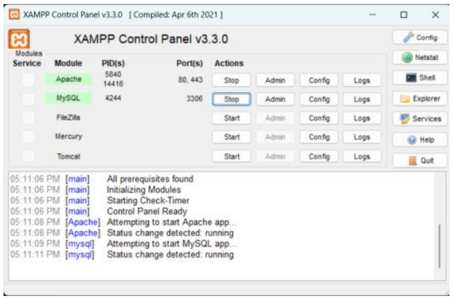
PROJECT OF A STUDENT ENTRY FORM *(PDO METHOD)*

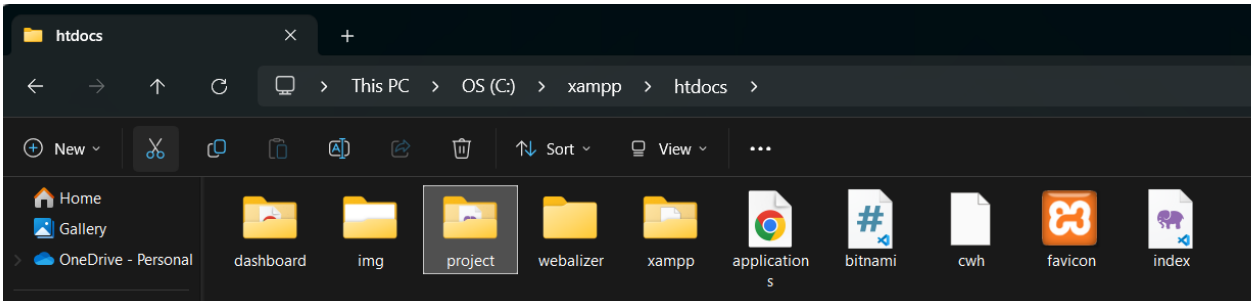
1. Download Set Up and Configuration of XAMPP Server


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Config
Netstat
Shell
Explorer
Services
Help
Quit

Modules Service	Module	PID(s)	Port(s)	Actions			
	Apache	5840 14416	80, 443	Stop	Admin	Config	Logs
	MySQL	4244	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
05:11:06 PM  [main]    All prerequisites found
05:11:06 PM  [main]    Initializing Modules
05:11:06 PM  [main]    Starting Check-Timer
05:11:06 PM  [main]    Control Panel Ready
05:11:08 PM  [Apache]  Attempting to start Apache app...
05:11:08 PM  [Apache]  Status change detected: running
05:11:09 PM  [mysql]   Attempting to start MySQL app...
05:11:11 PM  [mysql]   Status change detected: running
```

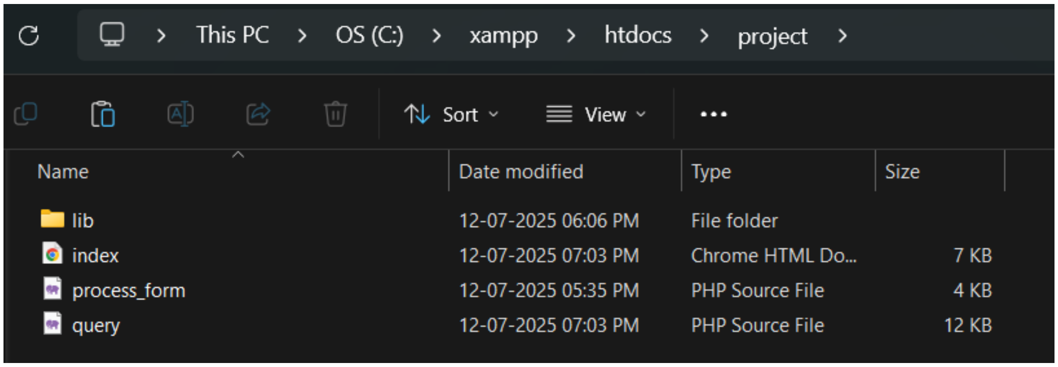
2. Create a file in *C:\xampp\htdocs*



Project Folder

Here we set up the files for the project.

3. File Over View: *C:\xampp\htdocs/project*



Name	Date modified	Type	Size
lib	12-07-2025 06:06 PM	File folder	
index	12-07-2025 07:03 PM	Chrome HTML Do...	7 KB
process_form	12-07-2025 05:35 PM	PHP Source File	4 KB
query	12-07-2025 07:03 PM	PHP Source File	12 KB

index.html with internal css: Contains the UI and form Structure.

process_form.php : Receives, validates, and inserts submitted student records into the database (students table).

Key Features:

- Connects to MySQL using **PDO** (secure & modern)
- Validates each entry:
  - Unique roll number
  - No digits in name
  - Valid semester (1–8)
  - Result must be one of *Pass*, *Fail*, or *Supplementary*
- Skips duplicates and invalid entries
- Shows a success message if all records are inserted
- Displays detailed error messages for each failed entry
-

query.php : To view**,** filter, and export student records from the students database table.

https://www.fpdf.org/en/download.php ***(download the latest version ZIP File)***

1. Extract the folder and move the fpdf file to the lib folder of the project Folder

C:\xampp\htdocs/project/lib/fpdf/fpdf.php

2. Also move the font folder from the downloaded ZIP file to the file location of

C:\xampp\htdocs\project\lib\fpdf\font

lib/ *here is the library for php PDF generation*

