CO551
Open Source Systems

CW2

21611431

1st May 2018

Computing & Web Development
Buckinghamshire New University

Summary

Section 1

Add Modules to the Database

The modules were added to the database by importing an SQL file into phpMyAdmin.

Figure 1.1: newmodules.sql

```
Insert into `module`(`modulecode`, `name`, `level`) values
('CO158', 'XML', '1'),
('CO265', 'Web Services', '2'),
('CO366', 'Mobile Systems', '3');
```

Section 2

Students Display Script

The student display script gets the students' data from the database and displays it on a table that is also a form, since we use a checkbox to select the students to be deleted from the system.

http://intweb.bucks.ac.uk/~21611431/students.php

Figure 2.1: students.php

```
<?php
  include("_includes/config.inc");
  include("_includes/dbconnect.inc");
  include("_includes/functions.inc");

  //check if logged in
  if(isset($_SESSION['id'])){

    echo template("templates/partials/header.php");
    echo template("templates/partials/nav.php");

    //Build SQL statement that selects students
    $result = $conn -> query("SELECT * FROM student");

    //prepare page content
    $data['content'] .= "<form action=
    'delete.php' method='post'>";
    $data['content'] .= "<div class='page-header'>";
    $data['content'] .= "<h1>Students</h1>";
    $data['content'] .= "</div>";
    $data['content'] .=
    "<table class='table table-hover table-condensed table-bordered'>";
    $data['content'] .= "<thead class='thead-light'>";
```

```
20      $data['content'] .= "<tr><th>Student ID</th>";
21      $data['content'] .= "<th>DOB</th><th>First Name</th>";
22      $data['content'] .= "<th>Last Name</th>";
23      $data['content'] .= "<th>House</th><th>Town</th>";
24      $data['content'] .= "<th>County</th><th>Country</th>";
25      $data['content'] .= "<th>Postcode</th><th>Selected</th>";
26      $data['content'] .= "</tr></thead><tbody>";

27      while($row = $result -> fetch()) {
28        $data['content'] .= "<tr><td>$row[studentid]</td>";
29        $data['content'] .= "<td>$row[dob]</td>";
30        $data['content'] .= "<td>$row[firstname]</td>";
31        $data['content'] .= "<td>$row[lastname]</td>";
32        $data['content'] .= "<td>$row[house]</td>";
33        $data['content'] .= "<td>$row[town]</td>";
34        $data['content'] .= "<td>$row[county]</td>";
35        $data['content'] .= "<td>$row[country]</td>";
36        $data['content'] .= "<td>$row[postcode]</td>";
37        $data['content'] .= "<td>";
38        $data['content'] .= "<div class='form-check'>";
39        $data['content'] .= "<input type='checkbox'";
40        $data['content'] .= "class='form-check-input";
41        $data['content'] .= "position-static' name='selected[]'";
42        $data['content'] .= "value='$row[studentid]'/>";
43        $data['content'] .= "</div></td></tr>";
44      }
45      $result -> closeCursor();
46      $data['content'] .= "</tbody></table>";
47      $data['content'] .= "<br/>";
48      $data['content'] .= "<div class='form-group'>";
49      $data['content'] .= "<a href='addstudents.php' class='btn";
50      $data['content'] .= "btn-primary btn-md mr-2'";
51      $data['content'] .= "role='button'>Add Student</a>";
52      $data['content'] .= "<input class='btn btn-danger btn-md'";
53      $data['content'] .= "type='submit' value='Delete'/>";
54      $data['content'] .= "</form>";
55      $data['content'] .= "</div>";
56      //render the template
57      echo template("templates/default.php", $data);
58    }
59    else{
60      header("Location: index.php");
61    }

62    echo template("templates/partials/footer.php");
```

```
63          ?>
```

Section 3

Delete Students Script

The delete students script runs on the background and deletes the students that were selected on the table, redirecting the user back to the students page afterwards.

http://intweb.bucks.ac.uk/~21611431/delete.php

Figure 3.1: delete.php

```php
<?php
  include("_includes/config.inc");
  include("_includes/dbconnect.inc");
  include("_includes/functions.inc");

  if(isset($_SESSION['id']))
  {
    $select = $_POST['selected'];
    //Building the query depending on the selected records
    if(!empty($select))
    {
      $lastElement = end($select);
      $sql = "Delete from student where ";
      foreach($select as $stu)
      {
        if(strcmp($stu, $lastElement) != 0)
        {
          $sql .= " studentid=? AND ";
        }
        else
        {
          $sql .= " studentid=$? ;";
        }
      }
```

```
24          $query = $conn -> prepare($sql);
25          $result = $query -> execute($select);
26          if($result)
27          {
28            header("Location: students.php");
29          }
30        }
31        else
32        {
33          header("Location: students.php");
34        }
35      }
36      else
37      {
38        header("Location: index.php");
39      }
40    ?>
```

Section 4

Add Students Script

The add students script opens a form for the user to fill up with the details for the new student, and, when submited, the form is checked by the same script and, when everything is correct, adds the new student to the database and redirects the user to the students page, and, if not, reopens the form with alerts displaying the data that was missing underneath the form inputs.

