Criterion C: Development

Table of Contents

Key Features Used

Feature	Purpose / Value
Dynamically created form elements based on SQL results with sorting	Allows for users to locate and go to events across the whole calendar.
Dynamically output records from SQL based on the secondary key of a database table	Keeps track of which notifications need to be displayed for which user. The notifications then need to be output in the main page.
Removal of a record in a SQL database based on the id of a HTML element	To remove an event chosen by the user on the main screen, with confirmation when deleting.
Dynamic creation of record in a SQL database depending on a HTML form	To create an event based on user input and insert the information into the database to be displayed on the main screen.
Dynamic creation of records in a SQL database based on dates in the gregorian calendar	To set up the dates table in my database with all the dates within a reasonable range (2 years)
Dynamically echo unordered list elements depending on PHP	To display all dates in a month with all events in each day in the correct order and places.

session variables	
Creation of cookies depending on a HTML form	To ensure that users will not have to log in every time if they have the remember me option on
Modifying PHP session variables in order to affect HTML elements	To go forward and backward a month and updating the main page depending on the month

Dynamically created form elements based on SQL results with sorting

```
$searchCount = 0;
$sql = "SELECT * FROM EVENTS WHERE Event_Name LIKE '%" . $searchArg . "%'";
$result = $conn->query($sql);
$count = 0;
while ($row = $result->fetch_assoc()) {
    echo "<form action='' method='POST' autocomplete='off'>";
    echo "<input type='hidden' name = 'goToVal' value='$count'>";
    echo $row['Event_Name'] . "  ";
    echo "<input type='submit' name = 'goTo' value='Go to event'>";
    echo "</form>";
    $sql2 = "SELECT * FROM `DATES` WHERE D_ID = '" . $row['D_ID'] . "'";
    $result2 = $conn->query($sql2);
    $date = $result2->fetch_assoc();
    $_SESSION["{'Event_Date'.$count}"] = $date['DATE'];
    $alrSearched[$searchCount] = $row['E_ID'];
    $searchCount ++;
    $count++;
}
$numWord = str_word_count($searchArg);
$wordArr = str_word_count($searchArg, 1);
for ($i = 0; $i < $numWord; $i++) {
    $sql = "SELECT * FROM EVENTS WHERE Event_Name LIKE '%" . $wordArr[$i] . "%'";
    $result = $conn->query($sql);
    $count = 0;
    $identicalCount = 0;
    while ($row = $result->fetch_assoc()) {
        for ($j = 0; $j < $searchCount; $j++){
        if ($row['E_ID'] == $alrSearched[$j]){
            $identicalCount ++;
        }
        }
        if ($identicalCount == 0){
            $alrSearched[$searchCount] = $row['E_ID'];
            $searchCount ++;
            echo "<form action='' method='POST' autocomplete='off'>";
        echo "<input type='hidden' name = 'goToVal' value='$count'>";
        echo $row['Event_Name'] . "  ";
        echo "<input type='submit' name = 'goTo' value='Go to event'>";
        echo "</form>";
        $sql2 = "SELECT * FROM `DATES` WHERE D_ID = '" . $row['D_ID'] . "'";
        $result2 = $conn->query($sql2);
        $date = $result2->fetch_assoc();
        $_SESSION["{'Event_Date'.$count}"] = $date['DATE'];
        $count++;
        }
```

Explanation: Displays events with the same/similar name as the search argument. Achieves the goal of showing users what they searched for with suggestions..
INGENUITY: Shows all results with each word of the search argument in them
COMPLEXITY: Keeps track of ids of results already displayed in an array so duplicate results are not shown

Cfs 4d: Users should be able to search for keywords on the calendar, with suggestions

Dynamically output records from SQL based on the secondary key of a database table

```
<div id='notif' class='notifModal'>
    <div class="notifModal-content">
        <span class="closeNotif">&times;</span>
        <?php
        echo "<h1 class = 'dayDetailName'> Notifications </h1>";
        echo " <ul class = 'dayDetail'><br>";
        $sql = "SELECT * FROM `NOTIFICATION` WHERE U_ID = '" . $_SESSION['U_ID'] . "'";
        $resultn = $conn->query($sql);
        if ($_SESSION["notifCount"] == 0) {
            echo "No notifications!";
            echo "<br><br>";
        } else {
            while ($rown = $resultn->fetch_assoc()) {
                echo "<li>" . $rown['Content'] . "</li><br>";
            }
            echo "<form action='' method='post'>
            <br><button type='submit' name='btn' value='CLEAR'>Clear all</button>
        </form>";
            echo "</ul>";
        }
        ?>

    </div>
</div>
```

