

## Lab 2: GPA Calculator

Course: Web Development  
Level: 2nd Year of LMD in Computer Science

### 1 Introduction

In this lab you will create a simple **GPA (Grade Point Average) Calculator** web application. The GPA is calculated as:

$$\text{GPA} = \frac{\sum_{i=1}^n (\text{Grade Points}_i \times \text{Credits}_i)}{\sum_{i=1}^n \text{Credits}_i}$$

The interpretation of GPA values is as follows:

- **Distinction:**  $\text{GPA} \geq 3.7$
- **Merit:**  $3.0 \leq \text{GPA} < 3.7$
- **Pass:**  $2.0 \leq \text{GPA} < 3.0$
- **Fail:**  $\text{GPA} < 2.0$

The letter-grade to grade-point mapping used in this lab is:

Letter Grade	Grade Points	Interpretation
A / A+	4.0	Excellent
B	3.0	Good
C	2.0	Average
D	1.0	Below Average
F	0.0	Fail

This lab is divided into several steps:

1. **Step 1: Separate Files (Without jQuery)** — Create separate files for the front-end and back-end. Front-end: an HTML form (fields: course name, credit hours, letter grade), CSS, and a separate JavaScript file for validation (all fields required, numeric credit check). Back-end: a PHP script that computes the weighted GPA, echoes the result, and displays a summary table.
2. **Step 2: Merged File** — Regroup the HTML and PHP code into one page so that the form always appears, even after submission, with the result shown above it.
3. **Step 3: Introduce jQuery and Bootstrap** — Enhance the application using jQuery for dynamic course-row management and AJAX submission, and Bootstrap to style the result using color-coded alert classes (**success**, **info**, **warning**, **danger**).
4. **Step 4: Homework** — Extend the project with additional features (see details below).

## 2 Step 1: Separate Files (Without jQuery)

In this step, create the following files.

### 2.1 Front-end: index.html

This file contains an HTML form. The JavaScript code is placed in a separate file (`script.js`).

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>GPA Calculator</title>
6   <link rel="stylesheet" href="style.css">
7   <script src="script.js"></script>
8 </head>
9 <body>
10  <h1>GPA Calculator</h1>
11  <form action="calculate.php" method="post"
12    onsubmit="return validateForm();">
13    <div id="courses">
14      <div class="course-row">
15        <label>Course:</label>
16        <input type="text" name="course[]"
17          placeholder="e.g. Mathematics" required>
18        <label>Credits:</label>
19        <input type="number" name="credits[]"
20          placeholder="e.g. 3" min="1" required>
21        <label>Grade:</label>
22        <select name="grade[]">
23          <option value="4.0">A</option>
24          <option value="3.0">B</option>
25          <option value="2.0">C</option>
26          <option value="1.0">D</option>
27          <option value="0.0">F</option>
28        </select>
29      </div>
30    </div>
31    <button type="button" onclick="addCourse()">
32      + Add Course
33    </button><br><br>
34    <input type="submit" value="Calculate GPA">
35  </form>
36 </body>
37 </html>

```

Listing 1: index.html

### 2.2 JavaScript: script.js (Vanilla JS for Step 1)

This file contains the validation logic and the dynamic row functionality.

```

1 function addCourse() {
2   var row = document.createElement('div');
3   row.className = 'course-row';
4   row.innerHTML =
5     '<label>Course:</label>' +
6     '<input type="text" name="course[]" ' +
7       'placeholder="e.g. Mathematics" required>' +
8     '<label>Credits:</label>' +
9     '<input type="number" name="credits[]" ' +
10      'placeholder="e.g. 3" min="1" required>' +
11     '<label>Grade:</label>' +
12     '<select name="grade[]">' +

```

```

13     '<option value="4.0">A</option>' +
14     '<option value="3.0">B</option>' +
15     '<option value="2.0">C</option>' +
16     '<option value="1.0">D</option>' +
17     '<option value="0.0">F</option>' +
18     '</select>' +
19     '<button type="button" ' +
20         'onclick="this.parentNode.remove()">Remove</button>';
21     document.getElementById('courses').appendChild(row);
22 }
23
24 function validateForm() {
25     var courses = document.querySelectorAll('[name="course[]"]');
26     var credits = document.querySelectorAll('[name="credits[]"]');
27
28     for (var i = 0; i < courses.length; i++) {
29         if (courses[i].value === "") {
30             alert("All course name fields are required.");
31             return false;
32         }
33     }
34     for (var j = 0; j < credits.length; j++) {
35         if (isNaN(credits[j].value) || credits[j].value <= 0) {
36             alert("Credit hours must be positive numbers.");
37             return false;
38         }
39     }
40     return true;
41 }

```

Listing 2: script.js (Step 1)

## 2.3 Style: style.css

A simple CSS file for basic styling.