```
└── fpdf/
    ├── fpdf.php
    └── font/
```

4. Run the localhost in the browser

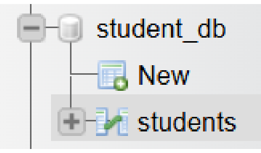
http://localhost/phpmyadmin/

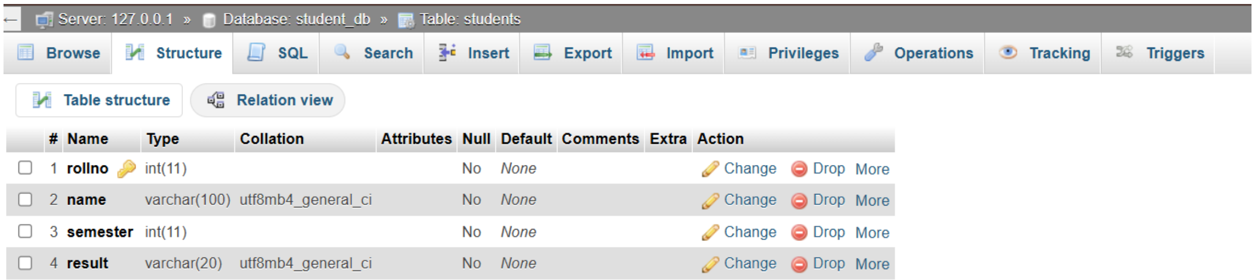
here we have to create a database :

