Authenticating User & Report Presentation

Chapter Outline

In this chapter, you will learn about:

- User Authentication
- Report Presentation

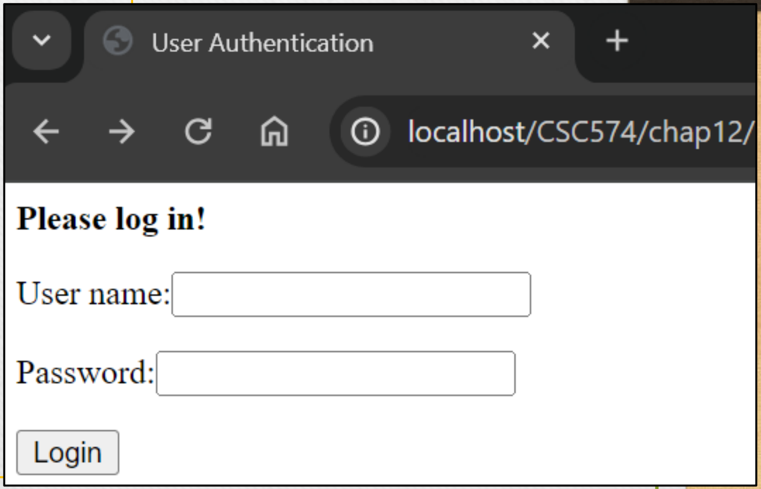
User Authentication

- Occasionally, users have to be authenticated by logging into a password-protected Web site.
- When building a site, developer may decide to require this of the visitors for several reasons.
- The first, and most obvious, is to protect secure information so that it can only be viewed by a select few.
- Developer may also choose to assign usernames and passwords to casual visitors.
- This may be done in order to keep track of who is viewing the site, or to provide personalized options and services to the visitors.

User Authentication

- The simplest way to authenticate users is by using a login form, such as follows:

```
<html>
    <head>
       <title>User Authentication</title>
    </head>
    <body>
       <?php
       if (isset($_POST["user"]) && isset($_POST["pwd"])
          && strtolower($_POST["user"])=="ali" && $_POST["pwd"]=="1234"){
       ?>
       Welcome: <?php echo $_POST["user"]; ?>
       <?php
       }
       else {
       ?> <b>Please log in!</b>
       <p>
       <form method="post">
           <p>User name:<input type="text" name="user"></p>
           <p>Password:<input type="password" name="pwd"></p>
           <p><input type="submit" name="Login" value="Login"></p>
       </form>
       </p>
       <?php
        }
       ?>
    </body>
</html>
```



Please log in!

User name:

Password:

User Authentication

- The username and password are hard coded, or you can validate the existence of the user against the data on a table in a database that contains the information on the user's login name and password.
- The limitation of using such form is that you can protect only one page at a time.
- Therefore, this is not a good practice.
- Another approaches to accomplish the task is using PHP sessions

Authenticating using PHP Sessions

- The information about whether a user is authenticated is saved in a session variable.
- Two important PHP session functions:
  - session_name() returns the name of the PHP session (for example, "PHPSESSID").
  - session_id() returns the session ID (for example, "18143b51ee37ac73cea81cd19ba20f2c").

Authenticating using PHP Sessions

- **Example: using a simple login form**

```
<?php //index.php
session_start();
?>
<html>
    <head>
        <title>Home Page</title>
    </head>
    <body>
        <h1>Home Page</h1>
        <?php
        if (isset($_SESSION["username"])){
            echo ("You are logged in!: ".$_SESSION["username"]);
            echo "<p><a href='logout.php'>Logout</a></p>";
        } else {
           echo ("You are NOT logged in!: ");
           echo "<p><a href='loginForm.php'>Goto Login Form</a></p>";
        }
        ?>
    </body>
</html>
```

```
<?php //logout.php
session_start();
session_destroy();//delete PHP session
?>
<b></b>You are now logout</b>
<p><a href='loginForm.php'>Goto Login Form</a>
```