http://intweb.bucks.ac.uk/~21611431/addstudents.php

Figure 4.1: addstudents.php

```php
<?php
  include("_includes/config.inc");
  include("_includes/dbconnect.inc");
  include("_includes/functions.inc");

  //Define variables for entered values as empty
  $studentid = $password = $firstname = $lastname = "";
  $house = $town = $county = $country = $postcode = "";
  $dob = date_create();

  //check if logged in
  if(isset($_SESSION['id']) && !($_SERVER["REQUEST_METHOD"]=="POST"))
  {
    //generate form
    echo template("templates/partials/header.php");
    echo template("templates/partials/nav.php");
    echo template("templates/addstudent.php");
  }
  else if(isset($_SESSION['id']) && $_SERVER["REQUEST_METHOD"] == "POST")
  {
    //Verifies that all the data has been entered and assigns
```

```
20      //the entered information into variables
21      if(empty($_POST['studentid']))
22      {
23        $data['validation']['studentid'] = true;
24      }
25      else{
26        $studentid = $_POST['studentid'];
27      }

28      if(empty($_POST['password']))
29      {
30        $data['validation']['password'] = true;
31      }
32      else
33      {
34        $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
35      }

36      if(empty($_POST['dob']))
37      {
38        $data['validation']['dob'] = true;
39      }
40      else{
41        $dob = $_POST['dob'];
42      }

43      if(empty($_POST['firstname']))
44      {
45        $data['validation']['firstname'] = true;
46      }
47      else
48      {
49        $firstname = $_POST['firstname'];
50      }

51      if(empty($_POST['lastname']))
52      {
53        $data['validation']['lastname'] = true;
54      }
55      else{
56        $lastname = $_POST['lastname'];
57      }

58      if(empty($_POST['house']))
59      {
```

```
60          $data['validation']['house'] = true;
61        }
62        else{
63          $house = $_POST['house'];
64        }

65        if(empty($_POST['town']))
66        {
67          $data['validation']['town'] = true;
68        }
69        else{
70          $town = $_POST['town'];
71        }

72        if(empty($_POST['county']))
73        {
74          $data['validation']['county'] = true;
75        }
76        else
77        {
78          $county = $_POST['county'];
79        }

80        if(empty($_POST['country']))
81        {
82          $data['validation']['country'] = true;
83        }
84        else
85        {
86          $country = $_POST['country'];
87        }

88        if(empty($_POST['postcode']))
89        {
90          $data['validation']['postcode'] = true;
91        }
92        else
93        {
94          $postcode = $_POST['postcode'];
95        }

96        //if any error message has been added to $data['content']
97        //prints the form again with the error messages
98        if(!empty($data['validation']))
99        {
```

```
100         echo template("templates/partials/header.php");
101         echo template("templates/partials/nav.php");
102         echo template("templates/addstudent.php", $data);
103       }
104       //if all the data is present, moves on to add the data to the database
105       else{
106         $sql = $conn -> prepare('INSERT INTO student(studentid,
107           `password`, dob, firstname, lastname, house,
108           town, county, country, postcode) VALUES(?, ?, ?,
109           ?, ?, ?, ?, ?, ?, ?)');
110         $result = $sql -> execute(
111           array(
112             $studentid,
113             $password,
114             $dob,
115             $firstname,
116             $lastname,
117             $house,
118             $town,
119             $county,
120             $country,
121             $postcode
122           )
123           );

124         //if the transaction has succeeded then send the user back
125         //to the students page
126         if($result)
127         {
128           header("Location: students.php");
129         }
130       }
131     }
132     else
133     {
134       header("Location: index.php");
135     }
136   ?>
```

The form is a template that is loaded and uses global data to know if the alerts are to be displayed or not.
http://intweb.bucks.ac.uk/~21611431/templates/addstudent.php

Figure 4.2: addstudent.php

```
1   <form action="addstudents.php" method="post">
```

```
<div class="form-row">
  <div class="form-group col-6">
    <label for="studentid">Student ID</label>
    <input type="text" id="studentid" class="form-control" name="studentid"/>
    <?php
      if(isset($validation['studentid']))
      {?>
        <div class="alert alert-danger" role="alert">
          You need to provide a student id number.
        </div>
      <?php }
    ?>
  </div>
  <div class="form-group col-6">
    <label for="password">Password</label>
    <input type="password" id="password" class="form-control"
    name="password"/>
    <?php
      if(isset($validation['password']))
      {?>
        <div class="alert alert-danger" role="alert">
          You need to provide a password.
        </div>
      <?php }
    ?>
  </div>
</div>
<div class="form-row">
  <div class="form-group col-6">
    <label for="firstname">First Name</label>
    <input type="text" id="firstname" class="form-control" name="firstname"/>
    <?php
      if(isset($validation['firstname']))
      {?>
        <div class="alert alert-danger" role="alert">
          You need to provide a First Name.
        </div>
      <?php }
    ?>
  </div>
  <div class="form-group col-6">
    <label for="lastname">Last Name</label>
    <input type="text" id="lastname" class="form-control" name="lastname"/>
    <?php
      if(isset($validation['lastname']))
```

```
        {?>
          <div class="alert alert-danger" role="alert">
            You need to provide a Last Name.
          </div>
        <?php }
      ?>
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-6">
      <label for="dob">Date Of Birth</label>
      <input type="date" id="dob" class="form-control" name="dob"/>
      <?php
        if(isset($validation['dob']))
        {?>
          <div class="alert alert-danger" role="alert">
            You need to provide a Date of Birth.
          </div>
        <?php }
      ?>
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-6">
      <label for="house">Number and Street</label>
      <input type="text" id="house" class="form-control" name="house"/>
      <?php
        if(isset($validation['house']))
        {?>
          <div class="alert alert-danger" role="alert">
            You need to provide a Number and Street.
          </div>
        <?php }
      ?>
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-6">
      <label for="town">Town</label>
      <input type="text" id="town" class="form-control" name="town"/>
      <?php
        if(isset($validation['town']))
        {?>
          <div class="alert alert-danger" role="alert">
            You need to provide a town.
```

```
            </div>
          <?php }
        ?>
      </div>
      <div class="form-group col-4">
        <label for="county">County</label>
        <input type="text" id="county" class="form-control" name="county"/>
        <?php
          if(isset($validation['county']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a county.
            </div>
          <?php }
        ?>
      </div>
      <div class="form-group col-2">
        <label for="postcode">Postcode</label>
        <input type="text" id="postcode" class="form-control" name="postcode"/>
        <?php
          if(isset($validation['postcode']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a postcode.
            </div>
          <?php }
        ?>
      </div>
    </div>
    <div class="form-row">
      <div class="form-group col-6">
        <label for="country">Country</label>
        <input type="text" id="country" class="form-control" name="country"/>
        <?php
          if(isset($validation['country']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a country.
            </div>
          <?php }
        ?>
      </div>
    </div>
    <input type="submit" class="btn btn-primary btn-md" value="Save"/>
  </form>
```

Section 5

Developing the UI/UX for the web app

The User Interface was developed using Bootstrap. It was applied throughout the pages and partial templates in order to give the app a similar feel and look and the same responsiveness throughout all the pages.

To start the UI, the Bootstrap CSS and JavaScripts had to be added to the header template:

Figure 5.1: header.php

```
<!DOCTYPE html>
<html lang="en">

   <head>
      <meta charset="utf-8">
      <meta
        name="viewport"
        content="width=device-width, initial-scale=1, shrink-to-fit=no"
      >
      <title>BNU Student Web Application</title>
      <link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.1
      0/css/bootstrap.min.css"
      integrity="sha384-9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jd
      IhOegiu1FwO5qRGvFXOdJZ4"
      crossorigin="anonymous"/>
      <link rel="stylesheet"
      href="../../css/customClasses.css"/>
      <script
      src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
      integrity="sha384-q8i/X+965DzO0rT7a
      bK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
```

```
22        crossorigin="anonymous">
23        </script>
24        <script
25        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.
26        0/umd/popper.min.js"
27        integrity="sha384-cs/chFZiN24E4KMATLdqdvsezGxaGsi4hLGOzlXwp
28        5UZB1LY//20VyM2taTB4QvJ"
29        crossorigin="anonymous">
30        </script>
31        <script
32        src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.
33        0/js/bootstrap.min.js"
34        integrity="sha384-uefMccjFJAIv6A+rW
35        +L4AHf99KvxDjWSu1z9VI8SKNVmz4sk7buKt/6v9KI65qnm"
36        crossorigin="anonymous">
37        </script>
38     </head>
```

The whole of the body is wrapped inside a container that sets the width and margins for the content inside the body of the page depending on the width of the device seeing the page. That div, as well as the declaration of the body, has been placed inside the nav template.
The navigation bar has also been completely rebuilt in order to build the design of a full width with background colour navigation bar.

