pathmanathan sayanthan

22845173

TASk-2

Table of Contents

[Existing Users 2](#_Toc136530511)

[Login.php 3](#_Toc136530512)

[Signup.php 4](#_Toc136530513)

[Style.css 5](#_Toc136530514)

[Validating.php 7](#_Toc136530515)

[courseSelectionform.php 9](#_Toc136530516)

[courseSelection.css 11](#_Toc136530517)

[courseSelection.js 14](#_Toc136530518)

[newCourse.php 18](#_Toc136530519)

[newCourse.js 21](#_Toc136530520)

[Layout.css 23](#_Toc136530521)

[Report.php 29](#_Toc136530522)

[Report.js 31](#_Toc136530523)

[Report.css 37](#_Toc136530524)

[mainFunctions.php 39](#_Toc136530525)

[Functions.php 44](#_Toc136530526)

[dbInstance.php 44](#_Toc136530527)

[Pdo.php 44](#_Toc136530528)

[Test Cases 48](#_Toc136530529)

# Existing Users

|  |  |
| --- | --- |
| **Username** | **Password** |
| sayan | 12 |
| meshi | 34 |

Only registered users can able to login and add and view report

To register new user we can able to use this link below so we can avoid to register new users.

* <http://localhost/signup.php?op=1&username=meshi&password=34>

# Login.php

<?php

session\_start();

session\_destroy();

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    include 'validating.php'; // include the PHP file that processes the form

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel='stylesheet' type='text/css' media='screen' href='style.css'>

    <title>Courses</title>

</head>

<body>

    <div class="main">

        <label>Login</label>

        <section id="form">

            <form action="validating.php?op=1" method="POST">

                <div class="input\_data">

                    <label for="username">Username:</label>

                    <input type="text" id="username" name="username"/>

                </div>

                <div class="input\_data">

                    <label for="password">Password:</label>

                    <input type="password" id="password" name="password"/>

                </div>

                <div>

                    <input class="Btns" type="submit" name="btn" value="Login"/>

                </div>

                <div>

                    <label for="message"><?php

                         // Receiving data

                        if(isset( $\_SESSION['error'])){

                            //if there is error meessge is set echo the error message inside label

                            echo("Username or Password is not valid!");

                        }

                    ?></label>

                </div>

            </form>

        </section>

</div>

</body>

</html>

# Signup.php

<?php

    require\_once './PDO/functions.php';//require functions.php to process further

    $response = array();// store the response values

        //check username and password value has been set

        if(isset($\_GET['username']) && !empty($\_GET['username']) && !empty($\_GET['password'])){

            $username=$\_GET['username'];//get username

            echo($username);

            $result=fetchARecordWithOneWhereClause("Users\_tbl","username",$username);//get the values if the username already exist

            if($result){//if its already exist

                    $response['error'] = true;

                    $response['message'] = "user already exist";//throw error message

            }

            else{//if user not exist already assign values

                //assign values to data array

                $data = [

                    "username" => $\_GET['username'],

                    "password" => hashPass($\_GET['password']),

                    ];

                    $result = insert("Users\_tbl", $data);//call insert function to insert the user details

                    if($result){//if its success

                        $response['error'] = false;//error is false

                        $response['message'] = true;//message is false

                    }

                    else{

                        $response['error'] = false;//error is false

                        $response['message'] = false;//message is false

                    }

            }

        }else{

            $response['error'] = true;//error is true

            $response['message'] = 'Required Parameters are missing';//set the message required parameter is missing

        }

    echo(json\_encode($response));

?>

Style.css  
  
\*{

    border: 0;

    margin: 0;

    box-sizing: border-box;

}

body{

    background-repeat: no-repeat;

    background-size: cover;

    background-attachment: fixed;

    display: flex;

    justify-content: center;

    align-items: center;

    overflow: auto;

    min-height: 100vh;

}

.main{

    display: flex;

    flex-direction: column;

    background-color: #ffff;

    text-align: center;

    align-items: center;

    position: relative;

    max-width: 300px;

    min-height: 200px;

    padding: 10px;

    box-shadow: rgba(0, 0, 0, 0.25) 0px 54px 55px, rgba(0, 0, 0, 0.12) 0px -12px 30px, rgba(0, 0, 0, 0.12) 0px 4px 6px, rgba(0, 0, 0, 0.17) 0px 12px 13px, rgba(0, 0, 0, 0.09) 0px -3px 5px;

    border-radius: 10px;

}

.main form{

    justify-content: space-between;

    margin-top: 40px;

}

.input\_data{

    margin-top: 10px;

    margin-bottom: 10px;

    padding: 5px;

    border: 1px solid rgb(36, 40, 42);

    border-radius: 5px;

    height: 100%;

}

.input\_data input{

    outline: none;

}

/\*Login button design\*/

.Btns {

  background: #0f0f0f;

  border-radius: 999px;

  box-shadow: #13134a 0 10px 20px -10px;

  box-sizing: border-box;

  color: #FFFFFF;

  cursor: pointer;

  font-family: Inter,Helvetica,"Apple Color Emoji","Segoe UI Emoji",NotoColorEmoji,"Noto Color Emoji","Segoe UI Symbol","Android Emoji",EmojiSymbols,-apple-system,system-ui,"Segoe UI",Roboto,"Helvetica Neue","Noto Sans",sans-serif;

  font-size: 15px;

  font-weight: 300;

  line-height: 20px;

  opacity: 1;

  outline: 0 solid transparent;

  padding: 8px 18px;

  user-select: none;

  -webkit-user-select: none;

  touch-action: manipulation;

  width: fit-content;

  word-break: break-word;

  border: 0;

}

# Validating.php

<?php

if(isset($\_POST['btn'])){//if login btn is pressed

            // Set the content type to JSON

    header('Content-Type: application/json');

    require\_once './PDO/functions.php';//require functions.php to connect with db

    $response = array();

    if(isset($\_GET['op'])){//check op has been set

    if(isset($\_POST['username'])){//check username has been set

        if(isset($\_POST['password'])){//check password has been set

            $username=$\_POST['username'];

            $password=$\_POST['password'];

            $result=fetchARecordWithOneWhereClause("Users\_tbl","username",$username);//get the details related to username from the Users\_tbl

            if($result){//if response is success

                $response['error'] = false;//make the error as false

                if(count(json\_decode(json\_encode($result)))==1){//and check wehther its only returns one record

                    //json encode will convert array to json string the decode the results to PHP associative array

                    $decodedVal=json\_decode(json\_encode($result),true);

                    if(passwordVerify($password,$decodedVal[0]['password'])){//verfiy the user inputed password and hashed password

                        $response['verified'] = true;//set verified as true

                        $response['user\_id']=$decodedVal[0]['user\_id'];//get user id

                        $response['username']=$decodedVal[0]['username'];//get username

                    }else

                    {

                        $response['verified'] = false;//set verified as false

                    }

                }

                else{

                    $response['verified'] = false;//set verified as false

                }

            }

            else{

                $response['error'] = true;

                $response['message'] = 'Unable to load data?data='.$response;//set error message if unable to load data from db

            }

        }

    }

    else{

        $response['error']=true;

        $response['message']="Required Parameters are missing";//set error message if required parameters are missing username and password

    }

        }else{

        $response['error']=true;

            $response['message']="Required Parameters are missing";//set error message if required parameters are op

