hr>

  <!--******************UPDATES******************-->
  <div class="w3-content" style="max-width:400px">


    <!-- Grid -->
    <div class="w3-row w3-padding w3-border">

      <!-- Blog entries -->
      <div class="w3-col 29 s12">

        <!-- Blog entry -->
        <div class="w3-container w3-white w3-margin w3-padding-large">
          <div class="w3-center">
            <h2><a href="index.php?action=home"> Updates </a></h2>
            <hr>
            <h5><strong><i>Be up to date with our developer update</i></strong></h5>
          </div>
```

```
                <div class="w3-justify">
            <p>See the updates about website and the latest and fastest news about the
games!</p>
                </div>
            </div>

        </div>

        <!-- END About/Intro Menu -->
    </div>

    <!-- END GRID -->
  </div>

  <!-- END w3-content -->
  <hr>
  <!--******************RATING******************-->
  <div class="w3-content" style="max-width:400px">


    <!-- Grid -->
    <div class="w3-row w3-padding w3-border">

        <!-- Blog entries -->
        <div class="w3-col 29 s12">

            <!-- Blog entry -->
            <div class="w3-container w3-white w3-margin w3-padding-large">
                <div class="w3-center">
                    <h2><a href="index.php?action=rating"> Rating </a></h2>
                    <hr>
                    <h5><strong><i>Star rating</i></strong></h5>
                </div>

                <div class="w3-justify">
                <p>Rate every single game and see which game is your favourite based on our
community rating!</p>

                </div>
            </div>
        </div>

        <!-- END About/Intro Menu -->
    </div>

    <!-- END GRID -->
  </div>

  <!-- END w3-content -->
  <hr>
<?php
//import footer
require_once __DIR__ . '/../templates/_footer.php';
```

8.6.12 The following is a commented listing of the templates/billing.php source code

```
<?php
require_once __DIR__ . '/../templates/_nav.php';
ob_start();
require 'db.php';
?>
<div style="clear:both"></div>


    <h2>Billing</h2>
<!--SHOPPING CART-->
    <div class="table-responsive" align="center">
      <table>
        <tr>
          <th width="40%">Product Name</th>
          <th width="10%">Quantity</th>
          <th width="20%">Price Details</th>
          <th width="15%">Order Total</th>
          <th width="5%">Action</th>
        </tr>
        <?php
        if(!empty($_SESSION["cart"]))
        {
          $total = 0;
          foreach($_SESSION["cart"] as $keys => $values)
          {
            ?>
            <tr>
              <!--SHOPPING DETAILS-->
              <th><?php echo $values["item_name"]; ?></th>

              <th><?php echo $values["item_quantity"] ?></th>

              <th>&euro; <?php echo $values["product_price"]; ?></th>

              <th>&euro; <?php echo number_format($values["item_quantity"] * $values["product_price"], 2); ?></th>

              <th><a href="index.php?action=delete&id=<?php echo $values["product_id"]; ?>"><span class="text-danger">X</span></a></th>


            </tr>


            <?php
            //total price
            $total = $total + ($values["item_quantity"] * $values["product_price"]);
          }
          ?>
          <tr>
            <td colspan="3" align="right"><strong>Total</strong></td>
            <td align="right"><u>&euro;<?php echo number_format($total, 2); ?></u></td>
            <td></td>
          </tr>


          <?php
        }
        ?>
<!--END SHOPPING CART-->


        <!--BILLING FORM-->
  <div align="left">
    <h4>Enter Your Detals</h4>
```

```
<fieldset>

<legend>Personal Detals</legend>

<label for="user">Name:</label>
<input type="text" name="firstName" required placeholder = "Enter first name" autofocus><strong>*</strong>
<br>

<label for="comments">Phone Number:</label>
<input type="text" name="phonenumber" required placeholder = "Enter your phone number" autofocus><strong>*</strong>

<br>

  <label for="date">Date of Birth:</label>
  <input type="date" name="quantity" min="0" max="100">
<br>
  <label for="gender">Gender: </label>
  <input type="radio" name="male" value = "male" >
  <span class = "radio"> Male </span>
  <input type="radio" name="female" value = "female" >
  <span class = "radio">Female</span>

</fieldset>

<br>
<fieldset>
  <legend>Payment</legend>
```

```
      <label for="user">Credit Card Number:</label>
      <input type="text" name="Number" required placeholder = "Credit Card Number" autofocus><strong>*</strong>
      <br>
      <label for="user">Card Valid Date</label>
      <input type="date" name="Valid"  autofocus><strong>*</strong>
      <br>
      <label for="user">Security Code:</label>
      <input type="text" name="code" required placeholder = "Security Code" autofocus><strong>*</strong>
      <br>


    </fieldset>
<!--END BILLING FORM-->
    <!--EMAIL ADDRESS CONFIRMATION -->
    <?php
    $games = $values["item_name"];
    // Check if form submitted with method="post"
    if ( $_SERVER['REQUEST_METHOD'] == 'POST' )
    {
      $email = $mysqli->escape_string($_POST['email']);
      $result = $mysqli->query("SELECT * FROM users WHERE email='$email'");
      if ( $result->num_rows == 0 ) // User doesn't exist
      {
        $_SESSION['message'] = "User with that email doesn't exist!";
        header("location: index.php?action=error");
      }
      else { // User exists (num_rows != 0)
```

```
        $user = $result->fetch_assoc(); // $user becomes array with user data


        $email = $user['email'];
        $hash = $user['hash'];
        $first_name = $user['first_name'];


        // Session message to display on success.php
        $_SESSION['message'] = "<p>Please check your email <span>$email</span>"
          . " for a receipt and list of games purchased!</p>";


        // Send registration confirmation link (reset.php)
        $to      = $email;
        $subject = 'Thank you for shopping with Portal Gaming! ( noreply)';
        $message_body = '
    Hello '.$first_name.',


    Thank you for shopping with Portal Gaming today!


    Purchased games:


    Games: '. print_r( $keys, true ) .
    ' Total: €' . $total;




        mail($to, $subject, $message_body);


        header("location: index.php?action=success");
      }
    }
    ?>
    <!--CONFIRM EMAIL FORM-->
    <!DOCTYPE html>
    <html>
    <head>
      <title>Reset Your Password</title>
      <style> @import "css/main.css"; </style>
    </head>
    <body>
    <div class="form">
      <h1>Confirm Address Email</h1>
      <form action="index.php?action=billing" method="post">
        <div class="field-wrap">
          <label>
            Email Address<span class="req">*</span>
          </label>
          <input type="email"required autocomplete="off" name="email"/>
        </div>
        <button class="button button-block"/>Purchase</button>
      </form>
    </div>
<!--END EMAIL CONFIRMATION-->
    </body>
    </html>
    <hr>
    <h2><center>Items in Cart</center></h2>
```

8.6.13 The following is a commented listing of the templates/cart.php source code

```
<?php

//implement header

require_once __DIR__ . '/../templates/_header.php';

?>

<body>

<?php

//implement navbar

require_once __DIR__ . '/../templates/_nav.php';

?>

<?php

//connect to database

$connect = mysqli_connect("localhost", "root", "root", "portalgaming");


if ( $_SESSION['logged_in'] != 1 ) {

   $_SESSION['message'] = "You must log in your before viewing your cart!";

   header("location: index.php?action=error");

}else {

   // Makes it easier to read

   $first_name = $_SESSION['first_name'];

   $last_name = $_SESSION['last_name'];

   $email = $_SESSION['email'];

   $active = $_SESSION['active'];

}


//if account not active give an error

if( !$active ){

   $_SESSION['message'] = "You must activate your account before viewing your cart!";
```

```
    header("location: index.php?action=errorActivate");
}
?>
<!doctype html>
<html>

<header class="w3-container w3-center w3-padding-48 w3-white">
    <h1 class="w3-xxxlarge"><b>SHOPPING CART</b></h1>
</header>