Explanation: Allows for the checking/outputting of notifications from the SQL database. Helps achieve the goal of showing all relevant notifications for the specific user.
INGENUITY: Notifications are inside a modal and are hidden until the user needs to see it.
COMPLEXITY:Checking and displaying different things depending on if the user has any notifications
Cfs 5: Users should be notified by a notification section in the website itself

Removal of a record in a SQL database based on the id of a HTML element

```
$eventid = $_GET['event_id'];
if (!empty($_POST["btn"])) {
if ($_POST["btn"] == "YES"){
        $sql = "SELECT Username FROM `USER` WHERE U_ID = '".$_SESSION['U_ID']."'";
        $result = $conn->query($sql);
        $uname = $result->fetch_assoc();

        $sql = "SELECT Event_Name FROM `EVENTS` WHERE E_ID = '$eventid'";
        $result = $conn->query($sql);
        $eventName = $result->fetch_assoc();

        $sql = "SELECT D_ID FROM `EVENTS` WHERE E_ID = '$eventid'";
        $result = $conn->query($sql);
        $temp = $result->fetch_assoc();


        $sql = "SELECT DATE FROM `DATES` WHERE D_ID = '".$temp['D_ID']."'";
        $result = $conn->query($sql);
        $date = $result->fetch_assoc();

        $sql = "SELECT * FROM `USER`";
        $result2 = $conn->query($sql);
        while ($row = $result2->fetch_assoc()) {
            $uidcurr = $row['U_ID'];
            $sql = "INSERT INTO `NOTIFICATION`(Content,U_ID)VALUES('".$uname['Username']."has removed an event:".$eventName['Event_Name']."for th
            $dosql = $conn->query($sql);
        }

        $sql = "DELETE FROM `EVENTS` WHERE E_ID='$eventid'";
        $result = $conn->query($sql);
```

Explanation: GETs the event id from the element it was redirected from, and deletes that event from the

database. Achieves the goal of being able to remove an event.
INGENUITY: By using an id of the element which is the same value as the event id, the correct event was selected and deleted
COMPLEXITY: Querying the database with the id that was retrieved from the element
Cfs 2 Users should be able to input/remove events

Dynamic creation of record in a SQL database depending on a HTML form

```php
$err = ' ';
if (!empty($_POST["date"])) {

    $startTime = filter_input(INPUT_POST, 'startTime');
    $endTime = filter_input(INPUT_POST, 'endTime');
    if ($startTime < $endTime) {
        $date = filter_input(INPUT_POST, 'date');
        $eventName = filter_input(INPUT_POST, 'eventName');
        $priority = filter_input(INPUT_POST, 'priority');
        $color = filter_input(INPUT_POST, 'color');
        $notes = filter_input(INPUT_POST, 'notes');
        $uid = $_SESSION['U_ID'];
        $sql = "SELECT D_ID FROM `DATES` WHERE DATE = '$date'";
        $result = $conn->query($sql);
        $recordData = $result->fetch_assoc();
        $did = $recordData["D_ID"];
        $sql = "INSERT INTO `EVENTS` (U_ID, D_ID, Notes, Start_Time, End_Time, Priority, Event_Name, Color) VALUES ('$uid','$did','$notes','$sta
        $dosql = $conn->query($sql);
        $sql = "SELECT Username FROM `USER` WHERE U_ID = '".$_SESSION['U_ID']."'";
        $result = $conn->query($sql);
        $uname = $result->fetch_assoc();
        $sql = "SELECT * FROM `USER`";
        $result2 = $conn->query($sql);
        while ($row = $result2->fetch_assoc()) {
            $uidcurr = $row['U_ID'];
            $sql = "INSERT INTO `NOTIFICATION` (Content, U_ID) VALUES ('".$uname['Username']." has created a new event: ".$eventName." for the d
            $dosql = $conn->query($sql);
        }
            $newURL = "http://localhost:8081";
            header('Location: ' . $newURL);
        } else {
            $err = "Error. Start time must be before end time.";
        }
    }
```

Explanation: Uses the data that is posted from a form, checks its validity and creates an element in the database with that data. Achieves the goal of being able to create an event.
INGENUITY: Setting variables with the posted information. Values are used multiple times without having to retrieve them again
COMPLEXITY: Error check for the starting time and ending time, with an error message.
Cfs 2 Users should be able to input/remove events

