**Task 2 Code Explanation**

**connection.php:**

<?php

$db\_host="db"; //localhost server

$db\_user="root"; //database username

$db\_password="csym019"; //database password

$db\_name="Recipe"; //database name

Try //Exception Handling

{

 $db=new PDO("mysql:host={$db\_host};dbname={$db\_name}",$db\_user,$db\_password);

 $db->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION); //A new PDO object is created and connected to MySQL using server, username and password

And setting it up so that database will show if there are any errors.

}

catch(PDOEXCEPTION $e)

{

 $e->getMessage(); //Getting the exception message

}

?>

**index.php:**

<?php

//Onlyxcodes.com (2022), Login and Register Script In PHP PDO with MySQL. Available at [https://www.onlyxcodes.com/2019/04/login-and-register-script-in-php-pdo.html](https://www.onlyxcodes.com/2019/04/login-and-register-script-in-php-pdo.html%20) (Accessed: 2 May 2022).

require\_once 'connection.php'; //Using the connection file

session\_start(); //Starting the session

if(isset($\_SESSION["user\_login"])) //check if session user\_login is set or not

{

 header("location: home.php"); //If session user\_login then redirect to home.php

}

if(isset($\_POST['login'])) //If login from the form is set and is not null

{

 $username =strip\_tags($\_POST["username"]); //Strips the string in username field of form from HTML,XML,PHP tags

 $email  =strip\_tags($\_POST["username"]); //Strips the string in username field of form from HTML,XML,PHP tags

 $password =strip\_tags($\_POST["password"]);   //Strips the string in password field of form from HTML,XML,PHP tags

 if(empty($username)){      //If username is empty

  $errorMsg[]="please enter username or email"; //Add the error message into errorMsg array

 }

 else if(empty($email)){ //If email is empty

  $errorMsg[]="please enter username or email"; //Add the error message into errorMsg array

 }

 else if(empty($password)){ //If password is empty

  $errorMsg[]="please enter password"; // Add the error message into errorMsg array

 }

 else

 {

  try //Exception Handling

  {

   $query=$db->prepare("SELECT \* FROM users WHERE username=? OR email=?"); //sql select query

   $query->execute(array($username,$email)); //execute query with bind parameter

   $row=$query->fetch(PDO::FETCH\_ASSOC); //Fetching the result as an associative array from database and storing in $row variable

   if($query->rowCount() > 0) //If Database record greater than 0

   {

    if($username==$row["username"] OR $email==$row["email"]) //check condition username or email matched from the database.

    {

     if($password== $row["password"]) //Check condition password matched from database password.

     {

      $\_SESSION["user\_login"] = $row["user\_id"]; //start the session with name "user\_login"

      $loginMsg = "Successfully Login...";  //Set user login success message

      header("refresh:2; home.php");  //Redirect to home.php

     }

     else

     {

      $errorMsg[]="wrong password"; //add the error message if passwords not matched

     }

    }

    else

    {

     $errorMsg[]="wrong username or email"; //add the error message if username or email not matched

    }

   }

   else

   {

    $errorMsg[]="wrong username or email"; //add the error message if username or email not matched

   }

  }

  catch(PDOException $e)

  {

   $e->getMessage(); //Handle any exceptions thrown and display the exception message

  }

 }

}

?>

<!DOCTYPE html> //HTML starts

<html>

  <head>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous"> //Using Bootstrap css

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script> //Using Jquery

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script>

//Using Bootstrap

    <link rel="stylesheet" href="layout.css"> //External stylesheet for the page

  </head>

  <body>

    <br> //Break

    <div class="login"> //Div element with class login

//GetBootstrap.com (2015), Getting started. Available at <https://getbootstrap.com/>(Accessed: 5 May 2022).

<form method="post" class="form-horizontal"> //Login Form with method Post

 <div class="form-group"> //Grouping label and input field

 <label class="col-sm-3 control-label">Username or Email</label> //Label for username or email

 <div class="col-sm-6"> //In small devices take width of 6 columns

 <input type="text" name="username" class="form-control" placeholder="Enter username or email" /> //Input field

 </div>

 </div>

 <div class="form-group"> //Grouping label and input field

 <label class="col-sm-3 control-label">Password</label> //Label for Password

 <div class="col-sm-6">

 <input type="password" name="password" class="form-control" placeholder="Enter passowrd" /> //input field

 </div>

 </div>

 <div class="form-group">

 <div class="col-sm-offset-3 col-sm-9 m-t-15"> //In sm devices set offset of 3 cols and take width of 9 cols

 <input type="submit" name="login" class="btn btn-success" value="Login"> //Login Button

 </div>

 </div>

 <div class="form-group">

 <div class="col-sm-offset-3 col-sm-9 m-t-15">

 You don't have a account register here? <a href="register.php"><p class="text-info">Register Account</p></a>   //Register Link

 </div>

 </div>

</form>

<?php

if(isset($errorMsg)) //If errorMsg is set or not null

{

//The below foreach is used to loop through the errorMsg array

 foreach($errorMsg as $error)

 {

 ?>

  <div class="alert alert-danger">

   <strong><?php echo $error; ?></strong> //Displaying the errors in red colour

  </div>

    <?php

 }

}

if(isset($loginMsg)) //If loginMsg is set or not null

{

?>

 <div class="alert alert-success">

  <strong><?php echo $loginMsg; ?></strong> //Display the login message in green color

 </div>

<?php

} //End of loop

?>

</div>

</body> //End of body

</html> //End of html

**register.php:**

<?php

require\_once "connection.php"; //Use connection.php

if(isset($\_POST['btn\_register'])) //If btn\_register in form is set or not null

{

 $username = strip\_tags($\_POST['username']); //Strips the string in username field of form from HTML,XML,PHP tags

 $email  = strip\_tags($\_POST['email']); //Strips the string in email field of form from HTML,XML,PHP tags

 $password = strip\_tags($\_POST['password']); //Strips the string in password field of form from HTML,XML,PHP tags

 if(empty($username)){ //If username is empty

  $errorMsg[]="Please enter username"; //Add the error message to errorMsg array

 }

 else if(empty($email)){ //If email is empty

  $errorMsg[]="Please enter email"; //Add the error message to errorMsg array

 }

 else if(empty($password)){ //If Password is empty

  $errorMsg[]="Please enter password"; //Add the error message to errorMsg array }

 else if(strlen($password) < 6){ //If password is less than 6 characters

  $errorMsg[] = "Password must be atleast 6 characters"; //Add the error message to errorMsg array }

 else

 {

  Try //Exception Handling

  {

   $query=$db->prepare("SELECT username, email FROM users

          WHERE username=? OR email=?"); // sql select query

   $query->execute(array($username, $email)); //execute query

   $row=$query->fetch(PDO::FETCH\_ASSOC); //Fetching the result as an associative array from database and storing in $row variable

   if($row["username"]==$username){ //Check if username already exists

    $errorMsg[]="Sorry username already exists"; //Add the error message to errorMsg array

   }

   else if($row["email"]==$email){ //Check if email already exists

    $errorMsg[]="Sorry email already exists"; //Add the error message to errorMsg array

   }

   else if(!isset($errorMsg)) //check if there’s no errorMsg

   {

    $insert\_stmt=$db->prepare("INSERT INTO users (username,email,password) VALUES

                (:uname,:uemail,:upassword)");   //sql insert query

    if($insert\_stmt->execute(array( ':uname' =>$username,

                                    ':uemail'=>$email,

                                    ':upassword'=>$password))){

     $registerMsg="Register Successfully..... Please Click On Login Account Link"; //Set execute query success message to registerMsg

    }

   }

  }

  catch(PDOException $e) //Catch any exceptions

  {

   echo $e->getMessage(); //Display the exception messsages

  }

 }

}

?>

<!DOCTYPE html> //HTML starts

<html>

  <head>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous"> //Using Bootstrap.css

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script> //Using JQuery

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script>

//Using Bootstrap

    <link rel="stylesheet" href="layout.css"> //Using external stylesheet for page

  </head>

  <body> //Start of Body

    <br> //Break

    <div class="login"> //Division Element with class login

<form method="post" class="form-horizontal"> //Registration Form with method Post

 <div class="form-group"> //grouping label and input field

 <label class="col-sm-3 control-label">Username</label> //Label for username

 <div class="col-sm-6">

 <input type="text" name="username" class="form-control" placeholder="Enter username" /> //Input field for username

 </div>

 </div>

 <div class="form-group"> //grouping label and input field

 <label class="col-sm-3 control-label">Email</label> //Label for email

 <div class="col-sm-6">

 <input type="text" name="email" class="form-control" placeholder="Enter email" /> //Input field for email

 </div>

 </div>

 <div class="form-group"> //grouping label and input field

 <label class="col-sm-3 control-label">Password</label> //Label for password

 <div class="col-sm-6">

 <input type="password" name="password" class="form-control" placeholder="Enter passowrd" /> //input field for password

 </div>

 </div>

 <div class="form-group">

 <div class="col-sm-offset-3 col-sm-9 m-t-15">

 <input type="submit"  name="btn\_register" class="btn btn-primary " value="Register"> //Button Register with classes btn btn-primary

 </div>

 </div>

 <div class="form-group">

 <div class="col-sm-offset-3 col-sm-9 m-t-15">

 You have a account register here? <a href="index.php"><p class="text-info">Login Account</p></a>  //Link to login account

 </div>

 </div>

</form> //Form ends

<?php

if(isset($errorMsg)) //If errorMsg is not null and is set

{

//The below foreach is used to loop through the errorMsg array

 foreach($errorMsg as $error)

 {

 ?>

  <div class="alert alert-danger">

   <strong><?php echo $error; ?></strong> //Displaying each error message in red color

  </div>

    <?php

 }

}

if(isset($registerMsg)) //If registerMsg is set or not null

{

?>

 <div class="alert alert-success">

  <strong><?php echo $registerMsg; ?></strong> //Display the error message in green color

 </div>

<?php

} //End of for loop

?>

</div>

</body> //End of body

</html> //End of HTML

**Home.php:**

<?php

 require\_once 'connection.php'; //Use connection.php

 session\_start(); //Start the session

 if(!isset($\_SESSION['user\_login'])) //check unauthorize user not access in "home.php" page

 {

  header("location: index.php"); //If session is not user\_login then redirect to index.php

 }

 $id = $\_SESSION['user\_login']; //Getting the id

 $select\_stmt = $db->prepare("SELECT \* FROM users WHERE user\_id=:uid"); //SQL select query

 $select\_stmt->execute(array(":uid"=>$id)); //Executing the query and fetching the data where user\_id = session id

 $row=$select\_stmt->fetch(PDO::FETCH\_ASSOC); //Fetching the result as an associative array from database and storing in $row variable

 if(isset($\_SESSION['user\_login'])) //If user\_login is set and not null

 {

 ?>

<!DOCTYPE html> //HTML starts

<html>

  <head>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous"> //Using Bootstrap.css

    <link rel="stylesheet" href="layout.css"> //Using External Stylesheet

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script> //using JQuery

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script>

//Using Bootstrap Js

  </head>

  <body>

  <header> //Start of header section

            <a class="home" href="home.php"><h3>CSYM019 - BBC GOOD FOOD RECIPES</h3></a>

            Welcome,

<?php

   echo $row['username']; //Prints welcome message

 }

?>

<?php

if(isset($errorMsg)) //If errorMsg is set or not null

{

 ?>

  <div class="alert alert-danger">

   <strong><?php echo $errorMsg; ?></strong> //Display error messages in red color

  </div>

    <?php

}

?>

            <a class="logout" href="logout.php">Logout</a> //Link for logout of session

        </header> //End of header section

 <nav> //Start of nav section

            <ul> //Unordered List

                <li><a href="./recipeSelectionForm.php">Recipe Report</a></li> //Link to Recipe Selection page

                <li><a href="./newRecipe.php">New Recipe</a></li> //Link to New Recipe page

            </ul>

        </nav> //End of navigation section

        <main> //Start of main section

<?php

        try //Exception Handling

  {

   $query=$db->prepare("SELECT \* FROM Recipes"); // sql select query

   $query->execute(); //execute query

   $results=$query->fetchAll(); //Fetch All the rows from the database and store in results variable

   if(empty($results)){ //If results is empty

    $errorMsg="Sorry No recipes Exists"; //Set the error message

   }

   else //check no "$errorMsg" show then continue

   {

     ?>

    <h1 class="titles">Recipes</h1> //H1 element with class “titles”

    <table> //Table start

    <thead class='tbl-header'> //Table Headings

      <tr>

        <th>No.</th> //No. heading

        <th>Name</th> //Name

        <th>Author</th> //Author

        <th>More Details</th> //More Details

        <th>Delete</th> //Delete

      </tr>

    </thead> //End of table headings

    <tbody> //Table Body

    <?php

    $c=1; //Set the counter to 1

//The below foreach loops through the results fetched from database

    foreach($results as $row){

      ?>

      <tr> //Table row

<?php

      echo '<td>' . $c++ . '</td>'; //Display and increment counter

      echo '<td>' . $row['name'] . '</td>'; //Display the name of recipe

      echo '<td>' . $row['author'] . '</td>'; //Display Author of Recipe

//Displays Prep Time, Cooking Time, Ratings, Description, Ingredients, Nutrition and Method

      echo '<td>Prep Time: ' . $row['prep'] . '<br>Cook Time: ' . $row['cook'] . '<br>Serves: ' . $row['serves']

      . '<br>Ratings: ' . $row['ratings'] . '<br>Description: ' . $row['description'] . '<br><br>Ingredients: ' .

      $row['ingredients'] . '<br><br>Kcal: ' . $row['kcal'] . '&emsp;Fat: ' . $row['fat'] . 'g&emsp;Saturates: ' . $row['saturates'] .

