Securing the admin panel through sessions

You may have your own approach, but this is one method of stopping users, who have not logged in as admins, being able to reach the admin panel (which is an assessment requirement).

The code used in this exercise is similar to the code we used to create sessions and login system in our previous exercises, so you should be able to adapt your previous code to fit.

First, create an admin table and enter admin credentials.

admin_login database table (note: admin_id is not required as we only check against admin username and password)

admin_id	admin_name	admin_password
1	admin	admin

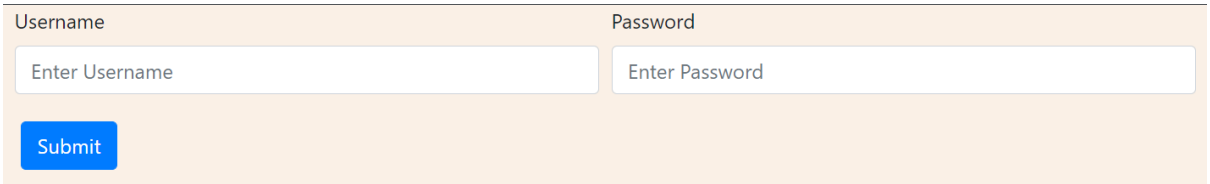
Then add a link on one of your webpages that admins can use to access their own admin login form.

```
<div class="col-12 text-right">
    <a href="adminlogin.php" class="btn btn-link">Admin Login</a>
</div>
```

Then create the admin login form – I have created a new page called adminlogin.php to hold the login form.

```
<form action="loginCheck.php" method="POST">
<div class="form-row">
  <div class="form-group col-6">
    <label for="Username">Username</label>
    <input type="text" class="form-control" name="username" placeholder="Enter Username">
  </div>
  <div class="form-group col-6">
    <label for="Password">Password</label>
    <input type="password" class="form-control" name="password" placeholder="Enter Password">
  </div>
</div>
  <button type="submit" name="submit" class="btn btn-primary">Submit</button>
</form><!-- end of admin login form -->
```

How the form looks.



The admin login form will take the users username and password and POST the data to another form called loginCheck.php which checks to see if use with correct username and password combination exists in the db table.

loginCheck.php file is based on the similar syntax we used when we created our first login forms earlier in the course.

