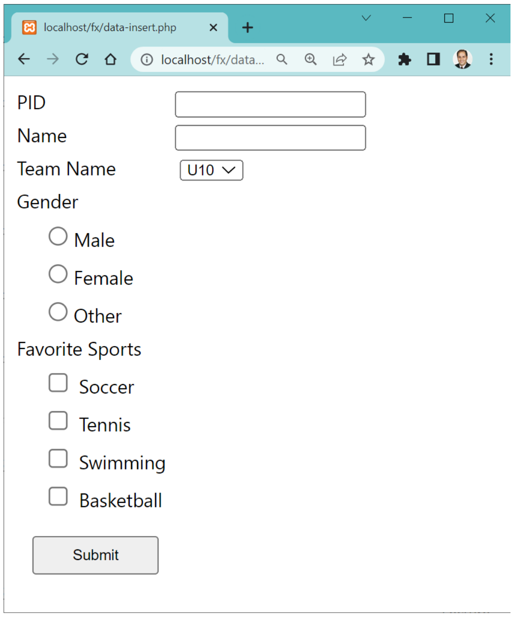
Web Forms, SESSION, and Databases

1. Create a HTML form as you can see in the screenshot below. To make it clear, I have included the HTML code for you as well 😉, All you must do is to type it to learn it!
2. The form has several fields, including PID (Player ID), Name, Team Name, Gender, and Favorite Sports.
3. The form uses the POST method to submit data to the same page ($_SERVER[PHP_SELF]).



