ECS417U Project Phase 2 Documentation

|  |  |
| --- | --- |
| **Name:** | Ewelike Daniel Chimezie |
| **Student Number:** | 220310576 |

|  |  |
| --- | --- |
| **Assessment Criteria 7**  Each time a new entry is added, the most recent post appears on top, followed by the next most recent post using a PHP-driven technique (writing a sorting algorithm) instead of a SQL Query. | |
| Instruction: use this column to paste the code from the files that have been used to develop this functionality.  **Paste the code directly from visual studio code, do not include a screenshot of the code as this document has a Turnitin requirement and this will lead to an automatic mark of zero. See example below.** | Instruction: use this column to address the following in your own words:   1. Describe the steps you took to develop this feature (including the prompts). 2. Describe how the sorting algorithm works.   Give reference to the code you have written when answering the questions above.  You can use diagrams (e.g. flow charts) or screenshots of your webpage output to illustrate your answer. |
| Filename: viewblog.php code snippet  //selection sort  //1. the outer loop iterates through the array, treating each element as the current maximum  //2. the inner loop iterates through the unsorted portion of the array to find the maximum element  //if a larger element is found, the index of the maximum element is updated  //3. after the inner loop, if the maximum element is not already in the correct position, it is swapped with the current index  //4. this process continues until the entire array is sorted in descending order  //5. the function returns the sorted array  //instead of using in values, im using the time of each post as a value converted by strtotime for easier comparison  function selectionSortByDate(array $posts): array  {      $n = count($posts);      for ($i = 0; $i < $n - 1; $i++) {          $maxIdx = $i;          for ($j = $i + 1; $j < $n; $j++) {              if (strtotime($posts[$j]['time']) > strtotime($posts[$maxIdx]['time'])) {                  $maxIdx = $j;              }          }          // Swap the found maximum with current index          if ($maxIdx !== $i) {              $temp = $posts[$i];              $posts[$i] = $posts[$maxIdx];              $posts[$maxIdx] = $temp;          }      }      return $posts;  } | 1. **Describe the steps you took to develop this feature (including the prompts).**   Before developing this algorithm, I was already very familiar with some sorting algorithm implementations from my other modules Procedural programming and OOP, so all I needed was an algorithm that uses some of the ones I already know as the base or the same one but modified to work for date time values comparison. So, I engineered a prompt using two of the best prompting technique—role and constraint prompting, the prompt goes like this:  *Act as a web developer with 50 years of experience and recommend 3 sorting algorithms for displaying blog posts by the most recent. This algorithm should compare time values fetched from a database. This algorithm will have to be written in PHP language by me so don't give me the code. Also, don't recommend using ORDER BY in SQL. These algorithms should be based on bubble sort or selection sort as am already familiar with it.*  The response of ChatGPT was broad but I particularly liked the following suggestion:  ***2. Selection Sort (Descending, Maximum First)***  ***Use case****: You're familiar with this too; just find the* ***maximum timestamp*** *instead of the minimum on each pass.* ***Logic****:*   * *Loop through unsorted part.* * *Find the post with the* ***largest timestamp****.* * *Swap it to the front of the unsorted segment.* * *Repeat for the rest.*   ***Why it's good****:*   * *Simple to code and reason through.* * *Minimizes swaps compared to Bubble Sort.*   This was particularly okay for me, so I got to work on it, first I loaded all the posts into an array variable and then I wrote selection sort function, I separated the function instead of implementing it directly on the post results in order to practice abstraction—we all know the benefits of it.  The function takes an array argument and returns the same type after sorting. The sorting does its work by implementing a nested for loop. A variable $n is declared outside to and assigned an int value with represents the number of rows or posts present in the array. The loops will only run for at most $n times. The outer loop loops takes the immediate value as the largest i.e. the newest and the inner for loop loops through the array to see if there is a post with time values newer than the initial one, if there is, it takes that one as the new newest one, at the end of the inner for loop it swaps the newest one position with the position of the initial one i.e. the current index, it only does the swapping if the current index is not the newest one. It repeats this process $n-1 number of times. The comparison is facilitated by using the inbuilt strtotime function to convert a string that is already in the format of date time to actual time values so they can be compared with greater and less than operators.   1. **Describe how the sorting algorithm works.**   This sorting algorithm works by using a nested for loop, an if condition, and a variable called maxindex that tracks the current index of the newest post at that particular time:  Step 1: outer loop through each post and assigned max index to the current post:  for ($i = 0; $i < $n - 1; $i++) {          $maxIdx = $i;  Step 2: inner loop loops through each post starting from the index of the outer loop +1:  for ($j = $i + 1; $j < $n; $j++) {  Step 3: in the inner loop, the outer loop current post time is compared to times of all the posts that is being looped through at each time:  if (strtotime($posts[$j]['time']) > strtotime($posts[$maxIdx]['time'])) {  Step 4: if a higher time is found, the max index is reassigned to be the index of that post.                  $maxIdx = $j;    Step 5: after the inner loop is done, the if statement checks if the current maxindex is the current index, if it is, it moves on to repeating everything again, else, it swaps the post at index maxindex with the post at the outer loop current index.  if ($maxIdx !== $i) {              $temp = $posts[$i];              $posts[$i] = $posts[$maxIdx];              $posts[$maxIdx] = $temp;          }  Step 6: after step 1 to 5 is repeated for $n-1 times, return the fully sorted array.      