Figure 5.2: nav.php

```
1  <body>
2    <nav class="navbar navbar-expand-sm navbar-dark bg-primary">
3      <a class="navbar-brand" href="#">BNU Student Application</a>
4      <button class="navbar-toggler" type="button"
5      data-toggle="collapse" data-target="#navbarSupportedContent"
6      aria-controls="navbarSupportedContent" aria-expand="false"
7      aria-label="Toggle navigation">
8        <span class="navbar-toggler-icon"></span>
9      </button>

10     <div class="collapse navbar-collapse" id="navbarSupportedContent">
11       <ul class="navbar-nav mr-auto">
12         <li class="nav-item active">
13           <a class="nav-link" href="index.php">Home</a>
14         </li>
15         <li class="nav-item">
16           <a class="nav-link" href="modules.php">My Modules</a>
17         </li>
18         <li class="nav-item">
```

```
          <a class="nav-link" href="assignmodule.php">Assign Module</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="details.php">My Details</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="students.php">Students</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="logout.php">Logout</a>
        </li>
      </ul>
    </div>
  </nav>
  <div class="container-fluid">
```

The footer also had to be changed so that it would close the div that is set inside nav for the container of the body.

Figure 5.3: footer.php

```
    </div>
  </body>

</html>
```

Most elements throughout the page, if not having a semantic meaning, were replaced with elements that would have a semantic meaning for accessibility purposes but also to allow the appropriate Bootstrap styling to be applied. Some extra divs might have also been added to separate the different sections and groups of elements, particularly for forms.
http://intweb.bucks.ac.uk/~21611431/

Figure 5.4: index.php

```
<?php

   include("_includes/config.inc");
   include("_includes/dbconnect.inc");
   include("_includes/functions.inc");

   echo template("templates/partials/header.php");

   if (isset($_GET['return'])) {
      $msg = "";
      if ($_GET['return'] == "fail") {$msg = "Login Failed.
        Please try again.";}
```

```
10          $data['message'] = "<p>$msg</p>";
11      }

12      if (isset($_SESSION['id'])) {
13          $data['content'] .= "<div class='row'>";
14          $data['content'] .= "<div class='jumbotron col mt-1'>";
15          $data['content'] .= "<h1>Welcome to your dashboard</h1>";
16          $data['content'] .= "</div>";
17          $data['content'] .= "</div>";
18          echo template("templates/partials/nav.php");
19          echo template("templates/default.php", $data);
20      } else {
21          echo template("templates/login.php", $data);
22      }

23      echo template("templates/partials/footer.php");

24  ?>
```

http://intweb.bucks.ac.uk/~21611431/modules.php

Figure 5.5: modules.php

```
1   <?php

2       include("_includes/config.inc");
3       include("_includes/dbconnect.inc");
4       include("_includes/functions.inc");


5       // check logged in
6       if (isset($_SESSION['id'])) {

7           echo template("templates/partials/header.php");
8           echo template("templates/partials/nav.php");

9           // Build SQL statment that selects a student's modules
10          $sql = $conn -> prepare('Select * from studentmodules sm,
11          module m where m.modulecode = sm.modulecode and
12          sm.studentid = ?');
13          $sql -> execute(array($_SESSION['id']));

14          // prepare page content
15          $data['content'] .= "<div class='page-header'>";
16          $data['content'] .= "<h1>My Modules</h1>";
17          $data['content'] .= "</div>";
```

```
        $data['content'] .= "<table class='table table-hover
        table-condensed table-bordered'>";
        $data['content'] .= "<thead class='thead-light'>";
        $data['content'] .=
        "<tr><th>Code</th><th>Type</th><th>Level</th></tr>";
        $data['content'] .= "</thead>";
        $data['content'] .= "<tbody>";
        // Display the modules within the html table
        while($row = $sql -> fetch()) {
           $data['content'] .= "<tr><td> $row[modulecode]
           </td><td> $row[name] </td>";
           $data['content'] .= "<td> $row[level] </td></tr>";
        }
        $sql -> closeCursor();
        $data['content'] .= "</tbody>";
        $data['content'] .= "</table>";

        // render the template
        echo template("templates/default.php", $data);

    } else {
        header("Location: index.php");
    }

    echo template("templates/partials/footer.php");

?>
```

http://intweb.bucks.ac.uk/~21611431/assignmodule.php

Figure 5.6: assignmodule.php

```
<?php

include("_includes/config.inc");
include("_includes/dbconnect.inc");
include("_includes/functions.inc");


// check logged in
if (isset($_SESSION['id'])) {

    echo template("templates/partials/header.php");
    echo template("templates/partials/nav.php");

    // If a module has been selected
```

```
10      if (isset($_POST['selmodule'])) {
11          $sql = $conn -> prepare('INSERT INTO studentmodules values
12          (?, ?)');
13          $sql -> execute(array($_SESSION['id'], $_POST['selmodule'])
14       );
15          $data['content'] .= "<p>The module " . $_POST['selmodule']
16          . " has been assigned to you</p>";
17      }
18      else  // If a module has not been selected
19      {

20        // Build sql statment that selects all the modules
21        $sql = $conn -> query('SELECT * FROM module');

22        $data['content'] .= "<h1>Assign Modules</h1>";
23        $data['content'] .= "<form name='frmassignmodule' action=''
24        class='mt-1' method='post' >";
25        $data['content'] .= "<div class='form-row'>";
26        $data['content'] .= "<div class='form-group col-6'";
27        $data['content'] .= "<label for='selmodule'>Select a module
28        to assign</label>";
29        $data['content'] .= "<select name='selmodule'
30        id='selmodule' class='form-control'>";
31        // Display the module name sin a drop down selection box
32        while($row = $sql -> fetch()) {
33            $data['content'] .= "<option value='$row[modulecode]
34            '>$row[name]</option>";
35        }
36        $sql -> closeCursor();
37        $data['content'] .= "</select>";
38        $data['content'].= "</div></div>";
39        $data['content'] .= "<input type='submit' class='btn
40        btn-primary btn-md' name='confirm' value='Save' />";
41        $data['content'] .= "</form>";
42      }

43      // render the template
44      echo template("templates/default.php", $data);

45  } else {
46      header("Location: index.php");
47  }

48  echo template("templates/partials/footer.php");
```

```
49 ?>
```