      'g&emsp;Carbs: ' . $row['carbs'] . 'g&emsp;Sugars: ' . $row['sugars'] . 'g&emsp;Fibre: ' . $row['fibre'] . 'g&emsp;Protein: '

      . $row['protein'] . 'g&emsp;Salts: ' . $row['salt'] . 'g<br><br>Method: ' . $row['method'] . '<br></td>';?>

      <td><a href="delete.php?id=<?php echo $row['id']?>" class="btn btn-danger btn-sm">Delete</a> </td> //Display Delete button and sending the id of recipe to delete.php

      </tr> //End of table row

<?php

    }

    ?>

    </tbody> //End of table body

    </table> //End of table

<?php

    }

   }

  catch(PDOException $e) //Catch any exceptions

  {

   echo $e->getMessage(); //Display the exception messages

  }

?>

</main> //End of main section

        <footer>&copy; CSYM019 2022</footer> //footer section

</div>

</body> //End of body

</html> //End of HTML

**delete.php:**

<?php

require\_once 'connection.php'; //Use connection.php

session\_start(); //start session

 if(!isset($\_SESSION['user\_login'])) //check unauthorize user not access in "home.php" page

 {

  header("location: index.php"); //Redirect if session is not set

 }

if (isset($\_GET['id'])){ //If Recipe ID is set or not null

    $ID = (int)$\_GET['id']; //Get the recipe id, convert to int and store in $ID

    $delete= $db->prepare("DELETE FROM Recipes WHERE id=$ID LIMIT 1"); //SQL query

    $delete->execute(); //Execute the query

    if ($delete){ //If deleted

        echo '<script>alert("Recipe deleted successfully !")</script>'; //Alert that Recipe is deleted

        echo '<script>window.location.href="home.php";</script>'; //Redirect to home.php

    }

}

?>

**layout.css:**

.login{ //Styles for elements with class “login”

  width: 650px; //sets width to 650px

    border: 1px solid #ccc; //sets border to 1px solid of color #ccc

    padding: 20px; //sets padding all sides to 20px

    border-radius: 15px; //Sets border-radius to 15 px

    margin-top: 200px; //Sets margin-top to 200px;

    box-shadow: 0 0 18px; //Sets box shadow to the element with blur-radius of 18px

margin-left: 200px; //sets margin-left to 200px}

.chartStyles{ //Styles for element with class “chartStyles”

  height: 250px; //sets height of element to 250px

  width: 250px; //sets width of element to 250px

text-align: center;margin-left: 37%; //Aligns the text to center and has a margin of 37% on left side}

.titles{ //Styles for element with class “titles”

  text-align: center; //aligns the text to center

}

.logout{ //Styles for element with class “logout”

  float:right; //Element floats at right end

  color:red; //sets the color to red

}

.home{//Styles for element with class “home”

  color:white; //Sets the color to white

}

.create{ //Styles for element with class “create”

  margin-left:45px; //Sets the margin of 45px on the left side

}

table{ //Styles for table element

    width: 90%; //Sets width to 90%

  margin: 50px; //Sets margin of 50px on all sides

 border-spacing: 0px; //has no border-spacing

  background-color:#372d2d; //background for table

}

  .tbl-content{ //styles for elements with class “tbl-content”

    overflow-x:auto; //Adds scrollbar when necessary

    margin-top: 0px; //Has no margin at the top

    border: 1px solid rgba(255,255,255,0.3); //Sets the border of 1px for element with color white with opacity 0.3

  }

  th{ //styles for th element

    padding: 20px 15px; //sets a padding of 20px at top and bottom and 15px padding at left and right

    text-align: center; //aligns the text to center

    font-weight: 500; //sets the weight of text to 500

    font-size: 12px; //Sets the font size to 12px

    color: #fff; //sets color to white

    text-transform: uppercase //make the text UPPERCASE

  }

  td{ //styles for td element

    padding: 15px; //sets the padding of 15px all sides

    text-align: left; //Aligns the text to left side

    vertical-align:middle; //vertically align the align in the middle

    font-weight: 300; //sets the weight of text to 300

    font-size: 12px; //sets the font size to 12px

    color: #fff;//sets the color to white

    border: solid 1px rgba(255,255,255,0.1); //Border of 1px solid color with opacity 0.1

  }

  body { //styles for body element

    display: grid; //Display to grid

    grid-template-columns: 15% 85%; //Line names and track sizing functions of grid columns

    grid-template-rows: auto; // Line names and track sizing functions of grid rows

    grid-template-areas: "header header"

"nav main" "footer footer"; //Cells in the grid and assigning the names

}

h4{ //styles of h4 element

  text-align: center; //aligns the text to center

}

//The below css sets padding of 5px on all sides to following elements

header, nav, section, aside, footer {

  padding: 5px;

}

header { //styles of header element

    background-color: #555; //sets the background-color to #555

    color: white; //sets text color to white

    padding: 10px; //sets padding of 10px to all sides

    grid-area: header; //sets area name

height: fit-content; //sets the height to fit the content}

nav { //styles of nav element

    background-color: whitesmoke; //sets the background color to whitesmoke

    color: black;//sets text color to black

    grid-area: nav; //sets area name

}

nav > ul { //styles of ul element inside nav

    list-style-type: none; //No list style

    margin: 0; //no margin

    padding: 0; //no padding

    border-right: 2%; //add 2% border to right side

    border-color: whitesmoke; //sets the border color to whitesmoke

    border-style: solid; //adds a solid border

    background-color: whitesmoke; //background color to whitesmoke

  }

nav > ul li a { //styles of a element inside nav with ul and li tag

    display: block; //Display of element to block

    color: #000; //color to black

    padding: 8px 16px; //padding of 8px top, bottom and 16px left, right

    text-decoration: none; //No text-decorations

}

/\* Change the link color on hover \*/

nav > ul li a:hover {

    background-color: #555; //sets background color to #555

    color: whitesmoke; //color to whitesmoke

}

main { //styles for main element

    background-color: whitesmoke; //sets the background color to whitesmoke

    color: black; //color of text to black

    margin-left: 1em; //sets the margin of 1em on left side

    padding-left: 1em; //sets the padding of 1em on left side

    grid-area: main; //shorthand property for grid-item’s properties

}

footer { //styles of footer element

  padding-top: 2em; //adds 2em padding at top

  background-color: #555; //sets background color to #555

  color: white; //sets color to white

  grid-area: footer; //sets area name

}

**Logout.php:**

<?php

session\_start(); //start session

header("location:index.php"); //redirect to index.php

session\_destroy(); //destroy the current session

?>

**newRecipe.php:**

<?php

 require\_once 'connection.php'; //use connection.php

 session\_start(); //start the session

 if(!isset($\_SESSION['user\_login'])) //check unauthorize user not access in "newRecipe.php" page

 {

  header("location: index.php"); //redirect to index.php

 }

 $id = $\_SESSION['user\_login']; //get the user id and store in $id

 $select\_stmt = $db->prepare("SELECT \* FROM users WHERE user\_id=:uid"); //SQL query to get details of user that’s logged in

 $select\_stmt->execute(array(":uid"=>$id)); //execute the query

 $row=$select\_stmt->fetch(PDO::FETCH\_ASSOC); //Fetching the result as an associative array from database and storing in $row variable

 if(isset($\_SESSION['user\_login'])) //if user\_login is set and not null

 {

 ?>

<!DOCTYPE html> //HTML starts

<html>

  <head>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous"> //Using Bootstrap.css

    <link rel="stylesheet" href="layout.css"> //Using external stylesheet for page

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script> //Using JQuery

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script> //Using Boostrap.js

  </head>

  <body> //Body starts

  <header>

            <a class="home" href="home.php"><h3>CSYM019 - BBC GOOD FOOD RECIPES</h3></a>

            Welcome,

<?php

   echo $row['username']; //Display welcome message

 }

 if(isset($\_POST['add'])) //button name "add" is set

{

 $name = strip\_tags($\_POST['name']); //Strips the string in name field of form from HTML,XML,PHP tags

 $author  = strip\_tags($\_POST['author']); //Strips the string in author field of form from HTML,XML,PHP tags

 $prep = strip\_tags($\_POST['prep']); //Strips the string in prep field of form from HTML,XML,PHP tags

 $cook = strip\_tags($\_POST['cook']); //Strips the string in cook field of form from HTML,XML,PHP tags

 $serves  = strip\_tags($\_POST['serves']);  //Strips the string in serves field of form from HTML,XML,PHP tags

 $ratings = strip\_tags($\_POST['ratings']); //Strips the string in ratings field of form from HTML,XML,PHP tags

 $foodType = strip\_tags($\_POST['foodType']); //Strips the string in foodType field of form from HTML,XML,PHP tags

 $difficulty  = strip\_tags($\_POST['difficulty']);  //Strips the string in difficulty field of form from HTML,XML,PHP tags

 $desc = strip\_tags($\_POST['desc']); //Strips the string in desc field of form from HTML,XML,PHP tags

 $ingredients = strip\_tags($\_POST['ingredients']); //Strips the string in ingredients field of form from HTML,XML,PHP tags

 $kcal  = strip\_tags($\_POST['kcal']);  //Strips the string in kcal field of form from HTML,XML,PHP tags

 $fat = strip\_tags($\_POST['fat']); //Strips the string in fat field of form from HTML,XML,PHP tags

 $saturates = strip\_tags($\_POST['saturates']); //Strips the string in saturates field of form from HTML,XML,PHP tags

 $carbs = strip\_tags($\_POST['carbs']); //Strips the string in carbs field of form from HTML,XML,PHP tags

 $sugars  = strip\_tags($\_POST['sugars']);  //Strips the string in sugars field of form from HTML,XML,PHP tags

 $fibre = strip\_tags($\_POST['fibre']); //Strips the string in fibre field of form from HTML,XML,PHP tags

 $protein = strip\_tags($\_POST['protein']); //Strips the string in protein field of form from HTML,XML,PHP tags

 $salt  = strip\_tags($\_POST['salt']);  //Strips the string in salt field of form from HTML,XML,PHP tags

 $method = strip\_tags($\_POST['method']); //Strips the string in method field of form from HTML,XML,PHP tags

//The below if code checks if any of the fields are empty and sets error message

 if(empty($name) OR empty($author) OR empty($prep) OR empty($cook) OR empty($serves) OR empty($ratings) OR

 empty($foodType) OR empty($difficulty) OR empty($desc) OR empty($ingredients) OR empty($kcal) OR

 empty($fat) OR empty($saturates) OR empty($carbs) OR empty($sugars) OR empty($fibre) OR empty($protein)

 OR empty($salt) OR empty($method) ){

  $errorMsg="Please enter All Fields";

 }

 else //No missing inputs

 {

  Try //Exception Handling

  {

   $query=$db->prepare("SELECT name FROM Recipes

          WHERE name=?"); // sql select query

   $query->execute(array($name)); //execute query

   $row=$query->fetch(PDO::FETCH\_ASSOC); //Fetching the row from database

//check recipe name already exists and sets error message

   if($row["name"]==$name){

    $errorMsg="Sorry Recipe Name already exists";

   }

   else if(!isset($errorMsg)) //check no "$errorMsg" show then continue

   {

    $insert\_stmt=$db->prepare("INSERT INTO Recipes (name,author,prep,cook,serves,ratings,foodType,difficulty,description,ingredients,kcal,fat,saturates,carbs,sugars,fibre,protein,salt,method) VALUES

                (:rname,:author,:prep,:cook,:serves,:ratings,:foodType,:difficulty,:desc,:ingredients,:kcal,:fat,:saturates,:carbs,:sugars,:fibre,:protein,:salt,:method)");   //sql insert query

    if($insert\_stmt->execute(array( ':rname' =>$name,

                                    ':author'=>$author,

                                    ':prep'=>$prep,

                                    ':cook' =>$cook,

                                    ':serves'=>$serves,

                                    ':ratings'=>$ratings,

                                    ':foodType' =>$foodType,

                                    ':difficulty'=>$difficulty,

                                    ':desc'=>$desc,

                                    ':ingredients' =>$ingredients,

                                    ':kcal'=>$kcal,

                                    ':fat'=>$fat,

                                    ':saturates' =>$saturates,

                                    ':carbs'=>$carbs,

                                    ':sugars'=>$sugars,

                                    ':fibre' =>$fibre,

                                    ':protein'=>$protein,

                                    ':salt'=>$salt,

                                    ':method' =>$method))){

     $addmMsg="Added Successfully....."; //sets the execute query success message

     ?>

     <script>setTimeout(function(){ window.location.href = "home.php"; //Redirecting to home.php after successful insertion},2000);</script>

<?php

    }

   }

  }

  catch(PDOException $e) //Catching the exceptions

  {

   echo $e->getMessage(); //Displaying the exception message

  }

 }

}

?>

<?php

if(isset($errorMsg)) //If any Error message

{

 ?>

  <div class="alert alert-danger">

   <strong><?php echo $errorMsg; ?></strong> //Display the error message in red color

  </div>

    <?php

}

//If addMsg is set

if(isset($addmMsg))

{

?>

 <div class="alert alert-success">

  <strong><?php echo $addmMsg; ?></strong> //Display the successful insertion message in green color

 </div>

<?php

}

?>

            <a class="logout" href="logout.php">Logout</a> //Link for logout

        </header> //End of header

 <nav> //Start of nav

            <ul>

//Link for Recipe Selection

                <li><a href="./recipeSelectionForm.php">Recipe Report</a></li>

//Link for New Recipe

                <li><a href="./newRecipe.php">New Recipe</a></li>

            </ul>

        </nav> //End of nav

        <main> //Start of main

            <br> //Break

        <h4>Add Recipe Details</h4> //h4 element

    <form method="post" class="form-horizontal"> //New recipe form with method post

      <div class="form-group"> //Grouping label and input field

        <label class="col-sm-3 control-label">Name</label> //Label for Name

        <div class="col-sm-6">

          <input type="text" name="name" class="form-control" placeholder="Enter Recipe Name" /> //Input field

        </div>

      </div>

//Label and input field for Author of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Author</label>

        <div class="col-sm-6">

          <input type="text" name="author" class="form-control" placeholder="Enter Author" />

        </div>

      </div>

//Label and input field for Preparation Time of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Prep Time</label>

        <div class="col-sm-6">

          <input type="text" name="prep" class="form-control" placeholder="Enter Preparation Time" />

        </div>

      </div>

//Label and input field for Cooking time of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Cook Time</label>

        <div class="col-sm-6">

          <input type="text" name="cook" class="form-control" placeholder="Enter Cooking Time" />

        </div>

      </div>

//Label and input field for no. of servings by recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Serves</label>

        <div class="col-sm-6">

          <input type="number" name="serves" min="1" class="form-control" placeholder="Enter Serves " />

        </div>

      </div>

//Label and input field for ratings of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Ratings</label>

        <div class="col-sm-6">

          <input type="number" name="ratings" min="0" class="form-control" placeholder="Enter Ratings" />

        </div>

      </div>

//Label and input field for Food type of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Food Type</label>

        <div class="col-sm-6">

          <input type="text" name="foodType" class="form-control" placeholder="Enter Food Type" />

        </div>

      </div>

//Label and input field for Difficulty of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Difficulty</label>

        <div class="col-sm-6">

          <input type="text" name="difficulty" class="form-control" placeholder="Enter Difficulty" />

        </div>

      </div>

//Label and input field for description of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Description</label>

        <div class="col-sm-6">

          <input type="text" name="desc" class="form-control" placeholder="Enter Description" />

        </div>

      </div>

//Label and textarea field for Ingredients of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Ingredients</label>

        <div class="col-sm-6">

        <textarea name="ingredients" rows="10" class="form-control" placeholder="Enter Ingredients"></textarea>

        </div>

      </div>

//Label and input field for kcal of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Kcal</label>

        <div class="col-sm-6">

          <input type="number" min="0" name="kcal" class="form-control" placeholder="Enter Kcal" />

        </div>

      </div>

//Label and input field for fat of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Fat</label>

        <div class="col-sm-6">

          <input type="number" step=0.01 name="fat" class="form-control" placeholder="Enter fat" />

        </div>

      </div>

//Label and input field for saturates of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Saturates</label>

        <div class="col-sm-6">

          <input type="number" step=0.01 name="saturates" class="form-control" placeholder="Enter Saturates" />

        </div>

      </div>

//Label and input field for carbs of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Carbs</label>

        <div class="col-sm-6">

          <input type="number" step=0.01 name="carbs" class="form-control" placeholder="Enter Carbs" />

        </div>

      </div>

//Label and input field for sugars of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Sugars</label>

        <div class="col-sm-6">

          <input type="number" step=0.01 name="sugars" class="form-control" placeholder="Enter Sugars" />

        </div>

      </div>

//Label and input field for fibre of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Fibre</label>

        <div class="col-sm-6">

          <input type="number" step=0.01 name="fibre" class="form-control" placeholder="Enter Fibre" />

        </div>

      </div>

//Label and input field for protein of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Protein</label>

        <div class="col-sm-6">

          <input type="number" step=0.01 name="protein" class="form-control" placeholder="Enter Protein" />

        </div>

      </div>

//Label and input field for salts of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Salt</label>

        <div class="col-sm-6">

          <input type="number" step=0.01 name="salt" class="form-control" placeholder="Enter Salt" />

        </div>

      </div>

//Label and textarea field for method of recipe

      <div class="form-group">

        <label class="col-sm-3 control-label">Method</label>

        <div class="col-sm-6">

        <textarea name="method" rows="10" class="form-control" placeholder="Enter Method"></textarea>

        </div>

      </div>

//Add button

      <div class="form-group">

        <div class="col-sm-offset-3 col-sm-9 m-t-15">

          <input type="submit"  name="add" class="btn btn-primary " value="Add">

        </div>

      </div>

</form> //End of form

</main> //End of main

        <footer>&copy; CSYM019 2022</footer> //Footer for the page

</body> //End of body

</html> //End of HTML

**recipeSelectionForm.php:**

<?php

 require\_once 'connection.php'; //Use connection.pho

 session\_start(); //start session

 if(!isset($\_SESSION['user\_login'])) //check unauthorize user not access in "recipeSelectionForm.php" page

 {

  header("location: index.php"); //redirect if any

 }

 $id = $\_SESSION['user\_login']; //Get the id of logged in user

 $select\_stmt = $db->prepare("SELECT \* FROM users WHERE user\_id=:uid"); //select query

