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:** GPA ≥ 3.7
- **Merit:** $3.0 \leq$ GPA < 3.7
- **Pass:** $2.0 \leq$ GPA < 3.0
- **Fail:** 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`).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>GPA Calculator</title>
  <link rel="stylesheet" href="style.css">
  <script src="script.js"></script>
</head>
<body>
  <h1>GPA Calculator</h1>
  <form action="calculate.php" method="post"
        onsubmit="return validateForm();">
    <div id="courses">
      <div class="course-row">
        <label>Course:</label>
        <input type="text" name="course[]"
               placeholder="e.g. Mathematics" required>
        <label>Credits:</label>
        <input type="number" name="credits[]"
               placeholder="e.g. 3" min="1" required>
        <label>Grade:</label>
        <select name="grade[]">
          <option value="4.0">A</option>
          <option value="3.0">B</option>
          <option value="2.0">C</option>
          <option value="1.0">D</option>
          <option value="0.0">F</option>
        </select>
      </div>
    </div>
    <button type="button" onclick="addCourse()">
      + Add Course
    </button><br><br>
    <input type="submit" value="Calculate GPA">
  </form>
</body>
</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.

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

```
      '<option value="4.0">A</option>' +
      '<option value="3.0">B</option>' +
      '<option value="2.0">C</option>' +
      '<option value="1.0">D</option>' +
      '<option value="0.0">F</option>' +
    '</select>' +
    '<button type="button" ' +
           'onclick="this.parentNode.remove()">Remove</button>';
  document.getElementById('courses').appendChild(row);
}

function validateForm() {
  var courses = document.querySelectorAll('[name="course[]"]');
  var credits = document.querySelectorAll('[name="credits[]"]');

  for (var i = 0; i < courses.length; i++) {
    if (courses[i].value === "") {
      alert("All course name fields are required.");
      return false;
    }
  }
  for (var j = 0; j < credits.length; j++) {
    if (isNaN(credits[j].value) || credits[j].value <= 0) {
      alert("Credit hours must be positive numbers.");
      return false;
    }
  }
  return true;
}
```

Listing 2: `script.js` (Step 1)

2.3 Style: `style.css`

A simple CSS file for basic styling.

```
body {
  font-family: Arial, sans-serif;
  margin: 20px;
  background-color: #e9ecef;
}
h1 {
  color: #007BFF;
}
.course-row {
  display: flex;
  gap: 10px;
  align-items: center;
  margin-bottom: 10px;
}
.course-row input,
.course-row select {
  padding: 6px;
  border: 1px solid #ccc;
  border-radius: 4px;
}
table {
  border-collapse: collapse;
  margin-top: 16px;
}
table th, table td {
  border: 1px solid #ccc;
  padding: 8px 14px;
```

```
  text-align: center;
}
table th {
  background-color: #007BFF;
  color: white;
}
```

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.

```
<?php
if (isset($_POST['course'], $_POST['credits'], $_POST['grade'])) {
  $courses      = $_POST['course'];
  $credits      = $_POST['credits'];
  $grades       = $_POST['grade'];
  $totalPoints  = 0;
  $totalCredits = 0;

  echo "<table>";
  echo "<tr>
          <th>Course</th>
          <th>Credits</th>
          <th>Grade</th>
          <th>Grade Points</th>
        </tr>";

  for ($i = 0; $i < count($courses); $i++) {
    $course = htmlspecialchars($courses[$i]);
    $cr     = floatval($credits[$i]);
    $g      = floatval($grades[$i]);
    if ($cr <= 0) continue;
    $pts           = $cr * $g;
    $totalPoints  += $pts;
    $totalCredits += $cr;
    echo "<tr>
            <td>$course</td>
            <td>$cr</td>
            <td>$g</td>
            <td>$pts</td>
          </tr>";
  }
  echo "</table>";

  if ($totalCredits > 0) {
    $gpa = $totalPoints / $totalCredits;
    if ($gpa >= 3.7) {
      $interpretation = "Distinction";
    } elseif ($gpa >= 3.0) {
      $interpretation = "Merit";
    } elseif ($gpa >= 2.0) {
      $interpretation = "Pass";
    } else {
      $interpretation = "Fail";
    }
    echo "<p>Your GPA is <strong>" . number_format($gpa, 2)
         . "</strong> ($interpretation).</p>";
  } else {
    echo "<p>No valid courses entered.</p>";
  }
```

```
} else {
  echo "Data not received.";
}
?>
```