http://intweb.bucks.ac.uk/~21611431/details.php

Figure 5.7: details.php

```
1  <?php

2  include("_includes/config.inc");
3  include("_includes/dbconnect.inc");
4  include("_includes/functions.inc");


5  // check logged in
6  if (isset($_SESSION['id'])) {

7     echo template("templates/partials/header.php");
8     echo template("templates/partials/nav.php");

9     // if the form has been submitted
10    if (isset($_POST['submit'])) {

11       // build an sql statment to update the student details
12       $sql = 'UPDATE student SET firstname = ?, lastname = ?,
13       house = ?, town = ?, county = ?, country = ?, postcode = ?
14       WHERE studentid = ?';
15       $query = $conn -> prepare($sql);
16       $result = $query -> execute(array(
17         $_POST['txtfirstname'],
18         $_POST['txtlastname'],
19         $_POST['txthouse'],
20         $_POST['txttown'],
21         $_POST['txtcounty'],
22         $_POST['txtcountry'],
23         $_POST['txtpostcode'],
24         $_SESSION['id']
25       ));

26       if($result)
27       {
28         $data['content'] .= '<p>Your details have been
29         updated.</p>';
30       }

31    }
32    else {
```

```
33          // Build a SQL statment to return the student record with
34          //the id that
35          // matches that of the session variable.
36          $sql = 'SELECT * from student where studentid=? ;';
37          $query = $conn -> prepare($sql);
38          $query -> execute(array($_SESSION['id']));
39          $row = $query -> fetch();
40          $query -> closeCursor();

41          // using <<<EOD notation to allow building of a multi-line
42          //string
43          // see
44          //http://stackoverflow.
45          //com/questions/6924193/what-is-the-use-of-eod-in-php for
46          //info
47          // also
48          //http://stackoverflow.
49          //com/questions/8280360/formatting-an-array-value-inside-a-
50          //heredoc
51          $data['content'] = <<<EOD

52       <h1>My Details</h2>
53       <form name="frmdetails" action="" method="post">
54       <div class="form-row">
55        <div class="form-group col-6">
56          <label for="txtfirstname">First Name</label>
57          <input name="txtfirstname" class="form-control"
58          type="text" value="{$row['firstname']}" />
59        </div>
60        <div class="form-group col-6">
61          <label for="txtlastname">Surname</label>
62          <input name="txtlastname" type="text" class="form-control"
63          value="{$row['lastname']}" />
64        </div>
65      </div>
66      <div class="form-row">
67        <div class="form-group col-6">
68          <label for="txthouse">Number and Street</label>
69          <input name="txthouse" type="text" class="form-control"
70          value="{$row['house']}" />
71        </div>
72      </div>
73      <div class="form-row">
74        <div class="form-group col-6">
75          <label for="txttown">Town</label>
```

```
76          <input name="txttown" type="text" class="form-control"
77          value="{$row['town']}" />
78        </div>
79        <div class="form-group col-4">
80          <label for="txtcounty">County</label>
81          <input name="txtcounty" type="text" class="form-control"
82          value="{$row['county']}" />
83        </div>
84        <div class="form-group col-2">

85        <label for="txtpostcode">Postcode</label>
86          <input name="txtpostcode" type="text" class="form-control"
87          value="{$row['postcode']}" />
88        </div>
89      </div>
90      <div class="form-row">
91        <div class="form-group col-6">
92          <label for="txtcountry">Country</label>
93          <input name="txtcountry" type="text" class="form-control"
94          value="{$row['country']}" />
95        </div>
96      </div>
97       <input type="submit" value="Save" class="btn btn-primary
98       btn-md" name="submit"/>
99       </form>

100 EOD;

101     }

102     // render the template
103     echo template("templates/default.php", $data);

104 } else {
105     header("Location: index.php");
106 }

107 echo template("templates/partials/footer.php");

108 ?>
```

http://intweb.bucks.ac.uk/~21611431/students.php

Figure 5.8: students.php

```
1 <?php
```

```php
include("_includes/config.inc");
include("_includes/dbconnect.inc");
include("_includes/functions.inc");

//check if logged in
if(isset($_SESSION['id'])){

  echo template("templates/partials/header.php");
  echo template("templates/partials/nav.php");

  //Build SQL statement that selects students
  $result = $conn -> query("SELECT * FROM student");

  //prepare page content
  $data['content'] .= "<form action=
  'delete.php' method='post'>";
  $data['content'] .= "<div class='page-header'>";
  $data['content'] .= "<h1>Students</h1>";
  $data['content'] .= "</div>";
  $data['content'] .=
  "<table class='table table-hover table-condensed table-bordered'>";
  $data['content'] .= "<thead class='thead-light'>";
  $data['content'] .= "<tr><th>Student ID</th>";
  $data['content'] .= "<th>DOB</th><th>First Name</th>";
  $data['content'] .= "<th>Last Name</th>";
  $data['content'] .= "<th>House</th><th>Town</th>";
  $data['content'] .= "<th>County</th><th>Country</th>";
  $data['content'] .= "<th>Postcode</th><th>Selected</th>";
  $data['content'] .= "</tr></thead><tbody>";

  while($row = $result -> fetch()) {
    $data['content'] .= "<tr><td>$row[studentid]</td>";
    $data['content'] .= "<td>$row[dob]</td>";
    $data['content'] .= "<td>$row[firstname]</td>";
    $data['content'] .= "<td>$row[lastname]</td>";
    $data['content'] .= "<td>$row[house]</td>";
    $data['content'] .= "<td>$row[town]</td>";
    $data['content'] .= "<td>$row[county]</td>";
    $data['content'] .= "<td>$row[country]</td>";
    $data['content'] .= "<td>$row[postcode]</td>";
    $data['content'] .= "<td>";
    $data['content'] .= "<div class='form-check'>";
    $data['content'] .= "<input type='checkbox'";
    $data['content'] .= "class='form-check-input";
    $data['content'] .= "position-static' name='selected[]'";
```

```
        $data['content'] .= "value='$row[studentid]'/>";
        $data['content'] .= "</div></td></tr>";
      }
      $result -> closeCursor();
      $data['content'] .= "</tbody></table>";
      $data['content'] .= "<br/>";
      $data['content'] .= "<div class='form-group'>";
      $data['content'] .= "<a href='addstudents.php' class='btn";
      $data['content'] .= "btn-primary btn-md mr-2'";
      $data['content'] .= "role='button'>Add Student</a>";
      $data['content'] .= "<input class='btn btn-danger btn-md'";
      $data['content'] .= "type='submit' value='Delete'/>";
      $data['content'] .= "</form>";
      $data['content'] .= "</div>";
      //render the template
      echo template("templates/default.php", $data);
    }
    else{
      header("Location: index.php");
    }

    echo template("templates/partials/footer.php");
    ?>
```