<!--SHOPPING CART-->
<div style="clear:both"></div>
<br>
<hr>
<h2>My Shopping Bag</h2>
<div class="table-responsive">
    <table class="table table-bordered">
        <tr>
            <th width="40%">Product Name</th>
            <th width="10%">Quantity</th>
            <th width="20%">Price Details</th>
            <th width="15%">Order Total</th>
            <th width="5%">Action</th>
        </tr>
        <?php

        if(isset($_GET["action"]))
        {
            if($_GET["action"] == "delete")
            {
                foreach($_SESSION["cart"] as $keys => $values)
                {
                    if($values["product_id"] == $_GET["id"])
                    {
                        unset($_SESSION["cart"][$keys]);
                        echo '<script>alert("Product has been removed")</script>';
                        echo '<script>window.location="index.php?action=cart"</script>';
                    }
                }
            }
        }
        //if empty cart
        if(empty($_SESSION["cart"])) {
            ?>
           <h1>Your shopping cart is empty!</h1>
            <p>purchse some games in <a href="index.php?action=store">store</a> </p>
            <?php
        }else{
            //if products added to cart
            if (!empty($_SESSION["cart"])) {
                $total = 0;
                foreach ($_SESSION["cart"] as $keys => $values) {
                    ?>
                    <tr>
                        <td><?php echo $values["item_name"]; ?></td>
                        <td><?php echo $values["item_quantity"] ?></td>
                        <td>&euro; <?php echo $values["product_price"]; ?></td>
                    <td>&euro; <?php echo number_format($values["item_quantity"] *
        $values["product_price"], 2); ?></td>
                    <td><a href="index.php?action=delete&id=<?php echo $values["product_id"];
        ?>"><span class="text-danger">Remove</span></a></td>
                    <td><input type="button" value="Check Out"
        onclick="window.location='index.php?action=billing'"></td>
                  </tr>


                  <?php
                  $total = $total + ($values["item_quantity"] * $values["product_price"]);
                }
                ?>
                <tr>
                  <td colspan="3" align="right"><strong>Total</strong></td>
                  <td align="right"> <br>Total: &euro; <?php echo number_format($total, 2);
        ?></td>
                  <td></td>
                   </tr>
                <?php
              }
            }
            ?>
          </table>
        </div>
        <!--SHOPPING CART END-->
        <?php
        //implement footer
        require_once __DIR__ . '/../templates/_footer.php';
```

8.6.14 The following is a commented listing of the templates/db.php source code

```
<?php

/* Database connection settings */

$host = 'localhost';

$user = 'root';

$pass = 'root';

$db = 'portalgaming';

$mysqli = new mysqli($host,$user,$pass,$db) or die($mysqli->error);
```

8.6.15 The following is a commented listing of the templates/error.php source code

```
<?php
/* Displays all error messages */
session_start();
?>
<!DOCTYPE html>
<html>
<head>
  <title>Error</title>
  <?php include 'css/css.html'; ?>
</head>
<body>
<div class="form">
    <h1>Error</h1>
    <p>
    <?php
    //if there is message passed display it
    if( isset($_SESSION['message']) AND !empty($_SESSION['message']) ):
        echo $_SESSION['message'];
    else: //if fail redirect to home
        header( "location: ?action=index" );
    endif;
    ?>
    </p>
    <!--redirect to home button-->
    <a href="?action=account"><button class="button button-block"/>Home</button></a>
</div>
</body>
```

8.6.16 The following is a commented listing of the templates/errorActivate.php source code

```
<?php
/* Displays all error messages */
session_start();
?>
<!DOCTYPE html>
<html>
<head>
  <title>Error</title>
  <?php include 'css/css.html'; ?>
</head>
<body>
<div class="form">
  <h1>Error</h1>
  <p>
    <?php//if there is message passed display it
    if( isset($_SESSION['message']) AND !empty($_SESSION['message']) ):
      echo $_SESSION['message'];
    else: //if fail redirect to home
      header( "location: ?action=home" );
    endif;
    ?>
  </p>
  <!--redirect to homepage when error is displayed-->
  <a href="?action=home"><button class="button button-block"/>Home</button></a>
</div>
</body>
</html>
```

8.6.17 The following is a commented listing of the templates/forgot.php source code

```
<?php
/* Reset your password form, sends reset.php password link */
require 'db.php';
session_start();

// Check if form submitted with method="post"
if ( $_SERVER['REQUEST_METHOD'] == 'POST' )
{
   $email = $mysqli->escape_string($_POST['email']);
   $result = $mysqli->query("SELECT * FROM users WHERE email='$email'");

   if ( $result->num_rows == 0 ) // User doesn't exist
   {
      $_SESSION['message'] = "User with that email doesn't exist!"; //sends message to the session
      header("location: index.php?action=error"); //redirect to error
   }
   else { // User exists (num_rows != 0)

      $user = $result->fetch_assoc(); // $user becomes array with user data

      $email = $user['email'];
      $hash = $user['hash'];
      $first_name = $user['first_name'];

      // Session message to display on success.php
      $_SESSION['message'] = "<p>Please check your email <span>$email</span>"
      . " for a confirmation link to complete your password reset!</p>";
```

```
    // Send registration confirmation link (reset.php)
    $to      = $email;
    $subject = 'Password Reset Link ( noreply)';
    $message_body = '
    Hello '.$first_name.',

    You have requested password reset!

    Please click this link to reset your password:

    http://localhost/index.php?action=reset&email='.$email.'&hash='.$hash;

    mail($to, $subject, $message_body);

    header("location: index.php?action=success");
  }
}
?>
<!--RESET PASSWORD FORM-->
<!DOCTYPE html>
<html>
<head>
  <title>Reset Your Password</title>
  <?php include 'css/css.html'; ?>
</head>
<body>

  <div class="form">
```

```
    <h1>Reset Your Password</h1>

    <form action="index.php?action=forgot" method="post">

     <div class="field-wrap">

      <label>

        Email Address<span class="req">*</span>

      </label>

      <input type="email"required autocomplete="off" name="email"/>

    </div>

    <button class="button button-block"/>Reset</button>

    </form>

  </div>


<script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>

<script src="js/index.js"></script>

</body>


</html>

<!--RESET PASSWORD FORM END-->
```

8.6.18 The following is a commented listing of the templates/function.php source code

```
<?php
require_once __DIR__ . '/../templates/_header.php';
?>


<?php
require_once __DIR__ . '/../templates/_nav.php';
?>


<?php
//connect to database
$connect = mysqli_connect("localhost", "root", "root", "portalgaming");



// Check if user is logged in using the session variable
if ( $_SESSION['logged_in'] != 1) {
    $_SESSION['message'] = "You must log in or activate your account before viewing shop!";
    header("location: index.php?action=error");
}
else {
    // Makes it easier to read
    $first_name = $_SESSION['first_name'];
    $last_name = $_SESSION['last_name'];
    $email = $_SESSION['email'];
    $active = $_SESSION['active'];


}


?>
```

```
<!doctype html>



   <header class="w3-container w3-center w3-padding-48 w3-white">
   <h1 class="w3-xxxlarge"><b>PORTAL GAMING SHOP</b></h1>
   <h6>Purchase all the new releases</h6>
   </header>

<div class="w3-content" style="max-width:1400px">

   <?php
   //select everything rom products table
          $query = "SELECT * FROM products ORDER BY id ASC";
   $result = mysqli_query($connect, $query);

          if(mysqli_num_rows($result) > 0)
          {
                    while($row = mysqli_fetch_array($result))
                    {
                              ?>




            <div class="w3-row w3-padding w3-border">

         <div class="thumbnail">
         <form method="post" action="index.php?action=add&id=<?php echo $row["id"]; ?>">
            <div class="w3-center">
```