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.

```
<?php
$result    = "";
$tableHtml = "";

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  $courses      = $_POST['course']  ?? [];
  $credits      = $_POST['credits'] ?? [];
  $grades       = $_POST['grade']   ?? [];
  $totalPoints  = 0;
  $totalCredits = 0;

  $tableHtml  = "<table>";
  $tableHtml .= "<tr>
                   <th>Course</th><th>Credits</th>
                   <th>Grade</th><th>Grade Points</th>
                 </tr>";

  for ($i = 0; $i < count($courses); $i++) {
    $course = htmlspecialchars($courses[$i]);
    $cr     = floatval($credits[$i]);
    $g      = floatval($grades[$i]);
    if ($cr <= 0) continue;
    $pts           = $cr * $g;
    $totalPoints  += $pts;
    $totalCredits += $cr;
    $tableHtml .= "<tr>
                     <td>$course</td><td>$cr</td>
                     <td>$g</td><td>$pts</td>
                   </tr>";
  }
  $tableHtml .= "</table>";

  if ($totalCredits > 0) {
    $gpa = $totalPoints / $totalCredits;
    if ($gpa >= 3.7) {
      $interpretation = "Distinction";
    } elseif ($gpa >= 3.0) {
      $interpretation = "Merit";
    } elseif ($gpa >= 2.0) {
      $interpretation = "Pass";
    } else {
      $interpretation = "Fail";
    }
    $result = "Your GPA is " . number_format($gpa, 2)
            . " ($interpretation).";
  } else {
    $result = "No valid courses entered.";
  }
}
?>
```

```php
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>GPA Calculator</title>
  <link rel="stylesheet" href="style.css">
  <script src="script.js"></script>
</head>
<body>
  <h1>GPA Calculator</h1>
  <?php if ($result != ""): ?>
    <?php echo $tableHtml; ?>
    <p><strong><?= $result ?></strong></p>
  <?php endif; ?>
  <form action="" method="post" onsubmit="return validateForm();">
    <div id="courses">
      <div class="course-row">
        <label>Course:</label>
        <input type="text" name="course[]"
               placeholder="e.g. Mathematics" required>
        <label>Credits:</label>
        <input type="number" name="credits[]"
               placeholder="e.g. 3" min="1" required>
        <label>Grade:</label>
        <select name="grade[]">
          <option value="4.0">A</option>
          <option value="3.0">B</option>
          <option value="2.0">C</option>
          <option value="1.0">D</option>
          <option value="0.0">F</option>
        </select>
      </div>
    </div>
    <button type="button" onclick="addCourse()">
      + Add Course
    </button><br><br>
    <input type="submit" value="Calculate GPA">
  </form>
</body>
</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.

```php
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>GPA Calculator with jQuery and Bootstrap</title>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
        bootstrap/4.5.2/css/bootstrap.min.css">
  <link rel="stylesheet" href="style.css">
```