```

1 body {
2     font-family: Arial, sans-serif;
3     margin: 20px;
4     background-color: #e9ecef;
5 }
6 h1 {
7     color: #007BFF;
8 }
9 .course-row {
10     display: flex;
11     gap: 10px;
12     align-items: center;
13     margin-bottom: 10px;
14 }
15 .course-row input,
16 .course-row select {
17     padding: 6px;
18     border: 1px solid #ccc;
19     border-radius: 4px;
20 }
21 table {
22     border-collapse: collapse;
23     margin-top: 16px;
24 }
25 table th, table td {
26     border: 1px solid #ccc;
27     padding: 8px 14px;

```

```

28     text-align: center;
29 }
30 table th {
31     background-color: #007BFF;
32     color: white;
33 }

```

Listing 3: style.css

## 2.4 Back-end: calculate.php

This PHP script computes the weighted GPA and echoes the student's result along with a course summary table and the GPA interpretation.

```

1  <?php
2  if (isset($_POST['course'], $_POST['credits'], $_POST['grade'])) {
3      $courses = $_POST['course'];
4      $credits = $_POST['credits'];
5      $grades = $_POST['grade'];
6      $totalPoints = 0;
7      $totalCredits = 0;
8
9      echo "<table>";
10     echo "<tr>
11         <th>Course</th>
12         <th>Credits</th>
13         <th>Grade</th>
14         <th>Grade Points</th>
15     </tr>";
16
17     for ($i = 0; $i < count($courses); $i++) {
18         $course = htmlspecialchars($courses[$i]);
19         $cr = floatval($credits[$i]);
20         $g = floatval($grades[$i]);
21         if ($cr <= 0) continue;
22         $pts = $cr * $g;
23         $totalPoints += $pts;
24         $totalCredits += $cr;
25         echo "<tr>
26             <td>$course</td>
27             <td>$cr</td>
28             <td>$g</td>
29             <td>$pts</td>
30         </tr>";
31     }
32     echo "</table>";
33
34     if ($totalCredits > 0) {
35         $gpa = $totalPoints / $totalCredits;
36         if ($gpa >= 3.7) {
37             $interpretation = "Distinction";
38         } elseif ($gpa >= 3.0) {
39             $interpretation = "Merit";
40         } elseif ($gpa >= 2.0) {
41             $interpretation = "Pass";
42         } else {
43             $interpretation = "Fail";
44         }
45         echo "<p>Your GPA is <strong>" . number_format($gpa, 2)
46             . "</strong> ($interpretation).</p>";
47     } else {
48         echo "<p>No valid courses entered.</p>";
49     }

```

```

50 } else {
51     echo "Data not received.";
52 }
53 ?>

```

Listing 4: calculate.php

### 3 Step 2: Merged HTML and PHP in One File

Now, combine the HTML and PHP into a single file (e.g., `index.php`). The form always appears—even after submission—with the result displayed above it.

```

1 <?php
2 $result      = "";
3 $tableHtml   = "";
4
5 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
6     $courses    = $_POST['course'] ?? [];
7     $credits     = $_POST['credits'] ?? [];
8     $grades      = $_POST['grade'] ?? [];
9     $totalPoints = 0;
10    $totalCredits = 0;
11
12    $tableHtml   = "<table>";
13    $tableHtml   .= "<tr>
14                    <th>Course</th><th>Credits</th>
15                    <th>Grade</th><th>Grade Points</th>
16                    </tr>";
17
18    for ($i = 0; $i < count($courses); $i++) {
19        $course = htmlspecialchars($courses[$i]);
20        $scr     = floatval($credits[$i]);
21        $g       = floatval($grades[$i]);
22        if ($scr <= 0) continue;
23        $pts     = $scr * $g;
24        $totalPoints += $pts;
25        $totalCredits += $scr;
26        $tableHtml .= "<tr>
27                    <td>$course</td><td>$scr</td>
28                    <td>$g</td><td>$pts</td>
29                    </tr>";
30    }
31    $tableHtml .= "</table>";
32
33    if ($totalCredits > 0) {
34        $gpa = $totalPoints / $totalCredits;
35        if ($gpa >= 3.7) {
36            $interpretation = "Distinction";
37        } elseif ($gpa >= 3.0) {
38            $interpretation = "Merit";
39        } elseif ($gpa >= 2.0) {
40            $interpretation = "Pass";
41        } else {
42            $interpretation = "Fail";
43        }
44        $result = "Your GPA is " . number_format($gpa, 2)
45                . " ($interpretation).";
46    } else {
47        $result = "No valid courses entered.";
48    }
49 }
50 ?>

```

