Universitat de Barcelona

Computació Orientada al Web

Práctica 2

Arthur Font Gouveia - 20222613

Barcelona

2022

Índice

1. Introducción

El objetivo de esta práctica fue añadir las funcionalidades de servidor y gestionar una base de datos usando PhpMyAdmin y MySQL, basándose en la página web desarrollada en la práctica anterior. Las nuevas funcionalidades fueron codificadas manualmente usando un editor de texto.

2. Apartado 1. Implementación de las funcionalidades de servidor (PHP)

En este apartado, he implementado la funcionalidad de servidor, simulando una búsqueda de alojamientos. Para eso, ha sido necesario crear dos páginas (**cliente.php** y **servidor.php**). En el fichero cliente.php (antiguo cliente.html), he añadido un formulario horizontal que contiene los campos *destination*, *checkin*, *checkout* y *guests* y un botón que es utilizado para submeter el formulario. A continuación están ilustradas las funcionalidades del código más relevantes:

```
…
<div class="row destFinder my-3 mx-1 p-3">
        <form action="http://localhost/COW/S2_3/A1/servidor.php" method="POST">
          <div class="form-row align-items-center">
            <div class="col form-group">
              <label for="destination">Destination</label>
              <input type="text" name="destination" id="destination"
             placeholder="e.g. Barcelona" size="20" minlength="1" maxlength="20">
            </div>
            …
            <div class="col form-group">
              <label for="guests">Guests</label><br>
              <input type="number" name="guests" min="1" max="20" id="guests"
value="1">
        …
        </form>
…
```

Los atributos **action** y **method** de la etiqueta **<form>** fueron utilizados para direccionar el formulario para un fichero *handler* y definir el método del formulario, respectivamente.

El control de los datos fue realizado en el lado del cliente y en el lado del servidor. Por parte del cliente fueron utilizados los atributos **minlength** y **maxlength**

para definir un número mínimo y un número máximo de caracteres. También fueron utilizados los atributos **min** y **max** para definir el valor mínimo y el máximo aceptado para el número de huéspedes y atributo **value** para mejorar la experiencia del usuario, configurando por definición el número de huéspedes a 1.

Por parte del servidor, se ha implementado la función **check_dates()** para validar las fechas de la búsqueda y se ha utilizado la función **htmlspecialchars()** del PHP para evitar la inyección de código en la variable *destination*. Además, fueron utilizados regex para las variables *destination* y *guests*, comprobados por medio de la función **preg_match()** del PHP. También se usaron las excepciones por medio de los comandos **try**, **catch** y **throw** para redirigir el usuario a otras páginas, notificando el tipo de error específico que ha ocurrido.

```
…
<?php
   function check_dates($d1, $d2) {
      return (strtotime($d2) > strtotime($d1) && strtotime($d1) >=
strtotime(date("Y/m/d")));
   }
   try {
      if ($_SERVER["REQUEST_METHOD"] == "POST") {
         $dest = htmlspecialchars($_POST["destination"]);
         …
         $regex_dest = "/^[a-zA-Z -]{1,20}$/"; // Max: 20 characters
         $regex_guests = "/^[1-9]{1,2}$/"; // Max: 100 guests
         …
         if (!preg_match($regex_dest, $dest)) {
            throw new Exception('Destination field must contain 1 to 20 characters!');
         }
         …
   } catch(Exception $e){
      die('Message: '.$e->getMessage());
   }
?>
…
```

3. Apartado 2. Gestión de una base de datos usando PhpMyAdmin

En este apartado se ha realizado la conexión la base de datos e implementado dos *queries* necesarias para proveer las nuevas funcionalidades.

La primera *query* implementó la funcionalidad de mostrar el resultado de la búsqueda de alojamientos disponibles en un destino específico. Para eso, recibió los parámetros enviados por el formulario de la sección anterior. La conexión fue realizada utilizando la librería **PDO** de PHP de la siguiente forma:

```
$conn = new PDO("mysql:host=$servername;dbname=world", $username, $password);
…
$rows = $conn->query("SELECT * FROM cities WHERE name = $dest");
```

El resultado de la *query* (**$rows**) fue mostrado en la página **servidor.php** utilizando una tabla. La función **foreach()** de PHP fue responsable por rellenarla de la siguiente forma:

```
$i = 1;
foreach ($rows as $row) {
echo '<tr>
        <th scope="row">'.$i.'</th>
        <td>'.$row['name'].'</td>
        <td>'.$row['district'].'</td>
        <td>'.$row['country_code'].'</td>
        <td>'.$row['population'].'</td>
    </tr>';
$i++;
```

La segunda *query* implementó la funcionalidad de adicionar un nuevo cliente a la base de datos. Para eso, fue creado un formulario con los campos *name*, *email* y *password* que se puede acceder desde el botón de *Sign up* del header de la página principal. El formulario de la página **sign_up.php** envía los datos a la página **new_user.php**, que es responsable por realizar la conexión con la base de datos y realizar la *query* representada abajo:

```
// Insert new user
$name_quo = $conn->quote($name);
$email = $conn->quote($email);
$pwd = $conn->quote($pwd)
$rows = $conn->exec("INSERT INTO students (id, name, email, password)
VALUES ($id, $name_quo, $email, $pwd)");
…
```

El método **quote()** fue utilizado para codificar las variables que se usan en la query para evitar inyección en el SQL y el método **exec()** fue utilizado para efectuar la *query* de inserción del nuevo cliente.