Authenticating using PHP Sessions

```php
<?php //loginForm.php
session_start();
if (isset($_POST["submit"])) {
    if ($_POST["user"]=="admin" && $_POST["pass"]=="1234") {
        $_SESSION["username"] = $_POST["user"];
    }
}
?>
<html>
    <head>
        <title>User Authentication</title>
    </head>
    <body>
        <?php
        if (isset($_SESSION["username"])){
            header("Location: index.php");
        } else {
        ?>
        <form method="post">
            Name: <input type="text" name="user" /><br><br>
            Password: <input type="password" name="pass" /><br>
            <input type="submit" name="submit" value="Login" />
        </form>
        <?php }?>
    </body>
</html>
```

Report Presentation

Report Presentation

- A **report presentation in PHP using a MySQL database** refers to the process of **retrieving data from a MySQL database with PHP, processing it, and displaying it in a structured, readable format** for users.
- Steps in a Report Presentation
  - **1. Data Retrieval -** PHP connects to a MySQL database and fetches data using SQL queries
  - **2. Data Processing -** PHP processes the data
    - ➢ Calculations (totals, averages)
    - ➢ Grouping or filtering
  - **3. Report Formatting -** the processed data is displayed as:
    - ➢ Tables (pagination)
    - ➢ Searching sections
    - ➢ Data visualization via Charts / graphs
    - ➢ Printable reports (PDF / HTML)

What is Pagination

- Pagination is the process of displaying the data on multiple pages rather than showing them on a single page.
- Pagination is done when there is a database with numerous records.
- Dividing those records increases the readability of the data.
- It can retrieve this data as per the user's requests.
- Example of pagination is the result page of a Google search.
- A search query entered into the search bar can fetch thousands of results. And it divides these results into multiple pages.
- You can click on any page number given at the bottom to visit that page.

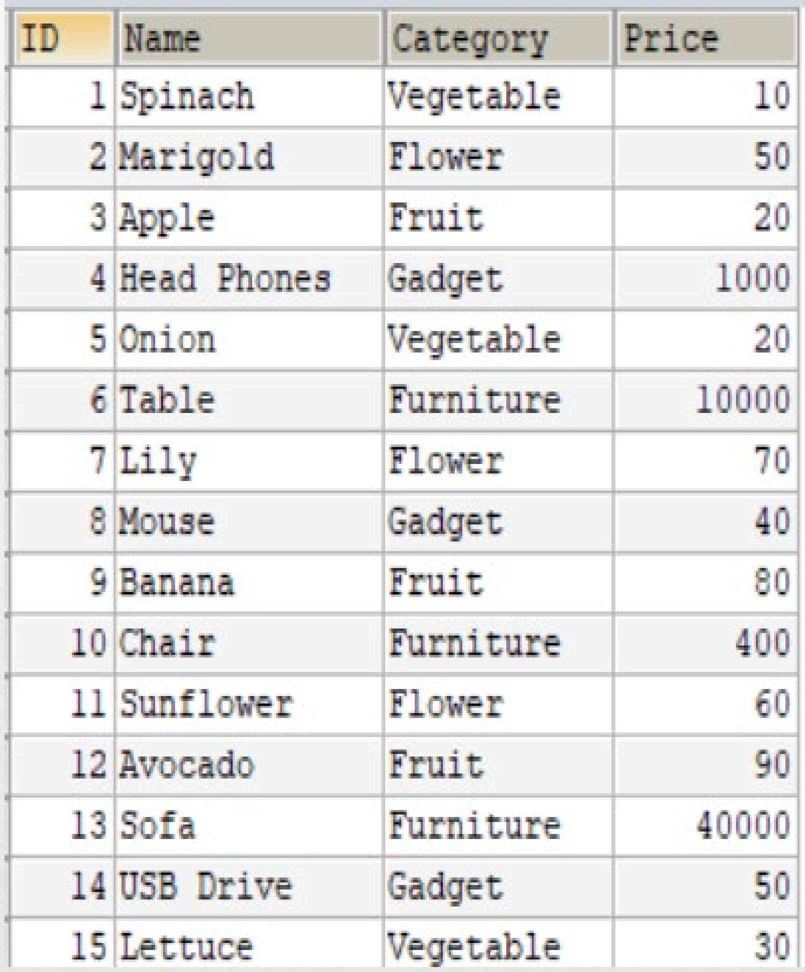
Pagination Implementation

- The steps to implement pagination with PHP and MySQL:
  - Step 1: Create database
  - Step 2: Connect the database.
  - Step 3: Set number rows and Retrieve the Currently Active Page Number
  - Step 4: Display the Table Records
  - Step 5: Show Page Number in the URL

Step 1: Create a Database.

