

Feature list



















































Feature One -import CSV file to SQL table	1
Feature Two - randomAssign Process	4
Feature Three Save meal through sessions	6
Feature Four - downloading meals	8
Feature Five - Live search	10

Feature One -import CSV file to SQL table

meal	category
beef	steak
burger	meat
tacos	mexican
mashed potato	carb
escalope	turkey
fajita	mexican
stirfry	asian
noodles	asian
bbq	meat
fish	fish
salmon	fish
shrimp	fish
fries	carb
lasagna	carb
pizza	carb
pasta	chicken

File uploaded - csv file contents

+ Options

					id	meal	category	Assigned
<input type="checkbox"/>	 Edit	 Copy	 Delete		50	beef	steak	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		67	burger	meat	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		68	tacos	mexican	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		69	mashed potato	carb	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		70	escalope	turkey	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		71	fajita	mexican	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		72	stirfry	asian	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		73	noodles	asian	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		74	bbq	meat	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		75	fish	fish	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		76	salmon	fish	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		77	shrimp	fish	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		78	fries	carb	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		79	lasagna	carb	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		95	pizza	carb	0
<input type="checkbox"/>	 Edit	 Copy	 Delete		96	pasta	chicken	0

Feature one outcome

```

1  <?php
2
3  $dbHost    = "localhost";
4  $dbUsername = "hi";
5  $dbPassword = "hello";
6  $dbName    = "IB";
7
8
9  $db = new mysqli($dbHost, $dbUsername, $dbPassword, $dbName);
10
11
12  if ($db->connect_error) {
13      die("Connection failed: " . $db->connect_error);
14  }
15
16  if(isset($_POST['importSubmit'])){
17      |
18
19
20      $csvMimes = array('text/x-comma-separated-values', 'text/comma-separated-values', 'application/octet-stream', 'application/vnd.
21
22      if(!empty($_FILES['file']['name']) && in_array($_FILES['file']['type'], $csvMimes)){
23          if(is_uploaded_file($_FILES['file']['tmp_name'])){
24
25              $csvFile = fopen($_FILES['file']['tmp_name'], 'r');
26
27
28              fgetcsv($csvFile);
29
30              while(($line = fgetcsv($csvFile)) !== FALSE){
31
32                  $meal    = $line[0];
33                  $category = $line[1];
34
35
36
37
38                  $db->query("INSERT INTO meals (meal, category) VALUES ('".$meal."', '".$category."')");
39
40              }
41
42
43
44
45              fclose($csvFile);
46
47          }
48      }
49
50  }
51  echo '<script>location.href="assing.html";</script>';
52  ?>

```

Feature one code

Explanation

It checks if html form was submitted by using the “isset” condition followed by the variable \$_POST in line 16. If true, an array of CSV MIMES is created, MIMES are a format for data transmission and dictates how the data should be encoded, CSV files are accompanied by mimes to help applications interpret the file correctly. In line 22 it checks if the CSV mime of the file that was uploaded is in the array. The code then parses the file from line 25- 38. In line 25 the uploaded file is opened. Fgetcsv in line 28 starts parsing the file and indicates to skip the first line (includes column names). With a while loop in line 30, the program goes through the file line by line and inserts the contents of the meals column in the spreadsheet into the meal column within the meals table and the same with category column.

Justification

File upload is the easiest way for users to upload mass data the quickest way possible. Another way the user could input data would be through a form, however this would be tedious for the user especially considering the large quantities of data. Importing CSV files into SQL is also possible if done by the admin, however this makes it difficult for the client to make changes, impacting the usability and ingenuity of the product. By using a file upload on the user end it abstracts the complexity of the program and leaves all the programming for the back end database.