 $select\_stmt->execute(array(":uid"=>$id)); //execute the query

 $row=$select\_stmt->fetch(PDO::FETCH\_ASSOC); //Fetch the row from database and store in $row variable

 if(isset($\_SESSION['user\_login'])) //If user\_login is set

 {

 ?>

<!DOCTYPE html> //HTML starts

<html>

  <head>

//Using Bootsrap.css

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous"> //Using External stylesheet for the page

    <link rel="stylesheet" href="layout.css">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script> //Using JQuery

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script> //Using bootstrap.js

  </head>

  <body> //Body starts

  <header> //Header starts

            <a class="home" href="home.php"><h3>CSYM019 - BBC GOOD FOOD RECIPES</h3></a>

            Welcome,

<?php

   echo $row['username']; //Display Welcome Message

 }

?>

<?php

if(isset($errorMsg)) //If errorMsg is set or not null

{

 ?>

  <div class="alert alert-danger">

   <strong><?php echo $errorMsg; ?></strong> //Display error message in red color

  </div>

    <?php

}

?>

            <a class="logout" href="logout.php">Logout</a> //Link to logout

        </header> //End of header

 <nav> //Nav starts

            <ul> //Unordered List

//Link to Recipe report

                <li><a href="./recipeSelectionForm.php">Recipe Report</a></li>

//Link to new recipe page

                <li><a href="./newRecipe.php">New Recipe</a></li>

            </ul>

        </nav> //End of nav

        <main> //Main starts

<?php

        Try //Exception Handling

  {

   $query=$db->prepare("SELECT \* FROM Recipes ORDER BY name"); // sql select query

   $query->execute(); //execute query

   $results=$query->fetchAll(); //Fetch all the rows from database and store in results

//If results are empty set the error message

   if(empty($results)){

    $errorMsg="Sorry No recipes Exists";

   }

   else //check no "$errorMsg" show then continue

   {

     ?>

    <script language="JavaScript">

//The below function is used to toggle the checkboxes values and disable submit button if no checkbox is checked

function toggle(source) {

    checkboxes = document.getElementsByName('recipeId[]');

  for(var i=0, n=checkboxes.length;i<n;i++) {

    checkboxes[i].checked = source.checked;

  }

  var checkboxes = $("input[type='checkbox']"),

  submitButt = $("button[type='submit']");

// checkboxes.click(function() {

    submitButt.attr("disabled", !checkboxes.is(":checked"));

}

//The below function is used to disable submit button if no checkbox is checked

function myFunc(){

    console.log("Inside");

var checkboxes = $("input[type='checkbox']"),

    submitButt = $("button[type='submit']");

    submitButt.attr("disabled", !checkboxes.is(":checked"));

}

</script>

<h1 class="titles">Recipes Selection</h1> //H1 element with class “titles”

<form action="recipeReport.php" method="POST"> //Form with method post

    <table> //Table to display recipes

    <thead class='tbl-header'> //table headings

      <tr>

//checkbox to check all checkboxes

          <th><input type="checkbox" onClick="toggle(this)" /></th>

        <th>Name</th>

        <th>Author</th>

        <th>More Details</th>

        <th>Delete</th>

      </tr>

    </thead>

    <tbody> //table body

    <?php

//Below foreach loops through the results

    foreach($results as $row){

      ?>

      <tr>

<?php

//Checkbox for each recipe with value as id of recipe

        echo '<td><input type="checkbox" onchange="myFunc()" name="recipeId[]" value="'. $row['id'] .'"></td>';

        echo '<td>' . $row['name'] . '</td>'; //Name of the recipe

        echo '<td>' . $row['author'] . '</td>'; //Author of the recipe

//Displays Prep Time, Cook Time, serves, Ratings ,Ingredients, Nutrition information and method of recipe

        echo '<td>Prep Time: ' . $row['prep'] . '<br>Cook Time: ' . $row['cook'] . '<br>Serves: ' . $row['serves']

        . '<br>Ratings: ' . $row['ratings'] . '<br>Description: ' . $row['description'] . '<br><br>Ingredients: ' .

        $row['ingredients'] . '<br><br>Kcal: ' . $row['kcal'] . '&emsp;Fat: ' . $row['fat'] . 'g&emsp;Saturates: ' . $row['saturates'] .

        'g&emsp;Carbs: ' . $row['carbs'] . 'g&emsp;Sugars: ' . $row['sugars'] . 'g&emsp;Fibre: ' . $row['fibre'] . 'g&emsp;Protein: '

        . $row['protein'] . 'g&emsp;Salts: ' . $row['salt'] . 'g<br><br>Method: ' . $row['method'] . '<br></td>';?>

        <!-- echo '<td><i class="material-icons" onclick="">delete</i></td>'; -->

        <td><a href="delete.php?id=<?php echo $row['id']?>" class="btn btn-danger btn-sm">Delete</a> </td> //Display Delete button and sending the id of recipe to delete.php

      </tr> //End of row

<?php

    }

    ?>

    </tbody> //end of table body

    </table> //end of table

    <button class="btn btn-primary create" type="submit"  value="submit" disabled>Create Recipe Report</button> //Create recipe report button

</form> //end of form

<br>

<?php

    }

   }

  catch(PDOException $e) //catch exceptions

  {

   echo $e->getMessage(); //display any exception messages

  }

?>

</main> //End of main

        <footer>&copy; CSYM019 2022</footer> //footer for page

</div>

</body> //End of body

</html> //End of HTML

**recipeReport.php:**

<?php

    require\_once 'connection.php'; //Use connection.php

    session\_start(); //start session

    if(!isset($\_SESSION['user\_login'])) //check unauthorize user not access in "recipeReport.php" page

    {

     header("location: index.php"); //redirect if any

    }

    $id = $\_SESSION['user\_login']; //get the id of logged in user

    $select\_stmt = $db->prepare("SELECT \* FROM users WHERE user\_id=:uid"); //SQL query

    $select\_stmt->execute(array(":uid"=>$id)); //Execute query

    $row=$select\_stmt->fetch(PDO::FETCH\_ASSOC); //Fetch the row from database

    if(isset($\_SESSION['user\_login'])) //If user\_login is set or not null

    {

    ?>

   <!DOCTYPE html> //HTML starts

   <html>

     <head>

//Using Bootstarp.css

       <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous"> //Using external stylesheet for the page

       <link rel="stylesheet" href="layout.css">

       <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script> //Using JQuery

       <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script> //Using Bootstrap.css

       <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.js"></script>

//Using Chart.js

     </head>

     <body> //Start of HTML body

     <header>

               <a class="home" href="home.php"><h3>CSYM019 - BBC GOOD FOOD RECIPES</h3></a>

               Welcome,

   <?php

      echo $row['username']; //Display the welcome message

    }

   ?>

   <?php

   if(isset($errorMsg)) //If errorMsg is set or not null

   {

    ?>

     <div class="alert alert-danger">

      <strong><?php echo $errorMsg; ?></strong> //Display error message in red color

     </div>

       <?php

   }

   ?>

               <a class="logout" href="logout.php">Logout</a> //Link to logout

           </header> //End of header

    <nav> //Start of nav

               <ul>

//Link to Recipe Selection page

                   <li><a href="./recipeSelectionForm.php">Recipe Report</a></li>

//link to New recipe page

                   <li><a href="./newRecipe.php">New Recipe</a></li>

               </ul>

           </nav> //End of nav

           <main> //Start of main

           <?php

$recipeID = $\_POST['recipeId']; //Getting the recipe ids that are checked

?>

   <?php

           try

     {

//implode returns string from the elements of array

      $query=$db->prepare('SELECT \* FROM Recipes WHERE id IN (' . implode(",", $recipeID) . ')');// sql select query where recipe id are in recipeId array

      $query->execute(); //execute query

      $results=$query->fetchAll(); //Fetch all the rows from database

      if(empty($results)){ //If results are empty set error message

       $errorMsg="Sorry No recipes Exists";

      }

      else //check no "$errorMsg" show then continue

      {

        ?>

        <script>

            var recipeLabels=[]; //Array to store Bar chart labels

            var kcal=[]; //Array to store kcal values of all recipes

            var fat=[]; //Array to store fat values of all recipes

            var saturates=[]; //Array to store saturates values of all recipes

            var carbs=[]; //Array to store carbs values of all recipes

            var sugars=[]; //Array to store sugars values of all recipes

            var fibre=[]; //Array to store fibre values of all recipes

            var protein=[]; //Array to store protein values of all recipes

            var salts=[];//Array to store salts values of all recipes

        </script>

<?php

      $c=1; //Counter

//The below foreach loops through results fetched from database

       foreach($results as $row){

         ?>

         <script>

            recipeLabels.push("<?php echo $row['name']?>"); //Adding recipe name to recipeLabels array

           kcal.push(<?php echo $row['kcal']?>); //Adding kcal info to kcal array

           fat.push(<?php echo $row['fat']?>); //Adding fat info to fat array

           saturates.push(<?php echo $row['saturates']?>); //Adding saturates info to saturates array

           carbs.push(<?php echo $row['carbs']?>); //Adding carbs info to carbs array

           sugars.push(<?php echo $row['sugars']?>); //Adding sugars info to sugars array

           fibre.push(<?php echo $row['fibre']?>); //Adding fibre info to fibre array

           protein.push(<?php echo $row['protein']?>); //Adding protein info to protein array

           salts.push(<?php echo $row['salt']?>); //Adding salts info to salt array

                   </script>

        <h4>Recipe <?php echo $c++ ?></h4> //Display Recipe No.

        <table> //Table

            <thead> //Table headings contains Name, Author and more details of recipe

            <tr>

        <th>Name</th>

        <th>Author</th>

        <th>More Details</th>

      </tr>

            </thead> /?End of table headings

            <tbody> //Table body

                <tr>

<?php

         echo '<td>' . $row['name'] . '</td>'; //Display recipe name

         echo '<td>' . $row['author'] . '</td>'; //display recipe author

//Displays Prep Time, Cook Time, Serves, Ratings, Ingredients and method

         echo '<td>Prep Time: ' . $row['prep'] . '<br>Cook Time: ' . $row['cook'] . '<br>Serves: ' . $row['serves']

         . '<br>Ratings: ' . $row['ratings'] . '<br>Description: ' . $row['description'] . '<br><br>Ingredients: ' .

         $row['ingredients'] . '<br><br>Method: ' . $row['method'] . '<br></td>';?>

         </tr>

       </tbody> //End of table body

       </table> //End of table

       <h5>Nutritions:</h5> //Nutritions Info

       <div class="chartStyles"> //Division element with class “chartStyles”

//Canvas for the chart

       <canvas id="<?php echo $row['id']?>" width="400" height="400"></canvas>

       </div>

       <script>

//Pie Chart

//Chartjs.org (2020), Getting Started. Available at <https://www.chartjs.org/docs/latest/getting-started/> (Accessed: 7 May 2022).

           var ctx = document.getElementById('<?php echo $row['id']?>').getContext('2d'); //Get the element

//Create Pie chart

var myChart = new Chart(ctx, {

type: 'pie',

data: {

    labels: ['Fat', 'Saturates', 'Carbs', 'Sugars', 'Fibre', 'Protein', 'Salts'], //Labels for pie chart

    datasets: [{

        label: '# g',

//Data inside Pie Chart displays fat, saturates, carbs, sugars, fibre, protein and salts info

        data: [<?php echo $row['fat']?>, <?php echo $row['saturates']?>, <?php echo $row['carbs']?>, <?php echo $row['sugars']?>, <?php echo $row['fibre']?>, <?php echo $row['protein']?>, <?php echo $row['salt']?>],

//Background colors for each labels data

        backgroundColor: [

            'rgba(0, 99, 132, 0.2)',

            'rgba(54, 162, 235, 0.2)',

            'rgba(0, 206, 86, 0.2)',

            'rgba(75, 192, 192, 0.2)',

            'rgba(153, 102, 255, 0.2)',

            'rgba(255, 159, 64, 0.2)',

            'rgba(152, 159, 64, 0.2)'

        ],

//Border colors for each labels data

        borderColor: [

            'rgba(255, 99, 132, 1)',

            'rgba(54, 162, 235, 1)',

            'rgba(255, 206, 86, 1)',

            'rgba(75, 192, 192, 1)',

            'rgba(153, 102, 255, 1)',

            'rgba(255, 159, 64, 1)',

            'rgba(152, 159, 64, 1)'

        ],

//Border width between data

        borderWidth: 1

    }]

},

options: {

    // resnponsive:true;

    // maintainAspectRatio: false;

    scales: {

        y: {

            beginAtZero: true //Y-axis begins at 0

        }

    }

}

});

       </script>

//Displays kcal info of recipe

              <p style="text-align:center;">Kcal:<?php echo $row['kcal']?></p>

              <br>

<?php

} //End of foreach

       ?>

        //Bar chart hidden

           <h3 id="bar" style="display:none;">Bar chart for Nutritions</h3>

//canvas for the chart hidden

    <canvas id="myChart" style="display:none;"></canvas>

    <script>

//Bar chart is displayed only when more than 1 recipes are selected

        if(<?php echo count($results)?>>1){

            document.getElementById("bar").style.display="block"; //Becomes visible

            var canvasElement= document.getElementById("myChart").style.display="block"; //Becomes visible

            canvasElement.width=300; //Setting width of canvas

            canvasElement.height=300; //Setting height of canvas

        var ctx = document.getElementById("myChart").getContext("2d");

var data = {

  labels: recipeLabels, //Labels for bar chart from recipeLabels array

  datasets: [{

//For Kcal set background color to blue and data is kcal array

    label: "Kcal",

    backgroundColor: "blue",

    data: kcal

  }, {

//For fat set background color to rgba(0, 99, 132, 1) and data is fat array

    label: "Fat",

    backgroundColor: 'rgba(0, 99, 132, 1)',

    data: fat

  }, {

//For saturates set background color to rgba(54, 162, 235, 1) and data is saturates array

    label: "Saturates",

    backgroundColor: 'rgba(54, 162, 235, 1)',

    data: saturates

  }, {

//For carbs set background color to rgba(0, 206, 86, 1) and data is carbs array

    label: "Carbs",

    backgroundColor: 'rgba(0, 206, 86, 1)',

    data: carbs

  }, {

//For sugars set background color to rgba(75, 192, 192, 1)and data is sugars array

    label: "Sugars",

    backgroundColor: 'rgba(75, 192, 192, 1)',

    data: sugars

  }, {

//For fibre set background color to rgba(153, 102, 255, 1)and data is fibre array

    label: "Fibre",

    backgroundColor: 'rgba(153, 102, 255, 1)',

    data: fibre

  }, {

//For protein set background color to rgba(255, 159, 64, 1) and data is protein array

    label: "Protein",

    backgroundColor: 'rgba(255, 159, 64, 1)',

    data: protein

  }, {

//For salts set background color to rgba(152, 159, 64, 1)and data is salts array

    label: "Salts",

    backgroundColor: 'rgba(152, 159, 64, 1)',

    data: salts

  }

]

};

//Create Bar chart with data above

var myBarChart = new Chart(ctx, {

  type: 'bar',

  data: data,

  options: {

    barValueSpacing: 20,

    scales: {

      yAxes: [{

        ticks: {

          stepSize: 50, //Step size in Y-axis is 50

          min: 0 //starts at 0

        }

      }]

    }

  }

});

