Esports Encyclopedia

Group 15, Section 3-4

Members: Aidan Buchanan (500819557), Minseung Jeon (500884277), Kevin Tran (500967982), Qingli Wang(500801570)

Goals of Project

We wanted to understand how an encyclopedia system works and is structured. We wanted to gain a greater understanding of the topics covered in class. This includes how PHP and HTML/CSS/JavaScript work together, as well as how PHP can access the database.

Types of Content

Static Pages

aboutUs.html | index.php | header.php | headerAdmin.php | listGames.php | listPlayers.php | listSponsors.php | listTeams.php

index.php: contains a search bar to search through the database
list{name}.php: lists all {name} entities in the database either as tiles or as a list

Dynamic Pages

connect.php | deleteGame.php | deletePlayer.php | deleteSponsor.php | deleteTeam.php | editGame.php | editSponsor.php | editTeam.php | index.php | insertGame.php | insertPlayer.php | insertTeam.php | login.php | signup.php | playerForm.php | teamForm.php | updateGame.php | updatePlayer.php | updateSponsor.php | updateTeam.php | viewGame.php | viewPlayer.php | viewSponsor.php | viewTeam.php

login.php: contains login function for user to be admin (login)
signup.php: contains signup function for user to sign up

delete{name}.php: deleting the {name} page from database
insert{name}.php: inserts the text/images in the database
{name}Form.php: form to submit a new profile/data
update{name}.php: editing and updating the existing profile/data
view{name}.php: viewing the details of each profile/data

Template Pages

viewGame.php | viewPlayer.php | viewSponsor.php | viewTeam.php

view{name}.php: used as templates to view information about a particular {name} entity

Design

Our website took inspiration from the Google search bar (Figure 1) and from the Steam tile system (Figure 2). Other than that, the design of our website was mostly made in a way to keep the feeling of the website very minimalistic and was created using our own ideas.

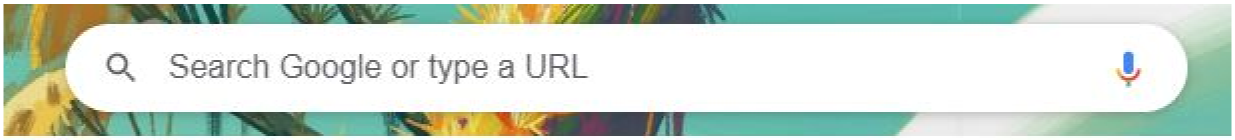

Figure 1

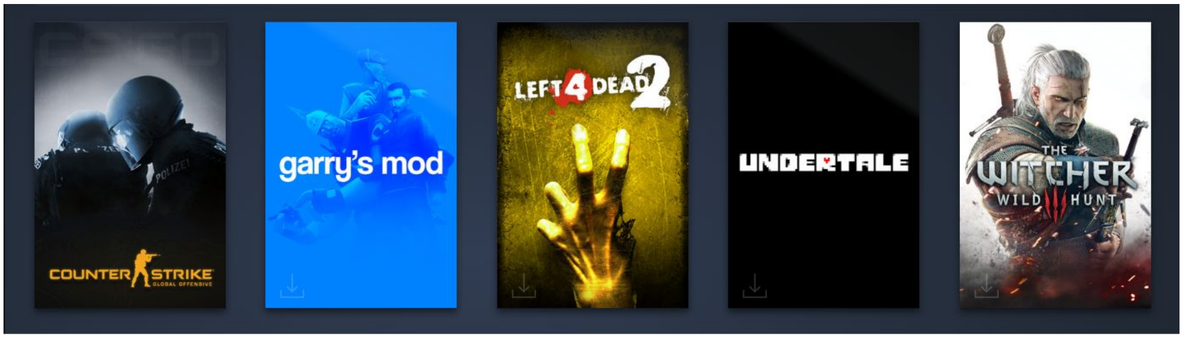

Figure 2

We have multiple form pages that allow "Admins" users to add to our database as well as a login and registration page. All of these pages use form validation using Javascript. The validation checks to see if the user's input is valid such as: containing the right amount of characters, capitalization, and checking to see if the user left the textbox blank.

Responsive Design

Our website does work the same across these major browsers: Firefox, Chrome, and Microsoft Edge.

The website takes into account the window size and adjusts accordingly. In a small window size, our header's options will collapse into a menu button which the user can press (Figure 3). In a medium and large-sized window, all web page content is adjusted accordingly.

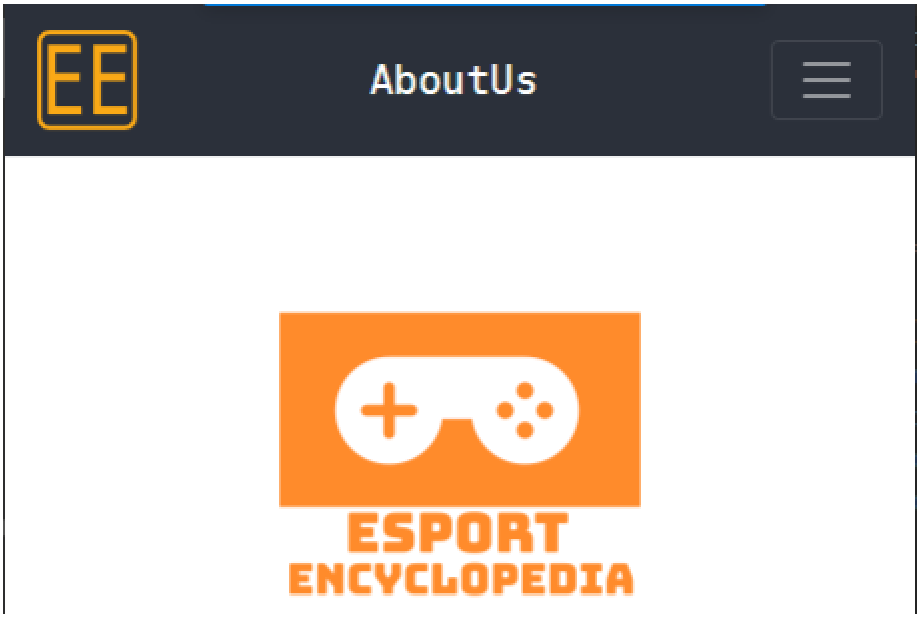

Figure 3

Accessibility

We did not take any steps into making our website accessible to those with disabilities. At most, we added alt-text to the images of our website, so those who use text-to-speech are able to understand what the images are showing. Our website utilizes minimal colouring, making it easier on the eyes when reading the content of the website.

APIs

We did not use any APIs.

Hosting Environment

We used AWS LightSail. When researching different hosting services, LightSail seemed to be the easiest in terms of creating a LAMP stack instance. LightSail provides a LAMP stack along with useful instructions on how to use it accordingly. It also provided a web-based terminal which made the server easy to access from anywhere.