http://intweb.bucks.ac.uk/~21611431/addstudents.php

Figure 5.9: addstudent.php

```
<form action="addstudents.php" method="post">
  <div class="form-row">
    <div class="form-group col-6">
      <label for="studentid">Student ID</label>
      <input type="text" id="studentid" class="form-control" name="studentid"/>
      <?php
        if(isset($validation['studentid']))
        {?>
          <div class="alert alert-danger" role="alert">
            You need to provide a student id number.
          </div>
        <?php }
      ?>
    </div>
    <div class="form-group col-6">
      <label for="password">Password</label>
      <input type="password" id="password" class="form-control"
      name="password"/>
```

```
        <?php
          if(isset($validation['password']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a password.
            </div>
          <?php }
        ?>
      </div>
    </div>
    <div class="form-row">
      <div class="form-group col-6">
        <label for="firstname">First Name</label>
        <input type="text" id="firstname" class="form-control" name="firstname"/>
        <?php
          if(isset($validation['firstname']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a First Name.
            </div>
          <?php }
        ?>
      </div>
      <div class="form-group col-6">
        <label for="lastname">Last Name</label>
        <input type="text" id="lastname" class="form-control" name="lastname"/>
        <?php
          if(isset($validation['lastname']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a Last Name.
            </div>
          <?php }
        ?>
      </div>
    </div>
    <div class="form-row">
      <div class="form-group col-6">
        <label for="dob">Date Of Birth</label>
        <input type="date" id="dob" class="form-control" name="dob"/>
        <?php
          if(isset($validation['dob']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a Date of Birth.
```

```
            </div>
          <?php }
        ?>
      </div>
    </div>
    <div class="form-row">
      <div class="form-group col-6">
        <label for="house">Number and Street</label>
        <input type="text" id="house" class="form-control" name="house"/>
        <?php
          if(isset($validation['house']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a Number and Street.
            </div>
          <?php }
        ?>
      </div>
    </div>
    <div class="form-row">
      <div class="form-group col-6">
        <label for="town">Town</label>
        <input type="text" id="town" class="form-control" name="town"/>
        <?php
          if(isset($validation['town']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a town.
            </div>
          <?php }
        ?>
      </div>
      <div class="form-group col-4">
        <label for="county">County</label>
        <input type="text" id="county" class="form-control" name="county"/>
        <?php
          if(isset($validation['county']))
          {?>
            <div class="alert alert-danger" role="alert">
              You need to provide a county.
            </div>
          <?php }
        ?>
      </div>
      <div class="form-group col-2">
```

```
109         <label for="postcode">Postcode</label>
110         <input type="text" id="postcode" class="form-control" name="postcode"/>
111         <?php
112           if(isset($validation['postcode']))
113           {?>
114             <div class="alert alert-danger" role="alert">
115               You need to provide a postcode.
116             </div>
117           <?php }
118         ?>
119       </div>
120     </div>
121     <div class="form-row">
122       <div class="form-group col-6">
123         <label for="country">Country</label>
124         <input type="text" id="country" class="form-control" name="country"/>
125         <?php
126           if(isset($validation['country']))
127           {?>
128             <div class="alert alert-danger" role="alert">
129               You need to provide a country.
130             </div>
131           <?php }
132         ?>
133       </div>
134     </div>
135     <input type="submit" class="btn btn-primary btn-md" value="Save"/>
136 </form>
```

Section 6

Protecting the SQL Queries against SQL Injection

In order to protect the SQL queries against SQL Injection, I will be using prepared statements with paramererized queries.

According to OWASP (Open Web Application Security Project) OWASP [b], parameterized queries allow the database to distinguish between code and data, independently of what the user supplies as an input, therefore preventing an attacker from changing the meaning of a statement by inserting SQL into the query.

In PHP, the best way to have prepared statements with parameterized queries is by using PDO (PHP Data Objects) to manage both the connection to the database and the process of building and executing the queries.

Because of that, the dbconnect.inc had to be modified so that the PDO object and the connection to the database would be set up.

Figure 6.1: dbconnect.inc

```
<?php
        //  Build PDO connection to database and display any errors
  //with the connection
  try{
    $conn = new PDO('mysql:host=localhost;dbname=oss-cw2;
    charset=utf8mb4', 'root', 'randomPass', array
    (PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
   }
   catch(Exception $error)
   {
     die('Error: ' . $error->getMessage());
   }
?>
```

The next files that were modified to support that change were all the files that needed to do queries to the database:

- functions.inc
- addstudents.php
- assignmodule.php
- delete.php
- details.php
- modules.php
- students.php

Figure 6.2: functions.inc

```php
<?php

include "passwordLib.php";

/**
 * Validate Login function
 *
 * @param $id        - user ID
 * @param $password  - plaintext password
 * @return boolean   - returns true or false depending on
 successful authentication of user.
 */
function validatelogin($id,$password) {
   global $conn;
   // Build an SQL statment that will return a record with a
   matching id
  $sql = $conn -> prepare("SELECT * FROM student where studentid
  = ?");
      $sql -> execute(array($id));
      $user = $sql->fetch();
      if ($user && password_verify($_POST['txtpwd'], $user
      ['password'])) {
         // valid
         $_SESSION['id'] = $id;
         return true;
      } else {
         // invalid
         unset($_SESSION['id']);
         return false;
```

```
28          }
29  }


30  /**
31   * Simple Templating function
32   *
33   * @param $file   - Path to the PHP file that acts as a template.
34   * @param $args   - (optional) Associative array of variables to
35   pass to the template file.
36   * @return string - Output of the template file. Likely HTML.
37   *
38   * Source:
39   http://www.daggerhart.com/create-simple-php-templating-function/
40   */
41  function template( $file, $args=array() ){
42     // ensure the file exists
43     if ( !file_exists( $file ) ) {
44        return '';
45     }

46     // Make values in the associative array easier to access by
47     extracting them
48     if ( is_array( $args ) ){
49        extract( $args );
50     }

51     // buffer the output (including the file is "output")
52     ob_start();
53     include $file;
54     return ob_get_clean();
55  }
56  ?>
```