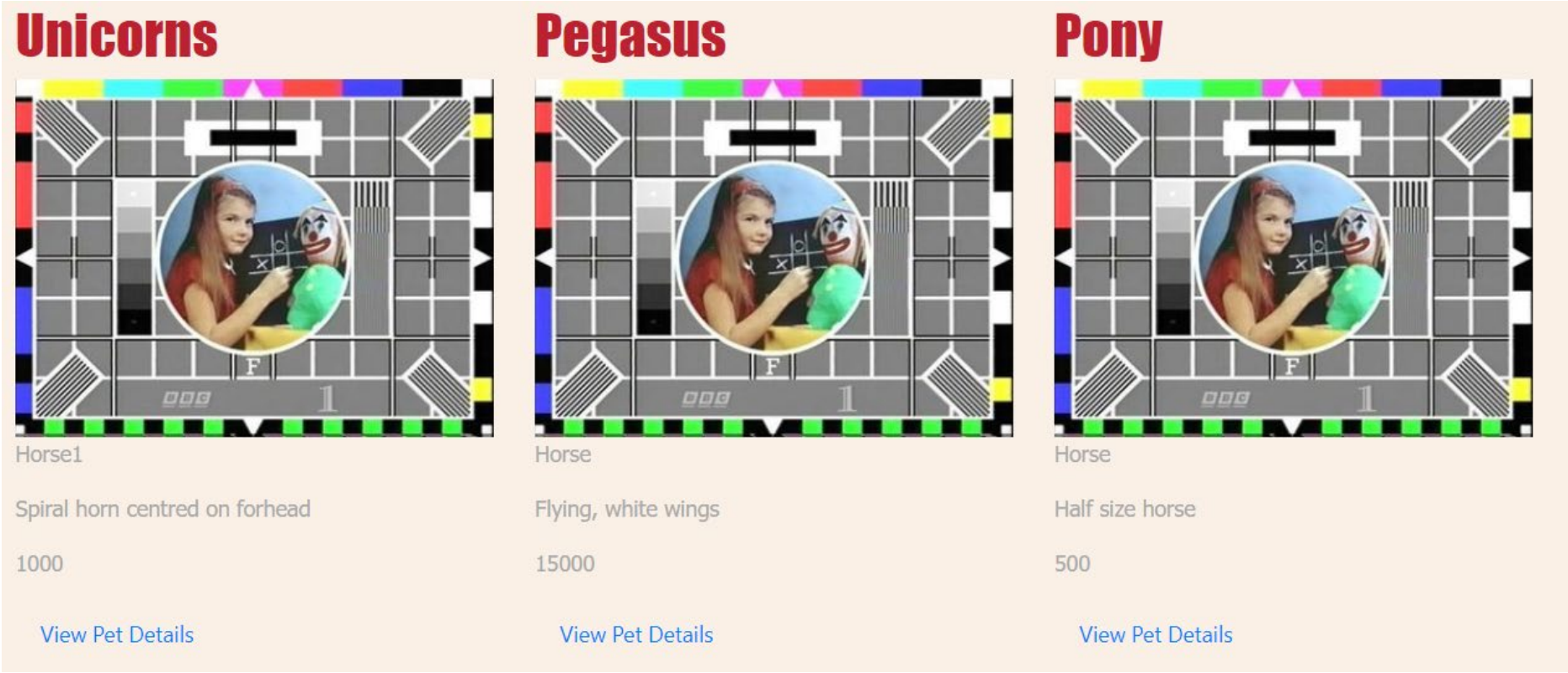
```php
<?php
//connection details
include('connection.php');
//posted values assigned to variables
$username = $_POST['username'];
$password = $_POST['password'];

//SQL query to check combination of username and password exist in db table
$query = "SELECT * FROM admin_login
          WHERE admin_name = '$username'
          AND admin_password = '$password'";

$result = mysqli_query($conn, $query)
or die ("couldn't run query");

//count number of rows returned and assign to a variable
$count = mysqli_num_rows($result);

//check username and password exist in db table
if($count==1)
{
    //set session called adminlogin
    session_start();
    $_SESSION['adminlogin']="admin";

    //header to redirect to the admin panel page
    header("Location:http://webdev.edinburghcollege.ac.uk/~HNDWEBXX/admin_panel.php");
}
else
{
    echo('wrong login details');
}
?>
```

On the admin_panel.php file we will have all the necessary code to perform the required admin panel functionality. I haven't added it here but the code below checks to see if a session called adminlogin is set. If it is set and the IF condition is true you can display the admin panel to users.

```php
<?php
session_start();
if(isset($_SESSION['adminlogin']))
{
echo("welcome to the admin area");
//INSERT ADMIN PANEL CODE HERE (BETWEEN BRACES)
}
else
{
echo("you are not allowed access here");
}
?>
```

Booking process using GET method and show book buttons if adult user is logged in

Here is the front of the website with dynamic content and ‘View Pet Details’ links available. It is up to you if you show or hide these if the user is not logged in.



Here is the code for the container section of the above dynamic webpage. Note: it does not have to be three column layout. It can be one column, one event on top of another if your prefer.

In the code below, we append dynamic content to our button. This will pass a unique identifier (from the db table) to a file called bookPet.php.

```
<div class="container" style="background-color:linen">
<!-- container class for content -->
<?php
session_start();
include('connection.php');

$query = "SELECT * FROM pet";
$result = mysqli_query($conn, $query)
or die ("couldn't run query");
echo'<div class="row">';//start of row
while($row = mysqli_fetch_array($result, MYSQLI_ASSOC))
{

    echo"<div class='col-md-6 col-lg-4'>";
      echo"<h1>" . $row['PetID'] . "</h1>";
      echo"<img src='images/". $row['image'] ."' class='img-fluid'>";
      echo"<p>" . $row['PetType'] . "</p>";
      echo"<p>" . $row['PetDescription'] . "</p>";
      echo"<p>" . $row['Price'] . "</p>";
      echo'<a class="button btn default centre details" href="bookPet.php?id=' . $row['PetID'] . '">View Pet Details</a>';
      echo'</div>';

}
echo'</div>';//end of row
echo'<hr/>';
?>
</div><!-- end of container -->
```

```
echo'<a class="button btn default centre details" href="bookPet.php?id=' . $row['PetID'] . '">View Details</a>';
```