- In “demo” database, create table “Items” with 4 columns:
  1. ID - int
  2. Name - varchar(20)
  3. Category - varchar(20)
  4. Price - int
- Insert 15 rows and populate data in the table.

```
CREATE TABLE items (
  ID INT(11) NOT NULL AUTO_INCREMENT,
  NAME VARCHAR(20) DEFAULT NULL,
  Category VARCHAR(20) DEFAULT NULL,
  Price DECIMAL(10,0) DEFAULT NULL,
  PRIMARY KEY (ID)
)
```



ID	Name	Category	Price
1	Spinach	Vegetable	10
2	Marigold	Flower	50
3	Apple	Fruit	20
4	Head Phones	Gadget	1000
5	Onion	Vegetable	20
6	Table	Furniture	10000
7	Lily	Flower	70
8	Mouse	Gadget	40
9	Banana	Fruit	80
10	Chair	Furniture	400
11	Sunflower	Flower	60
12	Avocado	Fruit	90
13	Sofa	Furniture	40000
14	USB Drive	Gadget	50
15	Lettuce	Vegetable	30

Step 2: Connect the Database

- The following code connects the database “demo” to the PHP file.

```
  <?php  //connection.php
     // Connect to the database
      $conn = mysqli_connect('localhost', 'root', '');
      if (! $conn) {
         die("Connection failed" . mysqli_connect_error());
         }
      else {
         mysqli_select_db($conn, ‘demo');
      }
?>
```

Step 3: Set number rows & retrieve the Currently Active Page Number

- Set number of rows per page and get the page number of the current page using the isset() method.

```
$limit =5; //to store number of rows per page
// update the active page number
  if (isset($_GET["page"])) {
     $page_number  = $_GET["page"];
  }
  else {
    $page_number=1;
  }
```

Step 4: Display the Table Records

- Display the table records of your database.

```
<?php
     // Table head
     while ($row = mysqli_fetch_array($result)) {
        ?>
        <tr>
          <td><?php echo $row["ID"]; ?></td>
          <td><?php echo $row["Name"]; ?></td>
          <td><?php echo $row["Category"]; ?></td>
          <td><?php echo $row["Price"]; ?></td>
        </tr>
 <?php
       };
?>
```

Step 5: Show Page Number in the URL

- Display the page numbers in the URL of each page. The code in step 4 updates the URL with page numbers. It also inserts a “previous” and a “next” link with the page numbers.

```
if($page_number>=2){
          echo "<a href='pagination.php?page=".($page_number-1)."'>  Prev </a>";
 }
 for ($i=1; $i<=$total_pages; $i++) {
         if ($i == $page_number) {
             $pageURL .= "<a class = 'active' href='pagination.php?page="
                                              .$i."'>".$i." </a>";
        }
         else  {
             $pageURL .= "<a href='pagination.php?page=".$i."'>
                                              ".$i." </a>";
         }
  };
  echo $pageURL;
  if($page_number<$total_pages){
          echo "<a href='pagination.php?page=".($page_number+1)."'>  Next </a>";
   }
```