Figure 6.3: addstudents.php

```
1  <?php
2    include("_includes/config.inc");
3    include("_includes/dbconnect.inc");
4    include("_includes/functions.inc");

5    //Define variables for entered values as empty
6    $studentid = $password = $firstname = $lastname = "";
7    $house = $town = $county = $country = $postcode = "";
8    $dob = date_create();
```

```php
//check if logged in
if(isset($_SESSION['id']) && !($_SERVER["REQUEST_METHOD"]=="POST"))
{
  //generate form
  echo template("templates/partials/header.php");
  echo template("templates/partials/nav.php");
  echo template("templates/addstudent.php");
}
else if(isset($_SESSION['id']) && $_SERVER["REQUEST_METHOD"] == "POST")
{
  //Verifies that all the data has been entered and assigns
  //the entered information into variables
  if(empty($_POST['studentid']))
  {
    $data['validation']['studentid'] = true;
  }
  else{
    $studentid = $_POST['studentid'];
  }

  if(empty($_POST['password']))
  {
    $data['validation']['password'] = true;
  }
  else
  {
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
  }

  if(empty($_POST['dob']))
  {
    $data['validation']['dob'] = true;
  }
  else{
    $dob = $_POST['dob'];
  }

  if(empty($_POST['firstname']))
  {
    $data['validation']['firstname'] = true;
  }
  else
  {
    $firstname = $_POST['firstname'];
```

```
50      }

51      if(empty($_POST['lastname']))
52      {
53        $data['validation']['lastname'] = true;
54      }
55      else{
56        $lastname = $_POST['lastname'];
57      }

58      if(empty($_POST['house']))
59      {
60        $data['validation']['house'] = true;
61      }
62      else{
63        $house = $_POST['house'];
64      }

65      if(empty($_POST['town']))
66      {
67        $data['validation']['town'] = true;
68      }
69      else{
70        $town = $_POST['town'];
71      }

72      if(empty($_POST['county']))
73      {
74        $data['validation']['county'] = true;
75      }
76      else
77      {
78        $county = $_POST['county'];
79      }

80      if(empty($_POST['country']))
81      {
82        $data['validation']['country'] = true;
83      }
84      else
85      {
86        $country = $_POST['country'];
87      }

88      if(empty($_POST['postcode']))
```

```
    {
      $data['validation']['postcode'] = true;
    }
    else
    {
      $postcode = $_POST['postcode'];
    }

    //if any error message has been added to $data['content']
    //prints the form again with the error messages
    if(!empty($data['validation']))
    {
      echo template("templates/partials/header.php");
      echo template("templates/partials/nav.php");
      echo template("templates/addstudent.php", $data);
    }
    //if all the data is present, moves on to add the data to the database
    else{
      $sql = $conn -> prepare('INSERT INTO student(studentid,
        `password`, dob, firstname, lastname, house,
        town, county, country, postcode) VALUES(?, ?, ?,
        ?, ?, ?, ?, ?, ?, ?)');
      $result = $sql -> execute(
        array(
          $studentid,
          $password,
          $dob,
          $firstname,
          $lastname,
          $house,
          $town,
          $county,
          $country,
          $postcode
        )
        );

      //if the transaction has succeeded then send the user back
      //to the students page
      if($result)
      {
        header("Location: students.php");
      }
    }
  }
```

```
132   else
133   {
134     header("Location: index.php");
135   }
136 ?>
```

Figure 6.4: assignmodule.php

```
1  <?php

2  include("_includes/config.inc");
3  include("_includes/dbconnect.inc");
4  include("_includes/functions.inc");


5  // check logged in
6  if (isset($_SESSION['id'])) {

7     echo template("templates/partials/header.php");
8     echo template("templates/partials/nav.php");

9     // If a module has been selected
10    if (isset($_POST['selmodule'])) {
11       $sql = $conn -> prepare('INSERT INTO studentmodules values
12       (?, ?)');
13       $sql -> execute(array($_SESSION['id'], $_POST['selmodule'])
14    );
15       $data['content'] .= "<p>The module " . $_POST['selmodule']
16       . " has been assigned to you</p>";
17    }
18    else  // If a module has not been selected
19    {

20      // Build sql statment that selects all the modules
21      $sql = $conn -> query('SELECT * FROM module');

22      $data['content'] .= "<h1>Assign Modules</h1>";
23      $data['content'] .= "<form name='frmassignmodule' action=''
24      class='mt-1' method='post' >";
25      $data['content'] .= "<div class='form-row'>";
26      $data['content'] .= "<div class='form-group col-6'";
27      $data['content'] .= "<label for='selmodule'>Select a module
28      to assign</label>";
29      $data['content'] .= "<select name='selmodule'
30      id='selmodule' class='form-control'>";
```

```
31        // Display the module name sin a drop down selection box
32        while($row = $sql -> fetch()) {
33           $data['content'] .= "<option value='$row[modulecode]
34           '>$row[name]</option>";
35        }
36        $sql -> closeCursor();
37        $data['content'] .= "</select>";
38        $data['content'].= "</div></div>";
39        $data['content'] .= "<input type='submit' class='btn
40        btn-primary btn-md' name='confirm' value='Save' />";
41        $data['content'] .= "</form>";
42     }

43     // render the template
44     echo template("templates/default.php", $data);

45  } else {
46     header("Location: index.php");
47  }

48  echo template("templates/partials/footer.php");

49  ?>
```

Figure 6.5: delete.php

```
1   <?php
2     include("_includes/config.inc");
3     include("_includes/dbconnect.inc");
4     include("_includes/functions.inc");

5     if(isset($_SESSION['id']))
6     {
7       $select = $_POST['selected'];
8       //Building the query depending on the selected records
9       if(!empty($select))
10      {
11        $lastElement = end($select);
12        $sql = "Delete from student where ";
13        foreach($select as $stu)
14        {
15          if(strcmp($stu, $lastElement) != 0)
16          {
17            $sql .= " studentid=? AND ";
18          }
```

```
          else
          {
            $sql .= " studentid=$? ;";
          }
        }
        $query = $conn -> prepare($sql);
        $result = $query -> execute($select);
        if($result)
        {
          header("Location: students.php");
        }
      }
      else
      {
        header("Location: students.php");
      }
    }
    else
    {
      header("Location: index.php");
    }
?>
```