        }

    </script>

   <?php

       }

      }

     catch(PDOException $e) //Catch any exceptions

     {

      echo $e->getMessage(); //Display exception messages

     }

   ?>

   </main>

           <footer>&copy; CSYM019 2022</footer> //Footer for the page

   </div>

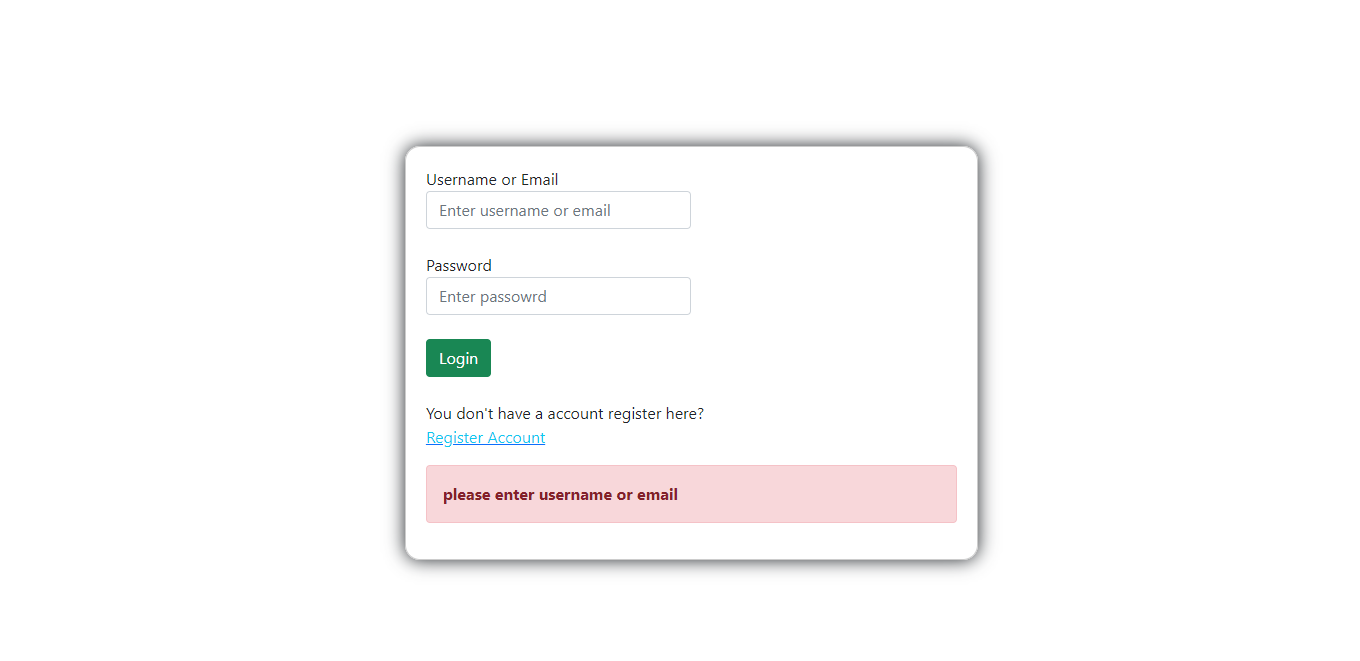
   </body> //End of Body

   </html> //End of HTML

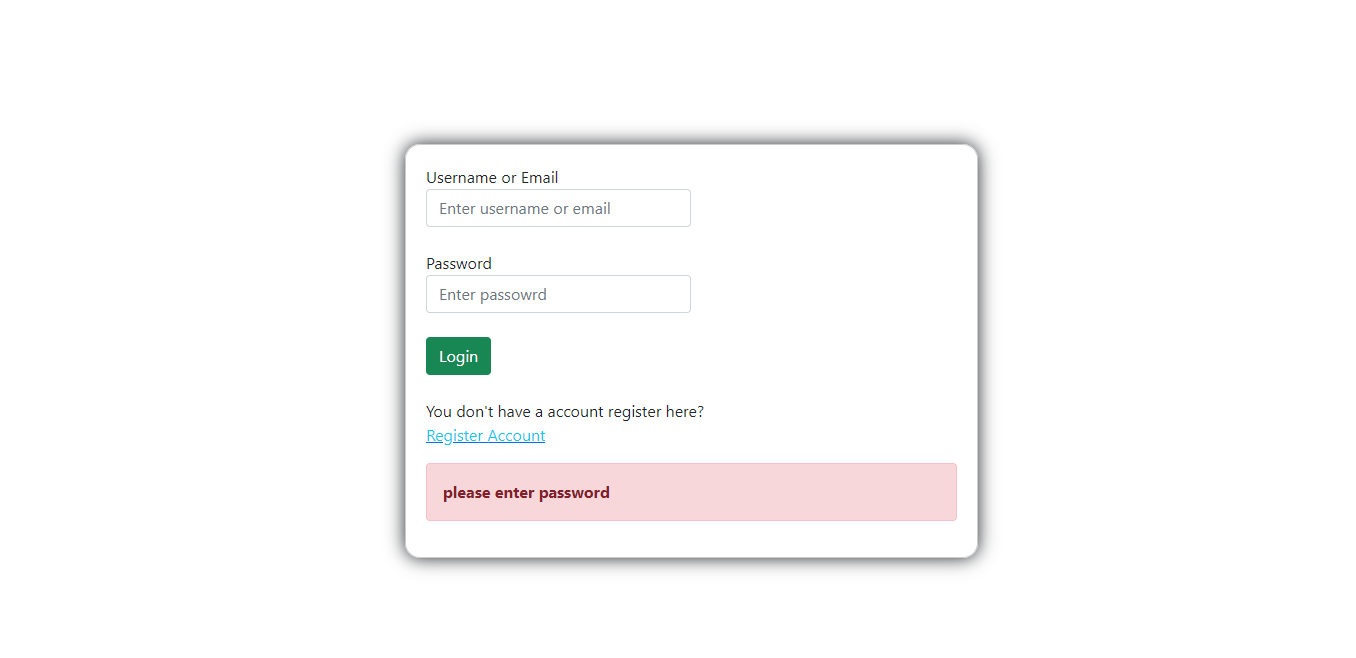
**Application Testing:**

Application starts at index.php where it prompts for username or email and password to login.

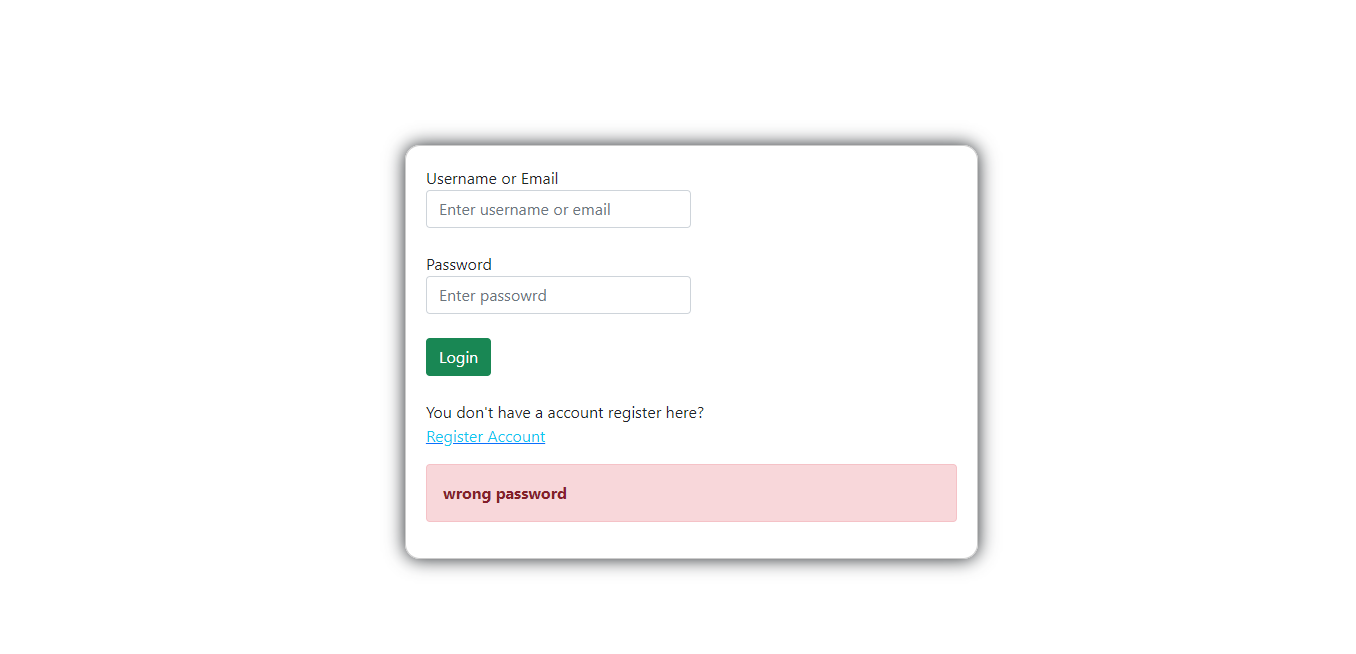
* If the username or email is not entered and login button is clicked, it gives Error Message “Please enter username or email”.



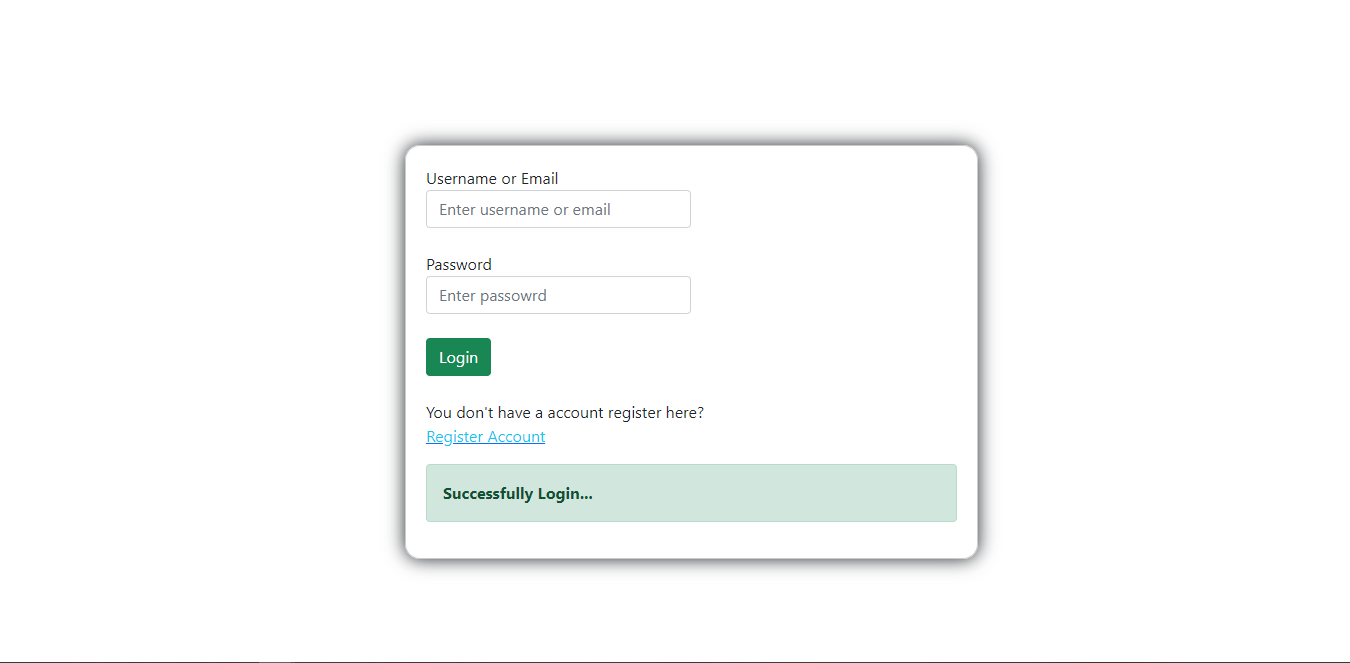
* If the password is not entered, it gives error message “please enter password”.



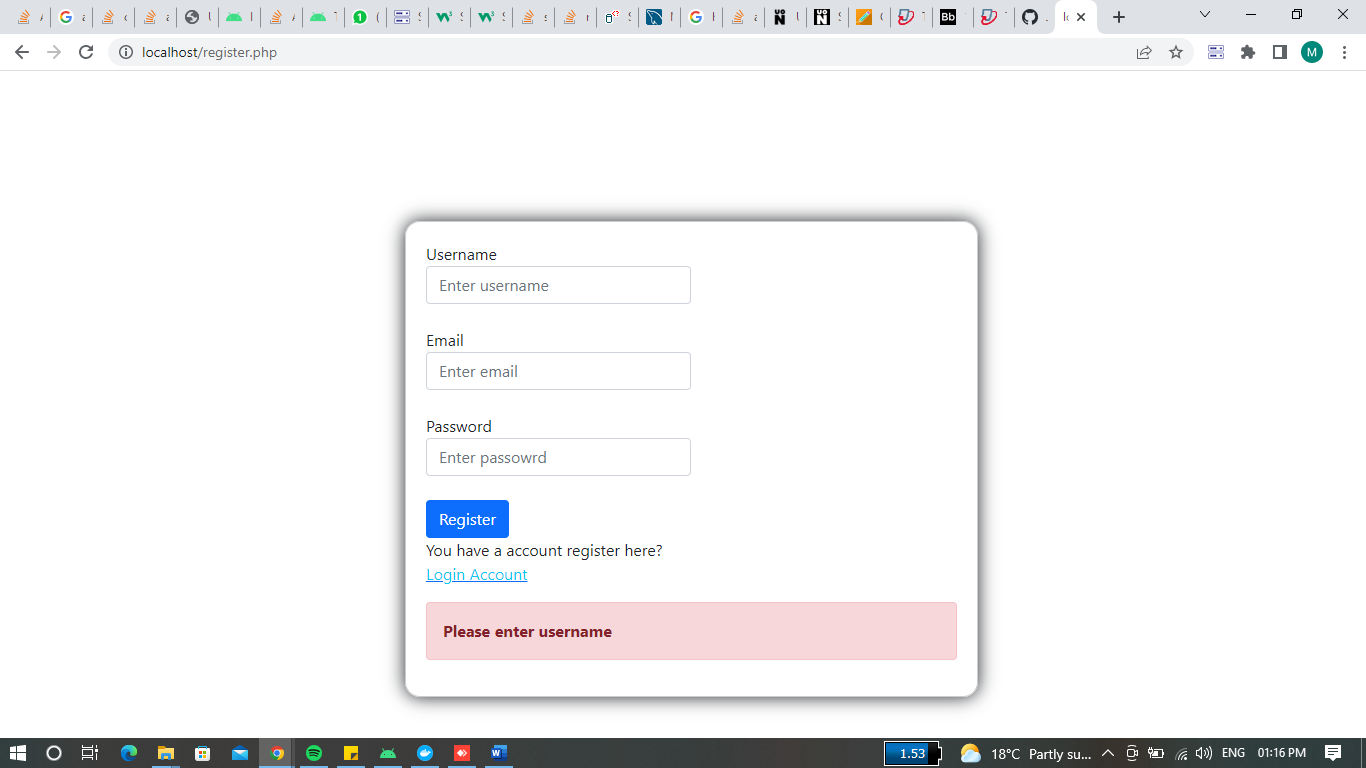
* If the password is incorrect, it gives error message “wrong password”.