The Complete Code

- The following code is the combined code with the CSS to format the webpage and the table. It fetches the data of the 15 rows from the database and displays 5 rows per page.
- Name file as pagination.php
- Copy code in the next slide

pagination.php

```
<html>
    <head>
        <title>Pagination in PHP</title>
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <style>
            table { border-collapse: collapse; }
            .inline {
               display: inline-block;
                float: right; margin: 20px 0px;
            }
            input,button { height: 34px; }
            .items { display: inline-block; }
            .items a {
                font-weight: bold; font-size: 18px;
                color: black; float: left;
                padding: 8px 16px; text-decoration: none;
                border: 1px solid black; margin: 2px;
            }
            .items a.active {
                background-color: rgba(175, 201, 244, 0.97);
            }
            .items a:hover:not(.active) {
                background-color: #87ceeb;
            }
        </style>
    </head>
    <body>
      <center>
        <?php
        // Connect to the database
        // Connect to the database
         include_once("connection.php");

        // variable to store number of rows per page
        $limit = 5;
        // update the active page number
        if (isset($_GET["page"])) {
            $page_number  = $_GET["page"];
        } else {
          $page_number=1;
        }
```

pagination.php

```
        // get the initial page number
        $initial_page = ($page_number-1) * $limit;
        // get data of selected rows per page
        //echo "initial: $initial_page  limit: $limit ";
        $getQuery = "SELECT * FROM items LIMIT $initial_page, $limit";
        $result = mysqli_query ($conn, $getQuery);
        echo $getQuery;
        ?>
        <div class="container">
            <br>
            <div>
                <h1>Pagination in PHP</h1>
                <table class="table table-striped table-condensed table-bordered">
                    <thead>
                        <tr>
                           <th width="10%">ID</th>
                            <th>Name</th>
                            <th>Category</th>
                            <th>Price</th>
                        </tr>
                    </thead>
                    <tbody>
                       <?php
                         while ($row = mysqli_fetch_array($result)) {
                         // Table head
                       ?>
                        <tr>
                            <td><?php echo $row["ID"]; ?></td>
                            <td><?php echo $row["Name"]; ?></td>
                            <td><?php echo $row["Category"]; ?></td>
                            <td><?php echo $row["Price"]; ?></td>
                        </tr>
                         <?php };?>
                    </tbody>
                </table>
```

pagination.php

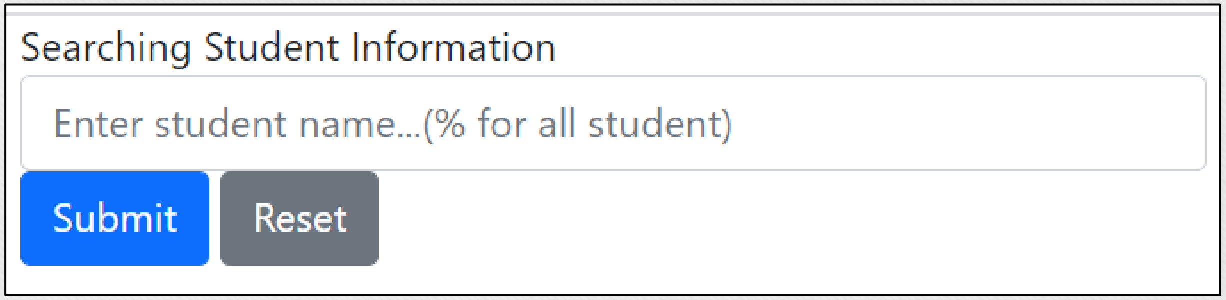
```
                <div class="Items">
                <?php
                    $getQuery = "SELECT COUNT(*) FROM Items";
                    $result = mysqli_query($conn, $getQuery);
                    $row = mysqli_fetch_row($result);
                    $total_rows = $row[0];
                    echo "</br>";
                    // get the required number of pages
                    $total_pages = ceil($total_rows / $limit);
                    $pageURL = "";
                    if($page_number>=2){
                        echo "<a href='pagination.php?page=".($page_number-1)."'>  Prev </a>";
                    }
                    for ($i=1; $i<=$total_pages; $i++) {
                        if ($i == $page_number) {
                            $pageURL .= "<a class = 'active' href='pagination.php?page=" .$i."'>".$i." </a>";
                        }
                        else  {
                            $pageURL .= "<a href='pagination.php?page=".$i."'> ".$i." </a>";
                        }
                    }
                    echo $pageURL;
                    if($page_number<$total_pages){
                        echo "<a href='pagination.php?page=".($page_number+1)."'>  Next </a>";
                    }
                ?>
                </div>
                <div class="inline">
                    <input id="page" type="number" min="1" max="<?php echo $total_pages?>"
                        placeholder="<?php echo $page_number." /".$total_pages; ?>" required>
                    <button onClick="go2Page();">Go</button>
                </div>
            </div>
        </div>
      </center>
      <script>
          function go2Page() {
            var page = document.getElementById("page").value;
            page = ((page><?php echo $total_pages; ?>)?<?php echo $total_pages; ?>:((page<1)?1:page));
            window.location.href = 'pagination.php?page='+page;
          }
     </script>
     </body>
</html>
```