```
CREATE DATABASE student_db;
    USE DATABASE student_db;
    CREATE TABLE students (
            rollno INT(11) PRIMARY KEY,
            name VARCHAR(100) NOT NULL,
            semester INT(11) NOT NULL,
            result VARCHAR(20) NOT NULL,
    );
```

The database name is student_db

The table name is students;



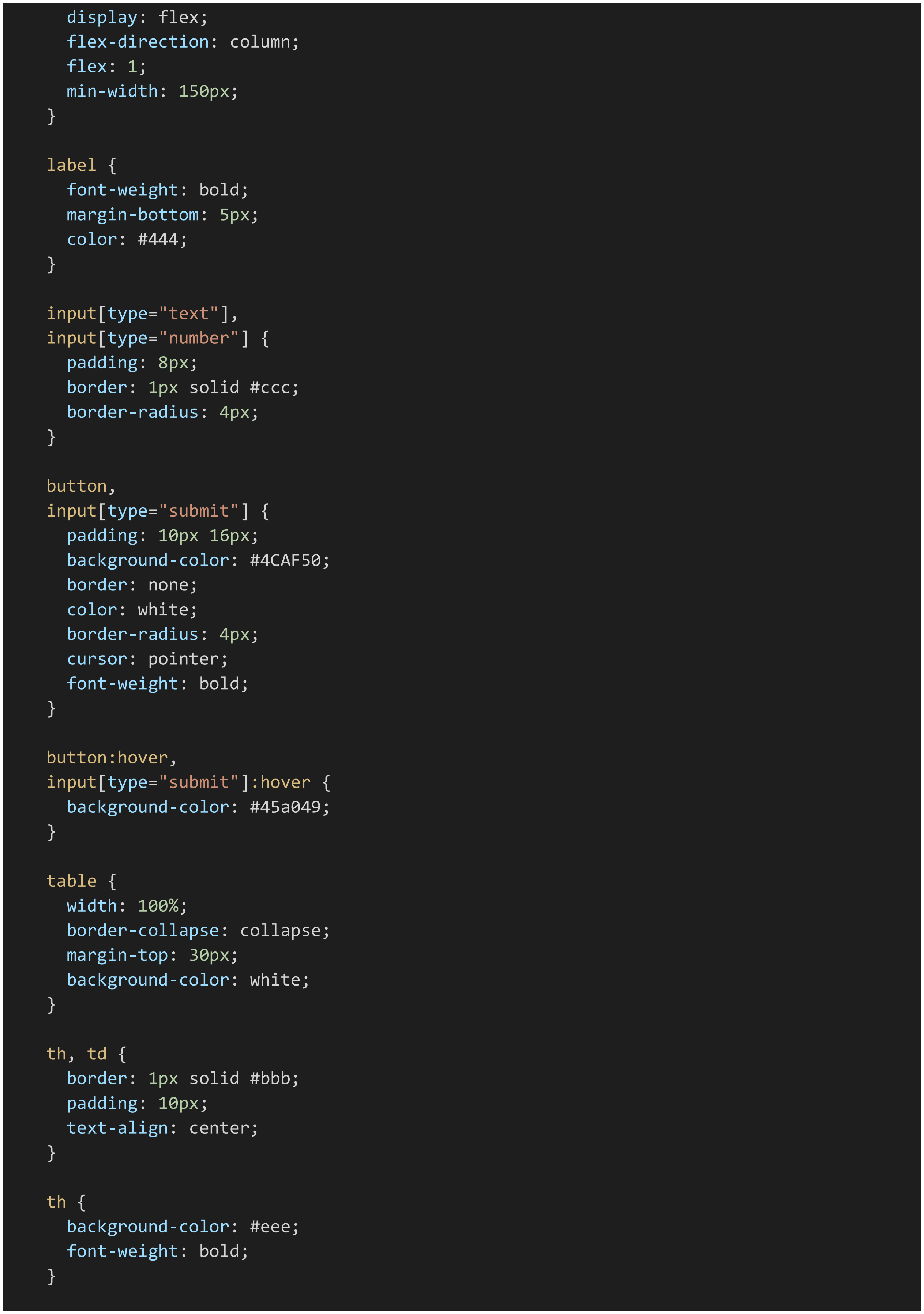


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**rollno** 🔑	int(11)			No	*None*			Change Drop More
□	2	**name**	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	**semester**	int(11)			No	*None*			Change Drop More
□	4	**result**	varchar(20)	utf8mb4_general_ci		No	*None*			Change Drop More

CODES

(i) index.html

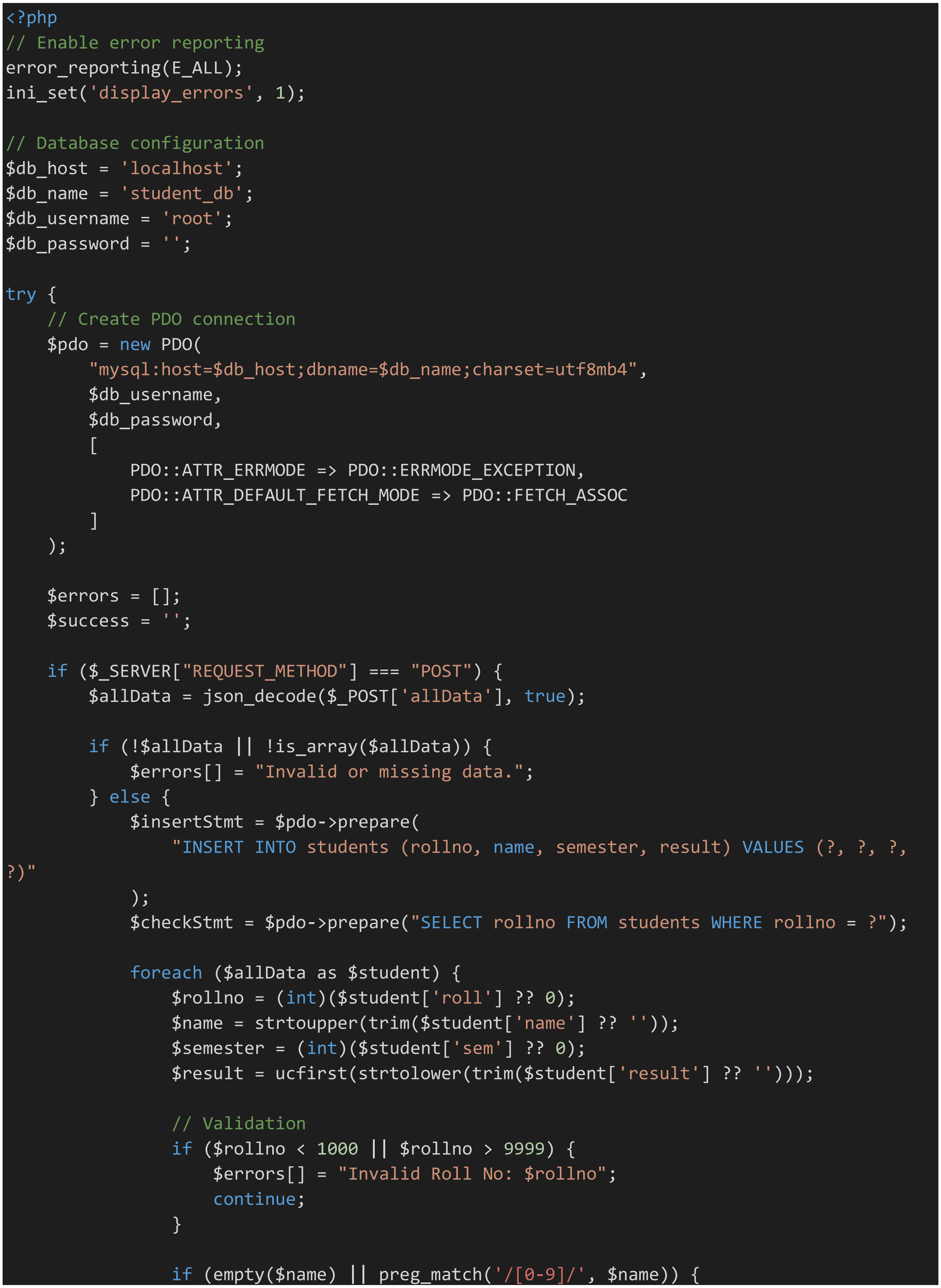
```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Student Database System</title>
  <style>
    body {
      font-family: 'Segoe UI', sans-serif;
      max-width: 900px;
      margin: 0 auto;
      padding: 20px;
      background-color: #f5f5f5;
    }