Feature 2 - randomAssign Process

sunday pasta
monday pizza
tuesday pasta
wednesday bbq
thursday fish
friday pizza
saturday salmon

refresh

save

Feature Two Outcome

```
19
20 $sql = "SELECT meal, category FROM meals";
21 $test = $con->query ($sql);
22 $meal = mysqli_fetch_all($test, MYSQLI_ASSOC);
23
24 $test = $con->query($sql);
25
26 $sql= mysqli_query($con,'select * from meals');
27 $check=mysqli_num_rows($sql);
28
29 $options=array();
30 for($i=0;$i<$check;$i++){
31     array_push($options,$i);
32 }
33 shuffle($options);
34
35 $sql = "SELECT meal, category FROM meals";
36 $test = $con->query ($sql);
37 $meal = mysqli_fetch_all($test, MYSQLI_ASSOC);
38
39
40 $sundaymeal = $options[0];
41 $mondaymeal = $options[1];
42 $tuesdaymeal = $options[2];
43 $wednesdaymeal = $options[3];
44 $thursdaymeal = $options[4];
45 $fridaymeal = $options[5];
46 $saturdaymeal = $options[6];
47
48 $name = $meal [$sundaymeal] ['meal'];
49 $category = $meal [$sundaymeal] ['category'];
```

Feature Two Code

Explanation

line 20 There's a query that retrieves meal and category columns from the meals table and stores it in “meal” associative array. To be able to randomize we need to know the number of data that we are working with, there's another query in line 26, that retrieves the contents of the meals column and the number of meals/rows of the table is held in var check.

array “option” is created, which holds integer values from 0 to int_val of \$check. a for loop in line 30 was created that uses array_push to add “\$i” into array “\$option”. The array will then be shuffled in line 33 so that for example the first index will not hold value 0, rather a random value from 0 to \$check.

Each index of options corresponds to a meal from the database, and then in line 40 to 46 a day of the week is associated with the meals.

```
51 - while ( ( $meal [$sundaymeal] ['category'] == $meal [$mondaymeal] ['category'] ) || ( $meal [$mondaymeal] ['category'] ==  
$meal [$tuesdaymeal] ['category'] ) || ( $meal [$tuesdaymeal] ['category'] == $meal [$wednesdaymeal] ['category'] ) || ( $meal  
[$tuesdaymeal] ['category'] == $meal [$wednesdaymeal] ['category'] ) || ( $meal [$wednesdaymeal] ['category'] == $meal  
[$thursdaymeal] ['category'] ) || ( $meal [$thursdaymeal] ['category'] == $meal [$fridaymeal] ['category'] ) || ( $meal  
[$fridaymeal] ['category'] == $meal [$saturdaymeal] ['category'] ) ) )  
  
52 - {  
    53- shuffle($options);  
    $sundaymeal = $options[0];  
    $mondaymeal = $options[1];  
    $tuesdaymeal = $options[2];  
    $wednesdaymeal = $options[3];  
    $thursdaymeal = $options[4];  
    $fridaymeal = $options[5];  
    $saturdaymeal = $options[6];  
}
```

The code is followed by a while loop with a conditional statement that ensures that there aren't any meal category duplicates.

Justification

I attempted using a for loop cycled through the days of the week and assigned a meal to it, this was not successful as the meals were not randomized and it was always the same outcome. I attempted to fix this using the “rand()” function, by setting the parameters from one - \$check. This didn't work as it would provide me with repeats and duplicates . A while loop with a similar condition as seen in line 52 was incorporated however this would take a lot of resources as the program is trying multiple possibilities to match the conditions and would even crash the code. Storing values in an array and shuffling it was the most resource friendly method, since values are taken directly from the database and not generated there will be no duplicates. Shuffling insures the order of the numbers is random rather than the numbers themselves which is more relevant to our desired outcome then using “rand()”

Feature Three Save meal through sessions

sunday pasta
monday pizza
tuesday pasta
wednesday bbq
thursday fish
friday pizza
saturday salmon