```
  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <!-- Bootstrap JS -->
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/
        4.5.2/js/bootstrap.min.js"></script>
  <script src="script.js" defer></script>
</head>
<body>
<div class="container">
  <h1 class="mt-5">GPA Calculator</h1>
  <div id="result" class="mt-3"></div>
  <form id="gpaForm" class="mt-3">
    <div id="courses">
      <div class="course-row form-row mb-2">
        <div class="col">
          <input type="text" name="course[]" class="form-control"
                 placeholder="Course name" required>
        </div>
        <div class="col-2">
          <input type="number" name="credits[]" class="form-control"
                 placeholder="Credits" min="1" required>
        </div>
        <div class="col-2">
          <select name="grade[]" class="form-control">
            <option value="4.0">A</option>
            <option value="3.0">B</option>
            <option value="2.0">C</option>
            <option value="1.0">D</option>
            <option value="0.0">F</option>
          </select>
        </div>
      </div>
    </div>
    <button type="button" id="addCourse"
            class="btn btn-secondary mb-3">
      + Add Course
    </button><br>
    <button type="submit" class="btn btn-primary">
      Calculate GPA
    </button>
  </form>
</div>
</body>
</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.

```
$(document).ready(function () {

  // Add a new course row
  $('#addCourse').click(function () {
    var row = $('.course-row').first().clone();
    row.find('input').val('');
    row.append(
      '<div class="col-auto">' +
        '<button type="button" ' +
                'class="btn btn-danger remove-row">X</button>' +
      '</div>'
```

```
    );
    $('#courses').append(row);
  });

  // Remove a course row
  $(document).on('click', '.remove-row', function () {
    if ($('.course-row').length > 1) {
      $(this).closest('.course-row').remove();
    }
  });

  // Submit via AJAX
  $('#gpaForm').submit(function (e) {
    e.preventDefault();

    // Client-side validation
    var valid = true;
    $('[name="course[]"]').each(function () {
      if ($(this).val().trim() === '') valid = false;
    });
    $('[name="credits[]"]').each(function () {
      if (isNaN($(this).val()) || parseFloat($(this).val()) <= 0) {
        valid = false;
      }
    });
    if (!valid) {
      $('#result').html(
        '<div class="alert alert-warning">' +
          'Please enter valid values in all fields.' +
        '</div>'
      );
      return;
    }

    $.ajax({
      url: 'calculate.php',
      type: 'POST',
      data: $(this).serialize(),
      dataType: 'json',
      success: function (response) {
        if (response.success) {
          var alertClass = 'alert-info';
          if (response.gpa >= 3.7) {
            alertClass = 'alert-success';
          } else if (response.gpa >= 3.0) {
            alertClass = 'alert-info';
          } else if (response.gpa >= 2.0) {
            alertClass = 'alert-warning';
          } else {
            alertClass = 'alert-danger';
          }
          $('#result').html(
            '<div class="alert ' + alertClass + '">' +
              response.message +
            '</div>' +
            response.tableHtml
          );
        } else {
          $('#result').html(
            '<div class="alert alert-danger">' +
              response.message +
            '</div>'
          );
```

```
        }
      },
      error: function () {
        $('#result').html(
          '<div class="alert alert-danger">' +
            'Server error occurred.' +
          '</div>'
        );
      }
    });
  });
});
```

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.

```
<?php
header('Content-Type: application/json');

if (isset($_POST['course'], $_POST['credits'], $_POST['grade'])) {
  $courses      = $_POST['course'];
  $credits      = $_POST['credits'];
  $grades       = $_POST['grade'];
  $totalPoints  = 0;
  $totalCredits = 0;

  $tableHtml  = '<table class="table table-bordered mt-3">';
  $tableHtml .= '<thead class="thead-dark">
                   <tr>
                     <th>Course</th><th>Credits</th>
                     <th>Grade</th><th>Grade Points</th>
                   </tr>
                 </thead><tbody>';

  for ($i = 0; $i < count($courses); $i++) {
    $course = htmlspecialchars($courses[$i]);
    $cr     = floatval($credits[$i]);
    $g      = floatval($grades[$i]);
    if ($cr <= 0) continue;
    $pts           = $cr * $g;
    $totalPoints  += $pts;
    $totalCredits += $cr;
    $tableHtml .= "<tr>
                     <td>$course</td><td>$cr</td>
                     <td>$g</td><td>$pts</td>
                   </tr>";
  }
  $tableHtml .= '</tbody></table>';

  if ($totalCredits > 0) {
    $gpa = $totalPoints / $totalCredits;
    if ($gpa >= 3.7) {
      $interpretation = "Distinction";
    } elseif ($gpa >= 3.0) {
      $interpretation = "Merit";
    } elseif ($gpa >= 2.0) {
      $interpretation = "Pass";
    } else {
      $interpretation = "Fail";
```

```php
    }
    $message = "Your GPA is " . number_format($gpa, 2)
             . " ($interpretation).";
    echo json_encode([
      'success'   => true,
      'gpa'       => $gpa,
      'message'   => $message,
      'tableHtml' => $tableHtml,
    ]);
  } else {
    echo json_encode([
      'success' => false,
      'message' => 'No valid courses entered.',
    ]);
  }
} else {
  echo json_encode([
    'success' => false,
    'message' => 'Data not received.',
  ]);
}
exit;
?>
```

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!*