```
    <h2 class="text-info"><?php echo $row["p_name"]; ?></h2>
        <a href="<?php echo $row["link"]; ?>"><img src="<?php echo $row["image"]; ?>" height="215" width="460" class="img-responsive"></a>
        <h5 class="text-danger"><strong>Price: &euro;<?php echo $row["p_price"]; ?></strong></h5>


        <div class="w3-justify">
    <p><?php echo $row["description"]; ?></p>
    </div>
    <br>


        <br>
        <!--SENT ITEMS TO CART-->
        <input type="text" name="quantity" class="w3-click" value="1">
        <input type="hidden" name="hidden_name" value="<?php echo $row["p_name"]; ?>">
        <input type="hidden" name="hidden_price" value="<?php echo $row["p_price"]; ?>">
        <input type="hidden" name="hidden_about" value="<?php echo $row["description"]; ?>">
        <input type="submit" name="add"  class="w3-button" value="Add to Bag">
        <!--END SENT ITEMS TO CART-->
    </form>
    </div>
    </div>
    </div>
    <br>



    <?php
```

```
                    }
            }
            ?>
        <!--PRODUCT ITEMS END-->

<br>

<?php
require_once __DIR__ . '/../templates/_footer.php';
?>
```

8.6.19 The following is a commented listing of the templates/home.php source code

```
<?php
//import header
require_once __DIR__ . '/../templates/_header.php';
?>
<?php
//import navbar
require_once __DIR__ . '/../templates/_nav.php';
?>
  <!-- Header -->
  <header class="w3-container w3-center w3-padding-48 w3-white">
    <h1 class="w3-xxxlarge"><b>PORTAL GAMING</b></h1>
    <h6>Welcome to Portal Gaming website</h6>
  </header>
<?php
//connect to database
$connect = mysqli_connect("localhost", "root", "root", "portalgaming");
			$query = "SELECT * FROM devupdate ORDER BY id DESC ";
			$result = mysqli_query($connect, $query);
			if(mysqli_num_rows($result) > 0)
  {
    while($row = mysqli_fetch_array($result))
    {
      ?>
  <div class="w3-content" style="max-width:1400px">
    <!-- Grid -->
    <div class="w3-row w3-padding w3-border">
      <!-- Blog entries -->
      <div class="w3-col 29 s12">
```

```
                <!-- Blog entry -->
                <div class="w3-container w3-white w3-margin w3-padding-large">
                    <div class="w3-center">
                        <h3><?php echo $row["title"]; ?></h3>
                        <hr>
                        <h5><?php echo $row["titledesc"]; ?>, <span class="w3-opacity"><i><?php echo $row["timestamp"]; ?></i></span></h5>
                    </div>

                    <div class="w3-justify">
                        <p><?php echo $row["description"]; ?></p>
                    </div>
                </div>
            </div>
                <!-- END About/Intro Menu -->
            </div>
            <!-- END GRID -->
        </div>
        <!-- END w3-content -->
<hr>
            <?php
        }
    }
?>
<?php
require_once __DIR__ . '/../templates/_footer.php';
```

8.6.20 The following is a commented listing of the templates/index.php source code

```
<?php

/* Main page with two forms: sign up and log in */

require 'db.php';

session_start();

?>

<!DOCTYPE html>

<html>

<head>


    <?php include 'css/css.html'; ?>

</head>


<?php

//if user already logged in, redirect to homepage

if ( $_SESSION['logged_in'] == 1 ) {

    $_SESSION['message'] = "Hey, How are you today?";

    header("location: ?action=profile");

}

else {




    if ($_SERVER['REQUEST_METHOD'] == 'POST') {

        if (isset($_POST['login'])) { //user logging in


            require 'login.php';


        } elseif (isset($_POST['register'])) { //user registering
            require 'register.php';


        }
    }
    ?>
    <!--NAVIGATION BAR-->
    <div class="w3-bar w3-black w3-hide-small">
        <a href="index.php?action=home" class="w3-bar-item w3-button"><i class="fa fa-home"> Home</i></a>
        <a href="index.php?action=profile" class="w3-bar-item w3-button"><i class="fa fa-user"> Profile</i></a>
        <a href="index.php?action=about" class="w3-bar-item w3-button"><i class="fa fa-info"> About</i></a>
        <a href="/forum" class="w3-bar-item w3-button"><i class="fa fa-pencil-square-o"> Forum</i></a>
        <a href="index.php?action=shop" class="w3-bar-item w3-button"><i class="fa fa-shopping-bag"> Shop</i></a>



        <a href="index.php?action=logout" class="w3-bar-item w3-button w3-right"><i class="fa fa-sign-out"> Sign Out</i></a>
        <a href="index.php?action=profile" class="w3-bar-item w3-button w3-right"><i class="fa fa-user-circle"> User</i></a>
        <a href="index.php?action=cart" class="w3-bar-item w3-button w3-right"><i class="fa fa-shopping-cart"> Cart</i></a>



    </div>
    <!--END NAVIGATION BAR-->

    <body>
```

```
    <!--LOGIN AND REGISTER FORM-->
    <div class="form">

        <ul class="tab-group">
            <li class="tab"><a href="#signup">Sign Up</a></li>
            <li class="tab active"><a href="#login">Log In</a></li>
        </ul>

        <div class="tab-content">
<!--LOGIN FORM-->
            <div id="login">
                <h1>Welcome Back!</h1>

                <form action="index.php" method="post" autocomplete="off">

                    <div class="field-wrap">
                        <label>
                            Email Address<span class="req">*</span>
                        </label>
                        <input type="email" required autocomplete="off" name="email"/>
                    </div>

                    <div class="field-wrap">
                        <label>
                            Password<span class="req">*</span>
                        </label>
                        <input type="password" required autocomplete="off" name="password"/>
                    </div>
```

```
          <p class="forgot"><a href="index.php?action=forgot">Forgot Password?</a></p>


          <button class="button button-block" name="login"/>
          Log In</button>
        </form>
      </div>
<!--END LOGIN FORM-->
<!--REGISTER FORM-->
      <div id="signup">
        <h1>Sign Up for Free</h1>
        <form action="index.php" method="post" autocomplete="off">
          <div class="top-row">
            <div class="field-wrap">
              <label>
                First Name<span class="req">*</span>
              </label>
              <input type="text" required autocomplete="off" name='firstname'/>
           </div>
            <div class="field-wrap">
              <label>
                Last Name<span class="req">*</span>
              </label>
              <input type="text" required autocomplete="off" name='lastname'/>
            </div>
          </div>
          <div class="field-wrap">
            <label>
              Email Address<span class="req">*</span>
            </label>
            <input type="email" required autocomplete="off" name='email'/>
          </div>
          <div class="field-wrap">
            <label>
              Set A Password<span class="req">*</span>
            </label>
            <input type="password" required autocomplete="off" name='password'/>
          </div>
          <button type="submit" class="button button-block" name="register"/>
          Register</button>

        </form>

      </div>
      <!--END REGISTER FORM-->
    </div><!-- tab-content -->

  </div> <!-- /form -->
  <script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>

  <script src="js/index.js"></script>

  </body>
  </html>
  <?php
  require_once __DIR__ . '/../templates/_footer.php';
}
?>
```

8.6.21 The following is a commented listing of the templates/login.php source code

```
<?php
/* User login process, checks if user exists and password is correct */
// Escape email to protect against SQL injections
$email = $mysqli->escape_string($_POST['email']);
$result = $mysqli->query("SELECT * FROM users WHERE email='$email'");
if ( $result->num_rows == 0 ){ // User doesn't exist
   $_SESSION['message'] = "User with that email doesn't exist!";
   header("location: index.php?action=error");
}
else { // User exists
   $user = $result->fetch_assoc();
   if ( password_verify($_POST['password'], $user['password']) ) {
      $_SESSION['email'] = $user['email'];
      $_SESSION['first_name'] = $user['first_name'];
      $_SESSION['last_name'] = $user['last_name'];
      $_SESSION['active'] = $user['active'];
      $_SESSION['admin'] = $user['admin'];
      $_SESSION['id'] = $user['id'];
      // This is how we'll know the user is logged in
      $_SESSION['logged_in'] = true;
      header("location: ?action=profile");
   }
   else {
      $_SESSION['message'] = "You have entered wrong password, try again!";
      header("location: index.php?action=error");
   }
}
```

8.6.22 The following is a commented listing of the templates/logout.php source code