Dynamic creation of records in a SQL database based on dates in the gregorian calendar

```sql
SET @StartDate = '2022-08-01';
SET @EndDate = '2024-08-01';
SET @DayNumWeek = 1;
SET @Curr = @StartDate;
SET @DayNameWeek = "Monday";

DELIMITER //
CREATE PROCEDURE while_loop()
BEGIN
   WHILE (@Curr <= @EndDate) DO
    SELECT CASE WHEN (@DayNumWeek = 1) THEN "Monday"
        WHEN (@DayNumWeek = 2) THEN "Tuesday"
        WHEN (@DayNumWeek = 3) THEN "Wednesday"
        WHEN (@DayNumWeek = 4) THEN "Thursday"
        WHEN (@DayNumWeek = 5) THEN "Friday"
        WHEN (@DayNumWeek = 6) THEN "Saturday"
        WHEN (@DayNumWeek = 7) THEN "Sunday"
    END INTO @DayNameWeek;
   INSERT INTO  `DATES` (`DATE`,`MONTH`,`DAY`,`YEAR`,`NAME_DAY`) VALUES (@Curr, MONTH(@Curr), DAY(@Curr), YEAR(@Curr), @DayNameWeek);
   SET @Curr =ADDDATE(@Curr, 1);
   SELECT CASE WHEN (@DayNumWeek = 7) THEN 1
   ELSE @DayNumWeek + 1
   END INTO @DayNumWeek;

END WHILE;
END//
DELIMITER ;
CALL while_loop();
```

Explanation: Loops through each day within range, inserting them into the SQL database. Achieves the goal of storing all days within an appropriate range to be displayed later or referenced later.
INGENUITY: Switch statement is used to determine the value of DayNumWeek, which is used for the value of `NAME_DAY`
COMPLEXITY: Use of 2 switch statements inside a while loop. Verifying the value of DayNumWeek before incrementing/changing it.
Cfs 4a: Users should be able to see as far as a year ahead

Dynamically echo unordered list elements depending on PHP session variables

```php
    echo "<ul class = 'days_ul'>";
    for ($count = 1; $count < $_SESSION["startDay"]; $count ++) {
        echo "<li class='dayempty'> </li>";
    };
    $_SESSION["date"] = 1;
    for ($daycount = 1; $daycount <= cal_days_in_month(CAL_GREGORIAN, $_SESSION["month"], $_SESSION["year"]); $daycount ++) {

        echo '<li class="dayfull">';
        echo $_SESSION["date"];
```

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Explanation: Echoes empty/full day slots depending on the session variables. Achieves the goal of displaying the correct amount of days for each month.
INGENUITY: Usage of session variables in order to calculate the number of days in the current month
COMPLEXITY: Use of multiple loops in order to display the correct amount of empty/full days
Cfs 4 Users should be able to view events on the calendar

Creation of cookies depending on a HTML form

```php
if ($remember == 'y') {
    $rememberToken = bin2hex(random_bytes(32));
    $sql = "UPDATE `USER` SET Remember_Token = '$rememberToken' WHERE Username = '$uname'";
    $dosql = $conn->query($sql);
    $options = [
        'expires' => time() + 60 * 60 * 24 * 365, // cookie expires after a year
        'path' => '/' //cookie good for everything under the domain
    ];
    setcookie('Remember_Token', $rememberToken, $options);
}
```

Explanation: Inserts a randomly generated token into the database and creates a cookie with that value. Achieves the goal of giving the choice of not having to log in every time.
INGENUITY: Usage of a randomised string as the value for a cookie
COMPLEXITY: Cookie has an expiration date of 1 year, preventing unauthorised access far into the future
Cfs 8 Users should be able to register/use accounts

Modifying PHP session variables in order to affect HTML elements

```php
if ($_SESSION["month"] < '12') {
    $_SESSION["month"] += '1';
} else {
    $_SESSION["month"] = '1';
    $_SESSION["year"] += '1';
}
```

```php
$_SESSION["startDay"] = date("N", strtotime('20' . $_SESSION["year"] . '-' . $_SESSION["month"] . '-01'));
$_SESSION["endDay"] = date("N", strtotime('20' . $_SESSION["year"] . '-' . $_SESSION["month"] . '-t'));
```