Figure 6.6: details.php

```
<?php

include("_includes/config.inc");
include("_includes/dbconnect.inc");
include("_includes/functions.inc");


// check logged in
if (isset($_SESSION['id'])) {

    echo template("templates/partials/header.php");
    echo template("templates/partials/nav.php");

    // if the form has been submitted
    if (isset($_POST['submit'])) {

        // build an sql statment to update the student details
        $sql = 'UPDATE student SET firstname = ?, lastname = ?,
        house = ?, town = ?, county = ?, country = ?, postcode = ?
        WHERE studentid = ?';
```

```
15          $query = $conn -> prepare($sql);
16          $result = $query -> execute(array(
17            $_POST['txtfirstname'],
18            $_POST['txtlastname'],
19            $_POST['txthouse'],
20            $_POST['txttown'],
21            $_POST['txtcounty'],
22            $_POST['txtcountry'],
23            $_POST['txtpostcode'],
24            $_SESSION['id']
25          ));

26          if($result)
27          {
28            $data['content'] .= '<p>Your details have been
29            updated.</p>';
30          }

31      }
32      else {
33          // Build a SQL statment to return the student record with
34          //the id that
35          // matches that of the session variable.
36          $sql = 'SELECT * from student where studentid=? ;';
37          $query = $conn -> prepare($sql);
38          $query -> execute(array($_SESSION['id']));
39          $row = $query -> fetch();
40          $query -> closeCursor();

41          // using <<<EOD notation to allow building of a multi-line
42          //string
43          // see
44          //http://stackoverflow.
45          //com/questions/6924193/what-is-the-use-of-eod-in-php for
46          //info
47          // also
48          //http://stackoverflow.
49          //com/questions/8280360/formatting-an-array-value-inside-a-
50          //heredoc
51          $data['content'] = <<<EOD

52      <h1>My Details</h2>
53      <form name="frmdetails" action="" method="post">
54      <div class="form-row">
55       <div class="form-group col-6">
```

```
      <label for="txtfirstname">First Name</label>
      <input name="txtfirstname" class="form-control"
      type="text" value="{$row['firstname']}" />
    </div>
    <div class="form-group col-6">
      <label for="txtlastname">Surname</label>
      <input name="txtlastname" type="text" class="form-control"
      value="{$row['lastname']}" />
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-6">
      <label for="txthouse">Number and Street</label>
      <input name="txthouse" type="text" class="form-control"
      value="{$row['house']}" />
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-6">
      <label for="txttown">Town</label>
      <input name="txttown" type="text" class="form-control"
      value="{$row['town']}" />
    </div>
    <div class="form-group col-4">
      <label for="txtcounty">County</label>
      <input name="txtcounty" type="text" class="form-control"
      value="{$row['county']}" />
    </div>
    <div class="form-group col-2">

    <label for="txtpostcode">Postcode</label>
      <input name="txtpostcode" type="text" class="form-control"
      value="{$row['postcode']}" />
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-6">
      <label for="txtcountry">Country</label>
      <input name="txtcountry" type="text" class="form-control"
      value="{$row['country']}" />
    </div>
  </div>
   <input type="submit" value="Save" class="btn btn-primary
   btn-md" name="submit"/>
   </form>
```

```
EOD;

    }

    // render the template
    echo template("templates/default.php", $data);

} else {
    header("Location: index.php");
}

echo template("templates/partials/footer.php");

?>
```

Figure 6.7: modules.php

```
<?php

    include("_includes/config.inc");
    include("_includes/dbconnect.inc");
    include("_includes/functions.inc");


    // check logged in
    if (isset($_SESSION['id'])) {

        echo template("templates/partials/header.php");
        echo template("templates/partials/nav.php");

        // Build SQL statment that selects a student's modules
        $sql = $conn -> prepare('Select * from studentmodules sm,
        module m where m.modulecode = sm.modulecode and
        sm.studentid = ?');
        $sql -> execute(array($_SESSION['id']));

        // prepare page content
        $data['content'] .= "<div class='page-header'>";
        $data['content'] .= "<h1>My Modules</h1>";
        $data['content'] .= "</div>";
        $data['content'] .= "<table class='table table-hover
        table-condensed table-bordered'>";
        $data['content'] .= "<thead class='thead-light'>";
        $data['content'] .=
```

```
        "<tr><th>Code</th><th>Type</th><th>Level</th></tr>";
        $data['content'] .= "</thead>";
        $data['content'] .= "<tbody>";
        // Display the modules within the html table
        while($row = $sql -> fetch()) {
           $data['content'] .= "<tr><td> $row[modulecode]
           </td><td> $row[name] </td>";
           $data['content'] .= "<td> $row[level] </td></tr>";
        }
        $sql -> closeCursor();
        $data['content'] .= "</tbody>";
        $data['content'] .= "</table>";

        // render the template
        echo template("templates/default.php", $data);

    } else {
        header("Location: index.php");
    }

    echo template("templates/partials/footer.php");

?>
```

Figure 6.8: students.php

```
<?php
  include("_includes/config.inc");
  include("_includes/dbconnect.inc");
  include("_includes/functions.inc");

  //check if logged in
  if(isset($_SESSION['id'])){

    echo template("templates/partials/header.php");
    echo template("templates/partials/nav.php");

    //Build SQL statement that selects students
    $result = $conn -> query("SELECT * FROM student");

    //prepare page content
    $data['content'] .= "<form action=
    'delete.php' method='post'>";
    $data['content'] .= "<div class='page-header'>";
    $data['content'] .= "<h1>Students</h1>";
```

```
16      $data['content'] .= "</div>";
17      $data['content'] .=
18      "<table class='table table-hover table-condensed table-bordered'>";
19      $data['content'] .= "<thead class='thead-light'>";
20      $data['content'] .= "<tr><th>Student ID</th>";
21      $data['content'] .= "<th>DOB</th><th>First Name</th>";
22      $data['content'] .= "<th>Last Name</th>";
23      $data['content'] .= "<th>House</th><th>Town</th>";
24      $data['content'] .= "<th>County</th><th>Country</th>";
25      $data['content'] .= "<th>Postcode</th><th>Selected</th>";
26      $data['content'] .= "</tr></thead><tbody>";

27      while($row = $result -> fetch()) {
28        $data['content'] .= "<tr><td>$row[studentid]</td>";
29        $data['content'] .= "<td>$row[dob]</td>";
30        $data['content'] .= "<td>$row[firstname]</td>";
31        $data['content'] .= "<td>$row[lastname]</td>";
32        $data['content'] .= "<td>$row[house]</td>";
33        $data['content'] .= "<td>$row[town]</td>";
34        $data['content'] .= "<td>$row[county]</td>";
35        $data['content'] .= "<td>$row[country]</td>";
36        $data['content'] .= "<td>$row[postcode]</td>";
37        $data['content'] .= "<td>";
38        $data['content'] .= "<div class='form-check'>";
39        $data['content'] .= "<input type='checkbox'";
40        $data['content'] .= "class='form-check-input";
41        $data['content'] .= "position-static' name='selected[]'";
42        $data['content'] .= "value='$row[studentid]'/>";
43        $data['content'] .= "</div></td></tr>";
44      }
45      $result -> closeCursor();
46      $data['content'] .= "</tbody></table>";
47      $data['content'] .= "<br/>";
48      $data['content'] .= "<div class='form-group'>";
49      $data['content'] .= "<a href='addstudents.php' class='btn";
50      $data['content'] .= "btn-primary btn-md mr-2'";
51      $data['content'] .= "role='button'>Add Student</a>";
52      $data['content'] .= "<input class='btn btn-danger btn-md'";
53      $data['content'] .= "type='submit' value='Delete'/>";
54      $data['content'] .= "</form>";
55      $data['content'] .= "</div>";
56      //render the template
57      echo template("templates/default.php", $data);
58    }
59    else{
```

```
60      header("Location: index.php");
61    }

62    echo template("templates/partials/footer.php");
63    ?>
```