return $posts;  Let’s say for example, there’s a post array like this  $exampleArray= [  [“title 1”,30/05/2024 15:00:23},  [“title 2”,30/07/2024 15:00:23},  [“title 3”,30/01/2024 15:00:23},  [“title 4”,30/01/2024 18:00:23}  }  **1ST RUN:**  At step 1: maxindex=0  At step 2 & 3: the array will be looped through and the dates of title 2, 3 and 4 will be compared to the date of title 1.  At step 4: maxindex will be = 1, because the date of title 2 is greater than the date of title 1.  At step 5: the exampleArray index 0 and 1 values will be swapped so it’ll be:  $exampleArray= [  [“title 2”,30/07/2024 15:00:23],  [“title 1”,30/05/2024 15:00:23],  [“title 3”,30/01/2024 15:00:23],  [“title 4”,30/01/2024 18:00:23]  ]  **2nd RUN:**  At step 1: maxindex=1  At step 2 & 3: the array will be looped through and the dates of title 3 and 4 will be compared to the date of title 1.  At step 4: maxindex will still be = 1, because the date of title 1 is greater than the dates of title 3 and 4.  At step 5: the exampleArray will remain the same and there will be no swaps:  $exampleArray= [  [“title 2”,30/07/2024 15:00:23],  [“title 1”,30/05/2024 15:00:23],  [“title 3”,30/01/2024 15:00:23],  [“title 4”,30/01/2024 18:00:23]  ]  **3rd RUN:**  At step 1: maxindex=2  At step 2 & 3: the array will be looped through and the date of title 3 will be compared to the date of title 4.  At step 4: maxindex will be = 3, because the time of title 4 is greater than the time of title 3.  At step 5: the exampleArray index 3 and 2 values will be swapped so it’ll be:  $exampleArray= [  [“title 2”,30/07/2024 15:00:23],  [“title 1”,30/05/2024 15:00:23],  [“title 4”,30/01/2024 18:00:23],  [“title 3”,30/01/2024 15:00:23]  ]  There will be no 4th Run because after the 3rd run or n-1th the last value will already be in its correct position.  At step 6: The now sorted array will be returned. |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Assessment Criteria 8 - Extra Feature 1**  (Note: delete any of the two features below)   * Allow blog viewers to log in and add comments to the entries. You, as the administrator of the blog should be able to delete entries or comments. | |
| Instruction: use this column to paste the code from the files that have been used to develop this functionality.  **Paste the code directly from visual studio code, do not include a screenshot of the code as this document has a Turnitin requirement and this will lead to an automatic mark of zero. See example below.** | Instruction: use this column to address the following in your own words:   1. How did you develop this feature? Describe the steps involved. 2. Describe the inner workings of the approach you developed.   Give reference to the code you have written when answering the questions above.  You can use diagrams (e.g. flow charts) or screenshots of your webpage output to illustrate your answer. |
| Filename: login.php  <?php  $servername = "127.0.0.1";  $username = "root";  $password = "";  $dbname = "ecs417";  $conn = new mysqli($servername, $username, $password, $dbname);  if ($conn->connect\_error) {      die("Connection failed: " . $conn->connect\_error);  }  session\_start();  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {      if (isset($\_POST["email"]) && isset($\_POST["password"])) {          $email = $\_POST["email"];          $password = $\_POST["password"];          $sql = "SELECT \* FROM portfolio\_admin WHERE email='$email' AND password='$password'";          $result = $conn->query($sql);          if ($result->num\_rows > 0) {              $\_SESSION["email"] = $email;              $\_SESSION["name"] = $result->fetch\_assoc()["name"];              $\_SESSION["loggedIn"] = true;              header("Location: addpost.php");              exit();          } else {              $password = hash("sha256", $password);              $sql = "SELECT \* FROM portfolio\_users WHERE email='$email' AND password='$password'";              $result = $conn->query($sql);              if ($result->num\_rows > 0) {                  $\_SESSION["email"] = $email;                  $\_SESSION["name"] = $result->fetch\_assoc()["firstName"];                  $\_SESSION["loggedIn"] = true;                  header("Location: viewblog.php");                  exit();              } else {                  echo "<script>alert('Invalid email or password');</script>";                  echo "<script>document.getElementById('form').reset</script>";                  echo "<script>window.location.href='login.php'</script>";              }          }      }      $conn->close();  }  **LOGIN FORM**  **/\***  **HTML login form here**  **\*/**  **Filename: signup.php**  <?php  $servername = "127.0.0.1";  $username = "root";  $password = "";  $dbname = "ecs417";  $conn = mysqli\_connect($servername, $username, $password, $dbname);  if (!$conn) {      die("Connection failed: " . mysqli\_connect\_error());  }  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {      $fname = htmlspecialchars($\_POST["fname"]);      $lname = htmlspecialchars($\_POST["lname"]);      $email = htmlspecialchars($\_POST["email"]);      $password = $\_POST["password"];      $cpassword = $\_POST["cpassword"];      if ($password != $cpassword) {          echo "<script>alert('Passwords do not match!');</script>";          echo "<script>window.location.href='signup.php';</script>";          exit();      }      $password = hash("sha256", $password);      $sql = "INSERT INTO portfolio\_users (firstName,lastName,email,password) VALUES ('$fname','$lname','$email','$password')";      if (mysqli\_query($conn, $sql)) {          echo "<script>alert('Registration successful!');</script>";          echo "<script>window.location.href='login.php';</script>";          exit();      } else {          echo "Error: " . $sql . "<br>" . mysqli\_error($conn);      }  }  ?