```
<?php
/* Log out process, unsets and destroys session variables */
session_start();
session_unset();
session_destroy();
?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Error</title>
  <?php include 'css/css.html'; ?>
</head>

<body>
    <div class="form">
          <h1>Thanks for stopping by Portal Gaming</h1>

          <p><?= 'You have been logged out!'; ?></p>

          <a href="index.php?action=index"><button class="button button-block"/>Home</button></a>

    </div>
</body>
</html>
```

8.6.23 The following is a commented listing of the templates/profile.php source code

```
<?php
require_once __DIR__ . '/../templates/_header.php';
?>


<body>
<form>
    <?php
    require_once __DIR__ . '/../templates/_nav.php';
    ?>
<?php


// Check if user is logged in using the session variable


    // Makes it easier to read
    $first_name = $_SESSION['first_name'];
    $last_name = $_SESSION['last_name'];
    $email = $_SESSION['email'];
    $active = $_SESSION['active'];




?>
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Welcome <?= $first_name.' '.$last_name ?></title>
  <?php include 'css/css.html'; ?>
</head>
```

```
<body>
  <div class="form">

          <h1>Welcome</h1>

          <p>
          <?php

          // Display message about account verification link only once
          if ( isset($_SESSION['message']) )
          {
              echo $_SESSION['message'];

              // Don't annoy the user with more messages upon page refresh
              unset( $_SESSION['message'] );
          }

          ?>
          </p>

          <?php

          // Keep reminding the user this account is not active, until they activate
          if ( !$active ){
              echo
              '<div class="info">
              Account is unverified, please confirm your email by clicking
              on the email link!
```

```
            </div>';
        }

        ?>

        <h2><?php echo $first_name.' '.$last_name; ?></h2>
        <p><?= $email ?></p>

        <a href="?action=logout" <button class="button button-block" name="logout"/>Log Out</button></a>

    </div>
<script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>
<script src="js/index.js"></script>

</body>
</html>
<?php
require_once __DIR__ . '/../templates/_footer.php';
```

8.6.24 The following is a commented listing of the templates/register.php source code

```
<?php
/* Registration process, inserts user info into the database
   and sends account confirmation email message
 */

// Set session variables to be used on profile.php page
$_SESSION['email'] = $_POST['email'];
$_SESSION['first_name'] = $_POST['firstname'];
$_SESSION['last_name'] = $_POST['lastname'];

// Escape all $_POST variables to protect against SQL injections
$first_name = $mysqli->escape_string($_POST['firstname']);
$last_name = $mysqli->escape_string($_POST['lastname']);
$email = $mysqli->escape_string($_POST['email']);
$password = $mysqli->escape_string(password_hash($_POST['password'],
PASSWORD_BCRYPT));
$hash = $mysqli->escape_string( md5( rand(0,1000) ) );

// Check if user with that email already exists
$result = $mysqli->query("SELECT * FROM users WHERE email='$email'") or die($mysqli-
>error());

// We know user email exists if the rows returned are more than 0
if ( $result->num_rows > 0 ) {

    $_SESSION['message'] = 'User with this email already exists!';
    header("location: ?action=error");
```

```
}
else { // Email doesn't already exist in a database, proceed...

  // active is 0 by DEFAULT (no need to include it here)
  $sql = "INSERT INTO users (first_name, last_name, email, password, hash) "
      . "VALUES ('$first_name','$last_name','$email','$password', '$hash')";

  // Add user to the database
  if ( $mysqli->query($sql) ){

    $_SESSION['active'] = 0; //0 until user activates their account with verify.php
    $_SESSION['admin'] = 0; //0 until admin sets 1 to admin it becomes admin account
    $_SESSION['logged_in'] = true; // So we know the user has logged in
    $_SESSION['message'] =

        "Confirmation link has been sent to $email, please verify
        your account by clicking on the link in the message!";

    // Send registration confirmation link (verify.php)
    $to      = $email;
    $subject = 'Portal Gaming Account Verification (noreply)';
    $message_body = '
    Hello '.$first_name.',

    Thank you for signing up on portalgaming.com

    Please click this link to activate your account:

    http://localhost/index.php?action=verify&email='.$email.'&hash='.$hash;
```

```
        mail( $to, $subject, $message_body );


        header("location: ?action=profile");


    }

    else {

        $_SESSION['message'] = 'Registration failed!';

        header("location: ?action=error");

    }


}
```

8.6.25 The following is a commented listing of the templates/reset.php source code

```
<?php
/* The password reset form, the link to this page is included
   from the forgot.php email message
*/
require 'db.php';
session_start();

// Make sure email and hash variables aren't empty
if( isset($_GET['email']) && !empty($_GET['email']) AND isset($_GET['hash']) && !empty($_GET['hash']) )
{
    $email = $mysqli->escape_string($_GET['email']);
    $hash = $mysqli->escape_string($_GET['hash']);

    // Make sure user email with matching hash exist
    $result = $mysqli->query("SELECT * FROM users WHERE email='$email' AND hash='$hash'");

    if ( $result->num_rows == 0 )
    {
        $_SESSION['message'] = "You have entered invalid URL for password reset!";
        header("location: index.php?action=error");
    }
}
else {
    $_SESSION['message'] = "Sorry, verification failed, try again!";
    header("location: index.php?action=error");
}
```

```
?>
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Reset Your Password</title>
  <?php include 'css/css.html'; ?>
</head>

<body>
    <div class="form">

          <h1>Choose Your New Password</h1>

          <form action="index.php?action=reset_password" method="post">

          <div class="field-wrap">
            <label>
              New Password<span class="req">*</span>
            </label>
            <input type="password"required name="newpassword" autocomplete="off"/>
          </div>

          <div class="field-wrap">
            <label>
              Confirm New Password<span class="req">*</span>
            </label>
            <input type="password"required name="confirmpassword" autocomplete="off"/>
          </div>
```

```
        <!-- This input field is needed, to get the email of the user -->

        <input type="hidden" name="email" value="<?= $email ?>">

        <input type="hidden" name="hash" value="<?= $hash ?>">


        <button class="button button-block"/>Apply</button>


        </form>


    </div>

<script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>

<script src="js/index.js"></script>


</body>

</html>
```

8.6.26 The following is a commented listing of the templates/reset_password.php source code

```
<?php
/* Password reset process, updates database with new user password */
require 'db.php';
session_start();
// Make sure the form is being submitted with method="post"
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Make sure the two passwords match
    if ( $_POST['newpassword'] == $_POST['confirmpassword'] ) {
        $new_password = password_hash($_POST['newpassword'], PASSWORD_BCRYPT);
        // We get $_POST['email'] and $_POST['hash'] from the hidden input field of reset.php
form
        $email = $mysqli->escape_string($_POST['email']);
        $hash = $mysqli->escape_string($_POST['hash']);

        $sql = "UPDATE users SET password='$new_password', hash='$hash' WHERE
email='$email'";
        if ( $mysqli->query($sql) ) {
        $_SESSION['message'] = "Your password has been reset successfully!";
        header("location: index.php?action=success");
        }
    }
    else {
        $_SESSION['message'] = "Two passwords you entered don't match, try again!";
        header("location: index.php?action=error");
    }
}
?>
```

8.6.27 The following is a commented listing of the templates/shop.php source code

```
<?php
/* Displays user information and some useful messages */
session_start();