        }

}

                // Send the response back to the client

                if($response['verified']===true){//if the username and password is verfied

                        session\_start();//start session

                        $\_SESSION['userData'] = $response;//store the userdata

                        header('Location: courseSelectionform.php');//navigate to courseSelectionform.php page

                        exit;

                    }

                    else{

                        session\_start();//start the session

                        $\_SESSION['error'] = $response;//store error response

                        header('Location: login.php');//navigate to login.php page

                        exit;

                    }

?>

# courseSelectionform.php

<?php

    session\_start();//start the session

    if(isset($\_SESSION['userData'])){//user data has been set load the page

        //load the page

    }

    else{//else navigate to login page

        header('Location: login.php');

        exit;

    }

?>

<!DOCTYPE html>

<html>

    <head>

        <title>Course List</title>

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

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

        <script src="https://code.jquery.com/jquery-3.4.1.min.js"

        integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo=" crossorigin="anonymous"></script>

        <script src='courseSelection.js' defer></script>

    </head>

    <body>

        <header>

            <h3>CSYM019 - UNIVERSITY COURSES</h3>

        </header>

        <nav>

            <ul>

                <li><a href="./courseSelectionForm.php">Course Report</a></li>

                <li><a href="./newCourse.php">New Course</a></li>

            </ul>

        </nav>

        <main>

            <h3>Course Selection Form</h3>

                <form action="report.php" method="post" class="addmore" target="\_blank">

                    <div class="table-container">

                            <table id="course\_Tbl">

                                <thead>

                                    <tr>

                                        <th style="display: flex; align-items: center; width:200px;">

                                            <input type="checkbox" id="selectAll" onclick="selectUnselctCheckboxes()" style="margin-right: 5px;">

                                            <label for="selectAll">Select All</label>

                                        </th>

                                        <th>Title</th>

                                        <th>Level</th>

                                        <th>Overview</th>

                                        <th>Location</th>

                                        <th>Month</th>

                                        <th>Fees Uk</th>

                                        <th>Fees International</th>

                                        <th>Entry Requirements</th>

                                        <th>Update</th>

                                        <th>Delete</th>

                                        <!-- Add more columns as needed -->

                                    </tr>

                                </thead>

                                <tbody id="courseDetailsBody">

                                </tbody>

                            </table>

                    </div>

                    <div class="inner\_btn">

                        <input type="submit" value="Create Report" class="newCourseBtns" name="report\_btn"/>

                    </div>

                </form>

                <div id="popup-container">

                    <div id="details">

                        <input type="text" id="courseTitle" placeholder="title" required></input>

                        <textarea id="overview" name="overview" rows="4" cols="30" required></textarea>

                        <input type="text" id="level" placeholder="level" required></input>

                        <input type="text" id="month" placeholder="month" required></input>

                        <input type="text" id="location" placeholder="location" required></input>

                        <input type="text" id="UK" placeholder="uk fee" required></input>

                        <input type="text" id="International" placeholder="internatinal fee" required></input>

                    </div>

                    <div class="pop\_up\_btn\_container">

                        <button class="edit" id="submit-button">Submit</button>

                        <button class="edit" id="close-button">Close</button>

                    </div>

                </div>

        </main>

        <footer>&copy; CSYM019 2023</footer>

    </body>

</html>

<?php

    if (isset($\_SESSION['deletesuccess'])) {

        $Message = $\_SESSION['deletesuccess'];

        if($Message){

            echo "<script>alert('Course deleted successfully');</script>";

        }

        else{

            echo "<script>alert('Unable to delete the course at the moment');</script>";

        }

        unset($\_SESSION['deletesuccess']); // Clear the session variable after displaying the alert

    }

?>

# courseSelection.css

main{

    display: flex;

    justify-content: center;

    flex-direction: column;

    align-items: center;

}

.addmore{

    display: grid;

    height: 100vh;

}

.table-container {

    max-height: 600px; /\* Set the maximum height of the table container \*/

    overflow-y: auto; /\* Enable vertical scrolling \*/

    justify-content: center;

    align-items: center;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);

  }

table{

    border-collapse: collapse;

    box-shadow: 0 5px 10px gray;

    background-color: white;

    text-align: left;

    overflow: hidden;

}

thead{

    box-shadow: 0 5px 10px gray;

}

th{

    padding: 1rem 2rem;

    text-transform: uppercase;

    letter-spacing: 0.1rem;

    font-size: 0.7rem;

    font-weight: 700;

}

td{

    padding: 1rem 2rem;

}

.table-cell-title {

    display: inline-block;

    position: relative;

    width: 300px;

    height: 50px; /\* Initial height \*/

    white-space: nowrap;

    overflow: hidden;

    text-overflow: ellipsis;

    vertical-align: top;

    transition: width 0.3s, height 0.3s; /\* Add transition for smooth animation \*/

  }

  .table-cell-title:hover {

    z-index: 1;

  width: 300px;

  height: auto;

  background-color: #9be6f73a;

  overflow: visible;

  white-space: normal;

}

.table-cell {

  display: inline-block;

  position: relative;

  width: 200px;

  height: 50px; /\* Initial height \*/

  white-space: nowrap;

  overflow: hidden;

  text-overflow: ellipsis;

  transition: width 0.3s, height 0.3s; /\* Add transition for smooth animation \*/

}

.table-cell:hover {

    z-index: 1;

  width: 200px;

  height: auto;

  background-color: #9be6f73a;

  overflow: visible;

  white-space: normal;

}

.edit{

    border-radius: 5px;

    background-color: black;

    padding: 0.2rem 1rem;

    text-align: center;

    color: white;

    cursor: pointer;

}

tr:nth-child(even){

    background-color: aliceblue;

}

#popup-container{

  display: none;

  position: fixed;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

  background-color: #fff;

  border: 1px solid #ccc;

  padding: 20px;

  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.3);

  min-width: 500px;

  min-height: 300px;

  overflow: auto;

  z-index: 1;

}

.pop\_up\_btn\_container{

  display: flex;

  flex-direction: row;

  justify-content: center;

}

#details{

  display: flex;

  flex-direction: column;

  justify-content: center;

}

#details input[type="text"],

#details textarea {

  padding: 5px;

  border-radius: 0;

  margin-bottom: 5px;

  -webkit-appearance: none;

  -moz-appearance: none;

}

# courseSelection.js

//load courses

let jsonVal;

//callBackFun function to request data from db every 200 milliseconds

$(document).ready(function () {

    //sending ajax request to mainFunctions.php to get allcourses details by assiging the actio value as allCourses

    $.ajax({

        url: 'mainFunctions.php',

        dataType: 'json',

        type: "POST",

        data: { action: 'allCourses' },

        success: function (response) {

            console.log(response);//for debugging purorse

            //call getDetails function to assign values to table body

            getDetails(response);

        },

        error: function (xhr, status, error) {

            console.log('Error:', error, " ", xhr.responseText);

        }

    });

});

//get the requiremnts for specific course

function getRequirements(id, callback) {

    var entryRequirement = "";//store the entryRequirement Value

    //sending ajax request to mainFunctions.php to get requirement details by assiging the action value as getRequirements and courseId

