School of Media Arts and Technology

BSc (Hons) **[Computing]**

Academic Year 2019-2020

[Evon Agatha Lobo]

[Analysis and Design Report]

Assessment 1

Developing for The Internet

Tutor: Nicholas Whitelegg

DATE: 01 May 2020

Analysis and Design Report

In this report, all the assessments tasks that are implemented will be discussed.

There will include snippets of html code and php scripts for better understanding. The link between every html file along with php scripts and how they operate will be established in a clear manner.

Tasks:

a) Allow a user to add a new point of interest (POI) along with providence of fields as you can see in the html form in figure 1. This should add a record to the pointsofinterest table.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="stylesheet.css">
    <title>Home Page: Hotspots </title>
</head>
<body>
<h2>Welcome to Hotspots! </h2>
    <h3>Add a New Point of Interest</h3>
        <div class= "container1">
          <form method="post" action="hotspot.php">
            <h5><label for="username">Username:</label></h5>
            <input type="text" id="username" name="username"/>
            <br/>
            <h5><label for="name">Name of Place:</label></h5>
            <input type="text" id="name" name="name"/>
            <br/>
            <h5><label for="type">Type of Place:</label></h5>
            <input type="text" id="type" name="type"/>
            <br/>
            <h5><label for="region">Region:</label></h5>
            <input type="text" id="region" name="region"/>
            <br/>
            <h5><label for="description">Description:</label></h5>
            <input type="text" id="description" name="description"/>
            <br/>
            <input type="submit" value="Done" />
          </form>
        </div>
        <br>
        <br>
```

Figure 1.:Hotspot.html

```
<?php
echo "<link rel='stylesheet' href='stylesheet.css'>";
$a= $_POST["username"];
$b= $_POST["name"];
$c= $_POST["type"];
$d= $_POST["region"];
$e= $_POST["description"];

$conn = new PDO ("mysql:host=localhost;dbname=assign246;", "assign246", "coefoase");
$statement = $conn->prepare("INSERT INTO pointsofinterest (username, name, type, region, description) VALUES (?, ?, ?, ?, ?)");
$statement->execute([$a, $b, $c, $d, $e]);
echo "<h5> You have added a point of interest with the following details: <br> Name: $b, <br> Username: $a, <br> Type: $c, <br> Region: $d, <br>
description: $e </h5>";
echo "<p><a href='hotspot.html'> Back to Home Page</a></p>";
?>
```

Figure 2.:Hotspot.php

Once a user logs in (through login.html and login.php), they are led to “Home Page: Hotspots” (hotspot.html). On this page, the user is allowed to add a point of interest by filling in the form (figure 1), which has an action of “hotspot.php” which verifies all fields filled in by the user and then adds all inputted details to the database.

In the PHP script (figure 2), the inputted values from the html form are retrieved and assigned with $a, $b, $c etc. variables. After connecting to the database (line 9 in figure 2), prepared statements are used along with SQL query. Using a SQL INSERT statement, information is asked to be inserted into the pointsofinterest table on the database, in columns named the “username”, “name”, “type”, “region” and “description” with values “?” (placeholder). However, the next prepared statement then executes variables $a, $b, $c, $d, $e etc which basically uses information stored in these variables to fill in columns in the database, hence adding a record. Once this is done, the user receives a confirmation message (as seen in the echo statement, line 12).

When the user then clicks the link “Back to Home Page” the user then returns to hotspot.html. Below the “Add a POI” form is another form which allows users to search for a POI by region. This leads to task b.

b) Allow a user to search for a POI by region. A user should be able to enter a region. Once they have entered the region, all POIs in that region should appear. Each search result should contain a hyperlink labelled "Recommend".

```
32      <h3>Search a Point of Interest</h3>
33          <div class="container2">
34          <form method="get" action="searchresults.php">
35              <h5><label for="POI">Search POI:</label></h5>
36              <input name="search" placeholder="Search For a Region" id="search"/>
37              <br/>
38              <br/>
39              <input type="submit" value="Search" />
40          </form>
41
42          </form>
43          </div>
44
45  </body>
46  </html>
```

Figure 3.: Hotspot.html

```php
<?php
echo "<link rel='stylesheet' href='stylesheet.css'>";
$a= $_GET["search"];
$conn = new PDO ("mysql:host=localhost;dbname=assign246;", "assign246", "coefoase");
 $statement = $conn->prepare("select * from pointsofinterest where region=?");
 $statement->execute([$a]);
try {
    while($row=$statement->fetch(PDO::FETCH_ASSOC))
    {
        echo "<p>";
        echo "<h2>Point of Interest</h2>";
        echo "<br>";
        echo " Name:". $row["name"] ."<br/> ";
        echo "<br>";
        echo " Type of Place:". $row["type"] ."<br/> ";
        echo "<br>";
        echo " Region:". $row["region"] ."<br/> ";
        echo "<br>";
        echo " Country:". $row["country"] ."<br/> ";
        echo "<br>";
        echo " Description:". $row["description"] ."<br/> ";
        echo "<br>";
        echo "<a href='recommend.php?ID=" . $row["ID"]  . "'>Recommend This Region</a>";
        echo "<br>";
        echo "<a href='review.php?ID=" . $row["ID"]  . "'>Check Reviews</a>";
        echo "</p>";