>  <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>SignUp</title>      <link rel="stylesheet" href="../css/reset.css" />      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">      <link rel="stylesheet" href="../css/header.css" />      <link rel="stylesheet" href="../css/mobile.css" />    </head>  <body class="bg-dark">      <header class="bg-dark text-light">          <nav>              <li><a href="index.php" id="logo"><span style="color: crimson">D</span>aniel.EC</a></li>              <li><a href="index.php">Home</a></li>              <li><a href="viewblog.php">Blog</a></li>              <li><a href="../html/about.html">About me</a></li>              <li><a href="../html/skills.html">Skills and Experience</a></li>              <li><a href="login.php">Sign In</a></li>          </nav>          <li id="logo-list"><a class="logo " href="index.html" id="logo1"><span                      style="color: crimson">D</span>aniel.EC</a></li>          <div class="dropdown">              <button class="btn btn-primary dropdown-toggle " type="button" id="dropdownMenuButton"                  data-bs-toggle="dropdown" aria-expanded="false">                  Menu</button>              <ul class="dropdown-menu bg-dark" aria-labelledby="dropdownMenuButton">                  <li><a class="dropdown-item text-light" href="index.php">Home</a></li>                  <li><a class="dropdown-item text-light" href="viewblog.php">Blog</a></li>                  <li><a class="dropdown-item text-light" href="../html/about.html">About me</a></li>                  <li><a class="dropdown-item text-light" href="../html/skills.html">Skills and Experience</a></li>                  <li><a class="dropdown-item text-light" href="login.php">Sign In</a></li>              </ul>          </div>      </header>      <div class="container mt-5">          <div class="row justify-content-center">              <div class="col-md-6 bg-light text-dark p-4 rounded shadow">                  <h2 class="text-center text-dark mb-4">User Registration Form</h2>                  <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST" id="form">                      <div class="mb-3">                          <label for="fname" class="form-label" name="firstName">First Name</label>                          <input type="text" class="form-control" id="fname" name="fname" placeholder="Enter First Name"                              required />                      </div>                      <div class="mb-3">                          <label for="lname" class="form-label" name="lastName">Last Name</label>                          <input type="text" class="form-control" id="lname" name="lname" placeholder="Enter Last Name"                              required />                      </div>                      <div class="mb-3">                          <label for="email" class="form-label" name="email">Email Address</label>                          <input type="email" class="form-control" id="email" name="email"                              placeholder="Enter Email address" required </div>                          <div class="mb-3">                              <label for="password" class="form-label" name="password">Password</label>                              <input type="password" class="form-control" id="password" name="password"                                  placeholder="Enter Password" required />                          </div>                          <div class="mb-3">                              <label for="cpassword" class="form-label" name="cpassword">Confirm Password</label>                              <input type="password" class="form-control" id="cpassword" name="cpassword"                                  placeholder="Confirm Password" required />                          </div>                          <button type="submit" class="btn btn-primary w-100">Sign Up</button>                  </form>              </div>          </div>      </div>      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>      <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  </body>  </html>  **Filename: comment.php**    <?php  session\_start();  $servername = "127.0.0.1";  $username = "root";  $password = "";  $dbname = "ecs417";  $conn = new mysqli($servername, $username, $password, $dbname);  if ($conn->connect\_error) {      die("Connection failed:" . $conn->connect\_error);  }  //check if post was clicked i.e postid was passed to it, if not redirect to viewblog.php  if (isset($\_GET["id"])) {      $id = $conn->real\_escape\_string($\_GET["id"]);      $sql = "SELECT \*  FROM portfolio\_post WHERE ID='$id'";      $result = $conn->query($sql);      if ($result->num\_rows > 0) {          $post = $result->fetch\_assoc();      }      //reads all comments from db for the particular post      $sql = "SELECT \* FROM portfolio\_comment WHERE postID='$id' ORDER BY date DESC";      $result2 = $conn->query($sql);      if ($result2->num\_rows > 0) {          $rows = $result2->fetch\_all(MYSQLI\_ASSOC);          $noComments = false;      } else {          $noComments = true;      }      //posts the comment to the db      if ($\_SERVER["REQUEST\_METHOD"] == "POST") {          $comment = htmlspecialchars($\_POST["comment"]);          $postID = htmlspecialchars($\_GET["id"]);          $firstName = $\_SESSION["name"];          $sql = "INSERT INTO portfolio\_comment (postID, username,content) VALUES ( '$postID', '$firstName', '$comment')";          if ($conn->query($sql) === TRUE) {              echo '<script>  alert("Comment posted successfully");</script>';              echo '<script>window.location.href="comment.php?id=' . $postID . '"</script>';          } else {              echo '<script>  alert("Error posting comment");</script>';              echo '<script>window.location.href="comment.php?id=' . $postID . '"</script>';          }      }  } else {      echo '<script>  alert("A post was not selected");</script>';      echo '<script>window.location.href="viewblog.php"</script>';  }  $conn->close();  ?>  <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Comments</title>      <link rel="stylesheet" href="../css/reset.css" />      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">      <link rel=" stylesheet" href="../css/header.css" />      <link rel="stylesheet" href="../css/mobile.css" />      <script src="../js/delete.js" defer></script>  </head>  <body>      <header>          <nav>              <li><a href="index.html" id="logo"><span style="color: crimson">D</span>aniel.EC</a></li>              <li>              <li><a href="../html/index.html">Home</a></li>              <li><a href="../php/viewblog.php">Blog</a></li>              <li><a href="../html/about.html">About me</a></li>              <li><a href="../html/skills.html">Skills and Experience</a></li>              <li><a href=logout.php>Logout</a></li>              </li>          </nav>          <li id="logo-list"><a class="text-dark" href="index.html" id="logo2"><span                      style="color: crimson">D</span>aniel.EC</a></li>          <div class="dropdown">              <button class="btn bg-dark btn-primary dropdown-toggle" type="button" id="dropdownMenuButton"                  data-bs-toggle="dropdown" aria-expanded="false">                  Menu</button>              <ul class="dropdown-menu bg-dark" aria-labelledby="dropdownMenuButton">                  <li><a class="dropdown-item text-white" href='index.php'>Home</a></li>                  <li><a class="dropdown-item text-white" href='viewblog.php'>Blog</a></li>                  <li><a class="dropdown-item text-white" href='../html/about.html' 1?