[Download](#)

return

Feature Three Outcome

```
4  session_start();
5
6  $server = "localhost";
7  $username = "hi";
8  $password = "hello";
9  $dbname = "IB";
10
11 $con = new mysqli($server, $username, $password, $dbname);
12 if ($con->connect_error) {
13     die("Connection failed: " . $con->connect_error);
14 }
15 else
16     { }
17
18 $dayone = " sunday\n ";
19 $sundaymeal = $_SESSION['sundaymeal'] ;
20 $mondaymeal = $_SESSION['mondaymeal'] ;
21 $daytwo = " monday\n ";
22 $tuesdaymeal = $_SESSION['tuesdaymeal'] ;
23 $daythree = " tuesday\n ";
24 $wednesdaymeal = $_SESSION['wednesdaymeal'];
25 $dayfour = " wednesday\n ";
26 $thursdaymeal = $_SESSION['thursdaymeal'];
27 $dayfive = " thursday\n ";
28 $fridaymeal = $_SESSION['fridaymeal'];
29 $daysix = " friday\n ";
30 $saturdaymeal = $_SESSION['saturdaymeal'];
31 $dayseven = " saturday\n ";
32
33 $week = array ($sundaymeal, $mondaymeal, $tuesdaymeal, $wednesdaymeal, $thursdaymeal, $fridaymeal, $saturdaymeal);
34 $day = array( "sunday", "monday", "tuesday", "wednesday", "thursday", "friday", "saturday" );
35 for ($x=0;$x<sizeof($week); $x++) {
36     $meal = $week[$x];
37     $days = $day[$x];
38     $sql = "INSERT INTO PAST (meal, day) VALUES ('$meal', '$days' )";
39     if (mysqli_query($con, $sql)) {
40
41     } else {
42
43     }
44 }
45
46 }
```

Feature Three Code

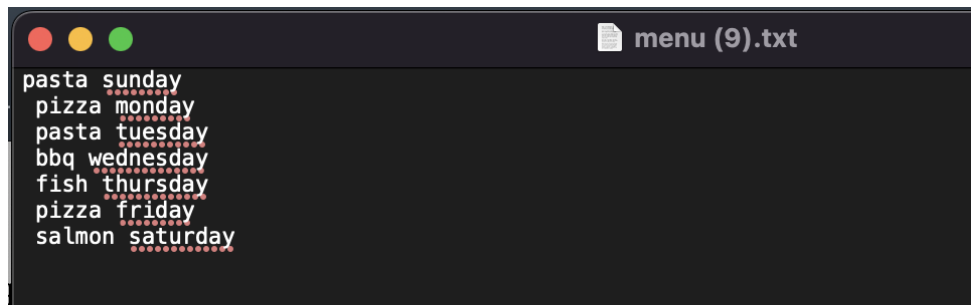
Explanation

first a session is created prior to establishing a connection with the database, Then from line 18 to 31, the days of the week and meals are assigned into variables as it makes it easier to handle. assigning variables makes the data easier to work with, especially when cycling through with a forloop. In lines 18 - 30 the \$_SESSION variables were re-written into variables that are more coherent. 2 arrays are created, one that holds the meals and one which holds days. A forloop is used to cycle through arrays and insert the values of each index into the PAST table indicating that this meal plan was chosen and saved.

Justification

This page comes after the user finalizes the meals they want. The transfer of the meal variable wouldn't be possible without sessions as they allow to store information about the user that is not limited to just one page, Another Way this could have been done is through cookies, however cookies are not scalable as they have a maximum file size. This means that it may not be able to transfer large arrays or variables through different pages. Cookies are also temporarily stored and can expire while sessions only end when the program tells them to overall provide more control over the data.

Feature Four - downloading meals



```
pasta sunday
pizza monday
pasta tuesday
bbq wednesday
fish thursday
pizza friday
salmon saturday
```

Feature Four Outcome


```

69  $myfile = fopen("testfile.txt", "w") or die("Unable to open file!");
70  $txt = $sundaymeal;
71  fwrite($myfile, $txt);
72
73  $txt = $dayone;
74  fwrite($myfile, $txt);
75
76  $txt = $mondaymeal;
77  fwrite($myfile, $txt);
78  $txt = $daytwo;
79  fwrite($myfile, $txt);
80  $txt = $tuesdaymeal;
81  fwrite($myfile, $txt);
82  $txt = $daythree;
83  fwrite($myfile, $txt);
84  $txt = $wednesdaymeal;
85  fwrite($myfile, $txt);
86  $txt = $dayfour;
87  fwrite($myfile, $txt);
88  $txt = $thursdaymeal;
89  fwrite($myfile, $txt);
90  $txt = $dayfive;
91  fwrite($myfile, $txt);
92  $txt = $fridaymeal;
93  fwrite($myfile, $txt);
94  $txt = $daysix;
95  fwrite($myfile, $txt);
96  $txt = $saturdaymeal;
97  fwrite($myfile, $txt);
98  $txt = $dayseven;
99  fwrite($myfile, $txt);
100 fclose($myfile);
101
102 |
103 ?>
104 <a href="testfile.txt" download="menu.txt">Download</a>
105
106 <form action="assing.html" method="post" enctype="multipart/form-data">
107 | | | | | <input type="submit" class="button" value="return">
108 | | | | | </form>