    $.ajax({

        url: 'mainFunctions.php',

        dataType: 'json',

        type: "POST",

        data: {

            action: 'getRequirements',

            courseid: id

        },

        success: function (requirementsResponse) {

            //loop through the requirementsResponse

            for (let x = 0; x < requirementsResponse.length; x++) {

                //assign value to entryRequirement as <Li>requirement</Li>

                entryRequirement += '<li>' + requirementsResponse[x].requirements + '</li>';

            }

            //callback is used to get the entry requirement value

            callback(entryRequirement);

        },

        error: function (xhr, status, error) {

            console.log('Error:', error, " ", xhr.responseText);

        }

    });

}

//distribute the reponse of json to table values

function getDetails(response) {

    var html = "";//store the html body value

    //make the table in ascending order

    response.sort((a, b) => a.title > b.title ? 1 : -1);

    let table = document.getElementById("course\_Tbl");//get course table by id

    var requirements = "";//store the requirement for the course

    //check whether number of rows and json response are equal otherwise update it

    if (response.length != (table.rows.length) - 1) {

        document.getElementById("courseDetailsBody").innerHTML = "";//get course table body by id

        for (let i = 0; i < response.length; i++) {//loop through the response

            //call getRequirements function to get requirement for the specific id

            getRequirements(response[i].id, function (entryRequirement) {

                $('#req-' + response[i].id).html(entryRequirement)//assign the valuue to the specific cell in the tabel using id

            });

            //create table body

            html += '<tr><td><input type="checkbox" name="checkbox[]" value="' + response[i].id + '"></td>';

            html += '<td><div class="table-cell-title">' + response[i].title + '</td>';

            html += '<td>' + response[i].type + '</td>';

            html += '<td ><div class="table-cell">' + response[i].overview + '</div></td>';

            html += '<td >' + response[i].location + '</td>';

            html += '<td >' + response[i].start\_month + '</td>';

            html += '<td >' + response[i].fees\_UK + '</td>';

            html += '<td >' + response[i].fees\_international + '</td>';

            html += '<td <div class="table-cell" id="req-' + response[i].id + '"><ul>' + requirements + '</ul></td>';

            html += '<td ><button  type="button" class="edit" value="' + response[i].id + '" name="course\_update" onclick="updateDetails(' + response[i].id + ')">Update</button></td>';

            html += '<td ><button class="edit" value="' + response[i].id + '" name="course\_delete">Delete</button></td></tr>';

            //assign the innser htm to the course body

            document.getElementById("courseDetailsBody").innerHTML = html;

        }

    }

};

//select and unselect all check boxes

function selectUnselctCheckboxes() {

    var selectAllCheckbox = document.getElementById('selectAll');//get selct all check box is checked or unchecked

    var checkboxes = document.getElementsByName('checkbox[]');//get all check boxes available

    for (var i = 0; i < checkboxes.length; i++) {//loop through all check boxes

        checkboxes[i].checked = selectAllCheckbox.checked;//check or uncheck it based on the selectAllCheckbox value

    }

}

//update data

function updateDetails(CourseId) {

    $.ajax({

        url: 'mainFunctions.php',

        dataType: 'json',

        type: "POST",

        data: {

            action: 'specificCourse',

            courseid: CourseId

        },

        success: function (Response) {

            $('#courseTitle').val(Response[0].title);//course title vale

            $('#overview').val(Response[0].overview);//overview value

            $('#level').val(Response[0].type);//level value

            $('#month').val(Response[0].start\_month);//month value

            $('#location').val(Response[0].location);// location value

            $('#UK').val(Response[0].fees\_UK);//uk fee valule

            $('#International').val(Response[0].fees\_international);//internation fee value

            //enable the pop up

            $('#popup-container').css({

                'display': 'block',

            });

        },

        error: function (xhr, status, error) {

            console.log('Error:', error, " ", xhr.responseText);

        }

    });

    //submit btn

    $('#submit-button').click(function () {

        var title = $('#courseTitle').val();

        var overview = $('#overview').val();

        var level = $('#level').val();

        var month = $('#month').val();

        var location = $('#location').val();

        var uk = $('#UK').val()

        var in\_fee = $('#International').val();

        //check all the details have the valid input without null value

        if (title.trim().length > 0 && overview.trim().length > 0 && level.trim().length > 0 && month.trim().length > 0 && location.trim().length > 0 && uk.trim().length > 0 && in\_fee.trim().length > 0) {

            $.ajax({

                url: 'mainFunctions.php',

                type: "POST",

                data: {

                    action: 'updateCourse',

                    courseid: CourseId,

                    title: title,

                    overview: overview,

                    type: level,

                    start\_month: month,

                    fees\_UK: uk,

                    fees\_international: in\_fee

                },

                success: function (Response) {

                    if (Response) {

                        alert("Succesfully updated", 200);//alert successful message if update wass successful

                    }

                    else {

                        alert("Unable to update course details", 200);//alert fail message if update was unsuccessful

                    }

                    //disable the update container

                    $('#popup-container').css({

                        'display': 'none',

                    });

                },

                error: function (xhr, status, error) {

                    console.log('Error:', error, " ", xhr.responseText);

                }

            });

        }

        else {

            alert("Please provide valid values!")

        }

    })

    //close btn

    $('#close-button').click(function () {

        //disable the update contaimer by disabling the diaplay field

        $('#popup-container').css({

            'display': 'none',

        });

    })

}

# newCourse.php

<?php

    session\_start();

    if(isset( $\_SESSION['userData'])){

            //load the page

    }

    else{

            header('Location: login.php');

            exit;

    }

    if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

        include 'mainFunctions.php'; // include the PHP file that processes the form

    }

?>

<!DOCTYPE html>

<html>

    <head>

        <title>Course Report</title>

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

        <script src="https://code.jquery.com/jquery-3.4.1.min.js"

        integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo=" crossorigin="anonymous"></script>

        <script src='newCourse.js' defer></script>

    </head>

    <body>

        <header>

            <h3>CSYM019 - UNIVERSITY COURSES</h3>

        </header>

        <nav>

            <ul>

                <li><a href="./courseSelectionForm.php">Course Report</a></li>

                <li><a href="./newCourse.php">New Course</a></li>

            </ul>

        </nav>

        <main>

            <h3 id="new\_course\_h3">New Course Entry Form</h3>

            <form method="post" action="./mainFunctions.php">

                <div class="addmore">

                    <div class="inner\_Blocks">

                        <label for="title">Title:</label>

                        <input type="text" id="title" name="title" required>

                    </div>

                    <div class="inner\_Blocks">

                        <label for="overview">overview:</label>

                        <textarea id="overview" name="overview" rows="4" cols="30" required></textarea>

                    </div>

                    <div class="inner\_Blocks">

                        <label for="level">Level:</label>

                        <label>

                            <input type="radio" name="level" value="Undergraduate" id="rd\_btn\_1" checked>

                            Undergraduate

                        </label>

                        <label>

                            <input type="radio" name="level" value="Postgraduate" id="rd\_btn\_2">

                            Postgraduate

                        </label>

                    </div>

                    <div class="inner\_Blocks">

                        <label for="level">Location:</label>

                        <label>

                            <input type="radio" name="location" value="Waterside" id="location\_rd\_btn\_1" checked>