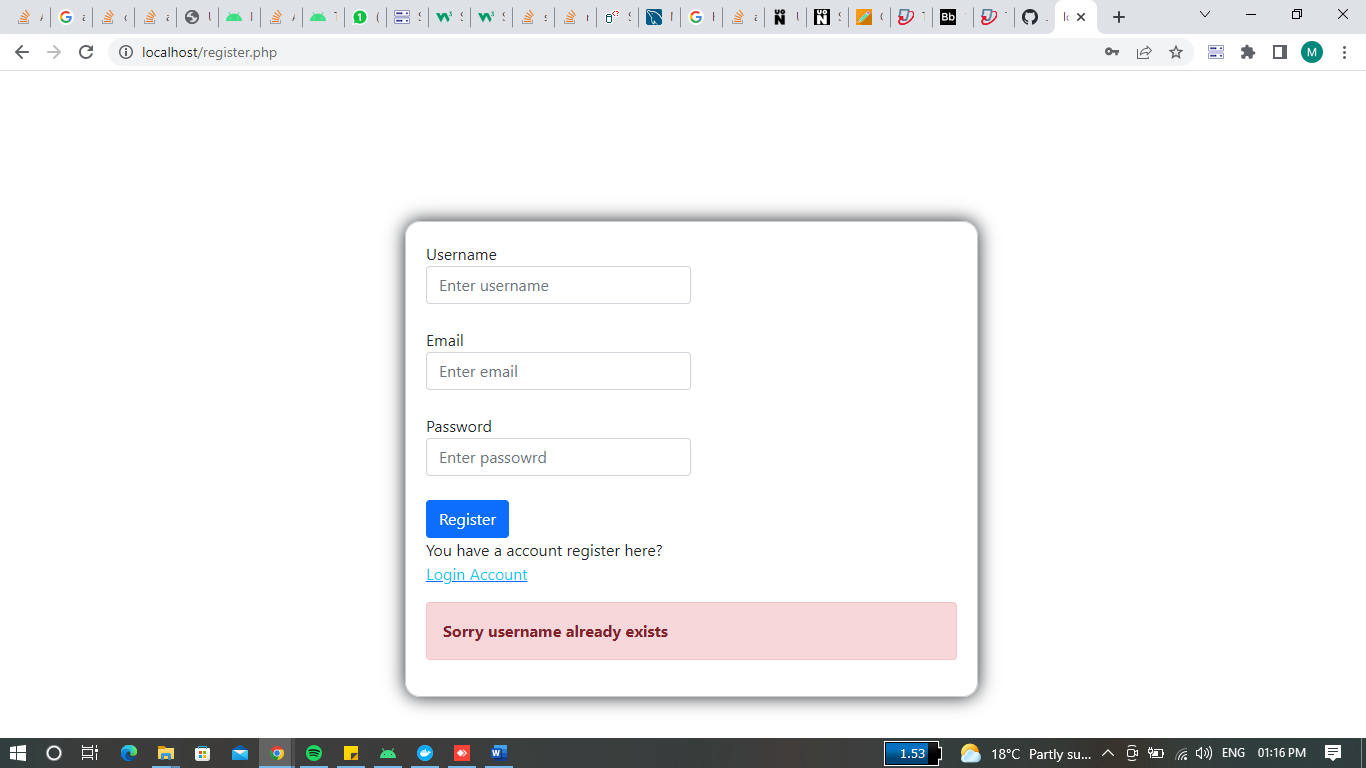
* If the details are entered correctly and login button is clicked, it displays “Successful Login” message and redirects to home.php after 2 seconds. If user don’t have a login credentials, click on Register Account link.



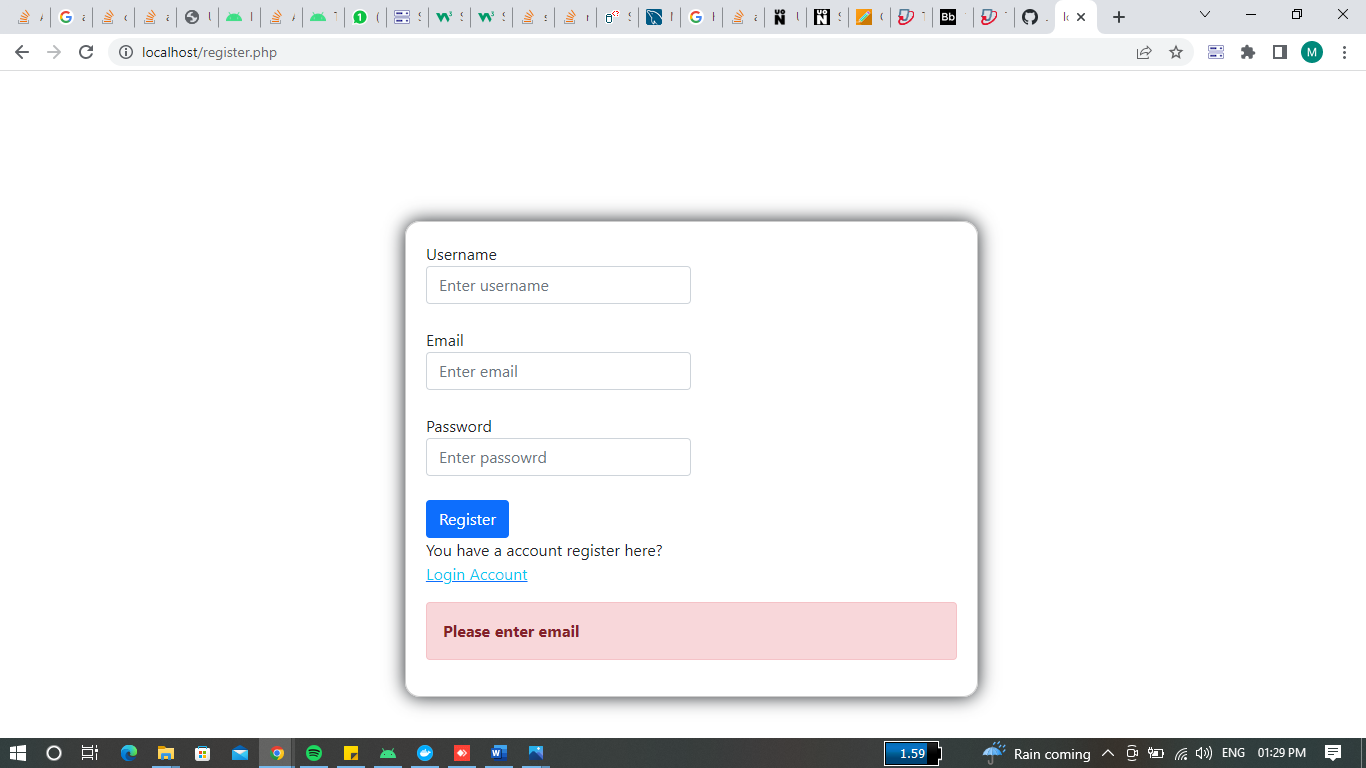
* Once the register.php opens, user has to enter username, email and password. If username is not entered and register button is clicked, it gives error message “Please enter username”.



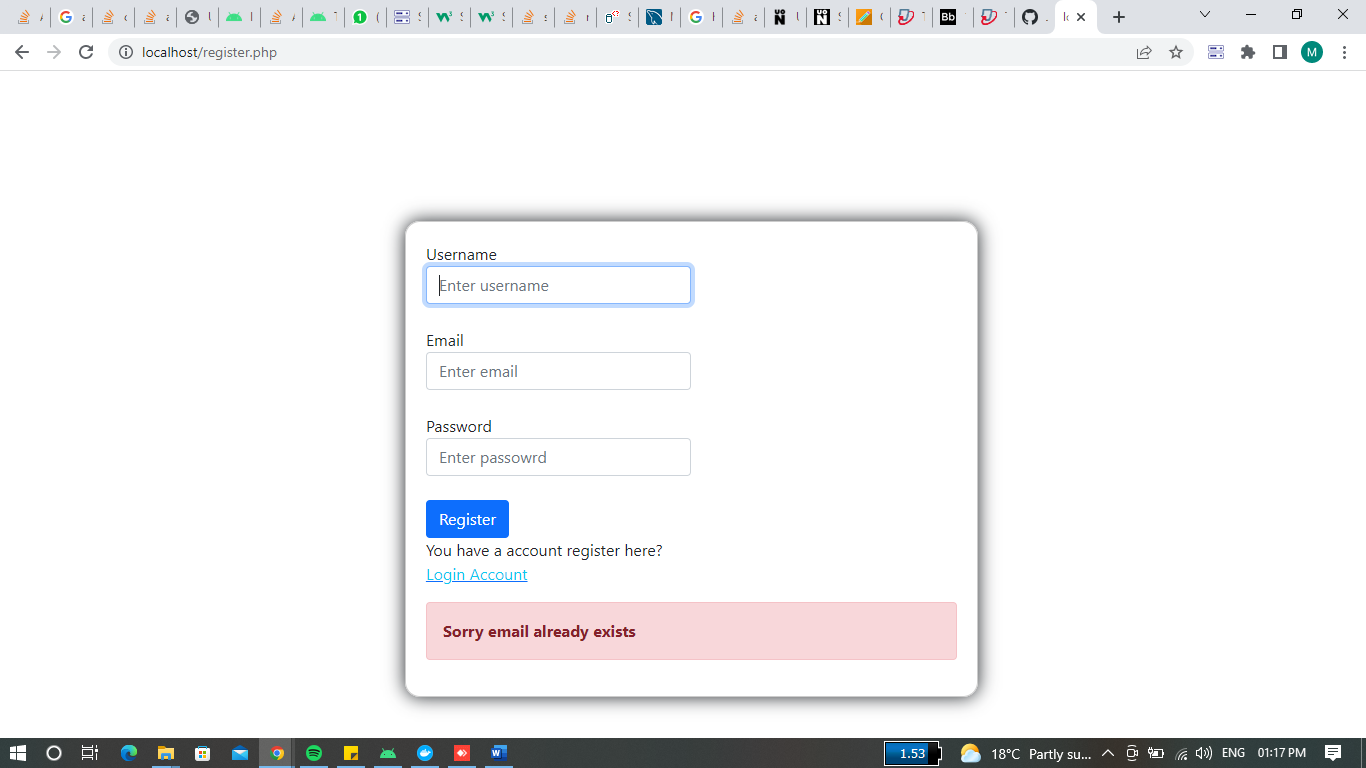
* If the username entered already exists in the database, it displays “Sorry username already exists”.



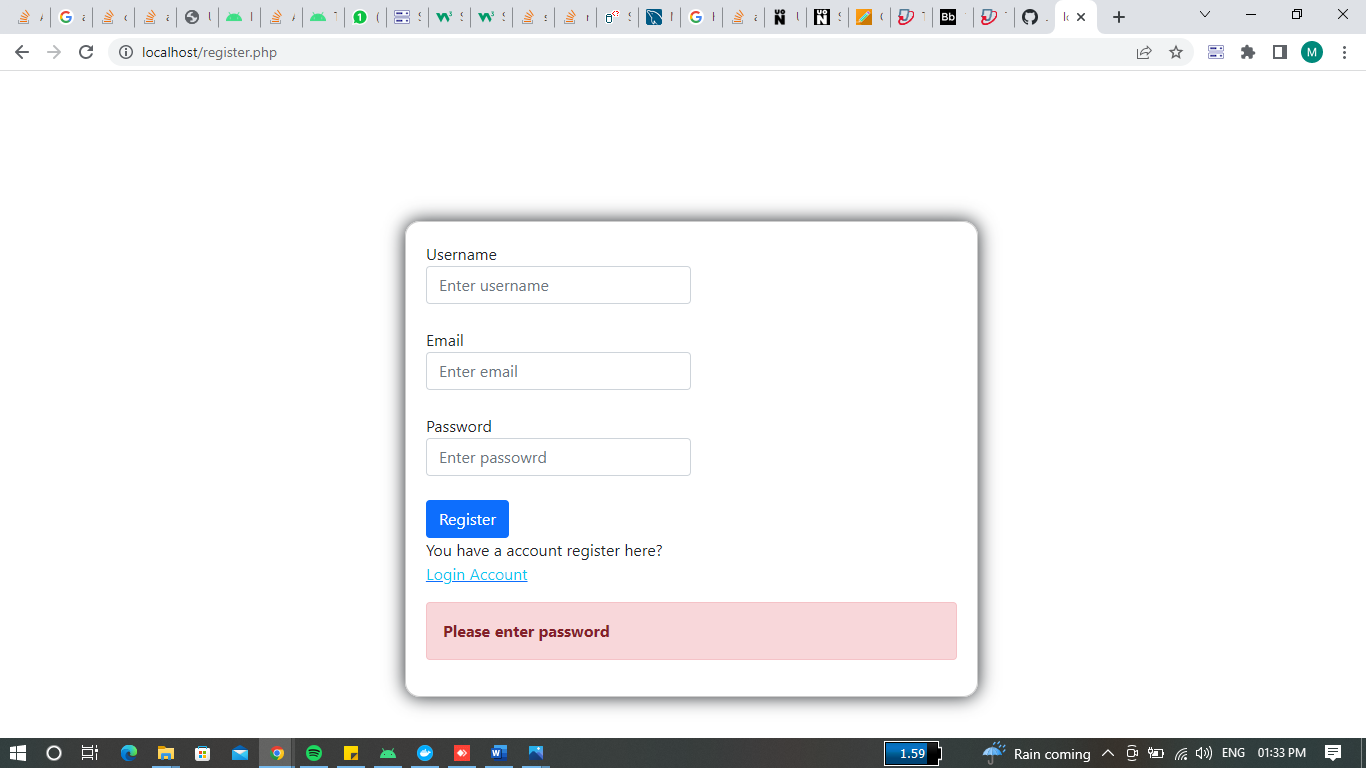
* If email is not entered, it gives error message “Please enter email”.



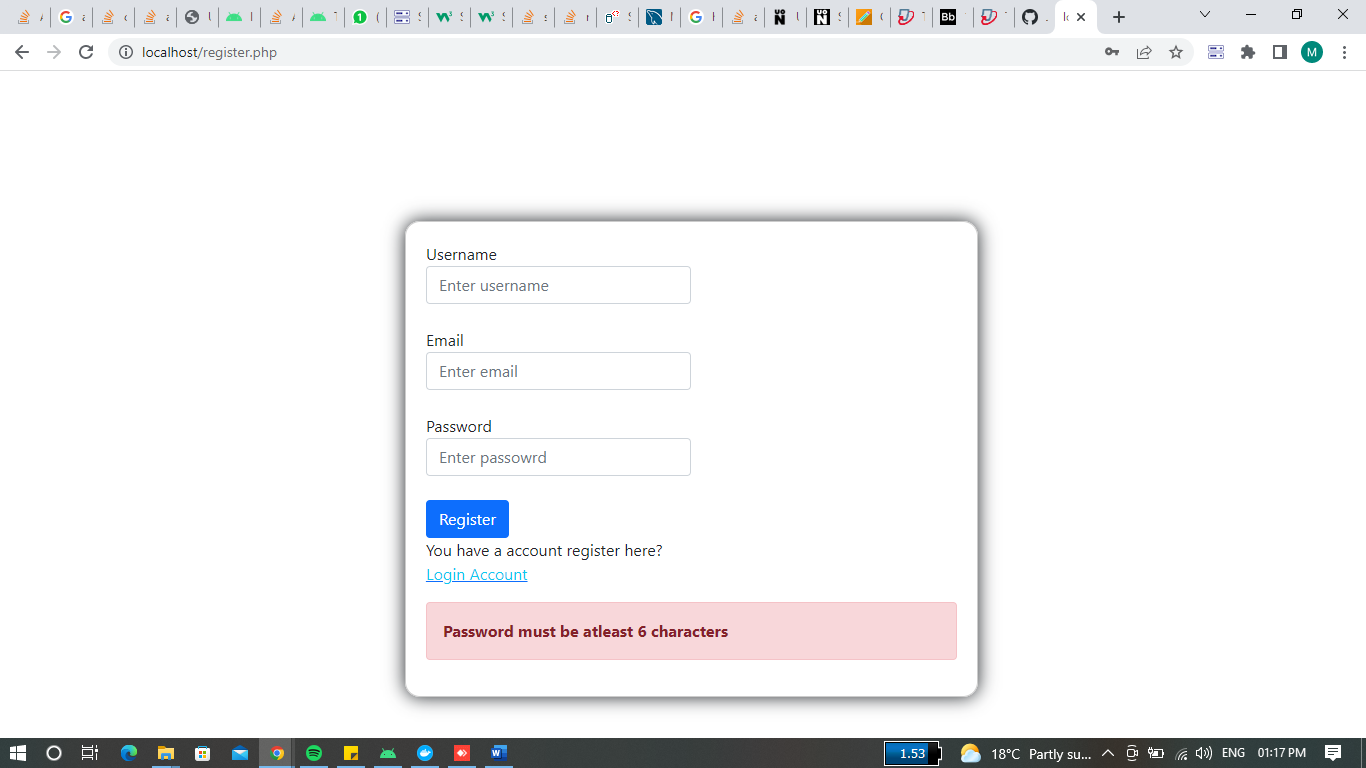
* If the email entered already exists in the database, it displays “Sorry email already exists”.