Search database and display results

- Search → to display data from database based on certain criteria.
- Steps
  1. Create a form for search input
  2. Write a PHP script based on the search input.

1. Create a form for search input

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Search Example</title>
        <link rel="stylesheet" href="css/bootstrap.min.css">
    </head>
    <body>
        <div class="container">
            Searching Student <br>
            <form class="form-inline" method="post" action="search.php">
                <input type="text" name="name" class="form-control" placeholder="Enter
student name..">
                <input type="submit" name="submit" class="btn btn-primary" />
                <input type="reset" name="reset" class="btn btn-secondary" />
            </form>
        </div>
    </body>
</html>
```

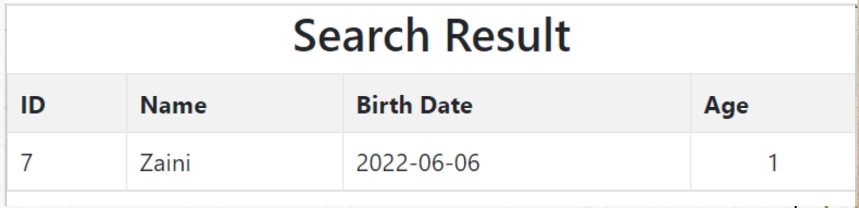

Searching Student Information

Enter student name...(% for all student)

2. Write a PHP script based on the search input.

```php
<?php
include 'connection.php'; //this include file connection.php
$totRow = 0;
if (isset($_POST["submit"])) {
    $name = $_POST["name"];
    if ($name == '%')
       $cond = "";
    else
       $cond = "where std_name = '".$name."' ";
    $sql = "select * from student ".$cond." order by std_name";
    $result = mysqli_query($conn,$sql);
    $totRow = mysqli_num_rows($result);//to get total rows return
?>
<html>
    <head>
        <title>Retrive data</title>
        <link rel="stylesheet" href="css/bootstrap.min.css">
    </head>
    <body>
        <?php if ($totRow > 0 ) { //if record found, display the data  ?>
        <div align="center">
            <h1>Search Result</h1>
            <table class="table table-striped table-condensed table-bordered">
                <tr>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Birth Date</th>
                    <th>Phone</th>
                    <th>Program</th>
                </tr>
                <?php
                while($row = mysqli_fetch_array($result)) { ?>
                <tr>
                    <td><?php echo $row['std_id']; ?></td>
                    <td><?php echo $row['std_name']; ?></td>
                    <td><?php echo $row['std_birthDate']; ?></td>
                    <td><?php echo $row['std_phone']; ?></td>
                    <td><?php echo $row['prg_code']; ?></td>
                </tr>
                <?php } ?>
            </table>
        </div>
    </body>
    <?php
    } else {
        echo "<br><h1>Record not found </h1>";
    }
} //end isset($_POST["submit"])
?>
</html>
```

Output



Search Result

ID	Name	Birth Date	Age
7	Zaini	2022-06-06	1

Data Visualization

- Visualizing data is a great way to make sense of it quickly and easily.
- This can also be incredibly helpful in presenting this data effectively to your audience, which is why bar and pie charts are so popular among developers.
- In this chapter, we will provide you with a step-by-step guide on how to create bar and pie charts using Chart.js!

What is Chart.JS

- Chart.js is an open-source JavaScript library for data visualization that allows you to create bar and pie charts in your web site or application.
- This Step-By-Step Guide will show you how to set up Chart.js and create your first bar chart theoretically.

1) Include the Chart.js library in your web page. You can either download the library from the Chart.js website, or include it from a CDN (Content Delivery Network).

2) Create a <canvas> element in your web page where the chart will be rendered. The <canvas> element needs to have an id attribute so that it can be referenced by Chart.js

3) Define the data that will be used to create the chart. This data can be hard-coded into the JavaScript, or retrieved dynamically from an external source such as a database

4) Create a JavaScript function that will initialize the chart.

Steps to set up Chart.js

1) Include the Chart.js library in your web page. You can either download the library from the Chart.js website, or include it from a CDN (Content Delivery Network).

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

2) Create a <canvas> element in your web page where the chart will be rendered. The <canvas> element needs to have an id attribute so that it can be referenced by Chart.js