                            Waterside

                        </label>

                        <label>

                            <input type="radio" name="location" value="online" id="location\_rd\_btn\_2">

                            online

                        </label>

                    </div>

                    <div class="inner\_Blocks">

                        <label for="selectMonth">Starting Month:</label>

                        <select id="selectMonth" name="selectMonth">

                            <option value="January">January</option>

                            <option value="April">April</option>

                            <option value="Septemper">Septemper</option>

                        </select>

                    </div>

                    <div class="inner\_Blocks\_col">

                        <label>Course Contents:</label><br>

                        <div id="modulesContainer">

                            <div class="inner\_Blocks\_content">

                                <label for="module">- Modules:</label><br>

                                <input type="text" id="module" name="module[]" required placeholder="module">

                                <input type="number" id="credit" name="credit[]" required placeholder="credit">

                                <button type="button" onclick="addModules()" >Add Module</button>

                            </div>

                        </div>

                    </div>

                    <div class="inner\_Blocks\_col">

                        <label>Entry Requirements:</label><br>

                            <div id="requirementContainer">

                                <div class="inner\_Blocks\_content">

                                    <label for="requirement"></label><br>

                                    <textarea id="requirement" name="requirement[]" rows="4" cols="30" required></textarea>

                                    <button type="button" onclick="addRequirement()" id="addRequirementBtn">Add Requirement</button>

                                </div>

                            </div>

                    </div>

                    <div class="inner\_Blocks">

                        <label>Fee and Funding:</label>

                        <div id="feesContainer">

                                <label for="uk">UK:</label><br>

                                <input type="text" id="uk" name="uk" required placeholder="UK"><br>

                                <label for="international">International:</label><br>

                                <input type="text" id="international" name="international" required placeholder="international">

                        </div>

                    </div>

                    <div class="inner\_Blocks\_col">

                        <label>FAQs:</label><br>

                            <div id="faqContainer">

                                <div class="inner\_Blocks\_content">

                                    <label></label><br>

                                    <input type="text" id="question" name="question[]" required placeholder="question">

                                    <input type="text" id="answer" name="answer[]" required placeholder="answer">

                                    <button type="button" onclick="addFaq()" >Add FAQ</button>

                                </div>

                            </div>

                    </div>

                    <div class="inner\_btn">

                        <input type="submit" value="Add Course" class="newCourseBtns" name="btn"/>

                    </div>

            </form>

        </main>

        <footer>&copy; CSYM019 2023</footer>

    </body>

</html>

<?php

if (isset($\_SESSION['error'])) {

    $errorMessage = $\_SESSION['error'];

    echo "<script>alert('$errorMessage');</script>";

    unset($\_SESSION['error']); // Clear the session variable after displaying the alert

}

if (isset($\_SESSION['success'])) {

    $Message = $\_SESSION['success'];

    if($Message){

        echo "<script>alert('Course Registered Successfully');</script>";

    }

    else{

        echo "<script>alert('Unable to register the course at the moment');</script>";

    }

    unset($\_SESSION['success']); // Clear the session variable after displaying the alert

}

?>

# newCourse.js

function addModules(){

    //get modules container

    var container=document.getElementById("modulesContainer");

    //create div and assign class

    var div = document.createElement("div");

      div.classList.add("inner\_Blocks\_content");

      //create label

      var label = document.createElement("label");

      label.textContent = "";

      div.appendChild(label);

      div.appendChild(document.createElement("br"));

      //create input requirement name field

      var module = document.createElement("input");

      module.type = "text";

      module.name = "module[]";

      module.required = true;

      module.placeholder = "module";

      div.appendChild(module);

      //create credit field

      var credit = document.createElement("input");

      credit.type = "number";

      credit.name = "credit[]";

      credit.required = true;

      credit.placeholder = "credit";

      div.appendChild(credit);

      // create delete button if we user want to delete those fields

      var moduledeleteButton = document.createElement("button");

      moduledeleteButton .type = "button";

      moduledeleteButton .textContent = "Delete";

      moduledeleteButton .addEventListener("click", function() {

        div.remove();

      });

      div.appendChild(moduledeleteButton );

      container.appendChild(div);

}

//add requirement function

function addRequirement(){

    //get requirements container

    var container=document.getElementById("requirementContainer");

    //create div and assign class

    var div = document.createElement("div");

      div.classList.add("inner\_Blocks\_content");

      //create label

      var label = document.createElement("label");

      label.textContent = "";

      div.appendChild(label);

      div.appendChild(document.createElement("br"));

      //create input requirement field

      var requirement = document.createElement("textarea");

      requirement.name = "requirement[]";

      requirement.required = true;

      requirement.placeholder = "requirement";

      requirement.rows=4,

      requirement.cols=30

      div.appendChild(requirement);

      // create delete button if we user want to delete those fields

      var requirementdeleteButton = document.createElement("button");

      requirementdeleteButton.type = "button";

      requirementdeleteButton.id="addRequirementBtn"

      requirementdeleteButton.textContent = "Delete";

      requirementdeleteButton.addEventListener("click", function() {

        div.remove();

      });

      div.appendChild(requirementdeleteButton);

      container.appendChild(div);

}

//add new faqs

function addFaq(){

        //get requirements container

        var container=document.getElementById("faqContainer");

        //create div and assign class

        var div = document.createElement("div");

          div.classList.add("inner\_Blocks\_content");

          //create label

          var label = document.createElement("label");

          label.textContent = "";

          div.appendChild(label);

          div.appendChild(document.createElement("br"));

        //create input question field

        var question = document.createElement("input");

        question.type = "text";

        question.name = "question[]";

        question.required = true;

        question.placeholder = "question";

        div.appendChild(question);

        //create answer field

        var answer = document.createElement("input");

        answer.type = "text";

        answer.name = "answer[]";

        answer.required = true;

        answer.placeholder = "answer";

        div.appendChild(answer);

          // create delete button if we user want to delete those fields

          var deleteButton = document.createElement("button");

          deleteButton.type = "button";

          deleteButton.textContent = "Delete";

          deleteButton.addEventListener("click", function() {

            div.remove();

          });

          div.appendChild(deleteButton);

          container.appendChild(div);

}

# Layout.css

body {

    display: grid;

    grid-template-columns: 15% 85%;

    grid-template-rows: auto;

    grid-template-areas: "header header"

                            "nav main"

                            "footer footer";

}

header {

    background-color: #000000;

    color: white;

    padding: 10px;

    grid-area: header;

}

nav {

    background-color: rgba(138, 155, 190, 0.173);

    color: black;

    grid-area: nav;

    box-shadow: 0 5px 10px gray;

}

nav > ul {

    list-style-type: none;

    margin: 0;

    padding: 0;

    border-right: 2%;

    border-color: rgba(138, 155, 190, 0.173);

    border-style: solid;

    /\*width: 10%;\*/

    background-color: rgba(138, 155, 190, 0.173);

  }

nav > ul li a {

    display: block;

    color: #000;

    padding: 8px 16px;

    text-decoration: none;

}

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

nav > ul li a:hover {

    background-color: #555;

    color: whitesmoke;

}

main {

    background-color: rgba(245, 245, 245, 0.316);

    color: black;

    margin-left: 1em;

    padding-left: 1em;

    grid-area: main;

}

footer {

    padding-top: 2em;

    background-color: #000000;

    color: white;

    grid-area: footer;

}

.note {

    color: red;

    font-size: 12pt;

    font-family: Arial, Helvetica, sans-serif;

    padding-top: 1em;

