Genevieve Saab­­­­

Final Project Documentation

Project Description

My final project is a simple pet simulator where you drag and drop items to your pet bunny to replenish its health bars. For example, dragging and dropping a carrot to the bunny would add another health block to the hunger bar. You can name your pet by submitting a name in the text box below the bunny. There’s a link to a separate page with the video on it.

Note: All references are contained in the code.

Link: <http://ec2-18-190-26-87.us-east-2.compute.amazonaws.com/FinalProject/>

Requirements

1. HTML5 and CSS3
   1. Note: All HTML pages contain the 5 required tags, specify the language, include the character set, and have something written in the title tag so they can pass through the validator, except header.html which is solely used to update the navigation bar.
   2. HTML can be found in these files:
      1. error.php (entire file): Displays login errors
      2. header.html (entire file, only includes <nav> tag): Stores navbar code
      3. index.html (entire file): Home page
      4. login\_form.php (entire file, except lines 23-27): The form for logging in
      5. petSim.php (entire file, except lines 1-13): Main content that users log into
      6. video.php (entire file, except lines 1-13): Page for YouTube video
   3. CSS can be found in these files:
      1. index.css (entire file): Used to style index.html
      2. myApp.css (entire file): Used to style login\_form.php
      3. petPage.css (entire file): Used to style petSim.php and video.php
      4. Very minimal CSS in style tags of error.php (lines 7-14)
   4. No tables used.
2. Consistent Design/Common Element
   1. Design: Each page uses a similar color scheme using alice blue, pink, and deep pink. All fonts are consistent.
   2. The nav bar is a common element on each page once the user is logged in. It is displayed on the top of the page and allows the user to easily flip through the pages. The logout button is also on the nav bar.
3. Organization
   1. Code is organized in their own files to keep different types of code separate and to keep the various purposes of the pages separate.
   2. CSS can easily be changed by modifying external style sheets.
   3. The nav bar can be edited in header.html, and changes will automatically be applied to all pages, and it is simple to add to new pages. This is done by having an empty div for the nav bar at the top of the body and using jQuery to load it in. This function can be found in petPage.js lines 2-4.
4. Public and Private Content
   1. Public:
      1. index.html is the first page that just sends the user to the login.
      2. login.php allows users to log into private content.
   2. Private
      1. petSim.php is the pet simulator, which must be logged into to use.
      2. video.php, which contains a YouTube video, also requires the user to be logged in to use.
   3. Login and Logout
      1. Pages that are logged into are indicated by the nav bar that appears at the top of the page and the presence of the logout button.
      2. The logout feature is kept on the nav bar for consistency.
   4. URL Hacking
      1. Protected pages contain php code in them that prevents the user from getting into the page if he/she isn’t logged in. This php code can be found in petSim.php and video.php both on lines 1-13.
5. Username/Password
   1. Username: test and Password: pass allows access into private pages.
6. PHP is included in these files:
   1. login.php (entire file): Code for logging in
   2. login\_form.php (lines 23-27): Minimal PHP is used for error checking
   3. logout.php (entire file): Code for logging out
   4. petPage.php (entire file): PHP is used to do a get request from the inputted form data and print out the data or an error.
   5. petSim.php and video.php (lines 1-13): Minimal PHP is used to make sure user can’t URL hack.
7. Get and Post
   1. Get: A get request is made in petPage.php on line 2 and line 5. On line two, the check is for the form. On line 5, the name of the pet is being retrieved.
   2. Get isn’t used for sensitive data.
   3. Post: A post request is made in login.php on line 26, 35, and 36. Line 26 is for the form. Line 35 is for the username, and line 36 is for the password. Post is used for the login to keep the user’s data private.
8. Form (other than login form)
   1. A form is used in petSim.php on lines 99-103. This provides an input box where the user can submit a name for their pet.
9. User Input Feedback
   1. If the login is incorrect, an error box appears that informs the user that the login wasn’t right. This is done in the handle\_login() function in login.php lines 34-55.
   2. If a user tries to submit a null name for their pet, doesn’t enter a string, or uses illegal characters an error is presented where the bunny’s name is supposed to be (on top of the middle box). These errors are handled in petPage.php lines 6-14.
10. Images
    1. There are many images on petSim.php. They can be found in the code on lines 39-49 and line 53. These images are for all the items and the picture of the bunny.
11. Video
    1. There is a YouTube video on video.php lines 36-37.
12. JavaScript
    1. In petPage.js, JavaScript can be found on lines 10-49.
       1. 10-12: JavaScript variables declared and defined.
       2. 14-26: The barCount() function deletes health blocks.
       3. 29: setInterval calls barCount() to delete the health blocks on a timer.
       4. 31-49: getContent() uses AJAX to get the input pet name and prints it in the correct place.
13. jQuery
    1. In petPage.js, jQuery can be found on lines 2-8 and 52-77.
       1. 2-4: This function loads the nav bar.
       2. 6-8: This makes objects draggable if they contain the class draggable.
       3. 52-77: This handles droppable.
14. jQuery UI
    1. In petPage.js draggable() (lines 6-8) and droppable() (lines 52-77) are both used.
15. AJAX
    1. In petPage.js, the getContent() function (lines 31-49) uses AJAX. This utilizes the code in petPage.php to get the form data and uses an XMLHttpRequest to send this data so it is printed for the bunny name.
16. Purpose
    1. The purpose of this web app is to provide a cute little game to take care of a pet bunny, and also contains all the technical requirements of this assignment.
17. Document
    1. This document contains the link and explains how I met all the criteria.