>About me</a></li>                  <li>                      <a class="dropdown-item text-white" href='../html/skills.html'>Skills and Experience</a>                  </li>                  <?php if ($\_SESSION["loggedIn"] == True) {                      echo "<li><a class='dropdown-item text-white' href=logout.php>Logout</a></li>";                  } else {                      echo "<li><a class='dropdown-item text-white' href=login.php>Sign In/Up</a></li>";                  } ?>                  </li>              </ul>          </div>      </header>      <!-- <?php echo $firstName ?> -->      <!-- //display the post -->      <div class=card shadow-sm>          <div class="card-body mb-3">              <h3 class=card-title><?php echo ($post["title"]); ?></h3>              <h6 class=card-subtitle mb-2 text-muted>                  By <strong><?php echo ($\_SESSION["admin"]); ?></strong> on                  <?php $zone = new DateTimeZone($post["zone"]);                  $date = new Datetime($post["time"], $zone);                  echo ($date->format("jS F Y g:i A T")); ?>              </h6>              <br>              <p class=card-text mt-3>                  <?php echo ($post["content"]); ?>              </p>          </div>      </div>      <hr>      <h4 class="display-6 text-dark">Comments</h4>      <hr>      <!-- //comments form, single line with bootstrap -->      <!-- the url encode adds the previous post id that was used to display the post , this is added to the post of the comment so that the else statement is not run -->      <form action="<?php echo htmlspecialchars($\_SERVER['PHP\_SELF']) . '?id=' . urlencode($\_GET['id']); ?>"          method="POST">          <div class="d-flex justify-content-center mt-4 w-100">              <input type="text" id="comment" name="comment" class="form-control w-50 me-2"                  placeholder="Add a comment...">              <input type="submit" class="btn btn-primary" value="Post">          </div>      </form>      <?php if ($noComments === true) {          echo '<h6 class="mt-3 text-dark">No Comments</h6>';      } ?>      <!--  //display all comments -->      <?php foreach ($rows as $row): ?>      <div class=card shadow-sm>          <div class=card-body>              <h6 class=card-subtitle mb-3 text-muted>                  By <strong> <?php echo ($row["username"]); ?> </strong> on <?php echo ($row["date"]); ?>              </h6>              <br>              <p class=card-text mt-3>                  <?php echo ($row["content"]); ?>              </p>              <?php if ($\_SESSION["loggedIn"] == true && $\_SESSION["email"] == "Daniel@admin") {                      echo "<a class=commentdel style='color:red' href=deleteComment.php?id=" . $row["ID"] . ">Delete</a> </>";                  } ?>          </div>      </div>      <hr>      <?php endforeach; ?>      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>      <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  </body>  </html>  **Filename: deletecomment.php**  <?php  session\_start();  $servername = "127.0.0.1";  $username = "root";  $password = "";  $dbname = "ecs417";  //create connection  $conn = new mysqli($servername, $username, $password, $dbname);  if ($conn->connect\_error) {      die("Connection failed:" . $conn->connect\_error);  }  $sql = "SELECT \* FROM portfolio\_comment WHERE ID='" . $\_GET["id"] . "'";  $result = $conn->query($sql);  if ($result->num\_rows > 0) {      $rows = $result->fetch\_all(MYSQLI\_ASSOC);      $postID = $rows[0]["postID"];  }  $sql = "DELETE FROM portfolio\_comment WHERE ID='" . $\_GET["id"] . "'";  if ($conn->query($sql) === TRUE) {      echo '<script>alert("Comment deleted successfully");</script>';      echo '<script>window.location.href="comment.php?id=' . $postID . '"</script>';  } else {      echo '<script>alert("Error deleting comment");</script>';      echo '<script>window.location.href="comment.php?id=' . $postID . '"</script>';  }  ?>  **Filename: deletepost.php**  <?php  $servername = "127.0.0.1";  $username = "root";  $password = "";  $dbname = "ecs417";  $conn = new mysqli($servername, $username, $password, $dbname);  if ($conn->connect\_error) {      die("Connection failed:" . $conn->connect\_error);  }  session\_start();  if ($\_SESSION["loggedIn"] == true && $\_SESSION["email"] = "Daniel@admin") {      if (isset($\_GET["id"])) {          $id = $conn->real\_escape\_string($\_GET["id"]);          $sql = "DELETE FROM portfolio\_post WHERE ID='$id'";          if ($conn->query($sql) === TRUE) {              echo '<script>alert("Post deleted Successfully");</script>';              header("Location:viewblog.php");          } else {              echo ('<script>alert("Post not deleted");</script>');          }      }  }  $conn->close();  ?>  **Filename: delete.js**  const posts=document.querySelectorAll(".postdel");  posts.forEach(function(post){      post.addEventListener("click",function(e){          if(confirm("Are you sure you want to delete post?")){              return;          }          else{              e.preventDefault();              return false;          }      });  });  const comments=document.querySelectorAll(".commentdel");  comments.forEach(function(comment){      comment.addEventListener("click",function(e){          if(confirm("Are you sure you want to delete comment?")){              return;          }          else{              e.preventDefault();              return false;          }      });  }); | **How did you develop this feature? Describe the steps involved.**  I developed this feature by creating six files named login.php, comment.php and signup.php, deletecomment.php, deletepost.php and delete.js for confirming the deletes. I also created two tables in the database to store user details and comments.  