CRUD - Intro - Text follow-up

Create Read Update Delete

First in mysql cli, create a database and table for the project “To Do List”

CREATE DATABASE CRUD;

Then use the database

USE CRUD;

Create the table

```
CREATE TABLE ToDos ( ToDoID INT NOT NULL AUTO_INCREMENT
                    ,ToDoTitle varchar(50) DEFAULT NULL
                    ,ToDoDescription varchar(1000) DEFAULT NULL
                    ,Complete boolean DEFAULT NULL
                    ,ToDueDate datetime DEFAULT NULL
                    ,EntryTS datetime DEFAULT NULL
                    ,UpdateTS datetime DEFAULT NULL
                    ,CompleteTS datetime DEFAULT NULL
                    ,PRIMARY KEY (ToDoID));
```

Make a test entry:

```
INSERT INTO ToDos (ToDoTitle , ToDoDescription)
VALUES ('Laundry' , 'Do the Laundry');
```

Create new folder ToDoCrud under workspace

Create a new file titled ToDoApp.php

Fill the file will the following content:

```
<?php
/*Connect to CRUD Database  mysqli(Server,User,Password,Database)*/
$link = new mysqli('localhost', 'root', '', 'CRUD');
/*Write error if they exist, otherwise, write success*/
if ($link->connect_error) {
  die("Connection failed: " . $link->connect_error);
} else {echo "<br>Connected successfully"; }
/*Close database connection*/
  mysqli_close ( $link );
?>
```

Click Run Project

Open the url from Apache

Open the ToDoCrud folder

Then open ToDoApp.php

The page should say “Connected successfully” confirming that you application was able to find the database and connect to it.

Next step, insert rows

Every time you access the page in your browser, it inserts a row into the table due to the code below that will be added now.

```
<?php
/*Connect to CRUD Database  mysqli(Server,User,Password,Database)*/
$link = new mysqli('localhost', 'root', '', 'CRUD');
if ($link->connect_error) {
  die("Connection failed: " . $link->connect_error);
} else {echo "<br>Connected successfully"; }
?>
//adding basic CSS color and border to the table, th, and td tags
<style>
table, th, td { border: 1px solid black;
                border-collapse: collapse; }
table th { background-color: black;
           color: white; }
table tr:nth-child(even) { background-color: #eee; }
table tr:nth-child(odd)  { background-color: #fff; }
</style>
<body>

<?php
/*Set Up the SQL statement*/
$sql = "INSERT INTO ToDos (ToDoTitle, ToDoDescription) VALUES ('Test',
'TestDescription');";
/*Exectute the statement, and write the results*/
if (mysqli_query($link, $sql)) {
    echo "<br>New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($link);
}

if (mysqli_error($link)) {
```

```
    echo '<br>Error: ' . mysqli_error($link);
    }else echo '<br>Success';
/*Close database connection*/
  mysqli_close ( $link );
?>
```

To reduce the amount of code we need to write and make it a bit more versatile going forward, let's make a config file and a functions file.

Create a new file titled config.php with the following content:

```
<?php
/*Configuration Settings*/
define('DB_HOST', 'localhost'); /*Database Server*/
define('DB_NAME', 'CRUD'); /*Database Name*/
define('DB_USER', 'root'); /*Database Username*/
define('DB_PWD',  ''); /*Database Password*/
?>
```

Then make an SQLFunctions.php file with the following content:

```
<?php
include('config.php');
/*Opens connection to database with credentials*/
function connectDB() {
    $link = new mysqli(DB_HOST, DB_USER, DB_PWD, DB_NAME);
  if ($link->connect_error) {
    die("Connection failed: " . $link->connect_error);
  }
  /*echo "<br>Connected successfully"; */
  return $link;
}
?>
```

We'll use these going forward to connect to the database.

Create action - Text lecture

Create a new file titled CreateToDo.php, this will be our form to create new to-do's from the browser

```
<HTML>
  <HEAD>
    <TITLE>CreateToDo</TITLE>
    <META http-equiv=Content-Type content="text/html; charset=utf-8">
```

```
    <script type="text/javascript">
      function validateForm(){
        //this is just a placeholder incase we wanted add additional javascript
type validations.
        return true;
      };
    </script>
  </HEAD>
  <BODY>
    <h1>New To-do</h1>
      <form action="CreateToDoSubmit.php" method="POST"
onsubmit='return validateForm()' />
        <p>To-do Title:  <input type="text" name="ToDoTitle"
maxlength='50' required/></p>
        <p>To-Due Date:  <input type="date" name="ToDueDate"></p>
        <p>Description:<br> <textarea cols="100" rows="5"
name="ToDoDescription" maxlength='1000'
required>       </textarea></p>
        <input type="submit">
      </form>
  </BODY>
</HTML>
```

Test it out by visiting the page.