}

.blueNote {

    color: blue;

    font-size: 12pt;

    font-family: Arial, Helvetica, sans-serif;

    padding-top: 1em;

    font-weight: bold;

}

.sketch img {

    width: 60%;

    height: auto;

}

.addmore{

    display: flex;

    flex-direction: column;

}

#new\_course\_h3{

    font-size: 25px;

    font-weight: 500;

    position: relative;

}

#new\_course\_h3::before{

    content: ' ';

    position: absolute;

    left: 0;

    bottom: 0;

    height: 3px;

    width: 30px;

    background: rgb(2,0,36);

    background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(9,9,121,1) 35%, rgba(0,212,255,1) 100%);

}

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

.addmore > input:hover {

    display:block;

    float: left;

    margin: 0 2em 2em 2em;

    padding: 0.5em 5em;

    text-decoration: none;

    background-color: #555;

    color: whitesmoke;

}

form{

  padding: 5px;

  background-color: whitesmoke;

  border-radius: 4px;

  border-style: none;

  box-sizing: border-box;

  box-shadow: 0 5px 10px gray;

}

/\*inner\_Blocks\*/

.inner\_Blocks {

    display: flex;

    flex-direction: row;

    margin-bottom: 15px;

    justify-content: flex-start;

  }

  .inner\_Blocks\_content{

    display: flex;

    flex-direction: row;

    margin-bottom: 15px;

    justify-content: flex-start;

    margin-left: 20px;

  }

  .inner\_Blocks label

  {

    flex: 0 0 140px; /\* Adjust the width as needed \*/

  }

  .inner\_Blocks\_content label {

    flex: 0 0 120px; /\* Adjust the width as needed \*/

  }

  .inner\_Blocks input[type="text"],

  .inner\_Blocks textarea {

    padding: 5px;

    border-radius: 0;

    -webkit-appearance: none;

    -moz-appearance: none;

  }

  .inner\_Blocks\_content button{

    background-color: #222;

    border-radius: 4px;

    border-style: none;

    box-sizing: border-box;

    color: #fff;

    cursor: pointer;

    display: inline-block;

    font-family: "Farfetch Basis","Helvetica Neue",Arial,sans-serif;

    font-size: 10px;

    font-weight: 300;

    line-height: 1.5;

    margin: 0;

    max-width: none;

    outline: none;

    overflow: hidden;

    padding: 9px 20px 8px;

    position: relative;

    text-align: center;

    text-transform: none;

    user-select: none;

    -webkit-user-select: none;

    touch-action: manipulation;

  }

  .inner\_Blocks\_content button,

  .inner\_Blocks\_content input

  {

    margin-right: 5px;

  }

  .inner\_Blocks input[type="text"],

  .inner\_Blocks input[type="number"]

  {

    width: 235px;

    border-radius: 0;

    -webkit-appearance: none;

    -moz-appearance: none;

  }

  .inner\_Blocks\_content input[type="text"],

  .inner\_Blocks\_content input[type="number"]

  {

    width: 235px;

    border-radius: 0;

    -webkit-appearance: none;

    -moz-appearance: none;

  }

  .inner\_Blocks\_content textarea

  {

    margin-right: 5px;

  }

  #addRequirementBtn

  {

    height: auto;

    max-height: 30px;

    overflow: hidden;

    white-space: nowrap;

    text-overflow: ellipsis;

  }

.inner\_btn{

    text-align: center;

}

.newCourseBtns {

  background-color: #222;

  border-radius: 4px;

  border-style: none;

  box-sizing: border-box;

  color: #fff;

  cursor: pointer;

  display: inline-block;

  font-family: "Farfetch Basis","Helvetica Neue",Arial,sans-serif;

  font-size: 16px;

  font-weight: 700;

  line-height: 1.5;

  margin: 0;

  max-width: none;

  min-height: 44px;

  min-width: 10px;

  outline: none;

  overflow: hidden;

  padding: 9px 20px 8px;

  position: relative;

  text-align: center;

  text-transform: none;

  user-select: none;

  -webkit-user-select: none;

  touch-action: manipulation;

  margin: 10px;

}

.newCourseBtns:hover,

.newCourseBtns:focus {

  opacity: .75;

}

# Report.php

<?php

    session\_start();//start the session

    if(isset( $\_SESSION['userData'])){//user data has been set load the page

            //load the page

    }

    else{

            header('Location: login.php');//else navigate to login page

            exit;

    }

    if(isset($\_POST['report\_btn'])) {//check if the generate report buttton has been clicked

        $btnValue = $\_POST['report\_btn'];//get the value

        if($btnValue==="Create Report"){//if it is generate report buttton

            // process selected courses

            $checkboxs = $\_POST['checkbox'];//get the selected values

            $courseIds=array();//create array variable

            if(is\_array($checkboxs)>0){

                // Loop through the arrays to access the values

                for ($i = 0; $i <count($checkboxs); $i++) {

                    $courseIds[$i]=$checkboxs[$i];//assign values to the variable

                }

            }

            else{

                header('Location: courseSelectionForm.php');//else navigate to courseSelection page

                exit;

            }

        }

    }else if(isset($\_POST['course\_delete'])){//check if the delete buttton has been clicked

        $btnValue = $\_POST['course\_delete'];//get the value

        require\_once './mainFunctions.php';//require mainFuncions.php to get methods

        session\_start();

        try{

            deleteCourse($btnValue);//call deleteCourse function to delete the course

            $\_SESSION['deletesuccess'] = true;//set as true if its success

        }catch (ExceptionType1 $e) {

            $\_SESSION['deletesuccess'] = false;//set as false if its unsuccess

        }

        //navigate to courseSelectionform.php

        header('Location: courseSelectionform.php');

        exit;

    }

?>

<!DOCTYPE html>

<html>

    <head>

        <title>Report</title>

        <link rel="stylesheet" href="report.css" />

        <script src="https://code.jquery.com/jquery-3.4.1.min.js"

        integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo=" crossorigin="anonymous"></script>

        <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

        <script src="report.js" defer></script>

        <script>

            //store the selected course id's in courseIds variable in js script to process in js file

            var courseIds = <?php echo json\_encode($courseIds); ?>;

        </script>

    </head>

    <body>

        <main id="main\_container">

            <div class="table-container">

                <table id="course\_Tbl">

                        <thead>

                            <tr>

                                <th>Title</th>

                                <th>Overview</th>

                                <th>Location</th>

                                <th>Month</th>

                                <th>Fees Uk</th>

                                <th>Fees International</th>

                            </tr>

                        </thead>

                    <tbody id="courseDetailsBody">

                    </tbody>

                </table>

            </div>

        </main>

    </body>

</html>

# Report.js

//load courses

let jsonVal;

$(document).ready(function () {

    var html = "";

    var i = 0;

    let table = document.getElementById("course\_Tbl");

    var barChartData = [];//store the barchart dataset

    var coursetitles=[];//store course titles to display in barchart

    var ProccesedCourseCount=0;//store the count of how many courses has been so far processed

    document.getElementById("courseDetailsBody").innerHTML = "";//make the table body empty

    //make ajax request to get all course details

    $.ajax({

        url: 'mainFunctions.php',

        dataType: 'json',

        type: "POST",

        data: { action: 'allCourses' },

        success: function (response) {

            console.log(response);

            //make the table in ascending order using title

            response.sort((a, b) => a.title > b.title ? 1 : -1);