I started by creating a table in my database named portfolio\_users, this table is different from the admin and only store information of users that are not admins. I then created a signup form that people can use to register to become users, this form posts information to the PHP part of itself, the information is then entered into the portfolio\_usesrs database. When the users go to the login page and enter their details, the self-processing file will take their email and password that have been submitted and check if it matches anyone in the database, if it does, they are redirected to the view blog page and will see special options like comments and also have access to the comment page where they can add comments and view comments.  **Describe the inner workings of the approach you developed.**  The signup.php file simply takes input from a form, the initial validations were done using the HTML5 validation attributes, this attributes makes it impossible to submit the form unless the correct format is matched, they include using required to make sure a field is never left empty, types to specify the type of input required on a field and making sure whatever was entered was of the right type, below is a screenshot of trying to submit the signup.php file when a field is left empty.    This happens for all fields.  The password validation was done using php, the script simply checks if the submitted password and confirm password are the same, if it is, it goes on to load the values in a new row in the database, else it sends an alert and redirects back to the signup page.  Below is a screenshot of the output for when the password and confirm password matches and when they do not.    The error message is shown and then user is redirected back to signup.php below after clicking ok.    It also goes without saying that after redirection, the user details are not stored in the table in the database, as can be seen below:    Now, when the valid details are entered and the form is submitted, The details are stored in the database, a successful registration message is shown by echoing a javascript alert and the user is redirected to the login page by setting window.location.href value using php echo, this is used instead of php header redirect so that the alert message can be shown.      In the login.php file, it should have been; the user enters their details in the form, if the password and email matches a record in the database, they are successfully logged in but I also used the same file and form to check for an admin login, so what I did was to first of all check if the entered details matches a record in the admin table of the database, if it does then the admin is logged in and no further check is required but if the details don’t match an admin record, it now checks if it matches a user record, if it does, the user is logged in and session starts with session variables set depending on who logged in and if the records is still not seen then an invalid password and email alert is echoed and user is still in the form    When it’s an admin login, they are redirected to the addpost.php.    And on the view blog they have access to delete post and delete comment.  When it’s a normal registered user, they are redirected to the viewblog page after a successful login;    The view blog page illustrates more clearly on the user that is currently accessing the page; below is a screenshot of its state depending on the user type.  Registered User:  Have access to the comments page, has their first name on the welcome section.  The nav bar shows logout, to indicate they are currently logged in.    Guest User:  Can only view blog, no access to comments page.  The navbar shows sign/up to show they are not currently logged in and can log in or register.  The welcome message says guest.    Admin:  Have access to comments, can delete post, can add post, plus all the functionalities of a registered user when logged in.    The comments functionality was achieved by first of all creating a table in my database named portfolio\_comments, this table has a time field that is default i.e. the SQL automatically inserts it, each comment also has an ID as a primary key and a postID as a secondary key that references the ID in the post table, I also added the DELETE ON CASCADE command that deletes the comment if the post is deleted.  When the logged in user clicks on comments link from the viewblog page they are taken to the comment page where the post they are trying to comment is fetched from the database and displayed and an input field is visible for them to add comment and post it. They can also view other peoples’ comment.  When they click on post after entering their comment, the form is submitted to the same page and entered into the database with the first name of the person who’s currently logged in.    Because I am currently logged in as an admin, I can delete comments:    When I click yes:    Confirmation:    The delete functionality for both the comment and post is achieved by simply taking the unique auto incremented id of the post or comment and running an sql query that deletes any row of comment or post with that same id. |

|  |  |
| --- | --- |
| **Assessment Criteria 9 - Extra Feature 2**  (Note: delete any of the two features below)   * Organise your blog entries into different months. Provide a drop-down menu for the blog viewers to view the entries of different months stored in an entry archive. | |
| Instruction: use this column to paste the code from the files that have been used to develop this functionality.  **Paste the code directly from visual studio code, do not include a screenshot of the code as this document has a Turnitin requirement and this will lead to an automatic mark of zero. See example below.** | Instruction: use this column to address the following in your own words:   1. How did you develop this feature? Describe the steps involved. 2. Describe the inner workings of the approach you developed.   Did you use any Generative AI tools to achieve this? If yes, can you provide the prompts you wrote to achieve this?  Give reference to the code you have written when answering the questions above.  You can use diagrams (e.g. flow charts) or screenshots of your webpage output to illustrate your answer. |