Now that we are using an html form to collect the data, it will submit it to another file we need to create called CreateToDoSubmit.php

Create the file CreateToDoSubmit.php

This file will receive the data from the html form and insert it into the database and then return the user back to the ToDoApp.php page

Insert the following into CreateToDoSubmit.php

```
<?php
include('SQLFunctions.php');
if ( !empty($_POST)) {
  // Store data from html form POST action into variables
  $tdTitle = $_POST['ToDoTitle'];
```

```
  $tdDate  = $_POST['ToDueDate'];
  $tdDescr = $_POST['ToDoDescription'];

/*Open the database connection based on config.php file settings*/
  $link = connectDB();
  /*Prepare the SQL INSERT Statement*/
  $sql = "INSERT INTO ToDos (ToDoTitle, ToDoDescription, ToDueDate,
EntryTS) VALUES ('".$tdTitle."','".$tdDescr."','".$tdDate."', NOW());";
  /*Insert values into the database*/
  if (mysqli_query($link, $sql)) {
  /*    echo "<br>New record created successfully";*/
  } else {
      echo  "<br>Error: " . $sql . "<br>" . mysqli_error($link);
  }
/*Close database connection*/
mysqli_close ( $link );
/*Forward User Back to Main View*/
header("Location: ToDoApp.php");
}

?>
```

Test this out, it should insert rows into the ToDo table, and then reroute the user to the ToDoApp.php

Read - Text lecture

Lets update ToDoApp.php to display all of the rows on the table

```
<?php
include('SQLFunctions.php');
?>
<html>
<!--The Style tag allows us to put some basic css shading and borders to
make the table a little easier to look at. Table, th and td are elements of an
html table.-->
<head>
<style>
```

```
  table, th, td { border: 1px solid black;
                  border-collapse: collapse; }
  table th { background-color: black;
             color: white; }
  table tr:nth-child(even) { background-color: #eee; }
  table tr:nth-child(odd)  { background-color: #fff; }
</style>
</head>
<body>
      <h1>To-do Main View</h1>
      <a href="CreateToDo.php"><button>New To-do</button></a>
<?php
  /*Create the SQL Statement, selecting the four columns were are
interested in*/
  /*format the date to display it easier*/
  $sql="SELECT ToDoTitle
               ,ToDoDescription
               ,DATE_FORMAT(ToDueDate,'%m-%d-%Y')
               ,ToDoID
          FROM ToDos;";
  echo '<br>sql :'.$sql.'<br>Comment this out, after testing<br><br>';

   /*Open the database connection based on config.php file settings*/
  $link = connectDB();
  /*Execute the sql and if there is a result, write out the table headers, then
rows*/
  if ($result = mysqli_query($link,$sql)){
      echo "<table>";
        //header
        echo "<tr>";
          echo "<th>Title</td>";
          echo "<th>Description</td>";
          echo "<th>DueDate</td>";
          echo "<th>Action</td>";
        echo "</tr>";

      //rows, use a while loop to write out each field in the result set.
      //mysqli_fetch_array() separates the results into an array named  $row
```

so

```
    //that each field can be referrenced using $row[x]
    while ($row = mysqli_fetch_array($result))  {
      echo "<tr>";
        echo "<td>{$row[0]}</td>";
        echo "<td>{$row[1]}</td>";
        echo "<td>{$row[2]}</td>";
        echo "<td>Link To Update Page for <br> ToDoID {$row[3]}</td>";
      echo "</tr>";
    }
    echo "</table>";
  }

 /*Close database connection*/
 mysqli_close ( $link );
?>
</body>
</html>
```

We have now Completed “READ”