// Check if user is logged in using the session variable
if ( $_SESSION['logged_in'] != 1) {
    $_SESSION['message'] = "You must log in before you can purchase!";
    header("location: index.php?action=error");

}else {
    // Makes it easier to read
    $first_name = $_SESSION['first_name'];
    $last_name = $_SESSION['last_name'];
    $email = $_SESSION['email'];
    $active = $_SESSION['active'];
}

if ( !$active ){
    $_SESSION['message'] = "You must activate your account before you can purchase!";
    header("location: index.php?action=errorActivate");
}

if(isset($_POST["add"]))
{
    if(isset($_SESSION["cart"]))
    {
        $item_array_id = array_column($_SESSION["cart"], "product_id");
        if(!in_array($_GET["id"], $item_array_id))
        {
            $count = count($_SESSION["cart"]);
            $item_array = array(
            'product_id' => $_GET["id"],
            'item_name' => $_POST["hidden_name"],
            'product_price' => $_POST["hidden_price"],
            'item_quantity' => $_POST["quantity"]
        );
            $_SESSION["cart"][$count] = $item_array;
            echo '<script>window.location="index.php?action=shop"</script>';
        }
        else
        {
            echo '<script>alert("Products already added to cart")</script>';
            echo '<script>window.location="index.php?action=shop"</script>';
        }
    }
    else
    {
        $item_array = array(
            'product_id' => $_GET["id"],
            'item_name' => $_POST["hidden_name"],
            'product_price' => $_POST["hidden_price"],
            'item_quantity' => $_POST["quantity"]
        );
        $_SESSION["cart"][0] = $item_array;
    }
}
if(isset($_GET["action"]))
```

```
{
   if($_GET["action"] == "delete")
   {
      foreach($_SESSION["cart"] as $keys => $values)
      {
         if($values["product_id"] == $_GET["id"])
         {
            unset($_SESSION["cart"][$keys]);
            echo '<script>alert("Product has been removed")</script>';
            echo '<script>window.location="index.php?action=cart"</script>';
         }
      }
   }
}
?>
```

8.6.28 The following is a commented listing of the templates/success.php source code

```
<?php
/* Displays all successful messages */
require 'db.php';
session_start();
?>
<html>
<head>
  <title>Success</title>
  <?php include 'css/css.html'; ?>
</head>
<body>
<div class="form">
  <h1><?= 'Success'; ?></h1>
  <p>
    <?php
    //receive message from session and use it
    if( isset($_SESSION['message']) AND !empty($_SESSION['message']) ):
      echo $_SESSION['message'];
    else:
      header( "location: index.php?action=home" );
    endif;
    ?>
  </p><a href="index.php?action=home"><button class="button button-block"/>Home</button></a>
</div>
</body>
</html>
```

8.6.29 The following is a commented listing of the templates/verify.php source code

```
<?php
/* Verifies registered user email, the link to this page
   is included in the register.php email message
*/
require 'db.php';
session_start();

// Make sure email and hash variables aren't empty
if(isset($_GET['email']) && !empty($_GET['email']) AND isset($_GET['hash']) && !empty($_GET['hash']))
{
    $email = $mysqli->escape_string($_GET['email']);
    $hash = $mysqli->escape_string($_GET['hash']);

    // Select user with matching email and hash, who hasn't verified their account yet (active = 0)
    $result = $mysqli->query("SELECT * FROM users WHERE email='$email' AND hash='$hash' AND active='0' AND admin='0'");

    if ( $result->num_rows == 0 )
    {
        $_SESSION['message'] = "Account has already been activated or the URL is invalid!";

        header("location: ?action=error");
    }
    else {
        $_SESSION['message'] = "Your account has been activated!";

        // Set the user status to active (active = 1)
        $mysqli->query("UPDATE users SET active='1' WHERE email='$email'") or die($mysqli->error);
        $_SESSION['active'] = 1;

        header("location: index.php?action=success");
    }
}
else {
    $_SESSION['message'] = "Invalid parameters provided for account verification!";
    header("location: ?action=error");
}
?>
```

8.6.30 The following is a commented listing of the templates/admin/add_update.php source code

```
<?php
if(count($_POST)>0) {
	require_once("db.php");
	$sql = "INSERT INTO devupdate (title, titledesc, description) VALUES ('" . $_POST["title"] . "','" . $_POST["titledesc"] . "','" . $_POST["description"] . "')";
	mysqli_query($conn,$sql);
	$current_id = mysqli_insert_id($conn);
	if(!empty($current_id)) {
		$message = "New User Added Successfully";
	}
}
?>
<html>
<head>
<title>Add New User</title>
<link rel="stylesheet" type="text/css" href="css/styles.css" />
</head>
<body>
<form name="frmUser" method="post" action="">
<div style="width:500px;">
<!--if successfully added send message-->
<div class="message"><?php if(isset($message)) { echo $message; } ?></div>
<div align="right" style="padding-bottom:5px;"><a href="index.php?action=admin" class="link"><img alt='List' title='List' src='/images/list.png' width='15px' height='15px'/> List User</a></div>
<table border="0" cellpadding="10" cellspacing="0" width="500" align="center" class="tblSaveForm">
<tr class="tableheader">
```

```
<td colspan="2">Add New User</td>
</tr>
<tr>
<td><label>Title</label></td>
<td><input type="text" name="title" class="txtField"></td>
</tr>
<tr>
<td><label>Tile Description</label></td>
<td><input type="text" name="titledesc" class="txtField"></td>
</tr>
<td><label>Description</label></td>
<td><input type="text" name="description" class="txtField"></td>

<tr>
<td colspan="2"><input type="submit" name="submit" value="Submit" class="btnSubmit"></td>
</tr>
</table>
</div>
</form>
</body></html>
```

8.6.31 The following is a commented listing of the templates/admin/add_user.php source code

```
<?php
if(count($_POST)>0) {
        require_once("db.php");
        $sql = "INSERT INTO products (p_name, image, p_price, description, link) VALUES ('" . $_POST["p_name"] . "','" . $_POST["image"] . "','" . $_POST["p_price"] . "','" . $_POST["description"] . "','" . $_POST["link"] . "')";
        mysqli_query($conn,$sql);
        $current_id = mysqli_insert_id($conn);
        if(!empty($current_id)) {
                $message = "New User Added Successfully";
        }
} ?>
<html><head>
<title>Add New User</title>
<link rel="stylesheet" type="text/css" href="css/styles.css" />
</head>
<body>
<form name="frmUser" method="post" action="">
<div style="width:500px;">
   <!--if successfully added send message-->
<div class="message"><?php if(isset($message)) { echo $message; } ?></div>
<div align="right" style="padding-bottom:5px;"><a href="index.php?action=admin" class="link"><img alt='List' title='List' src='/images/list.png' width='15px' height='15px'/> List User</a></div>
<table border="0" cellpadding="10" cellspacing="0" width="500" align="center" class="tblSaveForm">
<tr class="tableheader">
<td colspan="2">Add New User</td>
```

```
</tr>
<tr>
<td><label>Name</label></td>
<td><input type="text" name="p_name" class="txtField"></td>
</tr>
<tr>
<td><label>Image</label></td>
<td><input type="text" name="image" class="txtField" value="images/"></td>
</tr>
<tr>
<td><label>Price</label></td>
<td><input type="text" name="p_price" class="txtField"></td>
</tr>
<tr>
<td><label>Description</label></td>
<td><input type="text" name="description" class="txtField"></td>
</tr>
  <tr>
    <td><label>Link</label></td>
    <td><input type="text" name="link" class="txtField"></td>
  </tr>
<tr>
<td colspan="2"><input type="submit" name="submit" value="Submit" class="btnSubmit"></td>
</tr>
</table>
</div>
</form>
</body></html>
```

8.6.32 The following is a commented listing of the templates/admin/admin.php source code

```
<?php


require_once __DIR__ . '/../_header.php';

?>


<?php

require_once __DIR__ . '/../_nav.php';