```

Feature Four Code

Explanation

In line 69 a variable is created to open the text file “testfile.txt” in write mode, this will be used and called various times through this program. In line 70 - 99 the program stores a meal of the week and writes it into the file “ testfile.txt” using the fwrite function. The \$txt indicates to the function what to write and the \$myfile tells the function where to write to. After everything is written to the file the file is closed, then in line 104 the user has the option to download the file under the name of “menu.txt”.

Justification : fwrite function allows us to create, and write text to a file which is what we intend to do. However this could also be done with file_put_contents() the problem with file_put_contents() is that it loads the entire content of the file into memory before writing it which can be overwhelming for the primary memory , whereas fwrite loads the data in chunks giving it better memory management. Using file_put_contents() will require for the file to already exist, and to have access to the directory which isnt necessary for fwrite/fopen as it would create the file if it doesn't exist making it easier to use.

Feature Five - Live search

SEARCH FOR A MEAL

Meal Name

lasagna

Feature Five outcome

```

7 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
8 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
9 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12 <h1>SEARCH FOR A MEAL</h1>
13 </div>
14
15 <div class="container">
16 <div class="row">
17 <div class="col-sm-3">
18 </div>
19 <div class="col-sm-6">
20 <input type="text" class="form-control" id="search">
21 <table class="table table-hover">
22 <thead>
23 <tr>
24 <th>Meal Name</th>
25 </tr> </thead>
26 <tbody id="output">
27 </tbody>
28 </table>
29 </div>
30 <div class="col-sm-3">
31 </div>
32 </div>
33 </div>
34 <script type="text/javascript">
35 $(document).ready(function(){
36 $("#search").keypress(function(){
37 $.ajax({
38 type: 'POST',
39 url: 'search.php',
40 data: {
41 meal: $("#search").val(),
42 },
43 success: function(data){
44 $("#output").html(data);
45 }
46 });
47 });
48 });
49 </script>
50

```

Feature Five Code

Explanation:

In line 7 - 9 the jQuery and popper.js libraries are loaded as they are necessary for a live search, from line 15-33 the code divides the page into containers and rows to better structure the output and contents of the page. Line 20 has a input field for search term which goes under the ID search, in line 34 to 36 the script is executed and the code is run when something is typed into the search field which then creates an AJAX POST request and sends it to search.php

```

4  $conn = mysqli_connect("localhost", "hi", "hello", "IB");
5  $sql = "SELECT * FROM meals WHERE meal LIKE '%" . $_POST['meal'] . "%'";
6  $result = mysqli_query($conn, $sql);
7  if(mysqli_num_rows($result)>0){
8      while ($row=mysqli_fetch_assoc($result)) {
9          echo " <tr>
10             <td>". $row['meal'] . "</td>
11             </tr>";
12         }
13     }
14 }
15 else{
16     echo "<tr><td>0 result's found, click <a href=form.html here </a> here to add it</td></tr>";
17 }
18
19 ?>

```

a connection is established and a query with LIKE operator is called upon in line 5. The query with like %"var"% aims to find values from the meal table that have "var" in any position. this helps the auto complete part of the search bar. The ajax post request is sent as "meal" which is why it's referred to as \$_POST[meal]. The results are then outputted as they are stored in an associative array and echoed out with a while loop that goes through the whole array.

Followed tutorial for live search bar : <https://www.youtube.com/watch?v=wCsPAquMNVw>

Import button CSS from <https://codepen.io/giana>



Button has animation when hovered over