If we chose to hide this button (link) we can check if the **session called adult is set** or not (line 66). If the session is set, then echo the button code at line 68. If it is not set, it will not show the button. In the code below, we cannot see the session being set so we would not see the 'view pet details' buttons for each pet.

Note: The sessions should be set during the login process. If user is >= 18 then set adult session, or if user is < 18 then set junior session (see topic 14 on the Designing and Developing an Interactive Product Moodle page).

```
47  <div class="container" style="background-color:linen">
48  <!-- container class for content -->
49  <?php
50  session_start();
51  include('connection.php');
52
53  $query = "SELECT * FROM pet";
54  $result = mysqli_query($conn, $query)
55  or die ("couldn't run query");
56  echo'<div class="row">';//start of row
57  while($row = mysqli_fetch_array($result, MYSQLI_ASSOC))
58  {
59
60      echo"<div class='col-md-6 col-lg-4'>";
61        echo"<h1>" . $row['PetID'] . "</h1>";
62        echo"<img src='images/". $row['image'] ."' class='img-fluid'>";
63        echo"<p>" . $row['PetType'] . "</p>";
64        echo"<p>" . $row['PetDescription'] . "</p>";
65        echo"<p>" . $row['Price'] . "</p>";
66        if(isset($_SESSION['adult']))
67        {
68        echo'<a class="button btn default centre details" href="bookPet.php?id=' . $row['PetID'] . '">View Pet Details</a>';
69        }
70        echo'</div>';
71
72  }
73  echo'</div>';//end of row
74  echo'<hr/>';
75  ?>
```

Buttons are hidden for each pet.

Unicorns

Horse1

Spiral horn centred on forhead

1000

Pegasus

Horse

Flying, white wings

15000

Pony

Horse

Half size horse

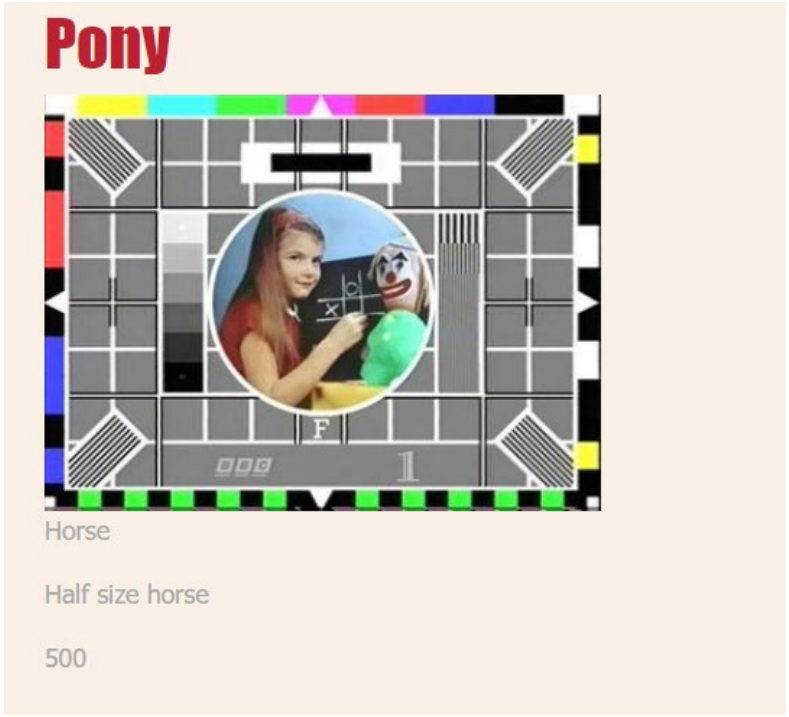
500

```php
session_start();
$_SESSION['adult'] = 'adultuser';//setting a session called adult
```

When the button is shown and the user clicks on it, it will pass the value of the PetID column to a file called bookPet.php.