    .navbar {
      background-color: #2c2c2c;
      overflow: hidden;
      margin-bottom: 30px;
      border-radius: 4px 4px 0 0;
    }

    .navbar a {
      float: left;
      display: block;
      color: white;
      text-align: center;
      padding: 14px 20px;
      text-decoration: none;
      font-weight: bold;
    }

    .navbar a.active {
      background-color: #4CAF50;
    }

    h2 {
      margin-bottom: 20px;
      color: #333;
    }

    .form-group {
      display: flex;
      flex-wrap: wrap;
      gap: 15px;
      align-items: flex-end;
    }

    .form-control {
```



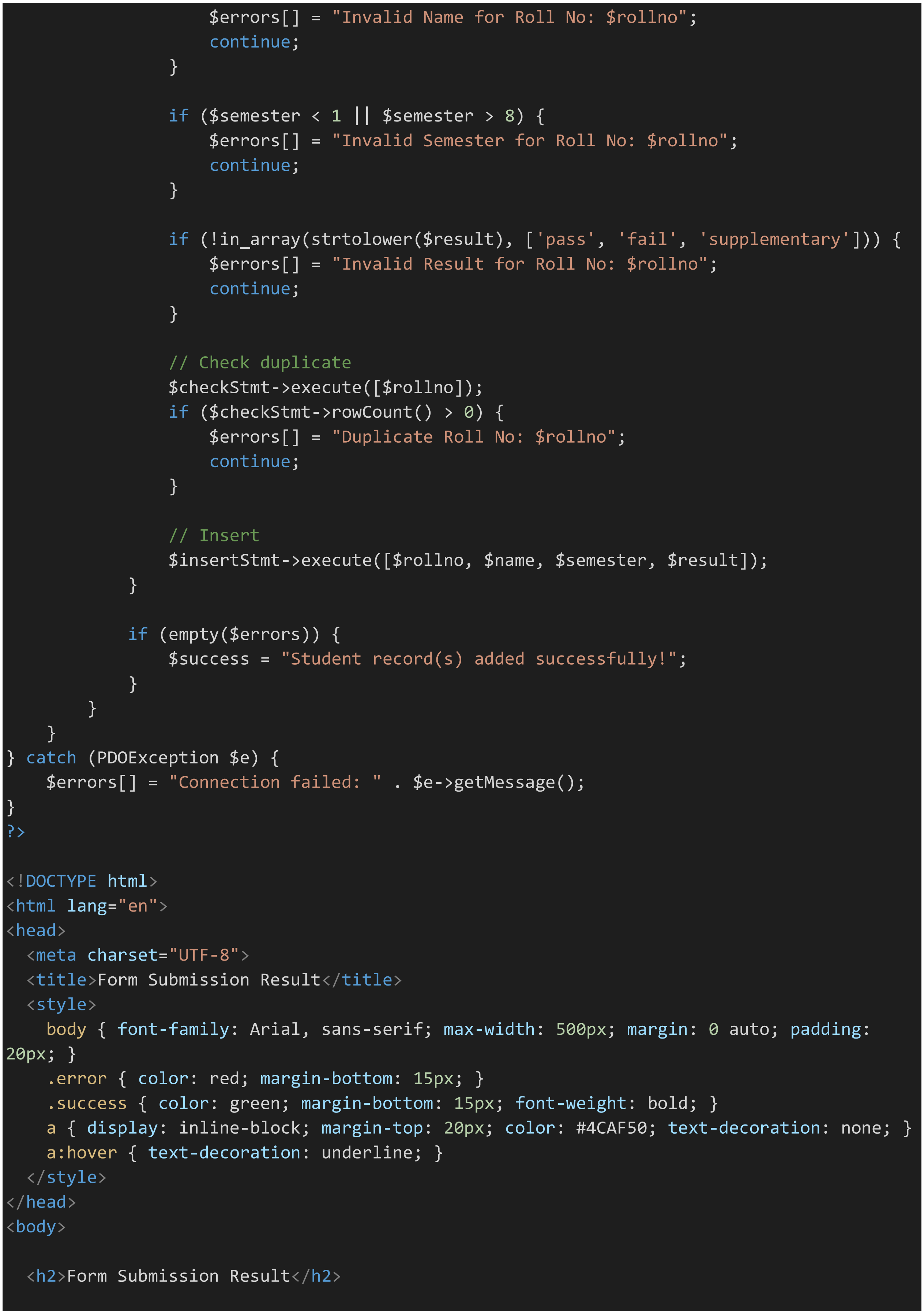
```
      display: flex;
      flex-direction: column;
      flex: 1;
      min-width: 150px;
    }