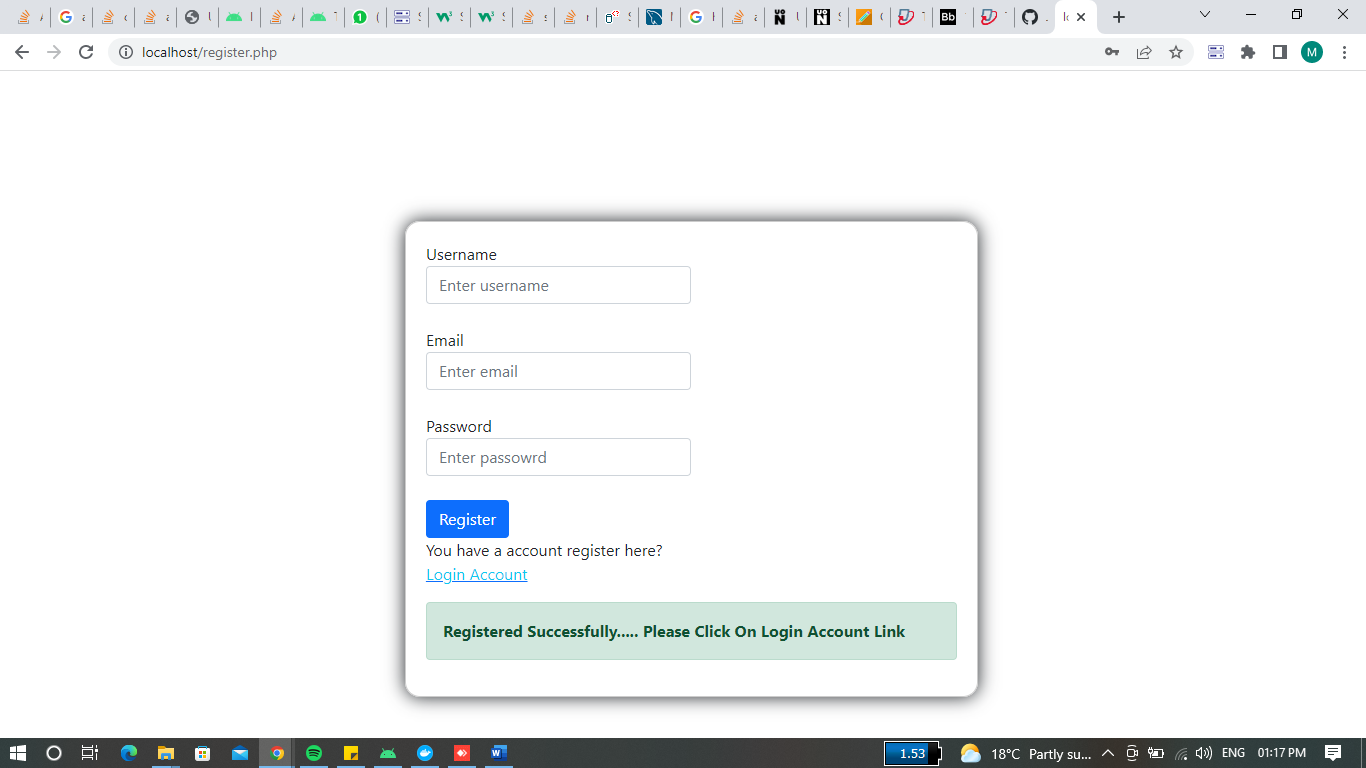
* If the above fields are entered and password is not entered, it gives error message “Please enter password”.



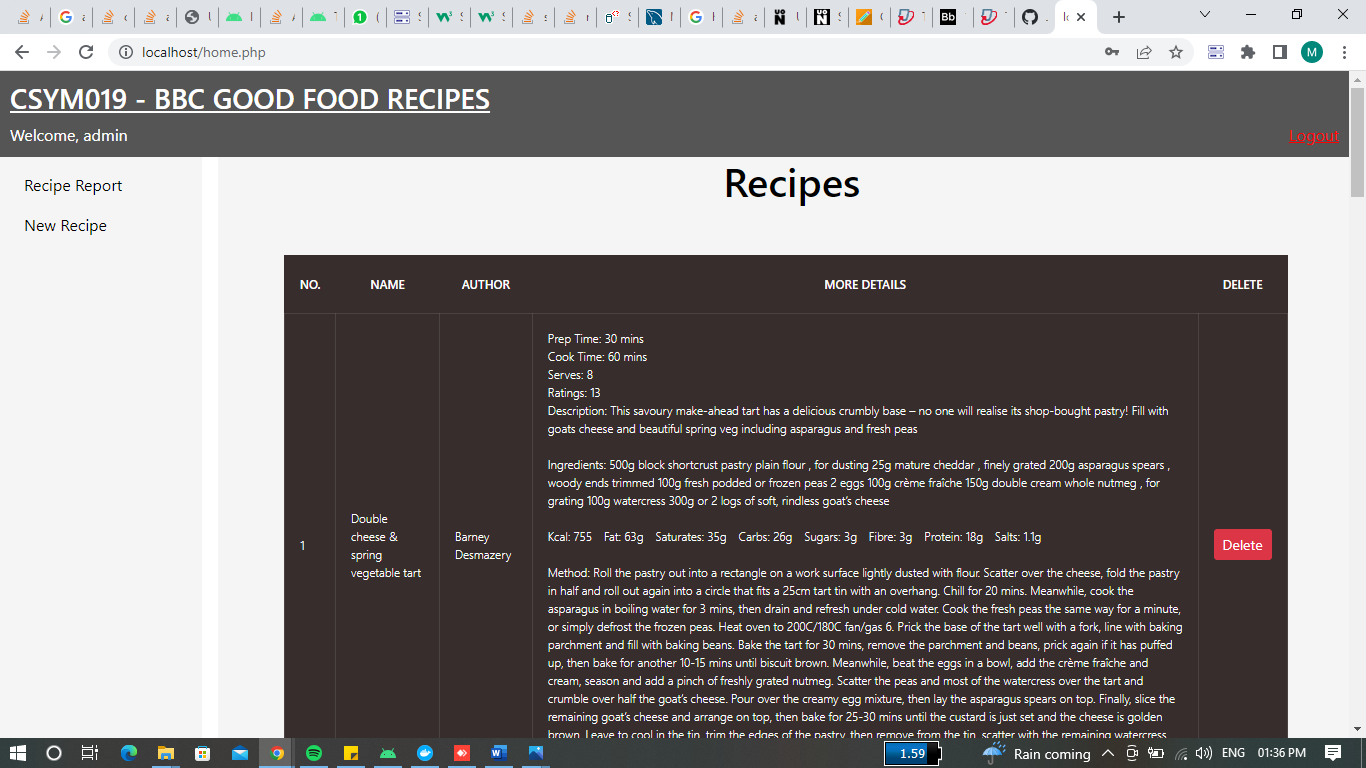
* If the password is less than 6 characters, it displays error message “Password must be atleast 6 characters”.



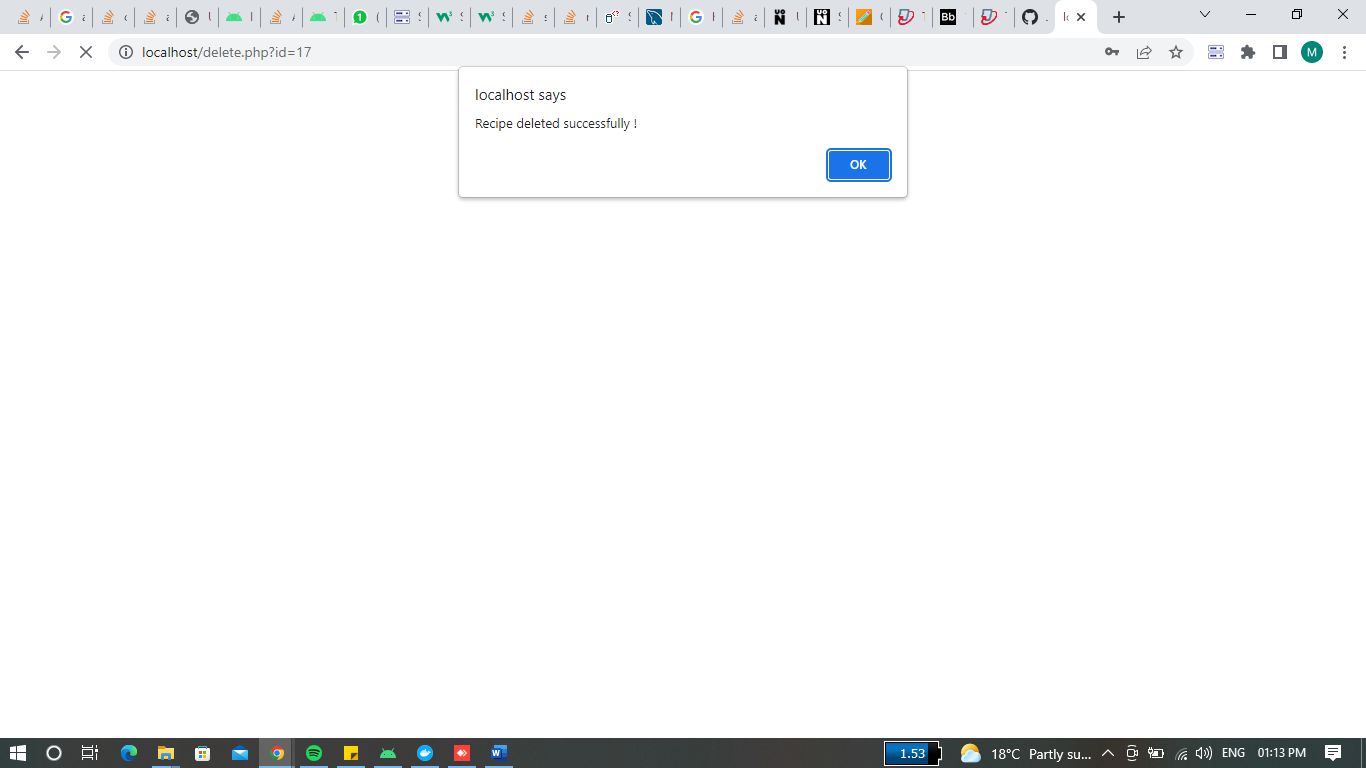
* If there are no errors then “Registered Successfully..... Please Click On Login Account Link” is displayed.



* The user has to click on Login Account for logging in.
* Once the user logs in the session starts and home page is displayed. CSYM019 – BBC GOOD FOOD RECIPES acts as the home icon and a welcome message is set with username. On the right hand side of the header, there’s a logout link for the user to logout.
* There’s a nav section on the left of the page to navigate to Recipe Report and New Recipe.
* If there are no recipes in the database, it displays “Sorry No recipes Exists” error message under the welcome message in header.
* Each recipe is displayed in the table that has a delete button associated with it. Recipe information such as No., Name, Author, More Details which consists of Prep Time, Cook Time, Serves, Ratings, Description, Ingredients, Nutrition Information and Method are displayed.



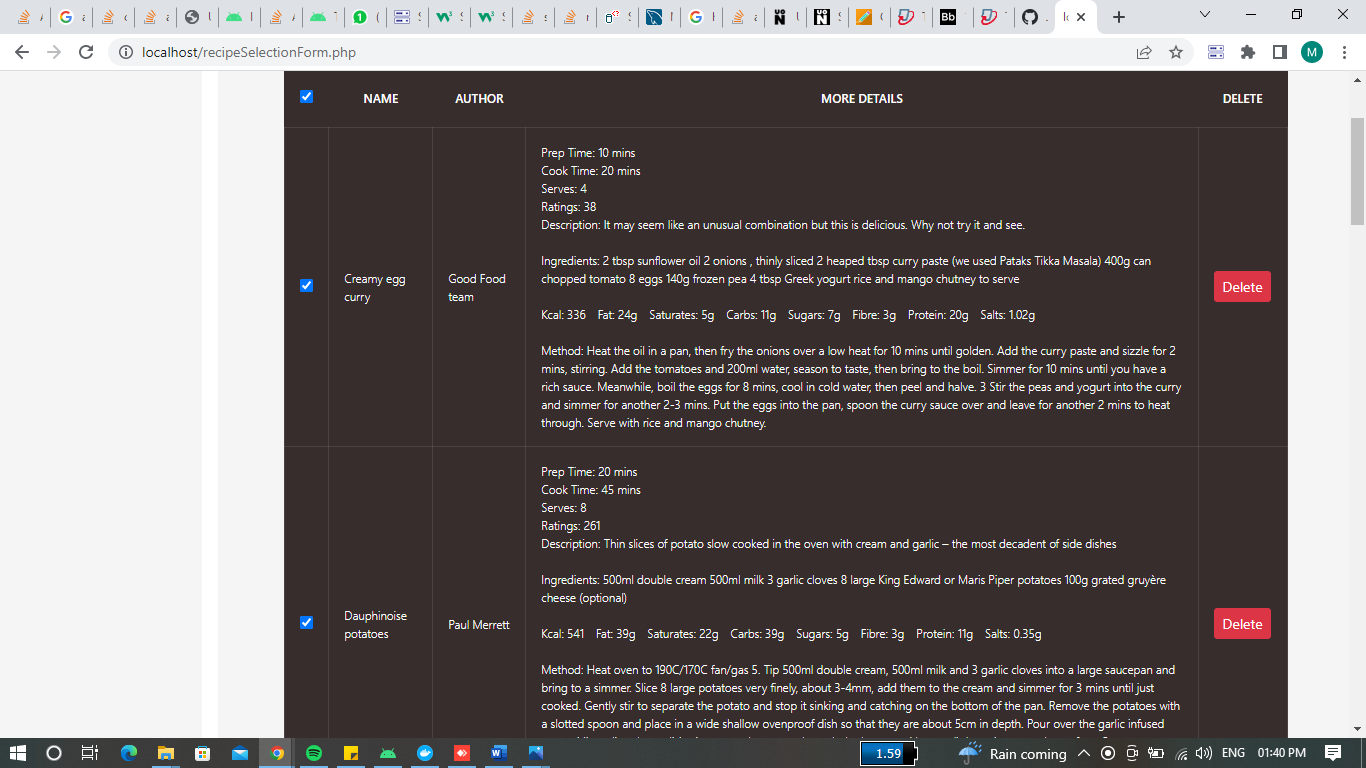
* If the delete button is clicked an alert box comes up which says “Recipe deleted succesfully”. Once OK button on the alert box is pressed, it redirects to home.php where the recipe details are fetched again and displayed in the table.



* When the user clicks on Recipe Report in the nav, it navigates to recipeSelectionForm.php where this time recipes are displayed with checkboxes as well.