```html
<canvas id="myChart" style="width:100%; max-width:600px"></canvas>
```

Steps to set up Chart.js

3) Define the data that will be used to create the chart.

```
const xValues = ["Kedah", "Kelantan", "Perak", "Perlis",
"Terengganu"];
const yValues = [55, 49, 44, 24, 35];
const barColors = ["red", "green","blue","orange","brown"];
```

Steps to set up Chart.js

4) Create a JavaScript function that will initialize the chart.

```
//type: "horizontalBar", "bar", "pie"
// "doughnut"
var chartType = "bar";
new Chart("myChart", {
  type: chartType,
  data: {
    labels: xValues,
    datasets: [{
      backgroundColor: barColors,
      data: yValues
    }]
  },
  options: {
    legend: {display: false},
    title: {
      display: true,
      text: "Padi Production 2020"
    }
  }
});
```

Complete Code

```
<html>
    <head>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.min.js"></script>
    </head>
    <body>
        <canvas id="myChart" style="width:100%;max-width:600px"></canvas>
        <script>
        const xValues = ["Kedah", "Kelantan", "Perak", "Perlis", "Terengganu"];
        const yValues = [55, 49, 44, 24, 35];
        const barColors = ["red", "green","blue","orange","brown"];
        //type: "horizontalBar", "bar", "pie", "doughnut"
        var chartType = "pie";
        new Chart("myChart", {
        type: chartType,
        data: {
            labels: xValues,
            datasets: [{
            backgroundColor: barColors,
            data: yValues
            }]
        },
        options: {
            legend: {display: true},
            title: {
            display: true,
            text: "Padi Production 2020"
            }
        }
        });
        </script>
    </body>
</html>
```

Creating Dynamic Data Graph using PHP and Chart.js

- Create a new project folder and name it **chart.**
- Note: feel free to name your project as per your choice.
- Inside the **chart** project folder create a subfolder and name it **js**. This will hold all the javascript files.

Creating Dynamic Data Graph using PHP and Chart.js

- In database (create db if not exist): **demo**
- Create table: **tbl_marks**

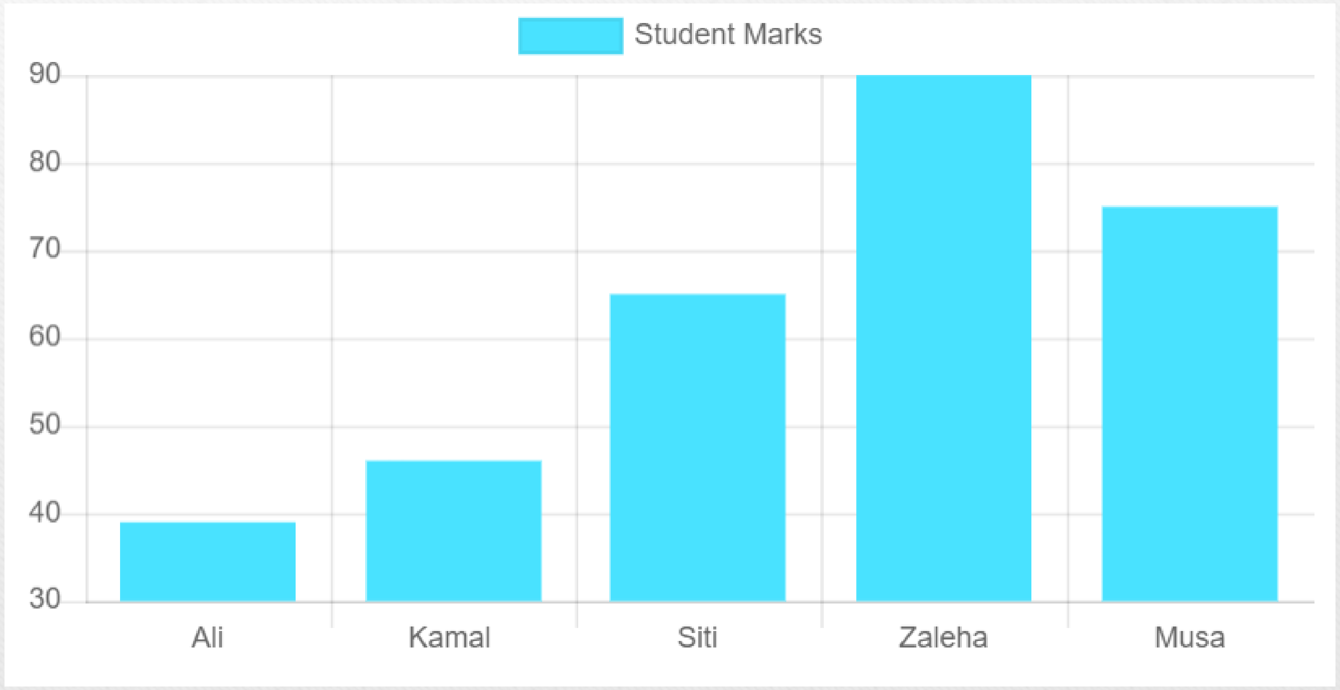
```
CREATE TABLE IF NOT EXISTS tbl_marks (
 student_id int(10) unsigned NOT NULL AUTO_INCREMENT,
 student_name varchar(35) NOT NULL,
 marks int(11) DEFAULT '0'
);
```

- Insert sample data:

```
INSERT INTO tbl_marks (student_name, marks) VALUES
('Ali', 39), ('Kamal ', 46), ('Siti', 65),
('Zaleha', 90),('Musa', 75);
```

Creating Dynamic Data Graph using PHP and Chart.js

- Read the mark data and supplied it to the Chart.js function to create the graph with the mark statistics.
- The following screenshot shows the graph output generated by Chart.js charting library with the dynamic data from the database.



Creating Dynamic Data Graph using PHP and Chart.js

- On loading the landing page, send an AJAX request to the PHP to read student marks from the database.
- This JSON response will be parsed and supplied as the parameter to the Chart.js function to create the graph.
- PHP gives limitless support for handling JSON file via programming with its built-in functions.

JSON File

- JSON is one of the popular data formats across all technologies. JSON handling with PHP is easier than it looks. Most of the APIs use JSON format for data interchange. For example, if you want to extract profile data from Facebook using its API, it returns it in JSON format. There is no option to ignore JSON.
- Data in JSON format can be read and parsed easily compared to other data formats. There are many core functions for JSON handling with PHP.
- Those built-in functions encode, write, parse, decode and convert JSON data. Those pre-defined PHP functions make our work easier

Landing Page (chartDB.php)

```
<!DOCTYPE html>
<html>
    <head>
    <title>Creating Dynamic Data Graph using PHP and Chart.js</title>
    <style type="text/css">
        BODY { width: 550PX; }

        #chart-container {
            width: 100%;
            height: auto; }
    </style>
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/Chart.min.js"></script>