//if not logged in display error

if ( $_SESSION['logged_in'] != 1 ) {

   $_SESSION['message'] = "Only For ADMINS!";

   header("location: index.php?action=error");

}else {

   // Makes it easier to read

   $first_name = $_SESSION['first_name'];

   $last_name = $_SESSION['last_name'];

   $email = $_SESSION['email'];

   $active = $_SESSION['active'];

   $admin = $_SESSION['admin'];

}


//if not admin display error

if(!$admin ){

   $_SESSION['message'] = "You must have admin privileges to enter!";

   header("location: index.php?action=error");

}


require_once("db.php");
```

```
$sql = "SELECT * FROM products ORDER BY id DESC";
$result = mysqli_query($conn,$sql);
?>
<html>

<head>
  <header class="w3-container w3-center w3-padding-48 w3-white">
    <h1 class="w3-xxxlarge"><b>ADMIN CURD</b></h1>
    <h6><i>Edit Portal Gaming website</i></h6>
  </header>
  <link rel="stylesheet" type="text/css" href="css/styles.css" />
</head>
<body>
<div class="w3-content" style="max-width:1400px">
  <!--SHOP CURD-->

  <div class="w3-row w3-padding w3-border">
    <div class="w3-col 29 s12">

      <div class="w3-container w3-white w3-margin w3-padding-large">
        <div class="w3-center">

<form name="frmUser" method="post" action="">

  <h1>GAMES EDIT</h1>
  <hr>
  <div class="message"><?php if(isset($message)) { echo $message; } ?></div>
```

```
 <div align="left" style="padding-bottom:5px;"><a href="index.php?action=add_user"
class="link"><img alt='Add' title='Add' src='images/add.png' width='15px' height='15px'/> Add
Game</a></div>
 <table border="0" cellpadding="10" cellspacing="1" class="tblListForm">
  <tr class="listheader">
   <td>ID</td>
   <td>Name</td>
   <td>Image</td>
   <td>Price</td>
   <td>Description</td>
   <td>CRUD Actions</td>
  </tr>
  <?php
  $i=0;
  while($row = mysqli_fetch_array($result)) {
   if($i%2==0)
    $classname="evenRow";
   else
    $classname="oddRow";
   ?>
   <tr class="<?php if(isset($classname)) echo $classname;?>">
    <td><?php echo $row["id"]; ?></td>
    <td> <?php echo $row["p_name"]; ?></td>
    <td><?php echo $row["image"]; ?></td>
    <td>&euro;<?php echo $row["p_price"]; ?></td>
    <td><?php echo $row["description"]; ?></td>
    <td><a href="index.php?action=edit_user&id=<?php echo $row["id"]; ?>"
class="link"><img alt='Edit' title='Edit' src='images/edit.png' width='15px' height='15px'
hspace='10' /></a>  <a href="index.php?action=delete_user&id=<?php echo $row["id"]; ?>"
class="link"><img alt='Delete' title='Delete' src='images/delete.png' width='15px' height='15px'
hspace='10' /></a></td>
         </tr>
         <?php
         $i++;
      }
      ?>

   </table>

</form>
            </div>
         </div>
      </div>
   </div>
</div>
   </div>
</div>
<br>
   <!--END SHOP CURD-->

   <!--DEV UPDATE CURD-->
<div class="w3-content" style="max-width:1400px">
   <div class="w3-row w3-padding w3-border">
      <div class="w3-col 29 s12">

         <div class="w3-container w3-white w3-margin w3-padding-large">
            <div class="w3-center">
            <?php
   $sql = "SELECT * FROM devupdate ORDER BY id DESC";
   $result = mysqli_query($conn,$sql);
   ?>


   <form name="frmUser" method="post" action="">
      <h1>DEV UPDATE EDIT</h1>
      <hr>
      <div class="message"><?php if(isset($message)) { echo $message; } ?></div>
      <div align="left" style="padding-bottom:5px;"><a href="index.php?action=add_update" class="link"><img alt='Add' title='Add' src='images/add.png' width='15px' height='15px'/> Add Update</a></div>
      <table border="0" cellpadding="10" cellspacing="1" class="tblListForm">
         <tr class="listheader">
            <td>ID</td>
            <td>Title</td>
            <td>Title Description</td>
            <td>Description</td>
            <td>TIMESTAMP</td>
            <td>CRUD Actions</td>
         </tr>
         <?php
         $i=0;
         while($row = mysqli_fetch_array($result)) {
            if($i%2==0)
               $classname="evenRow";
            else
               $classname="oddRow";
            ?>
            <tr class="<?php if(isset($classname)) echo $classname;?>">
              <td><?php echo $row["id"]; ?></td>
              <td> <?php echo $row["title"]; ?></td>
              <td><?php echo $row["titledesc"]; ?></td>
              <td><?php echo $row["description"]; ?></td>
              <td><?php echo $row["timestamp"]; ?></td>
              <td><a href="index.php?action=edit_update&id=<?php echo $row["id"]; ?>" class="link"><img alt='Edit' title='Edit' src='images/edit.png' width='15px' height='15px' hspace='10' /></a>  <a href="index.php?action=delete_update&id=<?php echo $row["id"]; ?>" class="link"><img alt='Delete' title='Delete' src='images/delete.png' width='15px' height='15px' hspace='10' /></a></td>
            </tr>
            <?php
            $i++;
          }
          ?>
        </table>
    </form>
<!--END UPDATE CURD-->
            </div>
          </div>
        </div>
    </div>
</div>
</form>
</div>
</body>
</html>
<?php
require_once __DIR__ . '/../_footer.php';
```

8.6.33 The following is a commented listing of the templates/admin/db.php source code

```
<?php
	$conn = mysqli_connect("localhost","root","root","portalgaming");
?>
```

8.6.34 The following is a commented listing of the templates/admin/delete_update.php source code

```
<?php
require_once("db.php");
$sql = "DELETE FROM devupdate WHERE id='" . $_GET["id"] . "'";
mysqli_query($conn,$sql);
header("Location: index.php?action=admin");
?>
```

8.6.35 The following is a commented listing of the templates/admin/delete_user.php source code

```
<?php
require_once("db.php");
$sql = "DELETE FROM products WHERE id='" . $_GET["id"] . "'";
mysqli_query($conn,$sql);
header("Location: index.php?action=admin");
?>
```

8.6.36 The following is a commented listing of the templates/admin/edit_update.php source code

```
<?php
require_once("db.php");
if(count($_POST)>0) {
	$sql = "UPDATE devupdate set title='" . $_POST["title"] . "', titledesc='" . $_POST["titledesc"] . "', description='" . $_POST["description"] . "' WHERE id='" . $_POST["id"] . "'";
	mysqli_query($conn,$sql);
	$message = "Record Modified Successfully";
}
$select_query = "SELECT * FROM devupdate WHERE id='" . $_GET["id"] . "'";
$result = mysqli_query($conn,$select_query);
$row = mysqli_fetch_array($result);
?>
<html>
<head>
   <title>Add New User</title>
   <link rel="stylesheet" type="text/css" href="css/styles.css" />
</head>
<body>
<form name="frmUser" method="post" action="">
   <div style="width:500px;">
      <div class="message"><?php if(isset($message)) { echo $message; } ?></div>
      <div align="right" style="padding-bottom:5px;"><a href="index.php?action=admin" class="link"><img alt='List' title='List' src='images/list.png' width='15px' height='15px'/> List User</a></div>
      <table border="0" cellpadding="10" cellspacing="0" width="500" align="center" class="tblSaveForm">
         <tr class="tableheader">
```