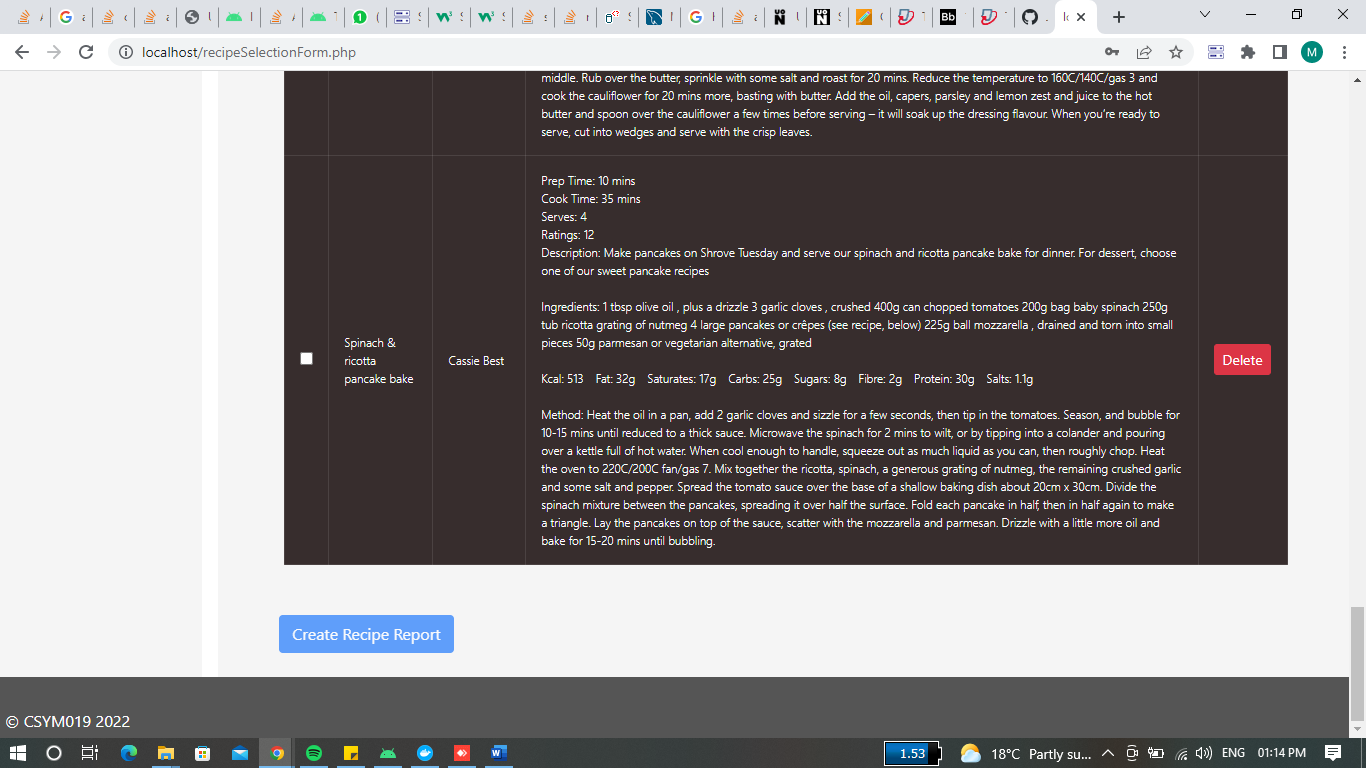


* The table consists of a checkbox in table heading where you can toggle the remaining checkboxes. Each recipe has its own checkbox, Name, Author, More details which consists of Prep Time, Cook Time, Serves, Ratings, Description, Ingredients, Nutrition Information, Method and delete button.
* When the checkbox is clicked in the table heading, all the remaining checkboxes are checked. When checkbox is clicked again, all the remaining checkboxes are unchecked.

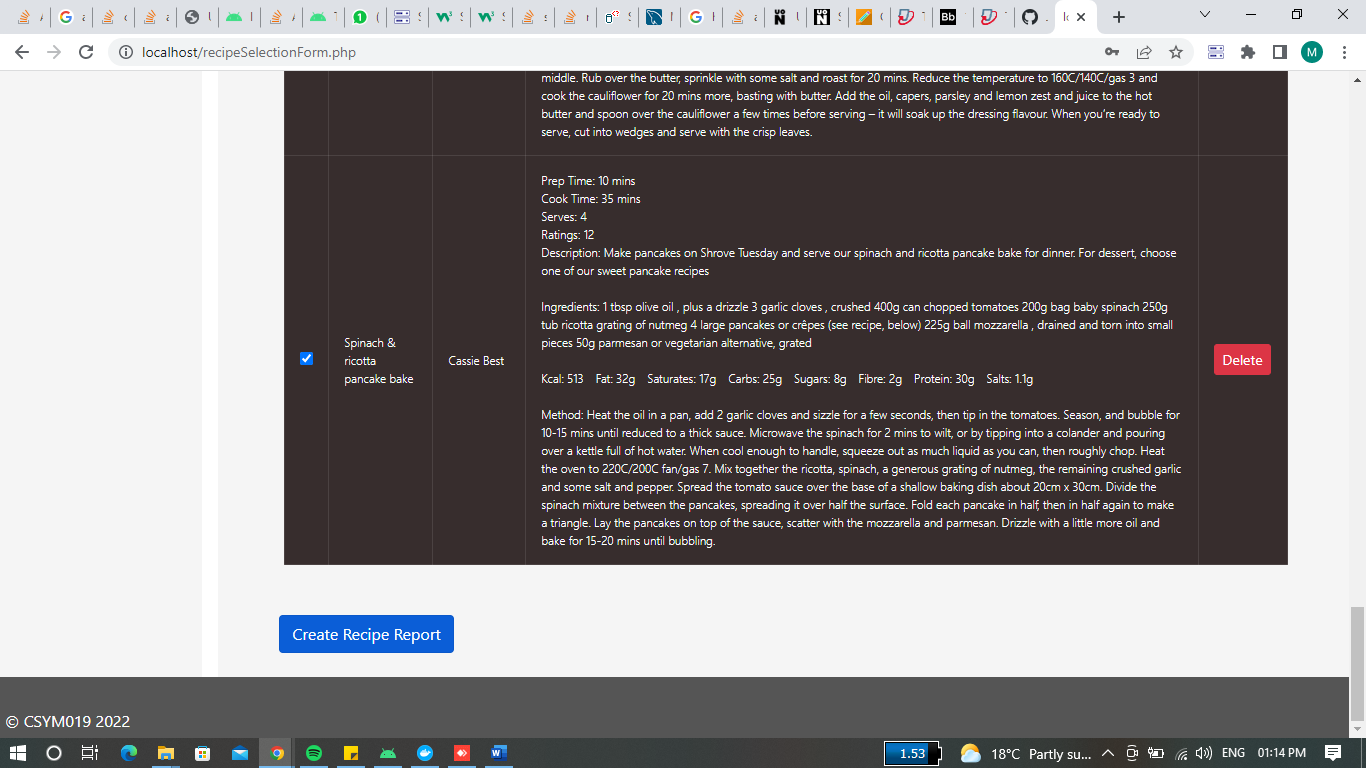


There’s a Create Recipe Report button at the end of the table.

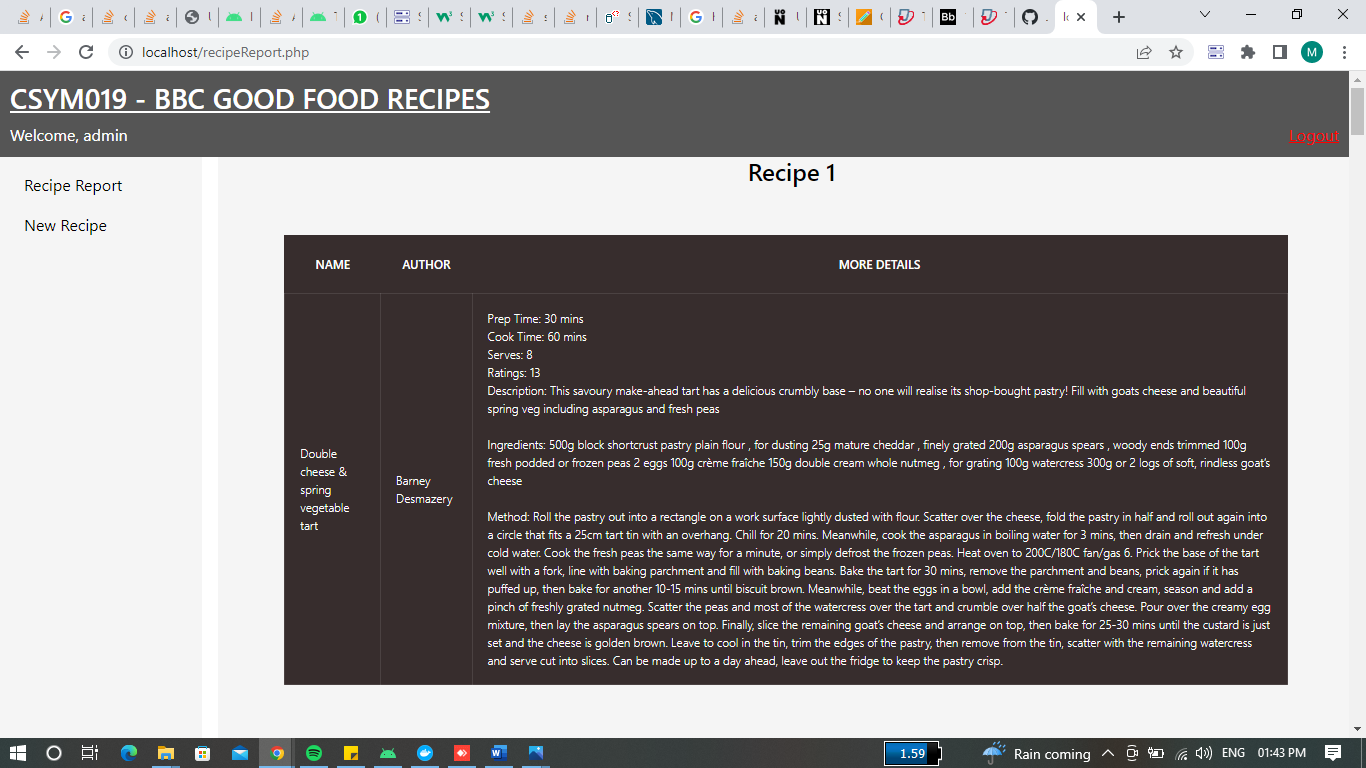
* When no checkbox is checked, the button will be disabled.



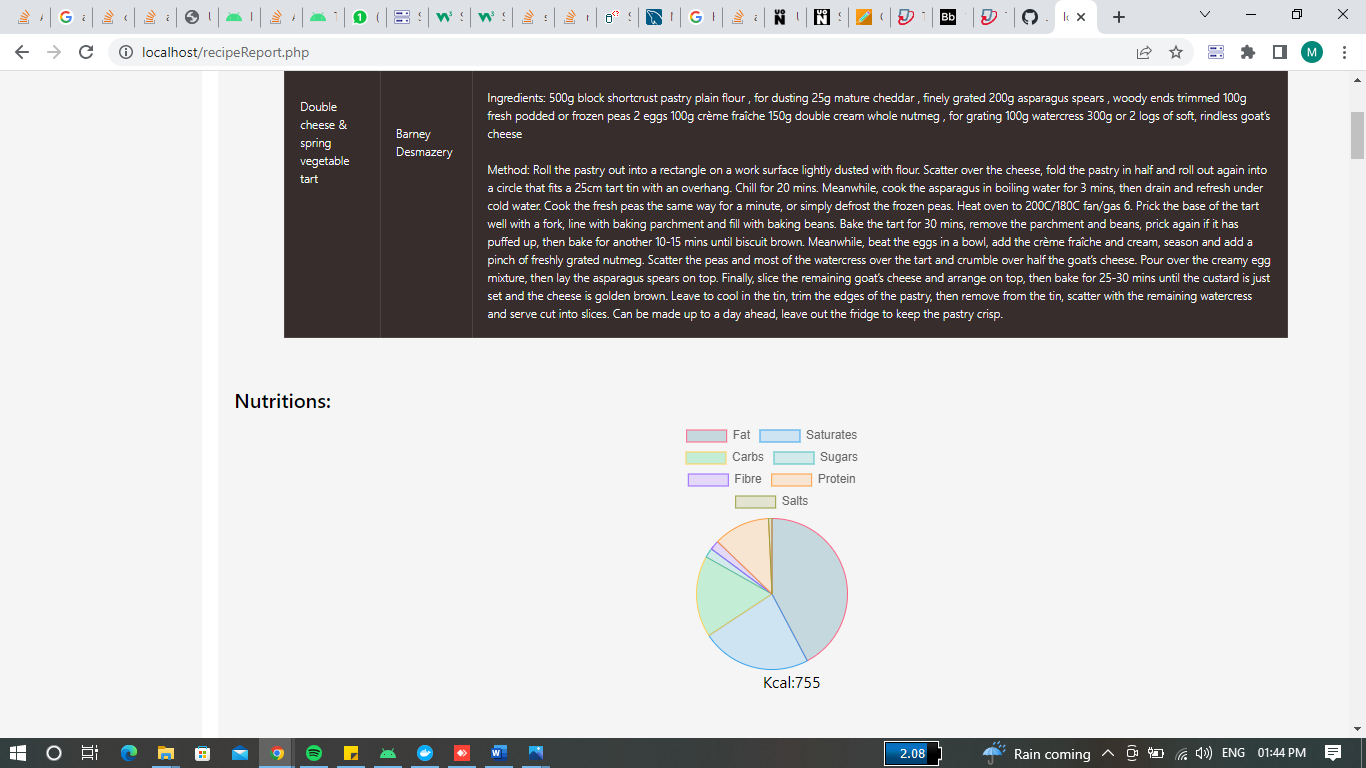
* If at least one checkbox is checked, the button is enabled and on click generates a recipe report of the selected recipes by navigating to recipeReport.php page.



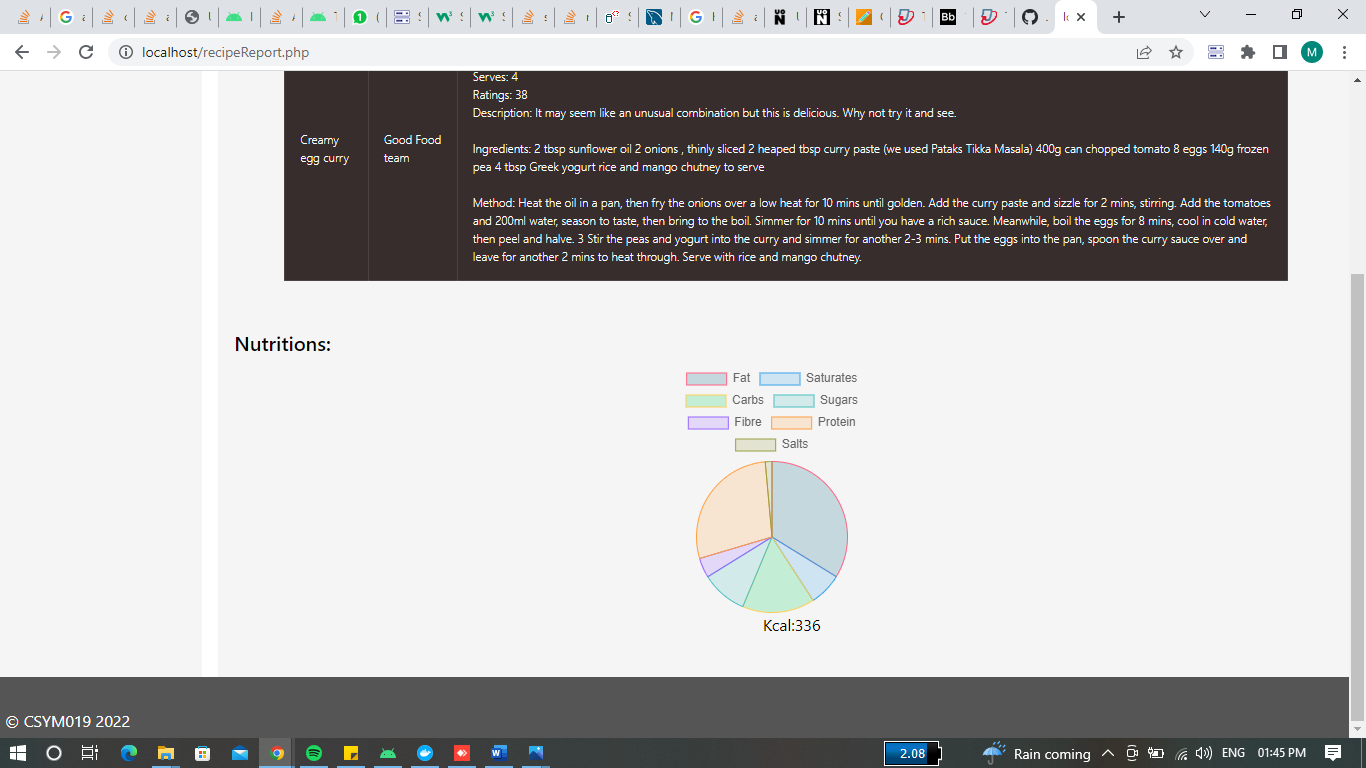
Once atleast one recipe is checked, it navigates to recipeReport.php page where it consists of Recipe No. A table which gives information about the following recipe except nutritions.