PID □

Name □

Team Name U10

Gender

○ Male

○ Female

○ Other

Favorite Sports

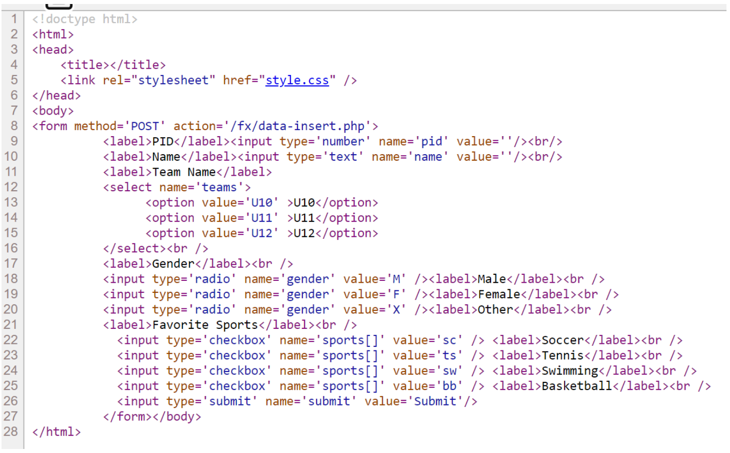
□ Soccer

□ Tennis

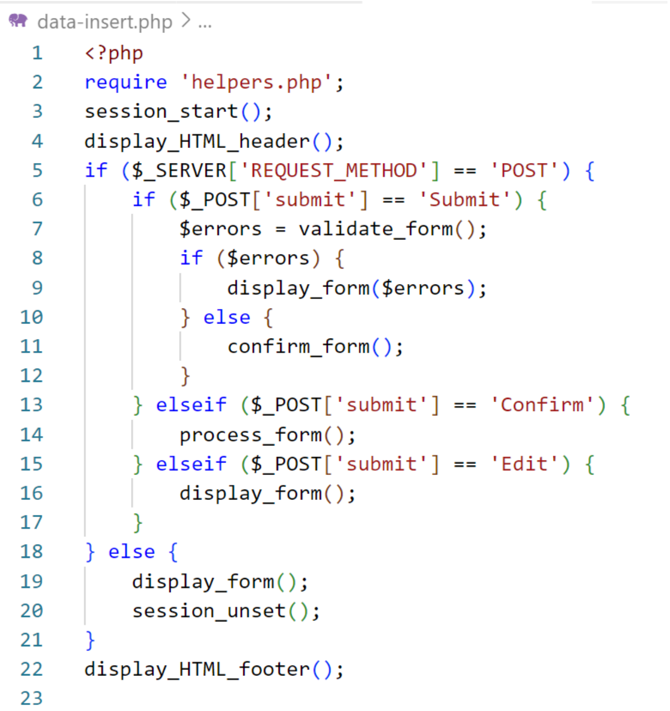
□ Swimming

□ Basketball

Submit



```
<!doctype html>
<html>
<head>
    <title></title>
    <link rel="stylesheet" href="style.css" />
</head>
<body>
<form method='POST' action='/fx/data-insert.php'>
          <label>PID</label><input type='number' name='pid' value=''/><br/>
          <label>Name</label><input type='text' name='name' value=''/><br/>
          <label>Team Name</label>
          <select name='teams'>
                <option value='U10' >U10</option>
                <option value='U11' >U11</option>
                <option value='U12' >U12</option>
          </select><br />
          <label>Gender</label><br />
          <input type='radio' name='gender' value='M' /><label>Male</label><br />
          <input type='radio' name='gender' value='F' /><label>Female</label><br />
          <input type='radio' name='gender' value='X' /><label>Other</label><br />
          <label>Favorite Sports</label><br />
            <input type='checkbox' name='sports[]' value='sc' /> <label>Soccer</label><br />
            <input type='checkbox' name='sports[]' value='ts' /> <label>Tennis</label><br />
            <input type='checkbox' name='sports[]' value='sw' /> <label>Swimming</label><br />
            <input type='checkbox' name='sports[]' value='bb' /> <label>Basketball</label><br />
            <input type='submit' name='submit' value='Submit'/>
          </form></body>
</html>
```



```
<?php
require 'helpers.php';
session_start();
display_HTML_header();
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if ($_POST['submit'] == 'Submit') {
        $errors = validate_form();
        if ($errors) {
            display_form($errors);
        } else {
            confirm_form();
        }
    } elseif ($_POST['submit'] == 'Confirm') {
        process_form();
    } elseif ($_POST['submit'] == 'Edit') {
        display_form();
    }
} else {
    display_form();
    session_unset();
}
display_HTML_footer();

```

4. You are provided with PHP code on the right which contains PHP functions that display a form for input, confirm the form input, and process the form input which is to insert data into a MySQL database.
5. You must use this PHP code and complete the assignment with the required functions in the **helpers.php** file.
6. Start with **display_HTML_header()** function. It has one input which is the title of the HTML page. Set a default value of empty string for the variable title. Then this parameter becomes optional if you do not pass any value to it when you call this function. Look at the HTML code above and write the code for printing all the HTML code up to <body> tag. Use *Heredoc* for this purpose.
7. Write some code for **validate_form()** function. Define an empty array for errors (error messages).
    a. Validate PID. It must be a positive integer (use **filter_input** function of PHP). The user must enter a value for PID. If there is a validation error, set the first element of the error array to the error message.

    For example: `$errors[0] = "PID must be a positive integer.";`

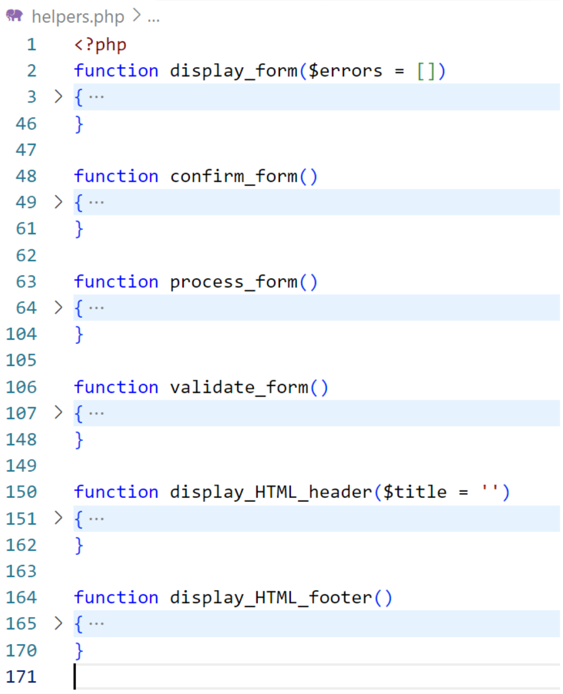
    If there is no error, set the user input as a value in the SESSION array. That is, `$_SESSION['PID'] = $pid;`

b. Validate Name. it must not be an empty string. If it is empty set an error message into $errors[1], otherwise, set the value to the $_SESSION array. That is: `$_SESSION['Name'] = $_POST['name'];`

c. Validate Team Names. The valid team names are: U10, U11, U12. Define an array and set those values in it then using **in_array** function of PHP, check if the value user has entered is among these 3 values. Otherwise, set an error message in **$errors[2]**. In case the input is valid, set a value in `$_SESSION['Team Name'] = $_POST['teams'];`

d. Validate Gender. The valid values are 'F', 'M', 'X'. similar to the team names, define an array, use in_array function to check if the value is among these 3 valid inputs and if it is not a valid input, set an error message in $errors[3] else set `$_SESSION['Gender'] = $_POST['gender'];`

e. Validating favorite sports is like validating Gender and Team Name but it is different as the user may select multiple sports. So, you must use a foreach loop to validate all the options that the user has selected.



```
<?php
function display_form($errors = [])
{⋯
}

function confirm_form()
{⋯
}

function process_form()
{⋯
}

function validate_form()
{⋯
}

function display_HTML_header($title = '')
{⋯
}

function display_HTML_footer()
{⋯
}