    label {
      font-weight: bold;
      margin-bottom: 5px;
      color: #444;
    }

    input[type="text"],
    input[type="number"] {
      padding: 8px;
      border: 1px solid #ccc;
      border-radius: 4px;
    }

    button,
    input[type="submit"] {
      padding: 10px 16px;
      background-color: #4CAF50;
      border: none;
      color: white;
      border-radius: 4px;
      cursor: pointer;
      font-weight: bold;
    }

    button:hover,
    input[type="submit"]:hover {
      background-color: #45a049;
    }

    table {
      width: 100%;
      border-collapse: collapse;
      margin-top: 30px;
      background-color: white;
    }

    th, td {
      border: 1px solid #bbb;
      padding: 10px;
      text-align: center;
    }

    th {
      background-color: #eee;
      font-weight: bold;
    }
    .action-btn {
      padding: 6px 10px;
      background-color: #f44336;
      border: none;
      color: white;
      border-radius: 4px;
      margin: 0 2px;
      cursor: pointer;
    }

    .action-btn.edit {
      background-color: #2196F3;
    }

    .action-btn:hover {
      opacity: 0.9;
    }

    marquee {
      margin-top: 20px;
      background-color: #fff3cd;
      color: #856404;
      padding: 10px;
      border: 1px solid #ffeeba;
      border-radius: 4px;
      font-weight: bold;
    }
  </style>
</head>
<body>

  <div class="navbar">
    <a href="index.html" class="active">Entry</a>
    <a href="query.php">Query</a>
    <a href="#" onclick="exitPage()" style="float: right; background-color:
#f44336;">Exit</a>


  </div>

  <h2>Student Entry Form</h2>

  <!-- Input Fields -->
  <div class="form-group">
    <div class="form-control">
      <label for="rollno">Roll No (1000-9999)</label>
      <input type="number" id="rollno" min="1000" max="9999" required>
    </div>
    <div class="form-control">
      <label for="name">Name</label>
      <input type="text" id="name" oninput="this.value=this.value.toUpperCase()"
required>
    </div>
    <div class="form-control">
      <label for="semester">Semester (1-8)</label>
      <input type="number" id="semester" min="1" max="8" required>
    </div>
    <div class="form-control">
      <label for="result">Result (Pass/Fail/Supp)</label>
      <input type="text" id="result" required>
    </div>
    <div class="form-control" style="margin-top: 25px;">
      <button onclick="addRow()">Add</button>
    </div>
  </div>

  <!-- Display Table -->
  <form action="process_form.php" method="post" onsubmit="return confirmSubmission()">
    <table id="studentTable">
      <thead>
        <tr>
          <th>Roll No.</th>
          <th>Name</th>
          <th>Semester</th>
          <th>Result</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody></tbody>
    </table>

    <br>
    <input type="submit" value="FINAL SUBMIT">
    <input type="hidden" name="allData" id="allData">
  </form>

  <marquee behavior="scroll" direction="left">
    Once the data is FINAL SUBMIT, it cannot be changed. Please double-check your entries before submission.
  </marquee>

  <script>
    const tableBody = document.querySelector('#studentTable tbody');
    let studentList = [];

    function addRow() {
      const roll = document.getElementById('rollno').value.trim();
      const name = document.getElementById('name').value.trim();
      const sem = document.getElementById('semester').value.trim();
      const result = document.getElementById('result').value.trim();

      if (!roll || !name || !sem || !result) return alert("All fields required!");
      if (roll < 1000 || roll > 9999) return alert("Roll No must be 1000-9999!");
      if (/\d/.test(name)) return alert("Name cannot contain numbers!");
      if (sem < 1 || sem > 8) return alert("Semester must be between 1-8");
      const student = { roll, name, sem, result };
      studentList.push(student);
      renderTable();
      clearInputs();
    }

    function renderTable() {
      tableBody.innerHTML = '';
      studentList.forEach((stu, index) => {
        const row = `<tr>
            <td>${stu.roll}</td>
            <td>${stu.name}</td>
            <td>${stu.sem}</td>
            <td>${stu.result}</td>
            <td>
              <button type="button" class="action-btn edit"
onclick="editRow(${index})">Edit</button>
              <button type="button" class="action-btn"
onclick="deleteRow(${index})">Delete</button>
            </td>
        </tr>`;
        tableBody.innerHTML += row;
      });
    }

    function clearInputs() {
      document.getElementById('rollno').value = '';
      document.getElementById('name').value = '';
      document.getElementById('semester').value = '';
      document.getElementById('result').value = '';
    }

    function deleteRow(index) {
      studentList.splice(index, 1);
      renderTable();
    }

    function editRow(index) {
      const stu = studentList[index];
      document.getElementById('rollno').value = stu.roll;
      document.getElementById('name').value = stu.name;
      document.getElementById('semester').value = stu.sem;
      document.getElementById('result').value = stu.result;
      deleteRow(index);
    }

    function confirmSubmission() {
      if (studentList.length === 0) {
        alert("No data to submit.");
        return false;
      }
      const confirmMsg = confirm("Are you sure you want to submit? Once submitted, data
cannot be changed.");
```