Section 7

Evaluation of ASP.NET and PHP

7.1 ASP.NET: Overview, Advantages and Disadvantages

According to Microsoft documentation, ASP.NET is a free web framework used to build websites and web applications using HTML, CSS and JavaScript. It can also be used to create Web APIs and to use real-time technologies like Web Socket.

For websites and web applications, ASP.NET offers three frameworks: Web Forms (built using drag-and-drop and is an event-driven model), ASP.NET MVC (pattern-based way to build sites with concise separation of different aspects of a web application) and ASP.NET Web Pages (Fast, approachable and lightweight way to combine server code with HTML).

All the ASP.NET code is compiled when the first time a user requests a resource from the Web site. The code is ran inside a Common Language Runtime, which allows developers to develop in different languages, as long as they all are supported by the .NET Framework.

The best points of this technologies are the following:

- The system comes as a whole (no need to install many different components to get everything to work)
- The templates allow for an easy set-up and also allow the user to focus on the functions of the code
- Many of the items are built automatically for the user
- Uses a strongly typed languages for better control of the application

When using ASP.NET, the whole process of building the website makes the developer feel like they are building the software that will serve and build the website for the user.

The down-sides of ASP.NET are:

- Is based on the .NET Framework, which only works on Windows (unless you use ASP.NET Core, the open-source multi-platform version of ASP.NET)
- There are not many servers that run it due to the costs of maintaining a Windows Server
- Requires you to install Visual Studio, which, for a production environment, means maintaining licenses that can be quite costly

7.2 PHP: Overview, Advantages and Disadvantages

PHP is a general-purpose scripting language that was built with web development in mind. It is also open-source and can be embedded right inside HTML. It is a back-end scripiting language, meaning that it is ran on the server to generate HTML and is afterwards sent to the client.

In order for a PHP application to work, there is a need to have three things:

- A PHP Parser
- A Web Server
- A Web Browser

The web server needs to have the php parser installed in it. In order for databases to interact with the web pages produced with PHP, the server must get both the database software and the add-on support for PHP installed.

Since PHP is not a full environment but only a scripting languages, no IDE is necessary, but also no templates are provided to the developer. A developer can use any environment they would like to develop PHP, but, when doing so, they would also need to write the entire system from scratch.

In the most common cases, a PHP web application would be built and deployed in what is commonly called as a LAMP stack, which is constitued of the following:

- Linux for the OS running the server
- Apache for the software managing the server
- MySQL for the Database Management System
- PHP for the parser present on the server

However, any of these components are open-source and therefore can be deployed on any computer platform.

With all this information in mind, the following are the advantages of PHP:

- Is Open-Source and Multi-Platform natively
- Allows the user to have full control of the code produced
- Is easy to learn and integrate to HTML
- The majority of web servers run PHP

And, like any other technologies, it also has its disadvantages, which are:

- Does not provide any templates or help for a beginner user
- Requires quite a lot of technology knowledge in order to troubleshoot problems related to the stack
- Requires installation of many different softwares to make a website work

7.3 Conclusion

To finish evaluating both server-side options, I would say that both have their advantages and disadvantages, and that it can all be resumed to a matter of preference, especially now that ASP.NET also has an open-source and multi-platform version to run websites on any type of server.

For a beginner however, ASP.NET would be much easier to grasp, since the person developing the web site would only need to focus on the functionality since everything else can be provided for them.

For an expert, PHP would probably be a better choice, especially if they know all the ins and outs of building websites using that scripting language.

7.4 Resources used for this task

Network [b], Network [a], Blixt, Team [b], Team [a], and Software.

Section 8

Key Security Concerns about Personal Data

There are many risks and issues that can arise when it comes to the security of personal data retained from people that use computerized systems, especially if they are stored on a remote server.

The OWASP [2017] group has conducted a project to evaluate which are the top ten risks in Web Application Security and those are the following:

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging and Monitoring

Out of these, many of them have easy and well-known remediations and ways to be prevented, so it can only lead to the conclusion that many developers tend to overlook these security measures.

Looking into the Proactive Controls project done by the OWASP [a], the following security concepts should be in the front of every developer's mind when developing a system:

1. Verify for Security Early and Often
2. Parameterize Queries
3. Encode Data
4. Validate All Inputs
5. Implement Identity and Authentication Controls
6. Implement Appropriate Access Controls
7. Protect Data
8. implement Logging and Intrusion Detection
9. Leverage Security Frameworks and Libraries
10. Error Handling and Exception Handling

References

A. Blixt. What are the advantages and disadvantages of using asp.net ? URL https://www.quora.com/What-are-the-advantages-and-disadvantages-of-using-ASP-NET.

M. D. Network. Asp.net compilation overview, a. URL https://msdn.microsoft.com/en-us/library/ms178466.aspx.

M. D. Network. Asp.net overview, b. URL https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx.

OWASP. Owasp proactive controls, a. URL https://www.owasp.org/index.php/OWASP_Proactive_Controls.

OWASP. Sql injection prevention cheat sheet, b. URL https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet.

OWASP. Owasp top 10 - 2017 (the ten most critical web application security risks). 2017. URL https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf.

O. Software. Asp.net vs php. URL https://www.orientsoftware.net/technologies/microsoft-net/aspnet-vs-php/.

P. D. Team. Php: Installation and configuration - manual, a. URL http://php.net/manual/en/install.php.

P. D. Team. Php: What is php? - manual, b. URL http://php.net/manual/en/intro-whatis.php.