            //loop throgh courses

            for (let i = 0; i < response.length; i++) {

                //check whether to process only the user selected courses

                if (courseIds.includes(response[i].id)) {

                    //create table body

                    html += '<td>' + response[i].title + '</td>';

                    html += '<td>' + response[i].overview + '</td>';

                    html += '<td>' + response[i].location + '</td>';

                    html += '<td>' + response[i].start\_month + '</td>';

                    html += '<td>' + response[i].fees\_UK + '</td>';

                    html += '<td>' + response[i].fees\_international + '</td></tr>';

                    //get modules for the specific selected courses using ajax request

                    $.ajax({

                        url: 'mainFunctions.php',

                        dataType: 'json',

                        type: "POST",

                        data: {

                            action: 'getModules',

                            courseid: response[i].id

                        },

                        success: function (moduleResponse) {

                            //assign course title

                            coursetitles[ProccesedCourseCount]=response[i].title;

                            //increment the processed course count

                            ProccesedCourseCount+=1;

                            var modules = [];//store the modules

                            var credits = [];//stores credits for that module

                            //loop thorugh modules

                            for (let i = 0; i < moduleResponse.length; i++) {

                                modules[i] = moduleResponse[i].module;

                                credits[i] = moduleResponse[i].credit;

                                //assign values for barchartdat aset

                                var labelExists = false;

                                var data={};//object to store new module values for courses

                                //loop through data array

                                for (var x = 0; x < data.length; x++) {

                                    //if the data object have that module already

                                    if (data[x].label === modules[i]) {

                                        //make it as label already exist in the array

                                        labelExists = true;

                                        data[x].data.push(parseInt(credits[i])); // Append the new value to existing data array inside data object

                                        break;

                                    }

                                }

                                //if the module was not exist in this course previously check for other courses have that module already

                                if(labelExists===false){

                                    //loop through barchart array

                                    for (var y = 0; y < barChartData.length; y++) {

                                        var currentLabel = modules[i];//get currently what module we are processing

                                        // Check if label already exists

                                        var existingIndex = barChartData.findIndex(obj => obj.label === currentLabel);

                                        //if already exist

                                        if (existingIndex !== -1) {

                                            // Append value to existing barChartData "data" array for the specific course using ProccesedCourseCount variable

                                            barChartData[existingIndex].data[ProccesedCourseCount-1]=parseInt(credits[i]);

                                            //make labelExists  true

                                            labelExists = true;

                                            break;

                                        }

                                    }

                                }

                                //if its new label create object

                                var creditArray=[];//to store credit values for new module

                                creditArray[ProccesedCourseCount-1]=parseInt(credits[i]);//assign values based on which course we are processing now

                                if(labelExists===false){// if the module is no already exist

                                    var randomColor = getRandomColor();//get the random color

                                    //assign values to data object

                                    data  = {

                                        label: modules[i],

                                        data: creditArray,

                                        backgroundColor: [

                                            randomColor

                                        ],

                                        borderColor: [

                                            randomColor

                                        ],

                                        borderWidth: 1

                                    }

                                }

                                // Check if label already exists

                                 var existingIndex = barChartData.findIndex(obj => obj.label === currentLabel);

                                if (existingIndex === -1) {

                                    //push the data to barchart array

                                    barChartData.push(data);

                                }

                            }

                            //load pie chat

                            loadPie(response[i].title, response[i].id, modules, credits);

                            //selected courses and data assigened are equal then call bar chart

                            console.log(JSON.stringify(barChartData));//for debugging purporse

                            //check all the selected courses has been processed and number of slected course should be freater than 1

                            if (courseIds.length===ProccesedCourseCount && courseIds.length>1) {

                                //call bar chart

                                barChart(coursetitles,barChartData);

                            }

                        },

                        error: function (xhr, status, error) {

                            console.log('Error:', error, " ", xhr.responseText);

                        }

                    }

                    );

                }

            }

            //assign table body

            document.getElementById("courseDetailsBody").innerHTML = html;

        },

        error: function (xhr, status, error) {

            console.log('Error:', error, " ", xhr.responseText);

        }

    });

}

);

//create modules table

function formModuleTable(title,div,modules, credits) {

    //create inner container

    var div2 = document.createElement("div");//create div element

    div2.classList.add("moudule-Table");//add class to div element

    var html = "";//to store html elements

    html += '<h3>'+title+'</h3>' ;

    html += '<table><thead><tr>' ;

    html += '<th>Modules</th>';

    html += '<th>Credits</th></tr></thead>';

    html += '<tbody>' ;

        //loop thorugh modules

        for (let i = 0; i < modules.length; i++) {

            html += '<tr><td>' + modules[i] + '</td>';//assign module name

            html += '<td>' + credits[i]+ '</td></tr>';//assign credit value

        }

    html += '</tbody>' ;

    html += '</table>' ;

    //append table

    div2.innerHTML = html

    //append pie chart

    div.appendChild(div2);

}

//pie chart

function loadPie(title, courseId, modules, credits) {

    //get modules container

    var container = document.getElementById("main\_container");

    //create div and assign class

    var div = document.createElement("div");

    div.classList.add("pie-container");

    //call module table creation

    formModuleTable(title,div,modules, credits);

    //create inner container

    var div2 = document.createElement("div");

    div2.classList.add("pie-chart");

    //create canvas

    var canvas = document.createElement("canvas");

    canvas.setAttribute("id", courseId);

    div2.appendChild(canvas);

    //append pie chart

    div.appendChild(div2);

    //div to cintainer

    container.appendChild(div);

    const ctx = document.getElementById(courseId);

    const chart = new Chart(ctx, {

        type: 'pie',

        data: {

            labels: modules,

            datasets: [{

                data: credits,

                borderWidth: 1

            }]

        },

        options: {

            scales: {

                y: {

                    beginAtZero: true

                }

            }

        }

    });

}

function barChart(courses,barDataSet) {

    //get modules container

    var container = document.getElementById("main\_container");

    //create div and assign class

    var div = document.createElement("div");

    div.classList.add("bar-container");

    //create Heading

    var h3 = document.createElement("h3");

    h3.textContent = "Descriptive Data";

    div.appendChild(h3);

    //create canvas

    var canvas = document.createElement("canvas");

    canvas.setAttribute("id", "barChart");//set id

    div.appendChild(canvas);//append it to div

    container.appendChild(div);//append it to main conatiner

    const ctx = document.getElementById('barChart').getContext('2d');

    var myChartConfig = {

        type: 'bar',

        data: {

            labels: courses,

            datasets: barDataSet

        },

        options: {

            scales: {

                y: {

                    beginAtZero: true,

                    title: {

                        display: true,

                        text: 'Credits',

                        font: {

                          weight: 'bold'

                        }

                      }

                }

            }

        }

    };

    const myChart = new Chart(ctx, myChartConfig);

}

//create random color

function getRandomColor() {

    var letters = '0123456789ABCDEF';//store string value

    var color = '#';

    for (var i = 0; i < 6; i++) {//loop until i is less than 6

      color += letters[Math.floor(Math.random() \* 16)];//create hexadecimal color value

    }

    return color;

  }

# Report.css

html, body { height: 100%; margin: 0; padding: 5px }

main{

    display: flex;

    justify-content: center;

    flex-direction: column;

    align-items: center;