```

a. Please note that the user may sunit the form without selecting gender and/or favorite sports. And in these cases your PHP code may throw errors. User **isset** function to check if the user has selected any value for these inputs before submitting the form. For example: `if (!isset($_POST['gender'])) {`

f. At the end of the Validate_form function, return the array of errors.

8. Complete **display_form** function. It has one input parameter which is the **$errors** array. Set the default value of the errors array as an empty array as you can see in the picture below. This way, the input parameter becomes optional. Note in the line 16 in **data-insert.php,** the display_form function is called without passing any input.

   a. Display error messages or the user entered values by writing code like:

```
echo "<form method='POST' action='$_SERVER[PHP_SELF]'>
      <label>PID</label><input type='number' name='pid' value='" . ($_SESSION['PID'] ?? '') . "'/>";


 if (isset($errors[0])) {
        echo "<span class='err'>$errors[0]</span>";
 }
```

The part of the code above that we use $_SESSION (`value='" . ($_SESSION['PID'] ?? '') . "`) is helpful when we want to display the form and display the entered values by the user, in case the user wants to edit their entries before submitting to the server.

You can write similar code for the Name input field of the form.

   b. For the team name you can write for example:

```
<option value='U10' " . (($_SESSION['Team Name'] ?? '') == 'U10' ? 'Selected' : '') . ">U10</option>
```

   c. For Gender you can write:

```
<input type='radio' name='gender' value='M' " . (($_SESSION['Gender'] ?? '') == 'M' ? 'Checked' : '') . "/><label>Male</label><br />
```

And if there is any error:

```
if (isset($errors[4])) {
  echo "<span class='err'>$errors[4]</span>";
}
```

   d. For the favorite sports, you can write:

```
    <input type='checkbox' name='sports[]' value='sc' " . (in_array('sc', ($_SESSION['Favorite
                Sports'] ?? [])) ? 'Checked' : '') . "/> <label>Soccer</label><br />
```

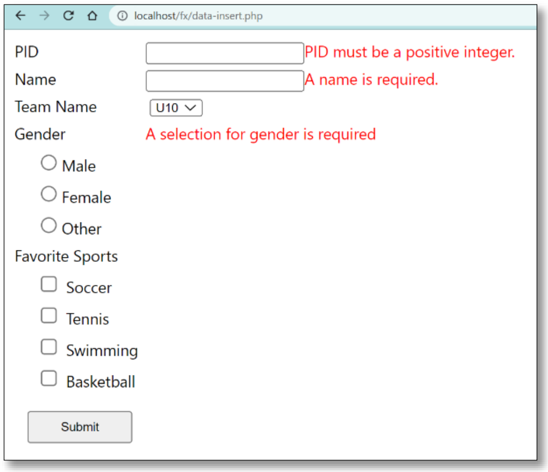
9. Complete **confirm_form** function. Use a foreach to go through the $_SESSION array and display the key and values to the user to get a confirmation from the user. For example:

   ```
   foreach ($_SESSION as $key => $value) {
   ```

   you can check also to find if any of the $value is an array. For example, Favorite Sports is an array, and you can use **implode** function to display an array . You can use **is_array** function to detect if a $value is an array.

   You want to let the user either Confirm or Edit the values. You can define a form with only two buttons:

   ```
    echo "<form method='POST' action='$_SERVER[PHP_SELF]'>
           <input type='submit' name='submit' value='Confirm'/>
            <input type='submit' name='submit' value='Edit'/>
          </form>";
   ```



localhost/fx/data-insert.php

PID [ ] PID must be a positive integer.

Name [ ] A name is required.