On the **bookPet.php** file, we need our connection credentials because it is going to retrieve data from the db table. Then we need to use the key from the button code (id) to get the value – the dynamic content from the db table (the PetID).

```
$id = $_GET['id'];
$query = "SELECT * FROM pet WHERE PetID = '$id'";
$result = mysqli_query($conn, $query)
```



```php
<div class="container" style="background-color:linen">
<!-- container class for content -->
<?php
include('connection.php');

$id = $_GET['id'];
$query = "SELECT * FROM pet WHERE PetID = '$id'";

$result = mysqli_query($conn, $query)
or die ("couldn't run query");
echo'<div class="row">';//start of div with row class
while($row = mysqli_fetch_array($result, MYSQLI_ASSOC))
{
      $altTag = $row['PetDescription'];//uses the description from db table to create dynamic alt tag - line 63
      echo"<div class='col-md-6 col-lg-4'>";
      echo"<h1>" . $row['PetID'] . "</h1>";
      echo"<img src='images/". $row['image'] ."' class='img-fluid' alt='Poster for the film $altTag'>";
      echo"<p>" . $row['PetType'] . "</p>";
      echo"<p>" . $row['PetDescription'] . "</p>";
      echo"<p>" . $row['Price'] . "</p>";
      echo'</div>';

}
echo'</div>';//end of div with row class
echo'<hr/>';
?>
</div><!-- end of container -->
```

To give this context for the Limelight cinema project, at this point, we have effectively chosen the film we want to see as part of our booking process.

Now, we can select how many tickets we want. Note: at this point, you could also ask the user what date and time they wish to see the selected film at.

So, now I will add a form to book how many tickets I need to my **bookPet.php** file.

Pegasus

Horse

Flying, white wings

15000

Number of Tickets

3

Book

```
<form action="book.php" method="POST">
<div class="form-group">
    <label>Number of Tickets</label>
    <input type="number" class="form-control" name="tickets">
</div>
<div class="form-group">
   <input type="hidden" class="form-control" name="hidden" value="<?php echo $id; ?>">
</div>
<button type="submit" class="btn btn-primary">Book</button>
</form>
</div><!-- end of container -->
```

The form has two form (input) fields, the first one has the number attribute and a key named “tickets”. This will pass form data based on how many tickets the user needs.

The second form field, has type attribute hidden and a key named hidden. As the form field is hidden we assign the value to be passed using the value attribute. In our example, it passes the value of the $id variable that is set using the GET method. `$id = $_GET['id'];`

Remember, this value is the unique identifier from the db table for the chosen PetID.

This from has now passed form data based on the PetID and number of tickets (filmID and number of tickets for your project) to a file called **book.php**.

book.php is the eTicket where information on, at a minimum, the film title and number of tickets is shown to the user.

On our book.php page the form data is posted and assigned to variables to create a simple eTicket.

```php
<?php
include('connection.php');
$id = $_POST['hidden'];
$tickets = $_POST['tickets'];
echo("You have booked " . $tickets . " for the showing of " . $id);
```

You have booked 3 for the showing of Pegasus

Hiding the Booking Form (ability to select number of tickets and book button)

We can use the same process for hiding the button, if an adult user is not logged in, for hiding the booking form. Check if the adult session is set and if it is set the true condition (in the braces {}) is run – this is where we will place the code for our HTML form.

See code below.

```php
if(isset($_SESSION['adult']))
{
?>
<form action="book.php" method="POST">
<div class="form-group col-md-2">
    <label>Number of Tickets</label>
    <input type="number" class="form-control" name="tickets">
</div>
<div class="form-group">
   <input type="hidden" class="form-control" name="hidden" value="<?php echo $id; ?>">
</div>
<button type="submit" class="btn btn-primary">Book</button>
</form>
</div><!-- end of container -->

<?php } ?>
```

Pegasus

Horse

Flying, white wings

15000