}

.table-container {

    max-height: 400px; /\* Set the maximum height of the table container \*/

    overflow-y: auto; /\* Enable vertical scrolling \*/

    justify-content: center;

    align-items: center;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);

  }

table{

    border-collapse: collapse;

    box-shadow: 0 5px 10px gray;

    background-color: white;

    text-align: left;

    overflow: hidden;

}

thead{

    box-shadow: 0 5px 10px gray;

}

th{

    padding: 1rem 2rem;

    text-transform: uppercase;

    letter-spacing: 0.1rem;

    font-size: 0.7rem;

    font-weight: 700;

}

td{

    padding: 1rem 2rem;

}

.edit{

    border-radius: 5px;

    background-color: black;

    padding: 0.2rem 1rem;

    text-align: center;

    color: white;

}

tr:nth-child(even){

    background-color: aliceblue;

}

.pie-container{

        display: flex;

        justify-content: space-between;

        margin: 5px;

        max-height:350px ;

        padding: 10px;

        box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);

        width: 82%;

        position: relative;

        border: 1px solid black;

}

.moudule-Table{

    flex: 1;

    display: flex;

    flex-direction: column;

    justify-content: center;

    align-items: center;

    max-height: 300px; /\* Set the maximum height of the table container \*/

    overflow-y: auto; /\* Enable vertical scrolling \*/

}

.moudule-Table td{

    padding: 0.5rem 1rem;

}

.bar-container{

    display: flex;

    flex-direction: column;

    justify-content: center;

    align-items: center;

    margin: 5px;

    padding: 10px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);

    width: 82%;

    position: relative;

    border: 1px solid black;

}

.pie-container h3 {

    margin: 0;

    padding: 10px;

  }

  .pie-container canvas {

    width: 100%;

    height: calc(100% - 45px); /\* Subtract the height of the h3 element \*/

    padding: 10px;

  }

# mainFunctions.php

<?php

// Set the content type to JSON

header('Content-Type: application/json');

//rquire functions.php

require\_once './PDO/functions.php';

$response = array();

//if submit button is clicked

if(isset($\_POST['btn'])) {

    $btnValue = $\_POST['btn'];//get the value

    if($btnValue==="Add Course"){//if the value of button is Add Course

        addNewCourse();//call addNewCourse function

    }

}

//function to add new course

function addNewCourse(){

    $data = [

        "title" => $\_POST['title'],

        "overview"=>$\_POST['overview'],

        "location"  =>$\_POST['location'],

        "start\_month"   =>$\_POST['selectMonth'],

        "fees\_UK"=>$\_POST['uk'],

        "fees\_international"=>$\_POST['international'],

        "type"=> $\_POST['level']

      ];

      //check whether with same name course exist or not

    $result=fetchARecordWithOneWhereClause("course\_tbl","title",$\_POST['title']);

      if($result){//if course already exist

        //navigate to newCourse.php

        session\_start();

        $\_SESSION['error'] = "course already exist ".$\_POST['title'];//set error message

        header('Location: newCourse.php');

        exit;

      }

      else{

        //process insertion

        echo($data["title"]);

        $courseId=insertAndgetId("course\_tbl",$data);

        echo($courseId);

        if($courseId>0){

            // process modules

            $modules = $\_POST['module'];

            $credits = $\_POST['credit'];

            // Loop through the arrays to access the values

            for ($i = 0; $i <count($modules); $i++) {

                $module = $modules[$i];

                $credit =  $credits[$i];

                $data = [

                    "course\_id"=>$courseId,

                    "module" => $module,

                    "credit"=>$credit

                ];

                $moduleResult=insert("modules\_tbl",$data);

                //if modules not added successfully delete the inserted course

                if($moduleResult!=true){

                    //delete course details

                    deleteRecord("course\_tbl","id",$courseId);

                    //navigate to newCourse.php

                    session\_start();

                    $\_SESSION['error'] = "unable to insert modules";//set error message

                    header('Location: newCourse.php');

                    exit;

                }

            }

            // process requirements

            $requirements = $\_POST['requirement'];

            // Loop through the arrays to access the values

            for ($i = 0; $i <count($requirements); $i++) {

                $requirement = $requirements[$i];

                $data = [

                    "course\_id"=>$courseId,

                    "requirements" => $requirement

                ];

                $requirementResult=insert("entry\_req\_tbl",$data);

                //if requirement not added successfully delete the inserted course

                if($requirementResult!=true){

                    //delete course details

                    deleteRecord("course\_tbl","id",$courseId);

                    //navigate to newCourse.php

                    session\_start();

                    $\_SESSION['error'] = "unable to insert entry requirements";//set error message

                    header('Location: newCourse.php');

                    exit;

                }

            }

            // process faqs

            $questions = $\_POST['question'];

            $answers = $\_POST['answer'];

            // Loop through the arrays to access the values

            for ($i = 0; $i <count($questions); $i++) {

                $question = $questions[$i];

                $answer =  $answers[$i];

                $data = [

                    "course\_id"=>$courseId,

                    "question" => $question,

                    "answer"=>$answer

                ];

                $faqResult=insert("faq\_tbl",$data);

                if($faqResult!=true){

                    //delete course details

                    deleteRecord("course\_tbl","id",$courseId);

                    //navigate to newCourse.php

                    session\_start();

                    $\_SESSION['error'] = "unable to insert modules";

                    header('Location: newCourse.php');

                    exit;

                }

            }

            // navigate with success message

            session\_start();

            $\_SESSION['success'] = true;

            header('Location: newCourse.php');

            exit;

        }

        else{

            // navigate with success message

            session\_start();

            $\_SESSION['success'] = false;

            header('Location: newCourse.php');

            exit;

        }

      }

}

//select which function to execute based on action

if (isset($\_POST['action'])) {

    switch ($\_POST['action']) {

        case 'allCourses'://if its all courses call allCourses function

            allCourses();

            break;

        case 'getModules'://if its get modules call getModules function with courseid parameter

            $courseid = $\_POST['courseid'];

            getModules($courseid);

            break;

        case 'getRequirements':

            $courseid = $\_POST['courseid'];//if its get requirements call getRequirements function with courseid parameter

            getRequirements($courseid);

            break;

        case 'deletCourse':

            $courseid = $\_POST['courseid'];//if its deleteCourse call getModules function with courseid parameter

            deletCourse($courseid);

            break;

        case 'specificCourse':

                $courseid = $\_POST['courseid'];//if its get course details call specificCourse function with courseid parameter

                specificCourse($courseid);

            break;

        case 'updateCourse':

            $courseid = $\_POST['courseid'];//if its get course details call specificCourse function with courseid parameter

            $data=[

                "title"=>$\_POST["title"],

                "overview"=>$\_POST["overview"],

                "type"=>$\_POST["type"],

                "start\_month"=>$\_POST["start\_month"],

                "fees\_UK"=>$\_POST["fees\_UK"],

                "fees\_international"=>$\_POST["fees\_international"]

            ];

            updateCourse($data,$courseid);

            break;

        }

}

//get all course details

function allCourses(){

    //get all course details from the table course\_tbl

    $result=fetchAllRecordsWithFetchAll("course\_tbl");

    $jsonResult = json\_encode($result);//encode is as json string

   echo $jsonResult;