Team Name [U10]

Gender A selection for gender is required

○ Male

○ Female

○ Other

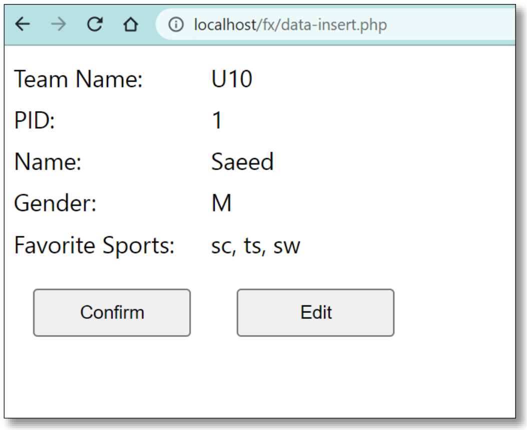
Favorite Sports

□ Soccer

□ Tennis

□ Swimming

□ Basketball



localhost/fx/data-insert.php

Team Name:	U10
PID:	1
Name:	Saeed
Gender:	M
Favorite Sports:	sc, ts, sw

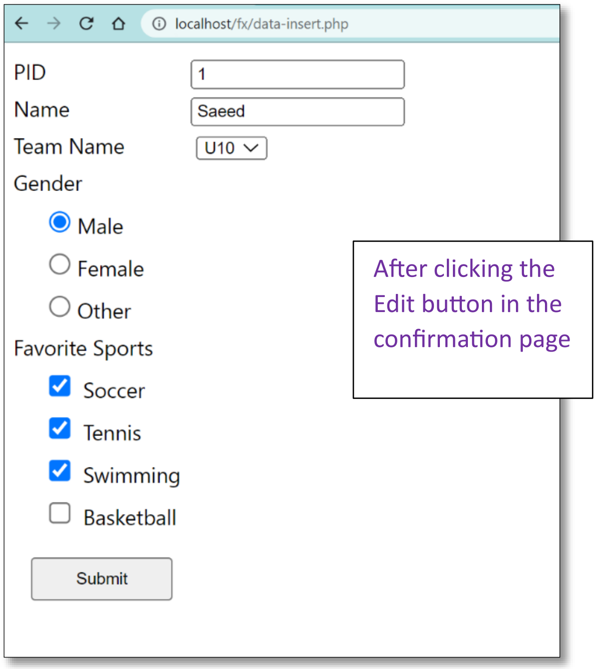
10. The process_form() function connects to a MySQL database using PDO (PHP Data Objects), creates a new database called players_db if it doesn't exist, and creates two tables: Player and Player_FavSports. The process_form() function then inserts the form data into the Player table and the selected favorite sports into the Player_FavSports table.

    First create a connection to the database using PDO.

```
    try {
         $conn = new PDO("mysql:host=localhost", "root", "");
         $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
     } catch (PDOException $e) {
         echo "<p class='err'>Error: " . $e->getMessage()
. "</p>";
     }
```

Then if the connection to the database was successful, execute your SQL statements:

```
    try {
        $sql = "CREATE DATABASE IF NOT EXISTS players_db;
                USE players_db;
                CREATE TABLE IF NOT EXISTS Player (
                    PID INT PRIMARY KEY,
                    PName VARCHAR(20),
                    TeamName CHAR(3),
                    Gender CHAR(1)
                );
                CREATE TABLE IF NOT EXISTS Player_FavSports (
```



localhost/fx/data-insert.php

PID [1]

Name [Saeed]

Team Name [U10]

Gender

● Male

○ Female

○ Other

Favorite Sports

☑ Soccer

☑ Tennis

☑ Swimming

□ Basketball

After clicking the Edit button in the confirmation page

```
                PID INT,
                FavSport CHAR(2),
                PRIMARY KEY(PID, FavSport),
                FOREIGN KEY(PID) REFERENCES Player(PID)
            );";
    $conn->exec($sql);
} catch (PDOException $e) {
    echo "<p class='err'>Error: " . $e->getMessage() . "</p>";
}
```

Then insert values to the database:

```
try {
    $insert = "INSERT INTO players_db.Player VALUES (?,?,?,?);";
    $stmt = $conn->prepare($insert);
    $stmt->execute(array($_SESSION['PID'], $_SESSION['Name'], $_SESSION['Team Name'],
   $_SESSION['Gender']));
    foreach ($_SESSION['Favorite Sports'] as $sport) {
        $insert = "INSERT INTO players_db.Player_FavSports VALUES (?,?);";
        $stmt = $conn->prepare($insert);
        $stmt->execute(array($_SESSION['PID'], $sport));
    }
    echo "<p style='color:green'>Data Inserted Successfully.</p>";
} catch (PDOException $e) {
    echo "<p class='err'>Error: " . $e->getMessage() . "</p>";
}
```

11. Test the form by submitting valid and invalid data, and verify that the data is inserted correctly into the database.
12. Document your code with comments to explain the logic and functionality of each function and block of code.
13. Do not forget to write your full name as a comment on top of ALL files that you submit.