| Filename: sort.js   document.getElementById("sort").addEventListener("change", function () {      const sortValue = this.value;      if (sortValue) {        window.location.href = `viewblog.php?sort=${sortValue}`;      }    });  Filename: viewblog.php  <?php  session\_start();  if (!isset($\_SESSION["loggedIn"])) {      $\_SESSION["loggedIn"] = false;      $\_SESSION["email"] = "Guest";      $\_SESSION["firstName"] = "Guest";  }  $servername = "127.0.0.1";  $username = "root";  $password = "";  $dbname = "ecs417";  //create connection  $conn = new mysqli($servername, $username, $password, $dbname);  if ($conn->connect\_error) {      die("Connection failed:" . $conn->connect\_error);  }  //if sort is not set, do default getting  $month = 0;  $desc = 0;  if (!isset($\_GET["sort"])) {      $month = 0;  } else {      if ($\_GET["sort"] == "1") {          $month = 1;      } else if ($\_GET["sort"] == "2") {          $month = "2";      } else if ($\_GET["sort"] == "3") {          $month = "3";      } else if ($\_GET["sort"] == "4") {          $month = "4";      } else if ($\_GET["sort"] == "5") {          $month = "5";      } else if ($\_GET["sort"] == "6") {          $month = "6";      } else if ($\_GET["sort"] == "7") {          $month = "7";      } else if ($\_GET["sort"] == "8") {          $month = "8";      } else if ($\_GET["sort"] == "9") {          $month = "9";      } else if ($\_GET["sort"] == "10") {          $month = "10";      } else if ($\_GET["sort"] == "11") {          $month = "11";      } else if ($\_GET["sort"] == "12") {          $month = "12";      } else if ($\_GET["sort"] == "-1") {          $desc = -1;      }  }  if ($month === 0) {      $sql = "SELECT ID, title, content, time, zone FROM portfolio\_post";  } else if ($desc === -1) {      $sql = "SELECT ID, title, content, time, zone FROM portfolio\_post ORDER BY ID ASC";  } else {      $sql = "SELECT ID, title, content, time, zone FROM portfolio\_post WHERE MONTH(time) = $month";  }  //whatever was gotten as the query code will be passed here  $result = $conn->query($sql);  if ($result->num\_rows > 0) {      $initialArray = $result->fetch\_all(MYSQLI\_ASSOC);      $invalid = false;  } else {      $invalid = true;  }  //selection sort  //1. the outer loop iterates through the array, treating each element as the current maximum  //2. the inner loop iterates through the unsorted portion of the array to find the maximum element  //if a larger element is found, the index of the maximum element is updated  //3. after the inner loop, if the maximum element is not already in the correct position, it is swapped with the current index  //4. this process continues until the entire array is sorted in descending order  //5. the function returns the sorted array  //instead of using in values, im using the time of each post as a value converted by strtotime for easier comparison  function selectionSortByDate(array $posts): array  {      $n = count($posts);      for ($i = 0; $i < $n - 1; $i++) {          $maxIdx = $i;          for ($j = $i + 1; $j < $n; $j++) {              if (strtotime($posts[$j]['time']) > strtotime($posts[$maxIdx]['time'])) {                  $maxIdx = $j;              }          }          // Swap the found maximum with current index          if ($maxIdx !== $i) {              $temp = $posts[$i];              $posts[$i] = $posts[$maxIdx];              $posts[$maxIdx] = $temp;          }      }      return $posts;  }  //pass the posts to be sorted if only the sort was set to a value not -1 i.e not oldest  if ($desc == -1) {      $rows = $initialArray;  } elseif (!$invalid) {      //sort the posts by date using selection sort      $rows = selectionSortByDate($initialArray);  }  $sql = "SELECT \* FROM portfolio\_admin WHERE ID=1";  $admin = $conn->query($sql);  if ($admin->num\_rows > 0) {      $admin = $admin->fetch\_all(MYSQLI\_ASSOC);      $\_SESSION["admin"] = $admin[0]["name"];  }  $admin = $admin[0];  ?>  <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Daniel's Blog</title>      <link rel="stylesheet" href="../css/reset.css" />      <link rel="stylesheet" href="../css/index.css" />      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">      <link rel="stylesheet" href="../css/mobile.css" />      <script src="../js/delete.js" defer></script>      <script src="../js/sort.js" defer></script>  </head>  <body id="view" style="background-color: black;">      <header>          <nav>              <li><a href="index.html" id="logo"><span style="color: crimson">D</span>aniel.EC</a></li>              <li>              <li><a href="../html/index.html">Home</a></li>              <li><a href="../php/viewblog.php">Blog</a></li>              <li><a href="../html/about.html">About me</a></li>              <li><a href="../html/skills.html">Skills and Experience</a></li>              <?php if ($\_SESSION["loggedIn"] == True) {                  echo "<li><a href=logout.php>Logout</a></li>";              } else {                  echo "<li><a href=login.php>Sign In/Up</a></li>";              } ?>              </li>          </nav>          <li id="logo-list"><a href="index.html" id="logo1"><span style="color: crimson">D</span>aniel.EC</a></li>          <div class="dropdown">              <button class="btn bg-dark btn-primary dropdown-toggle" type="button" id="dropdownMenuButton"                  data-bs-toggle="dropdown" aria-expanded="false">                  Menu</button>              <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton">                  <li><a class="dropdown-item " href='index.php'>Home</a></li>                  <li><a class="dropdown-item" href='viewblog.php'>Blog</a></li>                  <li><a class="dropdown-item" href='../html/about.html'>About me</a></li>                  <li>                      <a class="dropdown-item" href='../html/skills.html'>Skills and Experience</a>                  </li>                  <?php if ($\_SESSION["loggedIn"] == True) {                      echo "<li><a class='dropdown-item' href=logout.php>Logout</a></li>";                  } else {                      echo "<li><a class='dropdown-item' href=login.php>Sign In/Up</a></li>";                  } ?>                  </li>              </ul>          </div>      </header>      <div class="container bootstrap-container mt-5">          <h1 class="display-4 text-white">Daniel's Blog</h1>          <?php if ($\_SESSION["loggedIn"] == true && $\_SESSION["email"] == "Daniel@admin") {              echo "<h4 class='display-6 text-white'>Welcomes Admin</h4>";              echo "<a href='../php/addpost.php' class='btn btn-primary'>Add post +</a>";          } else if ($\_SESSION["loggedIn"] == true) {              echo "<h4 class='display-6 text-white'>Welcomes " . $\_SESSION["name"] . "</h4>";          } else {              echo "<h4 class='display-6 text-white'>Welcomes Guest</h4>";          } ?