```
   if (confirmMsg) {
    document.getElementById('allData').value = JSON.stringify(studentList);
    return true;
   } else {
    return false;
   }
  }

  //Exit Button
  function exitPage() {
 if (confirm("Are you sure you want to exit?")) {
  alert("Please click the browser's ⍰ button to close this tab.");
 }
}

 </script>

</body>
</html>
```

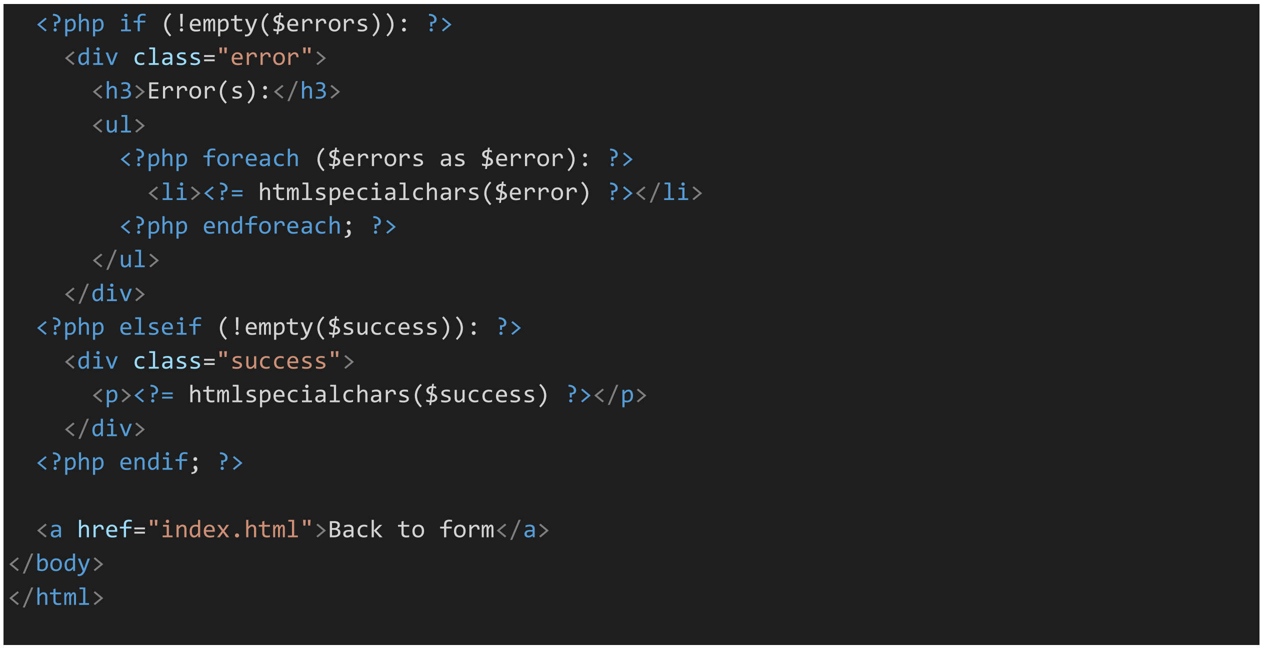
(ii) process_form.php



```
<?php
// Enable error reporting
error_reporting(E_ALL);
ini_set('display_errors', 1);

// Database configuration
$db_host = 'localhost';
$db_name = 'student_db';
$db_username = 'root';
$db_password = '';

try {
    // Create PDO connection
    $pdo = new PDO(
        "mysql:host=$db_host;dbname=$db_name;charset=utf8mb4",
        $db_username,
        $db_password,
        [
            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
            PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
        ]
    );

    $errors = [];
    $success = '';

    if ($_SERVER["REQUEST_METHOD"] === "POST") {
        $allData = json_decode($_POST['allData'], true);

        if (!$allData || !is_array($allData)) {
            $errors[] = "Invalid or missing data.";
        } else {
            $insertStmt = $pdo->prepare(
                "INSERT INTO students (rollno, name, semester, result) VALUES (?, ?, ?,
?)"
            );
            $checkStmt = $pdo->prepare("SELECT rollno FROM students WHERE rollno = ?");

            foreach ($allData as $student) {
                $rollno = (int)($student['roll'] ?? 0);
                $name = strtoupper(trim($student['name'] ?? ''));
                $semester = (int)($student['sem'] ?? 0);
                $result = ucfirst(strtolower(trim($student['result'] ?? '')));