Update action - Text lecture

Update

Create a new file titled UpdateToDo.php and put the following in it

```
<?php
 include('SQLFunctions.php');
  /*The Read page ToDoApp.php is going to link to this page by sending an html form POST
  with the ToDoID as the only input. We could name it anything. In this case, we named it "q" */
  $q=$_POST["q"];

  /*Open the database connection based on config.php file settings*/
  $link = connectDB();

   /*Create the Sql Statement*/
   $sql = "SELECT ToDoID, ToDoTitle, ToDoDescription, ToDueDate FROM
```

```
ToDos WHERE ToDoID = 7";
   /*$sql = "SELECT ToDoID, ToDoTitle, ToDoDescription, ToDueDate FROM
ToDos WHERE ToDoID =".$q;*/
   /*We will use the hard-coded sql statement for now*/
   echo '<br>sql :'.$sql.'<br>Comment this out, after testing<br><br>';


   /*If the $sql passes validation, exectute it*/
   if($stmt = $link->prepare($sql))
   {
      $stmt->execute();
      /*Assign the results into their respective php variables*/
      $stmt->bind_result($ToDoID, $ToDoTitle, $ToDoDescription,
$ToDueDate);
      while ($stmt->fetch())
      {
        /*reformat the date to html*/
        $newToDueDate = date("Y-m-d", strtotime($ToDueDate));
        echo "<BODY>";
        echo "  <div>";
        echo "    <div>";
        echo "    <h1>Update To-do</h1>";
        /*Create and prepopulate an html form with the values pulled from
the database.*/
        echo "    <form action='UpdateToDoCommit.php' method = 'POST'
onsubmit='' />";
        echo "      <input type='hidden' name='ToDoId'
value='".$ToDoID."'>";
        echo "      <p>To-do Title:  <input text='text' name='ToDoTitle'
maxlength='50'  required value='".$ToDoTitle."'/></p>";
        echo "      <p>To-Due Date:  <input type='date' name='ToDueDate'
value='".$newToDueDate."'></p>";
        echo "      <p>Description:<br> <textarea cols='100' rows='5'
name='ToDoDesr'
maxlength='1000'  required>".$ToDoDescription."</textarea></p>";
        echo "      <input type='submit'> ";
        echo "    </form>";
        echo "    <a href='ToDoApp.php'><button>Cancel</button></a>";
```

```
        echo "    </div>";
        echo "  </div>";
        echo "</BODY>";
      }
   }
   else  {
     echo 'Unable to connect';
     exit();
   }
?>
```

This should look almost exactly like the Create upon testing in the browser

The plan is to query the database, then populate these fields when the file is loaded.
Then when submit is clicked, the row is updated in the database.
To update the database, create a new file titled UpdateToDoCommit.php

```
<?php
include('SQLFunctions.php');
// If there is anything in the POST, store the data from the form into variables
if ( !empty($_POST)) {
  $tdID = $_POST['ToDoId'];
  $tdTitle = $_POST['ToDoTitle'];
  $tdDate  = $_POST['ToDueDate'];
  $tdDescr = $_POST['ToDoDesr'];

  /*Open the database connection based on config.php file settings*/
  $link = connectDB();
  /*Prepare the SQL INSERT Statement*/
  $sql = "UPDATE ToDos
        SET ToDoTitle = '".$tdTitle."'
          ,ToDoDescription = '".$tdDescr."'
          ,ToDueDate ='".$tdDate."'
          ,UpdateTS =  NOW()
        WHERE ToDoId = ".$tdID.";";
  echo $sql."<br>Comment this out, once tested";

  /*Insert values into the database*/
  if (mysqli_query($link, $sql)) {
```

```
    echo "<br>Update record successfully";
  } else {
    echo  "<br>Error: " . $sql . "<br>" . mysqli_error($link);
  }
/*Close database connection*/
mysqli_close ( $link );
/*Forwarded User Back to Main View*/
/*header("Location: ToDoApp.php");  Uncomment this after testing */
}

?>
```

Test it out by submitting an update from UpdateToDo.php

Update ToDoApp.php as follows:
Replace the last echo td with this:

```
        echo "<td><form action='UpdateToDo.php' method = 'POST'
onsubmit='' /> <input type='hidden' name='q' value='".$row[3]."'
/><input type='Submit' value='Update'></form></td>";
```