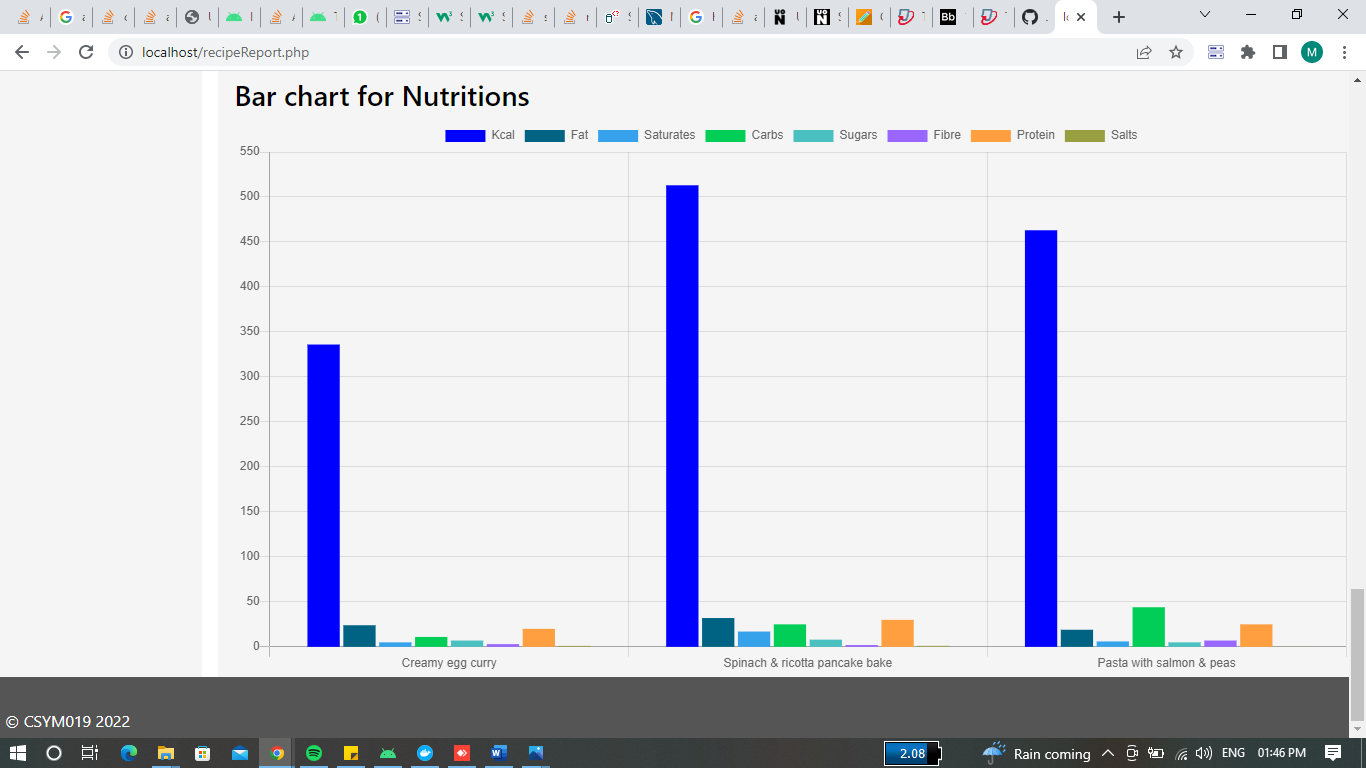
The nutrition information such as Fat, Saturates, Carbs, Sugars, Fibre, Protein, Salts is displayed in a pie chart. The Kcal information of recipe is displayed as a text under the pie chart.



If more than one recipes are selected, then the recipeReport also generates a Bar Chart(where the canvas is hidden in case of only 1 recipe).

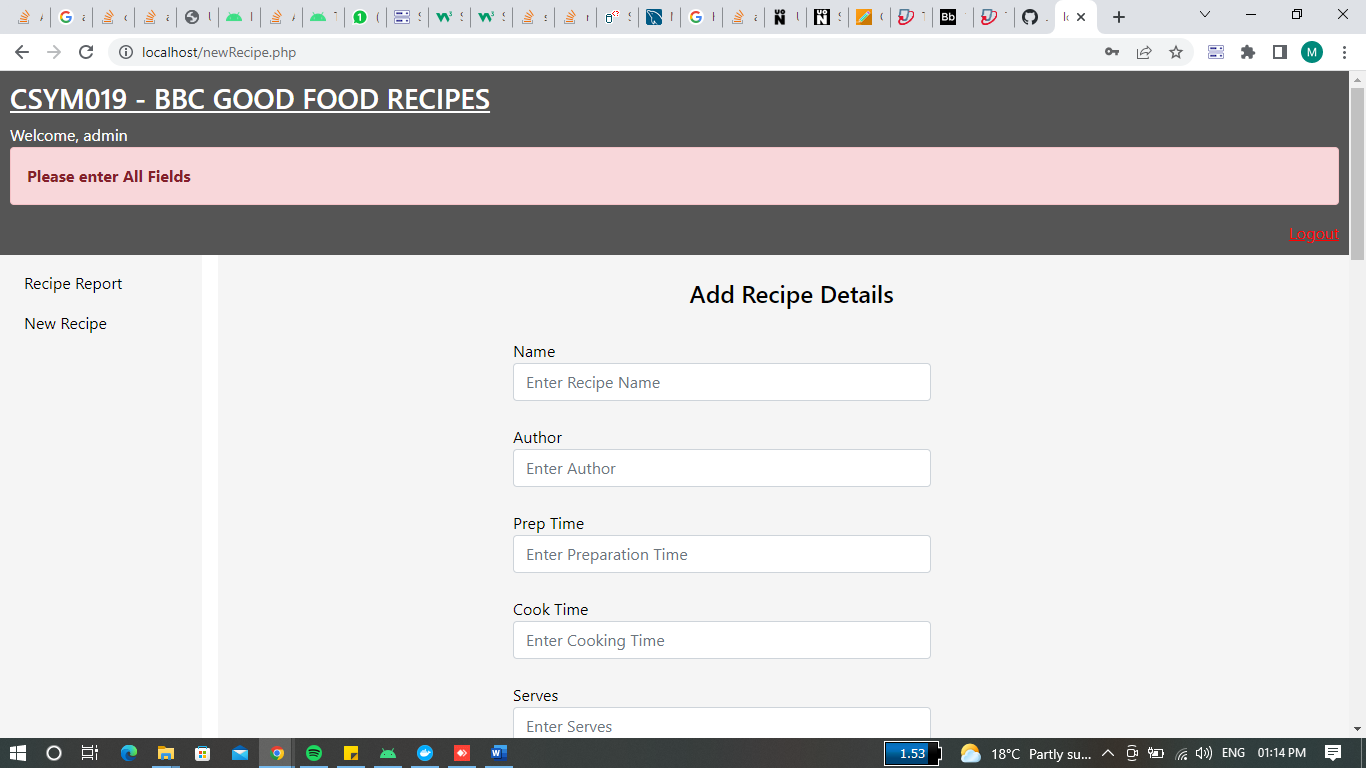


Bar chart gives information about nutrition info of all the selected recipes including Kcal and other nutrition such as fat, saturates, carbs, protein, fibre, sugar, and salts.

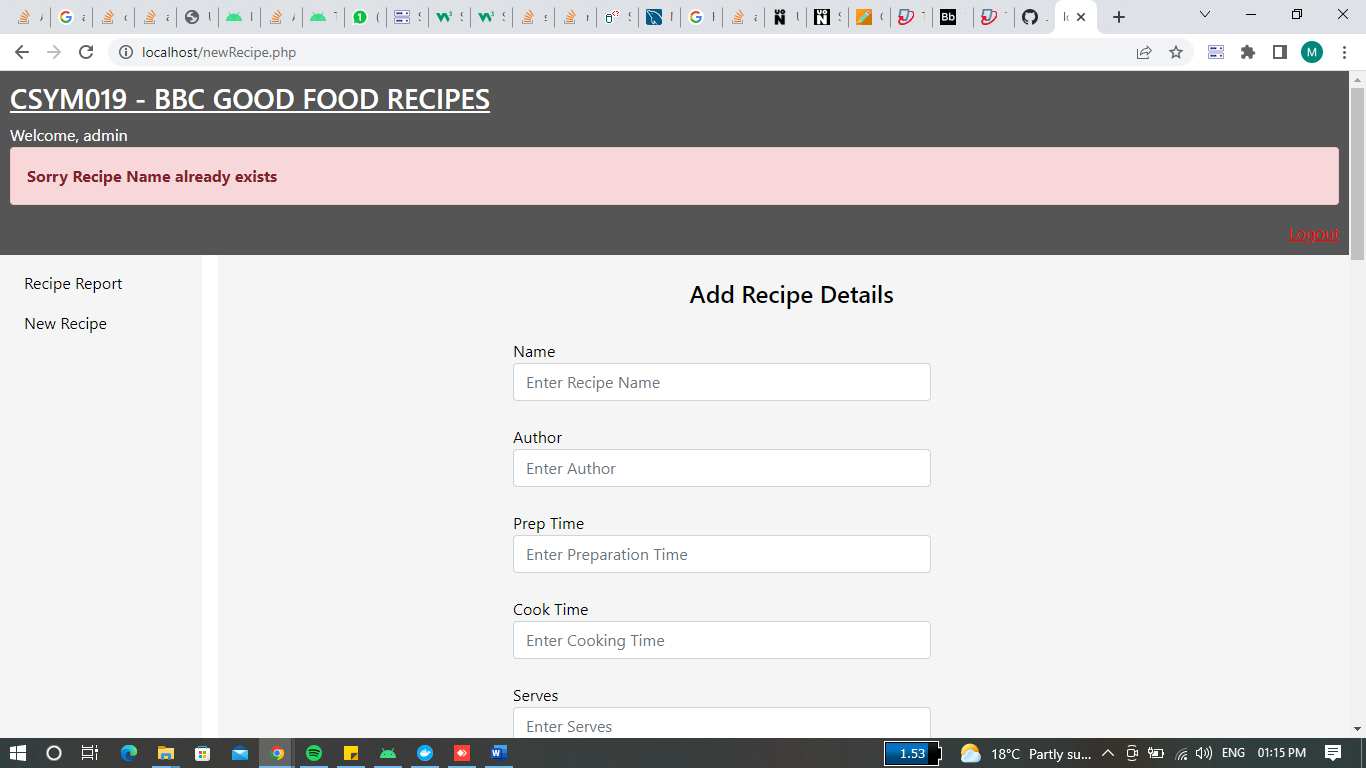


If the New Recipe is clicked, then it navigates to newRecipe.php where the user has to enter new recipe information.

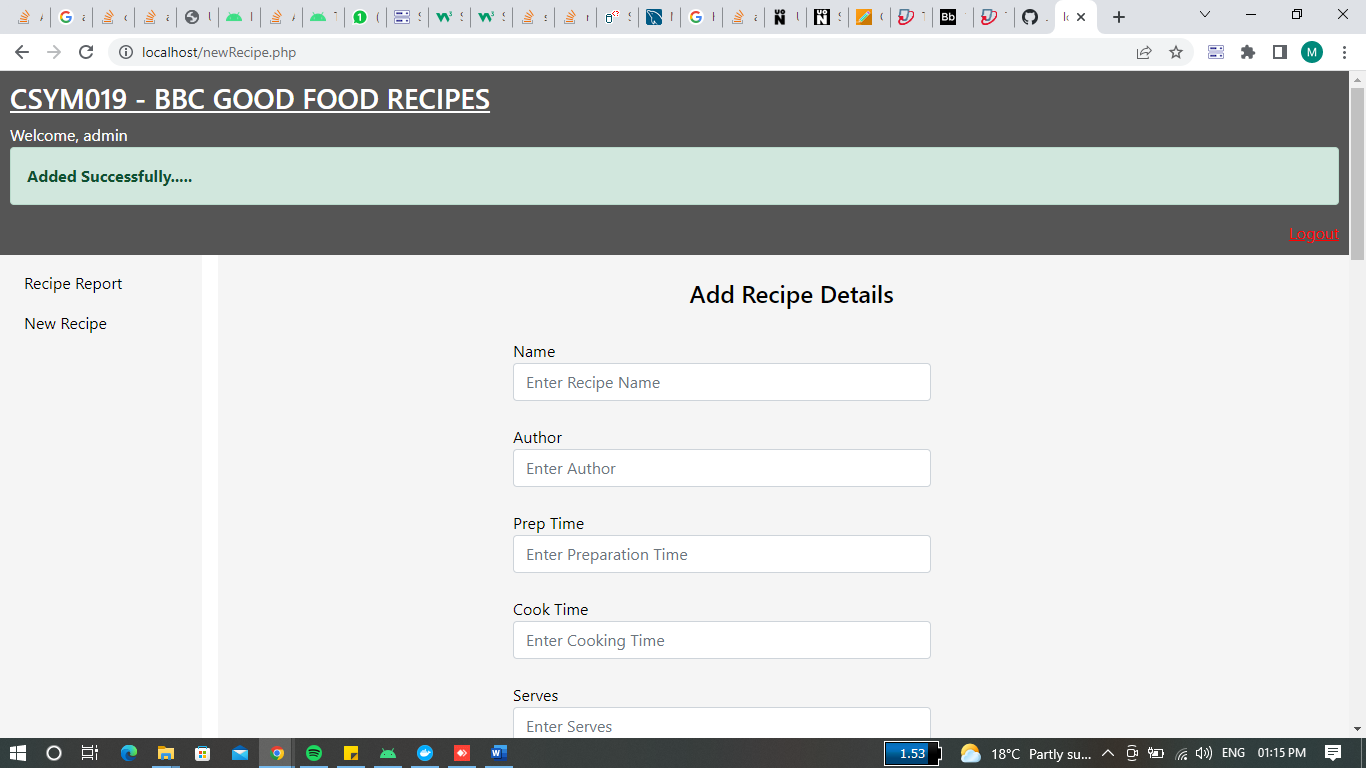
* The form has following fields Name, Author, Prep Time, Cook Time, Serves, Ratings, Food Type, Difficulty, Description, Ingredients, Kcal, fat, saturates, carbs, sugars, fibre, protein, salt, method.
* If the user has missed entering any of the fields and clicks on add button, it displays “Please enter all fields”.



* If the user enters the recipe name which already exists in the database, it displays “Sorry Recipe name already exists”.



* Once all the fields are entered and clicked on the “Add” button, it displays “Added Successfully” and redirects to home.php after 2 seconds.



When the user clicks on Logout link, the session is destroyed and the page is redirected to index.php. Even if user clicks on back button, it stays on index.php until user logs in again.

**References:**

Onlyxcodes.com (2022), Login and Register Script In PHP PDO with MySQL. Available at [https://www.onlyxcodes.com/2019/04/login-and-register-script-in-php-pdo.html](https://www.onlyxcodes.com/2019/04/login-and-register-script-in-php-pdo.html%20) (Accessed: 2 May 2022).

GetBootstrap.com (2015), Getting started. Available at <https://getbootstrap.com/>(Accessed: 5 May 2022).

Developers.google.com (2020), Google Hosted Libraries. Available at <https://developers.google.com/speed/libraries#jquery> (Accessed: 5 May 2022).

Chartjs.org (2020), Getting Started. Available at <https://www.chartjs.org/docs/latest/getting-started/> (Accessed: 7 May 2022).

**Video Link:**

<https://northampton.mediaspace.kaltura.com/media/1_l2po1b37>