                // Validation
                if ($rollno < 1000 || $rollno > 9999) {
                    $errors[] = "Invalid Roll No: $rollno";
                    continue;
                }

                if (empty($name) || preg_match('/[0-9]/', $name)) {
```



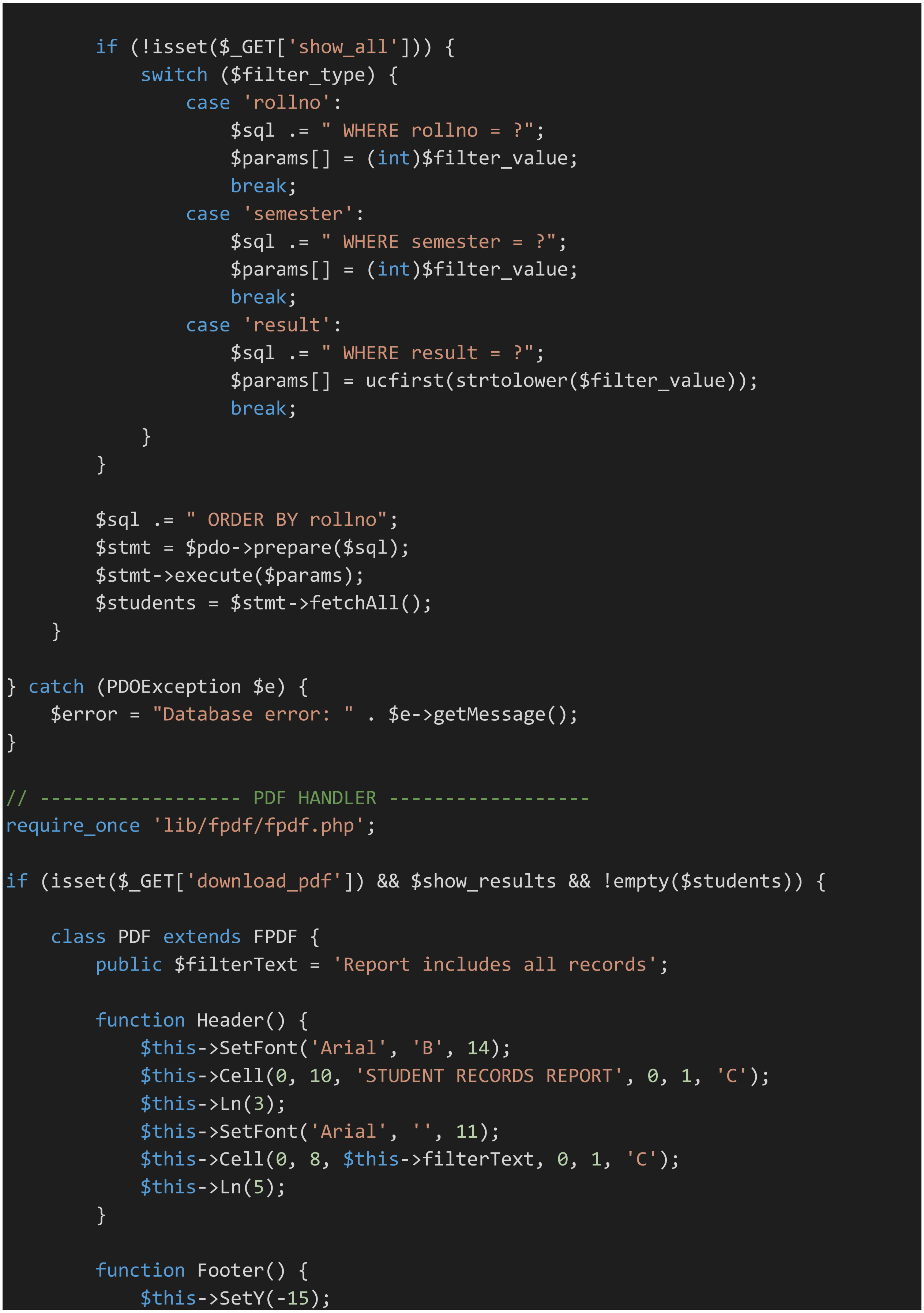
```
                    $errors[] = "Invalid Name for Roll No: $rollno";
                    continue;
                }

                if ($semester < 1 || $semester > 8) {
                    $errors[] = "Invalid Semester for Roll No: $rollno";
                    continue;
                }

                if (!in_array(strtolower($result), ['pass', 'fail', 'supplementary'])) {
                    $errors[] = "Invalid Result for Roll No: $rollno";
                    continue;
                }

                // Check duplicate
                $checkStmt->execute([$rollno]);
                if ($checkStmt->rowCount() > 0) {
                    $errors[] = "Duplicate Roll No: $rollno";
                    continue;
                }