```

51 <!DOCTYPE html>
52 <html lang="en">
53 <head>
54   <meta charset="UTF-8">
55   <title>GPA Calculator</title>
56   <link rel="stylesheet" href="style.css">
57   <script src="script.js"></script>
58 </head>
59 <body>
60   <h1>GPA Calculator</h1>
61   <?php if ($result != ""): ?>
62     <?php echo $tableHtml; ?>
63     <p><strong><?= $result ?></strong></p>
64   <?php endif; ?>
65   <form action="" method="post" onsubmit="return validateForm();">
66     <div id="courses">
67       <div class="course-row">
68         <label>Course:</label>
69         <input type="text" name="course[]"
70           placeholder="e.g. Mathematics" required>
71         <label>Credits:</label>
72         <input type="number" name="credits[]"
73           placeholder="e.g. 3" min="1" required>
74         <label>Grade:</label>
75         <select name="grade[]">
76           <option value="4.0">A</option>
77           <option value="3.0">B</option>
78           <option value="2.0">C</option>
79           <option value="1.0">D</option>
80           <option value="0.0">F</option>
81         </select>
82       </div>
83     </div>
84     <button type="button" onclick="addCourse()">
85       + Add Course
86     </button><br><br>
87     <input type="submit" value="Calculate GPA">
88   </form>
89 </body>
90 </html>

```

Listing 5: index.php (Combined File)

## 4 Step 3: Introducing jQuery and Bootstrap

Enhance the application by using jQuery for dynamic course-row management and AJAX submission, and Bootstrap for styling and color-coded alert messages.

### 4.1 Front-end: index.html with jQuery and Bootstrap

This HTML file includes Bootstrap and references a separate JavaScript file (script.js) containing jQuery code.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>GPA Calculator with jQuery and Bootstrap</title>
6   <!-- Bootstrap CSS -->
7   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
8     bootstrap/4.5.2/css/bootstrap.min.css">
9   <link rel="stylesheet" href="style.css">

```

```

10  <!-- jQuery -->
11  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
12  <!-- Bootstrap JS -->
13  <script src="https://stackpath.bootstrapcdn.com/bootstrap/
14      4.5.2/js/bootstrap.min.js"></script>
15  <script src="script.js" defer></script>
16  </head>
17  <body>
18  <div class="container">
19    <h1 class="mt-5">GPA Calculator</h1>
20    <div id="result" class="mt-3"></div>
21    <form id="gpaForm" class="mt-3">
22      <div id="courses">
23        <div class="course-row form-row mb-2">
24          <div class="col">
25            <input type="text" name="course[]" class="form-control"
26              placeholder="Course name" required>
27          </div>
28          <div class="col-2">
29            <input type="number" name="credits[]" class="form-control"
30              placeholder="Credits" min="1" required>
31          </div>
32          <div class="col-2">
33            <select name="grade[]" class="form-control">
34              <option value="4.0">A</option>
35              <option value="3.0">B</option>
36              <option value="2.0">C</option>
37              <option value="1.0">D</option>
38              <option value="0.0">F</option>
39            </select>
40          </div>
41        </div>
42      </div>
43      <button type="button" id="addCourse"
44        class="btn btn-secondary mb-3">
45        + Add Course
46      </button><br>
47      <button type="submit" class="btn btn-primary">
48        Calculate GPA
49      </button>
50    </form>
51  </div>
52 </body>
53 </html>

```

Listing 6: index.html with jQuery and Bootstrap

## 4.2 JavaScript: script.js (jQuery Version for Step 3)

This file contains the jQuery code for dynamic row management, AJAX submission, and Bootstrap alert styling based on the computed GPA.

```

1  $(document).ready(function () {
2
3    // Add a new course row
4    $('#addCourse').click(function () {
5      var row = $('.course-row').first().clone();
6      row.find('input').val('');
7      row.append(
8        '<div class="col-auto">' +
9        '<button type="button" ' +
10         'class="btn btn-danger remove-row">X</button>' +
11        '</div>'