    }
}
// Catch any exceptions (errors) thrown from the 'try' block
catch(PDOException $e)
{
    echo "Error: $e";
}
echo "<p><a href='hotspot.html'> Back to Home  Page</a>";
?>
```

Figure 4.: searchresults.php

Figure 3 depicts the search form created on hotspot.html (Home Page). This form has its action as “searchresults.php” which is seen in figure 4. This html form simply asks for a user to input a region and click search.

In the PHP script as seen in figure 4, $a variable is used to store the particular region searched by the user. The connection to the database is then established and prepared statements along with SQL queries are used to retrieve information. A SELECT SQL statement is used which, in this case, requests to select a region from the pointsofinterest table. The region is defined with a placeholder (?). The next statement executes $a variable, giving input to the place holder.

The next section begins with a try catch block in case any exception is thrown. Right after is a while loop which finally outputs all information that is retrieved from the database, using a region that the user entered. This while loop is meant to read the pointofinterest table and find all points of interests (all fields including Name, Type of Place, Country, Description along with region) with the region that was searched initially and display them. The loop runs until the last POI of this region is found.

As you can see in Figure 4, there is a link added below each POI displayed that says “Recommend This Region” which leads to task c).

c) Allow a user to recommend a POI. This should add one to the recommended column for that POI.

```php
<?php
echo "<link rel='stylesheet' href='stylesheet.css'>";
$conn = new PDO ("mysql:host=localhost;dbname=assign246;", "assign246", "coefoase");
$id = $_GET["ID"];
 $statement = $conn->prepare ("UPDATE pointsofinterest SET recommended=recommended+1 WHERE ID=? ");
 $statement->execute([$id]);
echo "<p>Thank you! This region has been added to recommended</p><br />";
echo "<p><a href='hotspot.html'> Back to Home  Page</a></p>";

?>
```

Figure 5.: Recommend.php

The “Recommend This Region” link in figure 4 leads to recommend.php as seen in figure 5. ID of this region is stored in the variable $id. Prepared statements are then used along with SQL statement within, to add 1 to the “Recommended” column in the pointofinterest table on the database. An UPDATE SQL statement fulfils this by setting recommended to “recommended+1” where ID=? (placeholder). This placeholder is then executed in the next line with the $id variable, hence adding one to the recommended column of a particular POI (selected by the user). The user will then be able to see a confirmation message as appears in figure 5.

On searchresults.php (figure 4), the user can also click on another link called “Check Reviews” which leads to task d).

d) Allow a user to view all reviews for a given POI.

```php
<?php
echo "<link rel='stylesheet' href='stylesheet.css'>";
$a= $_GET["ID"];
$conn = new PDO ("mysql:host=localhost;dbname=assign246;", "assign246", "coefoase");
 $statement = $conn->prepare("select * from poi_reviews where poi_id=?");
 $statement->execute([$a]);
try {
    while($row=$statement->fetch(PDO::FETCH_ASSOC))
    {
        echo "<p>";
        echo " Reviews For This POI:". $row["review"] ."<br/> ";
        echo "</p>";
    }
}
// Catch any exceptions (errors) thrown from the 'try' block
catch(PDOException $e)
{
    echo "Error: $e";
}
echo "<p><a href='hotspot.html'> Back to Home  Page</a></p>";

?>
```

Figure 6.: Review.php

Figure 6 displays review.php which a user can reach by clicking the “Check Reviews” link in figure 4. Once a user searches for a region on the “Search for a POI” form, they are able to view all POI’s of that particular region, and as stated above, they can recommend a region as well as check reviews for them.

When a user clicks the “Check Reviews” region, the review.php script is run. As seen in figure 6, $a stores the ID. Connection to the database is then established and prepared statements are used to retrieve data from the database. This prepared statement along with a SELECT SQL statement (line 5), is requesting data where poi_id=? (placeholder) from the poi_reviews table on the database. The placeholder is then inputted with data by executing $a in the next statement.

A try catch block is put in place in case any exception is thrown. A while loop is then used to retrieve all possible reviews for a particular poi_id. This loop runs until the last one is found. All retrieved reviews from the database are then displayed to the user.