```
        <td colspan="2">Edit User</td>
      </tr>
      <tr>
        <td><label>ID</label></td>
        <td><input type="hidden" name="id" class="txtField" value="<?php echo $row['id']; ?>"><input type="text" name="userName" class="txtField" value="<?php echo $row['userName']; ?>"></td>
      </tr>
      <tr>
        <td><label>Title</label></td>
        <td><input type="text" name="title" class="txtField" value="<?php echo $row['title']; ?>"></td>
      </tr>
      <td><label>Title Description</label></td>
      <td><input type="text" name="titledesc" class="txtField" value="<?php echo $row['titledesc']; ?>"></td>
      </tr>
      <td><label>Description</label></td>
      <td><textarea rows="6" cols="80" name="description"><?php echo $row['description']; ?> </textarea></td>
      </tr>
      <tr>
        <td colspan="2"><input type="submit" name="submit" value="Submit" class="btnSubmit"></td>
      </tr>
    </table>
  </div>
</form>
</body>
</html>
```

8.6.37 The following is a commented listing of the templates/admin/edit_user.php source code

```
<?php


require_once("db.php");

if(count($_POST)>0) {

	$sql = "UPDATE products set p_name='" . $_POST["p_name"] . "', image='" . $_POST["image"] . "', p_price='" . $_POST["p_price"] . "', description='" . $_POST["description"] . "', link='" . $_POST["link"] . "' WHERE id='" . $_POST["id"] . "'";

	mysqli_query($conn,$sql);

	$message = "Record Modified Successfully";

}

$select_query = "SELECT * FROM products WHERE id='" . $_GET["id"] . "'";

$result = mysqli_query($conn,$select_query);

$row = mysqli_fetch_array($result);

?>

<html>

<head>

<title>Add New User</title>

<link rel="stylesheet" type="text/css" href="css/styles.css" />

</head>

<body>

<form name="frmUser" method="post" action="">

<div style="width:500px;">

<div class="message"><?php if(isset($message)) { echo $message; } ?></div>

<div align="right" style="padding-bottom:5px;"><a href="index.php?action=admin" class="link"><img alt='List' title='List' src='images/list.png' width='15px' height='15px'/> List User</a></div>

<table border="0" cellpadding="10" cellspacing="0" width="500" align="center" class="tblSaveForm">
```

```
<tr class="tableheader">
<td colspan="2">Edit User</td>
</tr>
<tr>
<td><label>ID</label></td>
<td><input type="hidden" name="id" class="txtField" value="<?php echo $row['id']; ?>"><input type="text" name="userName" class="txtField" value="<?php echo $row['userName']; ?>"></td>
</tr>
<tr>
<td><label>Name</label></td>
<td><input type="text" name="p_name" class="txtField" value="<?php echo $row['p_name']; ?>"></td>
</tr>
<td><label>Image</label></td>
<td><input type="text" name="image" class="txtField" value="<?php echo $row['image']; ?>"></td>
</tr>
<td><label>Price</label></td>
<td><input type="text" name="p_price" class="txtField" value="<?php echo $row['p_price']; ?>"></td>
</tr>
<tr>
<td><label>Description</label></td>
<td><textarea rows="4" cols="50" name="description"><?php echo $row['description']; ?>
</textarea></td>
</tr>
<tr>
<td><label>Link</label></td>
```

```
<td><input type="text" name="link" class="txtField" value="<?php echo $row['link']; ?>"></td>
</tr>
<tr>
<td colspan="2"><input type="submit" name="submit" value="Submit" class="btnSubmit"></td>
</tr>
</table>
</div>
</form>
</body></html>
```

8.6.38 The following is a commented listing of the templates/admin/edit_user.php source code

```
<?php


require_once("db.php");

if(count($_POST)>0) {

	$sql = "UPDATE products set p_name='" . $_POST["p_name"] . "', image='" .
$_POST["image"] . "', p_price='" . $_POST["p_price"] . "', description='" .
$_POST["description"] . "', link='" . $_POST["link"] . "' WHERE id='" . $_POST["id"] . "'";

	mysqli_query($conn,$sql);

	$message = "Record Modified Successfully";

}

$select_query = "SELECT * FROM products WHERE id='" . $_GET["id"] . "'";

$result = mysqli_query($conn,$select_query);

$row = mysqli_fetch_array($result);

?>

<html>

<head>

<title>Add New User</title>

<link rel="stylesheet" type="text/css" href="css/styles.css" />

</head>

<body>

<form name="frmUser" method="post" action="">

<div style="width:500px;">

<div class="message"><?php if(isset($message)) { echo $message; } ?></div>

<div align="right" style="padding-bottom:5px;"><a href="index.php?action=admin"
class="link"><img alt='List' title='List' src='images/list.png' width='15px' height='15px'/> List
User</a></div>

<table border="0" cellpadding="10" cellspacing="0" width="500" align="center"
class="tblSaveForm">
```

```
<tr class="tableheader">
<td colspan="2">Edit User</td>
</tr>
<tr>
<td><label>ID</label></td>
<td><input type="hidden" name="id" class="txtField" value="<?php echo $row['id']; ?>"><input type="text" name="userName" class="txtField" value="<?php echo $row['userName']; ?>"></td>
</tr>
<tr>
<td><label>Name</label></td>
<td><input type="text" name="p_name" class="txtField" value="<?php echo $row['p_name']; ?>"></td>
</tr>
<td><label>Image</label></td>
<td><input type="text" name="image" class="txtField" value="<?php echo $row['image']; ?>"></td>
</tr>
<td><label>Price</label></td>
<td><input type="text" name="p_price" class="txtField" value="<?php echo $row['p_price']; ?>"></td>
</tr>
<tr>
<td><label>Description</label></td>
<td><textarea rows="4" cols="50" name="description"><?php echo $row['description']; ?> </textarea></td>
</tr>
<tr>
<td><label>Link</label></td>
```

```
<td><input type="text" name="link" class="txtField" value="<?php echo $row['link']; ?>"></td>
</tr>
<tr>
<td colspan="2"><input type="submit" name="submit" value="Submit" class="btnSubmit"></td>
</tr>
</table>
</div>
</form>
</body></html>
```

8.6.39 *The following is a commented listing of the templates/rating/add_rating.php source code*

```
<?php

session_start();
$id = (isset($_GET['id']));
$member_id = $_SESSION['id'];


if (! empty($_POST["rating"]) && ! empty($_POST["id"])) {
    require_once ("Rate.php");
    $rate = new Rate();

    $ratingResult = $rate->getRatingByTutorial($member_id, $member_id);
    //if not empty rating then update rating
    if (! empty($ratingResult)) {
        $rate->updateRating($_POST["rating"], $ratingResult[0]["id"]);
    } else { //if rating empty add new rating
        $rate->addRating($_POST["id"], $_POST["rating"], $member_id);
    }

    $postRating = $rate->getRatingByTutorial($_POST["id"]);
    //if rating is not empty post an average rating
    if (! empty($postRating[0]["rating_total"])) {
        $average = round(($postRating[0]["rating_total"] / $postRating[0]["rating_count"]), 1);
        echo "Average Star Rating is " . $average . " from the Total " .
$postRating[0]["rating_count"] . " Ratings";
    } else {
        echo "No Ratings";
    }
}
?>
```

8.6.40 The following is a commented listing of the templates/rating/DBController.php source code

```
<?php


class DBController
{


    private $host = "localhost";


    private $user = "root";


    private $password = "root";


    private $database = "portalgaming";


    private static $conn;


    function __construct()
    {
        $this->conn = mysqli_connect($this->host, $this->user, $this->password, $this->database);
    }
    //gets connection to database
    public static function getConnection()
    {
        if (empty($this->conn)) {
            new Database();
        }
    }
    //posts database
```

```
function getDBResult($query, $params = array())
{
    $sql_statement = $this->conn->prepare($query);
    if (! empty($params)) {
        $this->bindParams($sql_statement, $params);
    }
    $sql_statement->execute();
    $result = $sql_statement->get_result();