}

//get specific course details

function specificCourse($courseId){

    //get specific course details from the table course\_tbl

    $result=fetchARecordWithOneWhereClause("course\_tbl","id",$courseId);

    $jsonResult = json\_encode($result);//encode is as json string

   echo $jsonResult;

}

//get modules

function getModules($courseId){

    //get the modules from the modules\_tbl where course\_id = $courseId

    $result=fetchARecordWithOneWhereClause("modules\_tbl","course\_id",$courseId);

    $jsonResult = json\_encode($result);//encode is as json string

   echo $jsonResult;

}

//get requirements

function getRequirements($courseId){

    //get the requirements from the entry\_req\_tbl where course\_id = $courseId

    $result=fetchARecordWithOneWhereClause("entry\_req\_tbl","course\_id",$courseId);

    $jsonResult = json\_encode($result);//encode is as json string

   echo $jsonResult;

}

//delete Course

function deleteCourse($courseId){

    //delete the course record from the course\_tbl where course\_id = $courseId

    $result=deleteRecord("course\_tbl","id",$courseId);//encode is as json string

    return $result;

}

//update Course

function updateCourse($data, $id){

    //update the course record from the course\_tbl where course\_id = $courseId

    $result=update("course\_tbl",$data,$id,"id");//encode is as json string

    if($result){

        echo true;//return true if its successfully updated

    }else{

        echo false;//return false if its un successfully updated

    };

}

?>

# Functions.php

<?php

    require\_once 'dbInstance.php';

?>

# dbInstance.php

<?php

ob\_start();

if (session\_status() == PHP\_SESSION\_NONE) {

    session\_start();

}

require 'pdo.php';

?>

# Pdo.php

<?php

//

$host = 'db';

$user\_name = 'root';

$pass = 'csym019';

$schema= 'Courses';

global $pdo;

try {

 $pdo = new PDO("mysql:host=$host; dbname=$schema;", $user\_name, $pass);//create new pdo object

} catch (PDOException $e) {

  echo "Error!: " . $e->getMessage() . "<br/>";

 //echo "Ooooops something bad happened";

}

$current\_file  = $\_SERVER['SCRIPT\_NAME'];

$http\_referer = isset($\_SERVER['HTTP\_REFERER']) ? $\_SERVER['HTTP\_REFERER'] : "";

  //select all record from table

  function fetchAllRecords($table)

  {

    $stmt = $GLOBALS['pdo']->prepare('SELECT \* FROM ' . $table );

    $stmt->execute();

    return $stmt;

  }

  // insert new record into database

 function insert($table, $record)

 {

       $keys = array\_keys($record);

       $values = implode(', ', $keys);

       $valuesWithColon = implode(', :', $keys);

       $query = 'INSERT INTO ' . $table . ' (' . $values . ') VALUES (:' . $valuesWithColon . ')';

       $insert\_stmt = $GLOBALS['pdo']->prepare($query);

       $insert\_stmt->execute($record);

       return $insert\_stmt;

 }

  //insert and get the id

function insertAndgetId($table, $record)

{

      $keys = array\_keys($record);

      $values = implode(', ', $keys);

      $valuesWithColon = implode(', :', $keys);

      $query = 'INSERT INTO ' . $table . ' (' . $values . ') VALUES (:' . $valuesWithColon . ')';

      $insert\_stmt = $GLOBALS['pdo']->prepare($query);

      $insert\_stmt->execute($record);

      $last\_insert\_id = $GLOBALS['pdo']->lastInsertId();

     return $last\_insert\_id;

}

//fetch course record from database with where

function fetchARecordWithOneWhereClause($table, $field, $value)

{

  $stmt = $GLOBALS['pdo']->prepare('SELECT \* FROM ' . $table . ' WHERE ' . $field . ' = :value');

  $criteria = [

  'value' => $value

  ];

  $stmt->execute($criteria);

  $resultAll = $stmt->fetchAll();

  return $resultAll;

}

//insert with where condition

 function insertWhere($table, $field, $value, $record)

{

      $keys = array\_keys($record);

      $values = implode(', ', $keys);

      $valuesWithColon = implode(', :', $keys);

      $query = 'INSERT INTO ' . $table . ' (' . $values . ') VALUES (:' . $valuesWithColon . ') WHERE '.$field.' = '.$value.' ';

      $insert\_stmt = $GLOBALS['pdo']->prepare($query);

      $insert\_stmt->execute($record);

      return $insert\_stmt;

}

//delete course record from database and handle ajax request

function deleteRecord($table, $field, $value)

{

        $stmt = $GLOBALS['pdo']->prepare('DELETE FROM ' . $table . ' WHERE ' . $field . ' = :value');

        $criteria = [

        'value' => $value

        ];

        $stmt->execute($criteria);

        return $stmt;

}

//fetech all records

function fetchAllRecordsWithFetchAll($table)

{

 $stmt = $GLOBALS['pdo']->prepare('SELECT \* FROM ' . $table );

 $stmt->execute();

 $resultAll = $stmt->fetchAll();

 return $resultAll;

}

//hash the password

function hashPass($pass){

  return password\_hash($pass, PASSWORD\_DEFAULT);

}

//verify the password with hash password

function passwordVerify($inputPass,$hashPass){

  return password\_verify($inputPass,$hashPass);

}

//update table value with where clause

function update ($table, $data, $id, $whereClause)

{

    $setPart = array();

    $bindings = array();

    foreach ($data as $key => $value)

    {

        $setPart[] = "{$key} = '{$value}'";

    }

    $sql = "UPDATE {$table} SET ".implode(', ', $setPart)." WHERE $whereClause = $id ";

    $stmt = $GLOBALS['pdo']->prepare($sql);

    $stmt->execute($bindings);

    return $stmt;

}

  ?>

# Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected values** | **Actual output** | **Status** |
| Test 1 | Valid user name and password | Login success full | Login success full | pass |
| Test 2 | Valid user name and invalid password | Show error message | Show error message | pass |
| Test 3 | In Valid user name and valid password | Show error message | Show error message | pass |
| Test 4 | In Valid user name and in valid password | Show error message | Show error message | pass |
| Test 5 | Fill all details and create new course | Create course successful | Create course successful | pass |
| Test 6 | Keep some fields empty and try to create course | Show error message | Show error message | pass |
| Test 7 | Try to create already existing course title | Show error message | Show error message | pass |
| Test 8 | Try to add more modules | Modules added successfully | Modules added successfully | pass |
| Test 8 | Try to add more requirements | Requirements added successfully | Requirements added successfully | pass |
| Test 9 | View all course list In table | View all course details in table successfully | View all course details in table successfully | pass |
| Test 10 | Delete existing course | Delete existing course successfully | Deleted existing course successfully | pass |
| Test 11 | Updating course details | Update existing course details successfully | Updated existing course details successfully | pass |
| Test 12 | Select all check box | Select all check boxes | Select all check boxes | pass |
| Test 13 | Unselect all check box | unselect all check boxes | unselect all check boxes | pass |
| Test 14 | Select one course and click create report | Table with details and a pie chart | Table with details and a pie chart | pass |
| Test 15 | Select more than one course and click create report | Table with selected course details and the pie chart for each of them and additionally grouped bar chart | Table with selected course details and the pie chart for each of them and additionally grouped bar chart | pass |
| Test 16 | Without selecting any courses and try to create report | No report created and form | No report created and form | pass |