>          <div class="sort-container">              <label for="sort">Sort By:</label>              <select id="sort" name="sort">                  <option value="check">                      <?php if ($month == 0 && $desc == 0) {                          echo "Most recent";                      } else if ($month == 1)                          echo "January";                      else if ($month == 2)                          echo "February";                      else if ($month == 3)                          echo "March";                      else if ($month == 4)                          echo "April";                      else if ($month == 5)                          echo "May";                      else if ($month == 6)                          echo "June";                      else if ($month == 7)                          echo "July";                      else if ($month == 8)                          echo "August";                      else if ($month == 9)                          echo "September";                      else if ($month == 10)                          echo "October";                      else if ($month == 11)                          echo "November";                      else if ($month == 12)                          echo "December";                      else if ($desc == -1)                          echo "Oldest";                      ?>                  </option>                  <option value="0">Most Recent</option>                  <option value="-1">Oldest</option>                  <option value="1">January</option>                  <option value="2">February</option>                  <option value="3">March</option>                  <option value="4">April</option>                  <option value="5">May</option>                  <option value="6">June</option>                  <option value="7">July</option>                  <option value="8">August</option>                  <option value="9">September</option>                  <option value="10">October</option>                  <option value="11">November</option>                  <option value="12">December</option>              </select>          </div>          <?php if ($invalid): ?>              <div class="alert alert-danger" role="alert">                  No posts available for the selected month.              </div>          <?php else: ?>              <?php foreach ($rows as $row): ?>                  <div class="card w-100 my-4" style="max-width: 720px; margin-left: auto; margin-right: auto;">                      <div class="card-body">                          <h3 class="card-title"><?php echo htmlspecialchars($row["title"]); ?></h3>                          <h6 class="card-subtitle mb-2 text-muted">                              By <strong><?php echo htmlspecialchars($admin["name"]); ?></strong> on                              <?php                              $zone = new DateTimeZone($row["zone"]);                              $date = new DateTime($row["time"], $zone);                              echo $date->format("jS F Y g:i A T");                              ?>                          </h6>                          <p class="card-text mt-3"><?php echo htmlspecialchars($row["content"]); ?></p>                          <?php if ($\_SESSION["loggedIn"]): ?>                              <small><a href="comment.php?id=<?php echo $row["ID"]; ?>">Comments</a></small>&nbsp;                          <?php endif; ?>                          <?php if ($\_SESSION["loggedIn"] && $\_SESSION["email"] === "Daniel@admin"): ?>                              <small><a class="postdel text-danger" href="deletepost.php?id=<?php echo $row["ID"]; ?>">Delete                                      post</a></small>                          <?php endif; ?>                      </div>                  </div>              <?php endforeach; ?>          <?php endif; ?>          <?php $conn->close(); ?>      </div>      <footer class="bg-white text-dark py-4 mt-5">          <div class="container text-center">              <!-- Social Icons -->              <div class="d-flex justify-content-center gap-4 mb-3">                  <a href="https://www.linkedin.com/in/danielewelikechimezie" class="img-fluid social-icon"                      target="\_blank">                      <img src="../images/ln.webp" alt="LinkedIn" width="32" height="32">                  </a>                  <a href="https://github.com/Daniel-ec5" class="img-fluid social-icon" target="\_blank">                      <img src="../images/github-128.webp" alt="GitHub" width="32" height="32">                  </a>                  <a href="mailto:Daniel.ec5@Outlook.com?subject=Portfolio%20Contact" class="img-fluid social-icon">                      <img src="../images/mail.webp" alt="Mail" width="32" height="32">                  </a>                  <a href="https://snapchat.com/add/daniel\_el22" class="img-fluid social-icon" target="\_blank">                      <img src="../images/spo.webp" alt="Snapchat" width="32" height="32">                  </a>              </div>              <!-- Footer Text -->              <p class="mb-0 text-dark">&copy; 2025 Daniel Ewelike</p>          </div>      </footer>      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>  </body>  </html> | **How did you develop this feature? Describe the steps involved.**  The sorting by month is made possible by the sort.js  file that acts as the middleman and detects and  passes changes to the part of the viewblog.php  file that fetches the posts from the database,  the sort by recent is also by default applied to the posts fetched for each month so long as the selected  sort type is not by oldest.  **Describe the inner workings of the approach you**  **developed.**  The list with the different months and sort types were achieved by putting each of the options in an option tag that is nested in a select element, what this does is that it creates a menu where users can select the month they want to view, and this selection is identified through a JavaScript code in the sort.js file:  Filename: sort.js   document.getElementById("sort").addEventListener("change", function () {      const sortValue = this.value;      if (sortValue) {        window.location.href = `viewblog.php?sort=${sortValue}`;      }    });  The code snippet above simply listens for any changes in the selection element options, so if an option is clicked, the value of that option is sent to viewblog.php as a sort value, the values are numbered from -1 representing sort by oldest, 0 representing sort by recent and the values 1 to 12 representing all the months of the year in ascending order.  