Also, comment out the echo $SQL

```
<?php
include('SQLFunctions.php');
?>
<html>
<!--The Style tag allows us to put some basic css shading and borders to
make the table a little easier to look at.  Table, th and td are elements of an
html table.-->
<style>
  table, th, td { border: 1px solid black;
                  border-collapse: collapse; }
  table th { background-color: black;
             color: white; }
  table tr:nth-child(even) { background-color: #eee; }
  table tr:nth-child(odd)  { background-color: #fff; }
</style>
  <body>
    <h1>To-do Main View</h1>
    <a href="CreateToDo.php"><button>New To-do</button></a>
```

```
<?php
  /*Create the SQL Statement, selecting the four columns were are
interested in*/
  /*format the date to display it easier*/
  $sql="SELECT ToDoTitle
              ,ToDoDescription
              ,DATE_FORMAT(ToDueDate,'%m-%d-%Y')
              ,ToDoID
          FROM ToDos;";
  /*echo '<br>sql :'.$sql.'<br>Comment this out, after testing<br><br>';*/

   /*Open the database connection based on config.php file settings*/
  $link = connectDB();
  /*Execute the sql and if there is a result, write out the table headers, then
rows*/
  if ($result = mysqli_query($link,$sql)){
      echo "<table>";
        //header
        echo "<tr>";
          echo "<th>Title</td>";
          echo "<th>Description</td>";
          echo "<th>DueDate</td>";
          echo "<th>Action</td>";
        echo "</tr>";

      //rows, use a while loop to write out each field in the result set.
      //mysqli_fetch_array() separates the results into an array named  $row
so
      //that each field can be referrenced using $row[x]
      while ($row = mysqli_fetch_array($result))  {
        echo "<tr>";
          echo "<td>{$row[0]}</td>";
          echo "<td>{$row[1]}</td>";
          echo "<td>{$row[2]}</td>";
          echo "<td><form action='UpdateToDo.php' method = 'POST'
onsubmit='' />  <input type='hidden' name='q' value='".$row[3]."'
/><input type='Submit' value='Update'></form></td>";
        echo "</tr>";
```

```
    }
    echo "</table>";
  }

 /*Close database connection*/
 mysqli_close ( $link );
?>
 </body>
</html>
```

Back on UpdateToDo.php
Comment out the hard coded $sql row, and uncomment the dynamic one.
Test it out by updating a few existing ToDo records.

Delete action - Text lecture
Delete ->

Create a new field titled DeleteToDo.php
This file will take the input from a POST, and delete the corresponding ToDo row.

```
<?php
 include('SQLFunctions.php');
/*if anything is in the Post, assign the $tdID variable with the ID from the post*/
if ( !empty($_POST)) {
 $tdID = $_POST['q'];
  /*Open the database connection based on config.php file settings*/
 $link = connectDB();
 /*Prepare the SQL Delete Statement using the ID from the POST*/
 $sql = "DELETE
        FROM ToDos
        WHERE ToDoId = ".$tdID.";";
 echo "sql:".$sql." Comment this out after testing";

 /*Attempt Delete*/
 if (mysqli_query($link, $sql)) {
    echo "<br>Delete record successfully";
```

```
  } else {
      echo  "<br>Error: " . $sql . "<br>" . mysqli_error($link);
  }
/*Close database connection*/
mysqli_close ( $link );
/*Forwared User Back to Main View*/
/*header("Location: ToDoApp.php");  uncomment this after testing*/
}
?>
```

To Test it, we will need to update ToDoApp.php
Update the while statement on ToDoApp.php and add a delete button under update, ensure you've moved the </td> to the end of the delete button from the update

```
      while ($row = mysqli_fetch_array($result))  {
        echo "<tr>";
          echo "<td>{$row[0]}</td>";
          echo "<td>{$row[1]}</td>";
          echo "<td>{$row[2]}</td>";
          echo "<td><form action='UpdateToDo.php' method = '_POST'
onsubmit='' /> <input type='hidden' name='q' value='".$row[3]."'
/><input type='Submit' value='Update'></form>";
          echo "<form action='DeleteToDo.php' method = '_POST'
onsubmit='' /> <input type='hidden' name='q' value='".$row[3]."'
/><input type='Submit' value='Delete'></form></td>";
        echo "</tr>";
      }
```

Now test it out. You will quickly find that you can delete rows with the touch of a button.

Once tested, Update the DeleteToDo.php
Comment out the echo sql
Uncomment the header Location ToDoApp.php at the end. Test out the functions of the ToDo app and enjoy!