```

```

12     );
13     $('#courses').append(row);
14 });
15
16 // Remove a course row
17 $(document).on('click', '.remove-row', function () {
18     if ($('#course-row').length > 1) {
19         $(this).closest('.course-row').remove();
20     }
21 });
22
23 // Submit via AJAX
24 $('#gpaForm').submit(function (e) {
25     e.preventDefault();
26
27     // Client-side validation
28     var valid = true;
29     $('[name="course[]"]').each(function () {
30         if ($(this).val().trim() === '') valid = false;
31     });
32     $('[name="credits[]"]').each(function () {
33         if (isNaN($(this).val()) || parseFloat($(this).val()) <= 0) {
34             valid = false;
35         }
36     });
37     if (!valid) {
38         $('#result').html(
39             '<div class="alert alert-warning">' +
40             'Please enter valid values in all fields.' +
41             '</div>'
42         );
43         return;
44     }
45
46     $.ajax({
47         url: 'calculate.php',
48         type: 'POST',
49         data: $(this).serialize(),
50         dataType: 'json',
51         success: function (response) {
52             if (response.success) {
53                 var alertClass = 'alert-info';
54                 if (response.gpa >= 3.7) {
55                     alertClass = 'alert-success';
56                 } else if (response.gpa >= 3.0) {
57                     alertClass = 'alert-info';
58                 } else if (response.gpa >= 2.0) {
59                     alertClass = 'alert-warning';
60                 } else {
61                     alertClass = 'alert-danger';
62                 }
63                 $('#result').html(
64                     '<div class="alert ' + alertClass + '">' +
65                     response.message +
66                     '</div>' +
67                     response.tableHtml
68                 );
69             } else {
70                 $('#result').html(
71                     '<div class="alert alert-danger">' +
72                     response.message +
73                     '</div>'
74                 );

```



```

75     }
76   },
77   error: function () {
78     $('#result').html(
79       '<div class="alert alert-danger">' +
80       'Server error occurred.' +
81       '</div>'
82     );
83   }
84 });
85 });
86 });

```

Listing 7: script.js (Step 3)

### 4.3 Back-end: calculate.php (AJAX Version)

This PHP script now returns a JSON response containing the GPA value, the interpretation message, and a Bootstrap-styled HTML table.

```

1 <?php
2 header('Content-Type: application/json');
3
4 if (isset($_POST['course'], $_POST['credits'], $_POST['grade'])) {
5     $courses      = $_POST['course'];
6     $credits       = $_POST['credits'];
7     $grades        = $_POST['grade'];
8     $totalPoints   = 0;
9     $totalCredits  = 0;
10
11     $tableHtml    = '<table class="table table-bordered mt-3">';
12     $tableHtml    .= '<thead class="thead-dark">
13         <tr>
14             <th>Course</th><th>Credits</th>
15             <th>Grade</th><th>Grade Points</th>
16         </tr>
17     </thead><tbody>';
18
19     for ($i = 0; $i < count($courses); $i++) {
20         $course = htmlspecialchars($courses[$i]);
21         $scr     = floatval($credits[$i]);
22         $g       = floatval($grades[$i]);
23         if ($scr <= 0) continue;
24         $pts     = $scr * $g;
25         $totalPoints += $pts;
26         $totalCredits += $scr;
27         $tableHtml .= "<tr>
28             <td>$course</td><td>$scr</td>
29             <td>$g</td><td>$pts</td>
30         </tr>";
31     }
32     $tableHtml .= '</tbody></table>';
33
34     if ($totalCredits > 0) {
35         $gpa = $totalPoints / $totalCredits;
36         if ($gpa >= 3.7) {
37             $interpretation = "Distinction";
38         } elseif ($gpa >= 3.0) {
39             $interpretation = "Merit";
40         } elseif ($gpa >= 2.0) {
41             $interpretation = "Pass";
42         } else {
43             $interpretation = "Fail";

```

```
44     }
45     $message = "Your GPA is " . number_format($gpa, 2)
46               . " ($interpretation).";
47     echo json_encode([
48         'success' => true,
49         'gpa'     => $gpa,
50         'message' => $message,
51         'tableHtml' => $tableHtml,
52     ]);
53 } else {
54     echo json_encode([
55         'success' => false,
56         'message' => 'No valid courses entered.',
57     ]);
58 }
59 } else {
60     echo json_encode([
61         'success' => false,
62         'message' => 'Data not received.',
63     ]);
64 }
65 exit;
66 ?>
```

Listing 8: calculate.php (AJAX version)

## 5 Step 4: Homework

For your homework, extend this lab project by:

- Adding a feature to store and display **previous GPA calculations**, identified by a student name and semester label.
- Integrating a **MySQL database** to save user data (student name, courses, credits, grades, computed GPA, and timestamp).
- Displaying a **Bootstrap progress bar** (0–4 scale) that changes color based on the GPA range: green for Distinction, blue for Merit, yellow for Pass, and red for Fail.
- Adding a **CSV export** button so the user can download the course table as a `.csv` file via a PHP endpoint.
- Enhancing the UI with additional **Bootstrap components** such as modals, cards, or tooltips.
- Adding further **validations and error handling** on both the client and server sides (e.g., maximum credit limit, duplicate course names).

*Be creative and feel free to add any extra features you find interesting!*