In the viewblog file, the php first of all checks if any value was passed  i.e.  if (!isset($\_GET["sort"])) {      $month = 0;  }  From the snippet above, if nothing is set as the sort-type, it defaults to 0 which represent sort by most recent, if a value is passed, then it uses a bunch of if..if-else statements to assign the  $month variable a number representing that value.  An SQL query is then run with the command being:      $sql = "SELECT ID, title, content, time, zone FROM portfolio\_post WHERE MONTH(time) = $month";  This is not done when sort by oldest is selected because -1 cannot be represented as a month, instead, the sorting by recent function is not called on the query result like below:  if ($desc == -1) {      $rows = $initialArray;  } elseif (!$invalid) {      //sort the posts by date using selection sort      $rows = selectionSortByDate($initialArray);  }  The $desc variable here also represents the if sort by oldest was selected.  The screenshot below shows example of the output of the above implemented extra feature:    The first option simply displays the current sort being applied, and changes depending on a selected month. It is achieved by adding a bunch of if statements that outputs the month depending on the value stored in the $month variable in the php code at the beginning of the file.                  <option value="check">                      <?php if ($month == 0 && $desc == 0) {                          echo "Most recent";                      } else if ($month == 1)                          echo "January";                      else if ($month == 2)                          echo "February";                      else if ($month == 3)                          echo "March";                      else if ($month == 4)                          echo "April";                      else if ($month == 5)                          echo "May";                      else if ($month == 6)                          echo "June";                      else if ($month == 7)                          echo "July";                      else if ($month == 8)                          echo "August";                      else if ($month == 9)                          echo "September";                      else if ($month == 10)                          echo "October";                      else if ($month == 11)                          echo "November";                      else if ($month == 12)                          echo "December";                      else if ($desc == -1)                          echo "Oldest";                      ?>                  </option>  From the screenshot above, it can be seen that if the month selected has no posts, a message is displayed in danger red color telling the user that there was no post available for selected month, this is achieved by using a Boolean flag $invalid , that is assigned true if the sql query for the specific month returns 0 rows i.e. no posts for that month and in the html part that displays the post is only ran if invalid is false, else it displays this error message. |
| **Assessment Criteria 14 - Web technologies/framework**  For this project, you have made use of HTML, CSS, JavaScript and PHP. If you were to conduct this project again, which other web technologies/frameworks would you use to develop your portfolio site? Justify your choice. | |
| For this project, I have made use of vanilla CSS, JavaScript and html for the part 1 and added a bit of bootstrap CSS classes for the part 2. The little I added has given me a glimpse on the importance of frameworks and other modern web technologies in web developing. These frameworks will save time, promote code reuse and provide a professional and clean user interface to my websites in the future. On my remake of this project and for further projects, I’ll use the following frameworks:  **REACT**  This is a powerful JavaScript Library that will let me build a clean and professional website in less time, this is achieved because of its component-based structure that lets me reuse its already built functionalities for some common tools like navbars, forms, blog cards, etc. In comparison with bootstrap JavaScript plugins, it has a virtual DOM that renders only the updates and not the whole page, making it way better than the normal DOM. I chose it also because of its JSX structure, where I can write html markup inside JavaScript, this will make it easier to visualize when coding as it looks close to what’s rendered on the screen and the numerous other benefits it provides. Part of my main reasons for using it is also because of its growing huge online support and materials which makes it easier to learn, understand and apply.  **TAILWIND CSS**  This will probably be my future go to CSS framework for styling my web pages, I’ll prefer this over bootstrap simply because it’s high customizability. Unlike bootstrap, which offers precompiled codes and styles for the different components needed in web programming, Tailwind has utility classes that assists in building a better UI. From my limited understanding with bootstrap, most of the components are already created with the styles applied to them, manually changing and adding a bit of your own code sometimes breaks it out of flow, particularly for the classes that implement a grid like layout. The above statement was made from experience. Tailwind is more utility based rather than component based i.e. it supports you to build a better and cleaner styling through its micro classes rather than giving you it’s already assembled styling with little customizability. Because of this too, it has less unused class styling imports.  **LARAVEL**  I am probably biased towards Laravel as opposed to Express.js just because of the ability to write queries in the PHP language rather than in SQL and this query is translated to SQL on its own, from my study, this is achieved by its ORM(Object Relational Mapper) which lets developers interact with their database using classes and objects instead of writing raw SQL queries thereby streamlining CRUD operations. This functionality also automatically prevents malicious access through SQL injections. It’s inbuilt authentication system provides a quick, secure and CUSTOMIZABLE way to handle user logins, registrations and managements. It provides easy setups for password and email recovery that is popular in login forms. It offers all this and more while still being highly customizable. | |