    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $resultset[] = $row;
        }
    }

    if (! empty($resultset)) {
        return $resultset;
    }
}
//updates database
function updateDB($query, $params = array())
{
    $sql_statement = $this->conn->prepare($query);
    if (! empty($params)) {
        $this->bindParams($sql_statement, $params);
    }
    $sql_statement->execute();
}
```

```
    function bindParams($sql_statement, $params)
    {
        $param_type = "";
        foreach ($params as $query_param) {
            $param_type .= $query_param["param_type"];
        }

        $bind_params[] = & $param_type;
        foreach ($params as $k => $query_param) {
            $bind_params[] = & $params[$k]["param_value"];
        }

        call_user_func_array(array(
            $sql_statement,
            'bind_param'
        ), $bind_params);
    }
}
```

8.6.41 The following is a commented listing of the templates/rating/Rate.php source code

```
<?php
require_once "DBController.php";

class Rate extends DBController
{
   //get data from database
   function getAllPost()
   {
      $query = "SELECT products.*, COUNT(tbl_member_rating.rating) as rating_count, 
SUM(tbl_member_rating.rating) as rating_total FROM products LEFT JOIN tbl_member_rating 
ON products.id = tbl_member_rating.tutorial_id GROUP BY tbl_member_rating.tutorial_id";

      $postResult = $this->getDBResult($query);
      return $postResult;
   }
//gets rating from database
   function getRatingByTutorial($tutorial_id)
   {
      $query = "SELECT products.*, COUNT(tbl_member_rating.rating) as rating_count, 
SUM(tbl_member_rating.rating) as rating_total FROM products LEFT JOIN tbl_member_rating 
ON products.id = tbl_member_rating.tutorial_id WHERE tbl_member_rating.tutorial_id = ? 
GROUP BY tbl_member_rating.tutorial_id";

      $params = array(
         array(
            "param_type" => "i",
            "param_value" => $tutorial_id
         )
      );

      $postResult = $this->getDBResult($query, $params);
      return $postResult;
   }

   function getRatingByTutorialForMember($tutorial_id, $member_id)
   {
      $query = "SELECT * FROM tbl_member_rating WHERE tutorial_id = ? AND member_id 
= ?";

      $params = array(
         array(
            "param_type" => "i",
            "param_value" => $tutorial_id
         ),
         array(
            "param_type" => "i",
            "param_value" => $member_id
         )
      );

      $ratingResult = $this->getDBResult($query, $params);
      return $ratingResult;
   }
```

```
    //add new rating
    function addRating($tutorial_id, $rating, $member_id)
    {
        $query = "INSERT INTO tbl_member_rating (tutorial_id,rating,member_id) VALUES (?,
?, ?)";

        $params = array(
            array(
                "param_type" => "i",
                "param_value" => $tutorial_id
            ),
            array(
                "param_type" => "i",
                "param_value" => $rating
            ),
            array(
                "param_type" => "i",
                "param_value" => $member_id
            )
        );

        $this->updateDB($query, $params);
    }
    //update rating
    function updateRating($rating, $rating_id)
    {
        $query = "UPDATE tbl_member_rating SET  rating = ? WHERE id= ?";

        $params = array(
            array(
                "param_type" => "i",
                "param_value" => $rating
            ),
            array(
                "param_type" => "i",
                "param_value" => $rating_id
            )
        );

        $this->updateDB($query, $params);
    }
}
```

8.6.42 The following is a commented listing of the templates/rating/rating.php source code

```
<?php
// Here the member id is harcoded.
// You can integrate your authentication code here to get the logged in member id

require_once __DIR__ . '/../../templates/_nav.php';
$id = (isset($_GET['id']));
$member_id = $_SESSION['id'];

require_once ("Rate.php");
$rate = new Rate();
$result = $rate->getAllPost();
?>
<HTML>
<HEAD>
   <header class="w3-container w3-center w3-padding-48 w3-white">
      <h1 class="w3-xxxlarge"><b>PORTAL GAMING RATING</b></h1>
      <h6>Rate and browse games!</h6>
   </header>

   <link href="css/rating.css" rel="stylesheet" type="text/css" />
<script src="js/jquery-3.2.1.min.js" type="text/javascript"></script>
<script>function highlightStar(obj,id) {
	removeHighlight(id);
	$('.demo-table #tutorial-'+id+' li').each(function(index) {
		$(this).addClass('highlight');
		if(index == $('.demo-table #tutorial-'+id+' li').index(obj)) {
			return false;
		}
	});
}
function removeHighlight(id) {
	$('.demo-table #tutorial-'+id+' li').removeClass('selected');
	$('.demo-table #tutorial-'+id+' li').removeClass('highlight');
}

function addRating(obj,id) {
	$('.demo-table #tutorial-'+id+' li').each(function(index) {
		$(this).addClass('selected');
		$('#tutorial-'+id+' #rating').val((index+1));
		if(index == $('.demo-table #tutorial-'+id+' li').index(obj)) {
			return false;
		}
	});
	$.ajax({
	url: "index.php?action=add_rating",
	data:'id='+id+'&rating='+$('#tutorial-'+id+' #rating').val(),
	type: "POST",
   success: function(data) {
      $("#star-rating-count-"+id).html(data);
      }
	});
}
```

```
function resetRating(id) {
        if($('#tutorial-'+id+' #rating').val() != 0) {
                $('.demo-table #tutorial-'+id+' li').each(function(index) {
                        $(this).addClass('selected');
                        if((index+1) == $('#tutorial-'+id+' #rating').val()) {
                                return false;
                        }
                });
        }
} </script>
<title>PORTAL GAMING</title>
</HEAD>
<BODY>
    <table class="demo-table">
        <tbody>
            <tr>
                <th><strong>List Of Games</strong></th>
            </tr>
<?php
if (! empty($result)) {
    $i = 0;
    foreach ($result as $tutorial) {
        $ratingResult = $rate->getRatingByTutorialForMember($tutorial["id"], $member_id);
        $ratingVal = "";
        if (! empty($ratingResult[0]["rating"])) {
            $ratingVal = $ratingResult[0]["rating"];
        }
        ?>
<tr>
                <td valign="top">
                    <div class="feed_title"><?php echo $tutorial["p_name"]; ?></div>
                    <a href="<?php echo $tutorial["link"]; ?>"><img src="<?php echo
$tutorial["image"]; ?>" height="215" width="460" class="img-responsive"></a>

                    <div id="tutorial-<?php echo $tutorial["id"]; ?>"
                        class="star-rating-box">
                        <input type="hidden" name="rating" id="rating"
                            value="<?php echo $ratingVal; ?>" />
                        <ul
                            onMouseOut="resetRating(<?php echo $tutorial["id"]; ?>);">
  <?php
        for ($i = 1; $i <= 5; $i ++) {
            $selected = "";
            if (! empty($ratingResult[0]["rating"]) && $i <= $ratingResult[0]["rating"]) {
                $selected = "selected";
            }
            ?>
  <li class='<?php echo $selected; ?>'
                                onmouseover="highlightStar(this,<?php echo $tutorial["id"]; ?>);"
                                onmouseout="removeHighlight(<?php echo $tutorial["id"]; ?>);"
                                onClick="addRating(this,<?php echo $tutorial["id"]; ?>);">&#9733;</li>
  <?php }  ?>
</ul>
                        <div
                            id="star-rating-count-<?php echo $tutorial["id"]; ?>"
                            class="star-rating-count">
                        <?php

      if (! empty($tutorial["rating_total"])) {
         $average = round(($tutorial["rating_total"] / $tutorial["rating_count"]), 1);
         echo "Average Star Rating is " . $average . " from the Total " . $tutorial["rating_count"] .
" Ratings";
         ?>

                        <?php } else { ?>
                        No Ratings
                        <?php  } ?>
                        </div>

               </div>
               <div><?php echo $tutorial["description"]; ?></div>
            </td>
         </tr>
<?php
   }
}
?>
</tbody>
   </table>
</BODY>
</HTML>
```