    </head>
    <body>
        <div id="chart-container"> <canvas id="graphCanvas"></canvas>
        </div>
        <script>
            const barColors = "#49e2ff";
             $(document).ready(function () {
                showGraph();
            });

            function showGraph() {
                $.post("data.php", //an AJAX request to the PHP to read student marks from the db
                function (data) {
                    console.log(data); //to see data in json file
                    var name = [];
                    var marks = [];

                    for (var i in data) {
                        name.push(data[i].student_name);
                        marks.push(data[i].marks);
                    }

                    var chartdata = {
                        labels: name, //x values
                        datasets: [ {
                                label: 'Student Marks',
                                backgroundColor: barColors,
                                borderColor: '#46d5f1',
                                hoverBackgroundColor: '#CCCCCC',
                                hoverBorderColor: '#666666',
                                data: marks // y values
                            }
                        ]
                    };
                    var graphTarget = $("#graphCanvas");
                    var barGraph = new Chart(graphTarget, {
                        type: 'bar',
                        data: chartdata
                    });
                });
            }
        </script>
        </body>
</html>
```

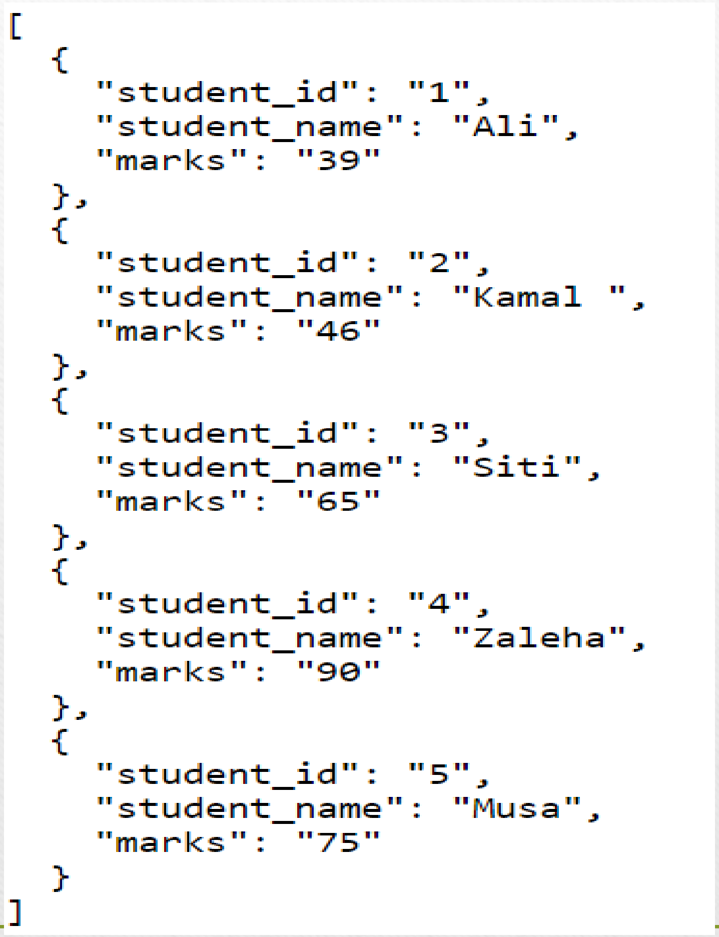
data.php

- The PHP file data.php is requested via AJAX to access database to read the student marks.
- After reading the records it returns it as JSON response

```
<?php
//indicates that the content being sent or received is JSON data
header('Content-Type: application/json');

$conn = mysqli_connect("localhost","root","","demo");

$sqlQuery = "SELECT student_id,student_name,marks FROM tbl_marks
ORDER BY student_name";

$result = mysqli_query($conn,$sqlQuery);

$data = array();
foreach ($result as $row) { //store record set array data[]
    $data[] = $row;
}
//print_r($data);
mysqli_close($conn);

echo json_encode($data); //json_encode: TO OUTPUT JSON STRUCTURED
DATA
?>
```

JSON Response

- See this in the browser **developer tool** of the **AJAX** response



```
[
  {
    "student_id": "1",
    "student_name": "Ali",
    "marks": "39"
  },
  {
    "student_id": "2",
    "student_name": "Kamal ",
    "marks": "46"
  },
  {
    "student_id": "3",
    "student_name": "Siti",
    "marks": "65"
  },
  {
    "student_id": "4",
    "student_name": "Zaleha",
    "marks": "90"
  },
  {
    "student_id": "5",
    "student_name": "Musa",
    "marks": "75"
  }
]
```

References

- https://www.w3schools.com/ai/ai_chartjs.asp
- https://www.chartjs.org/