Explanation: Modifies the session variable month depending on its value, and setting the startDay and endDay for using the variables. This affects the display of the number of days and the events within those days.
INGENUITY: Using a session variable to set the value of another session variable
COMPLEXITY: If statement checking if the month is december, adjusting the session variable year accordingly
Cfs 4 Users should be able to view events on the calendar

Sources

https://www.w3schools.com/php/

https://www.w3schools.com/sql/default.asp

https://www.php.net/docs.php

https://stackoverflow.com

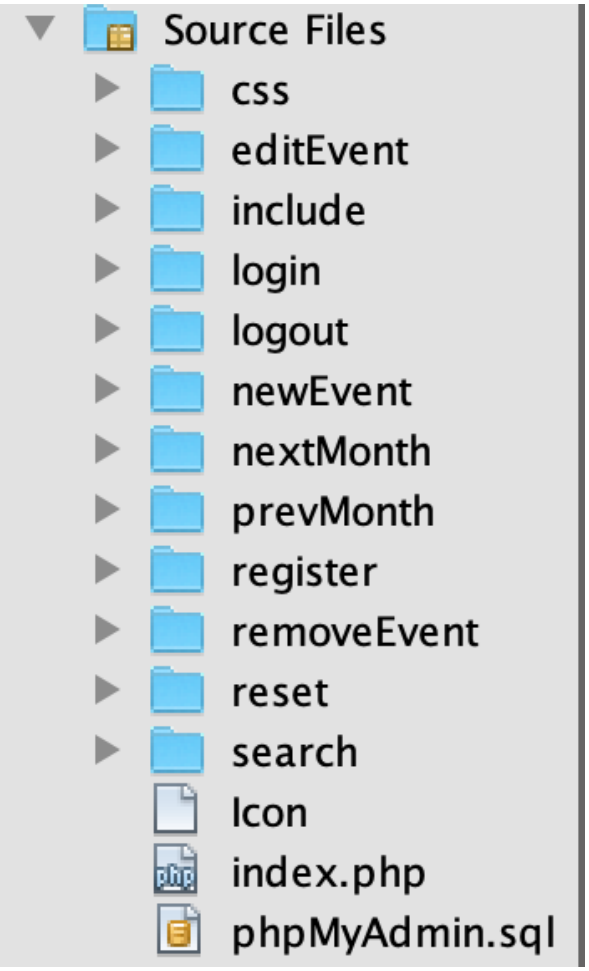
Extensibility

My variable/database table/function names are clear and easy to understand. There are comments about the purpose of some of my code which helps with returning to development in the future. My code is also formatted with indentations, and is separated into different segments/files for different functions/pages, making it easier to develop further in the correct places.

```
    <head>
        <meta charset="UTF-8">
        <title>Remove an event</title>
        <link rel="stylesheet" type="text/css" href="../css/styles.css">
    </head>
    <body>
<?php
// include 'template html elements'
$path = "/include/header.php";
// Include the file
include_once($root . $path);
$path = "/include/menu.php";
include_once($root . $path);
?>
        <h1 class="crev_header">Remove Event</h1>
            <div class="crev_container">
            <form action='' method='post'>
                        <p>You are about to remove an event. Are you sure?</p>
                        <br><br><br><button type='submit' name='btn' value='YES'>Yes, remove</button>
                        <br><br><br><button type='submit' name='btn' value='NO'>No, don't remove</button>
                    </form>
                </div>

<?php
// include 'template html elements'
$path = "/include/footer.php";
// Include the file
include_once($root . $path);
?>
```

Indentation and comments



Code organised into folders

```php
if ($_SESSION["init"] == FALSE) {
    setcookie('Remember_Token', null, $options);
    $_SESSION["LOGIN"] = FALSE;
    $_SESSION["month"] = intval(date('m'));
    $_SESSION["year"] = intval(date('y'));
    $_SESSION["today"] = date();
    $_SESSION["init"] = TRUE;
}
```

Variable names

```javascript
// open the modal
function showModal(clicked_id) {
    modal = document.getElementById("d" + clicked_id);
    modal.style.display = "block";
}

// close the modal
function closeModal(clicked_id) {
    modal = document.getElementById("d" + clicked_id);
    modal.style.display = "none";
}
```

Function names

```sql
CREATE TABLE `USER` (
  `U_ID` int(11) NOT NULL,
  `Username` text NOT NULL,
  `Password` text NOT NULL,
  `Remember` text NOT NULL,
  `Remember_Token` text NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Field names