                // Insert
                $insertStmt->execute([$rollno, $name, $semester, $result]);
            }

            if (empty($errors)) {
                $success = "Student record(s) added successfully!";
            }
        }
    }
} catch (PDOException $e) {
    $errors[] = "Connection failed: " . $e->getMessage();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Form Submission Result</title>
  <style>
    body { font-family: Arial, sans-serif; max-width: 500px; margin: 0 auto; padding:
20px; }
    .error { color: red; margin-bottom: 15px; }
    .success { color: green; margin-bottom: 15px; font-weight: bold; }
    a { display: inline-block; margin-top: 20px; color: #4CAF50; text-decoration: none; }
    a:hover { text-decoration: underline; }
  </style>
</head>
<body>

  <h2>Form Submission Result</h2>
```



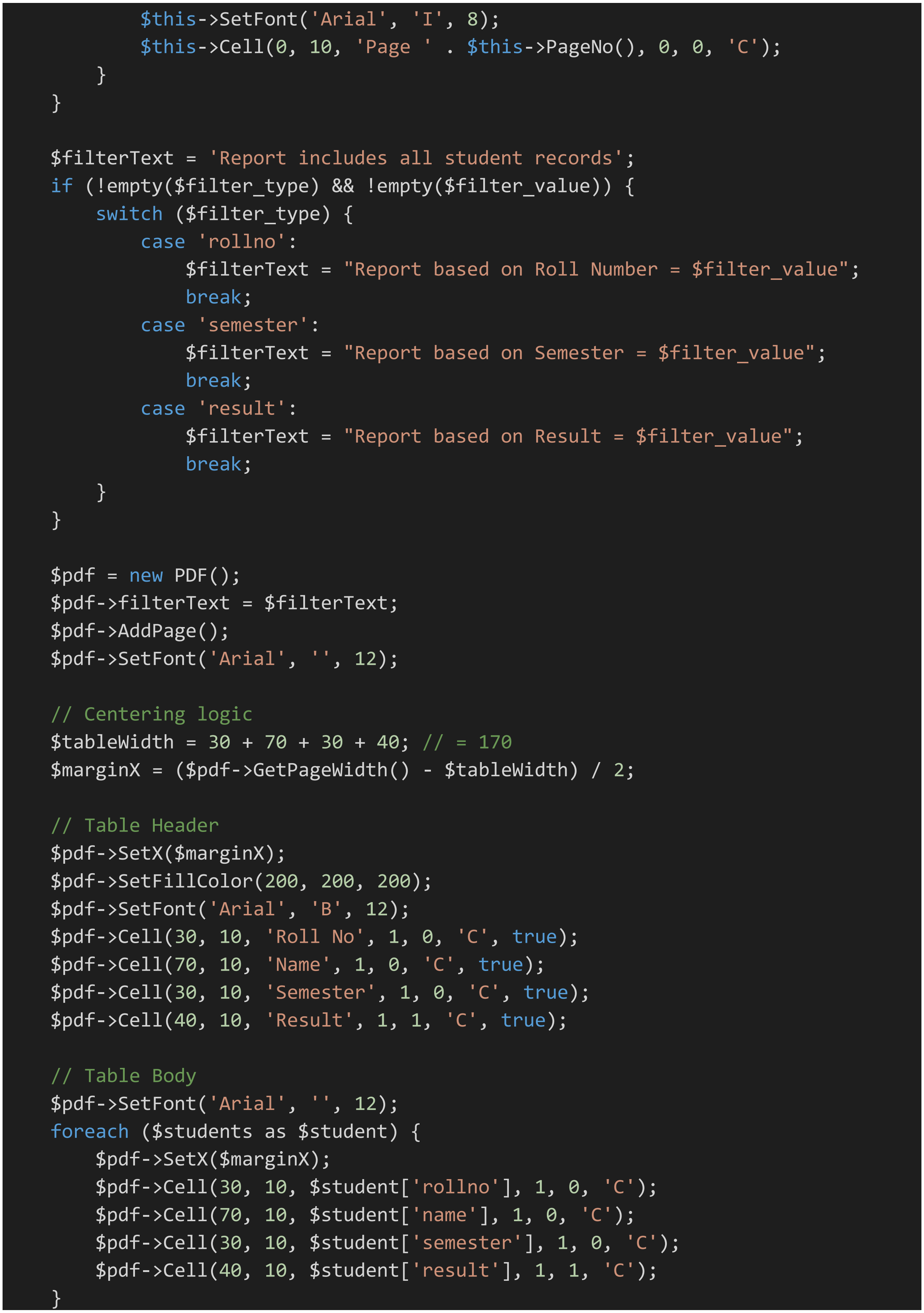
```
  <?php if (!empty($errors)): ?>
    <div class="error">
      <h3>Error(s):</h3>
      <ul>
        <?php foreach ($errors as $error): ?>
          <li><?= htmlspecialchars($error) ?></li>
        <?php endforeach; ?>
      </ul>
    </div>
  <?php elseif (!empty($success)): ?>
    <div class="success">
      <p><?= htmlspecialchars($success) ?></p>
    </div>
  <?php endif; ?>

  <a href="index.html">Back to form</a>
</body>
</html>
```

(iii) query.php

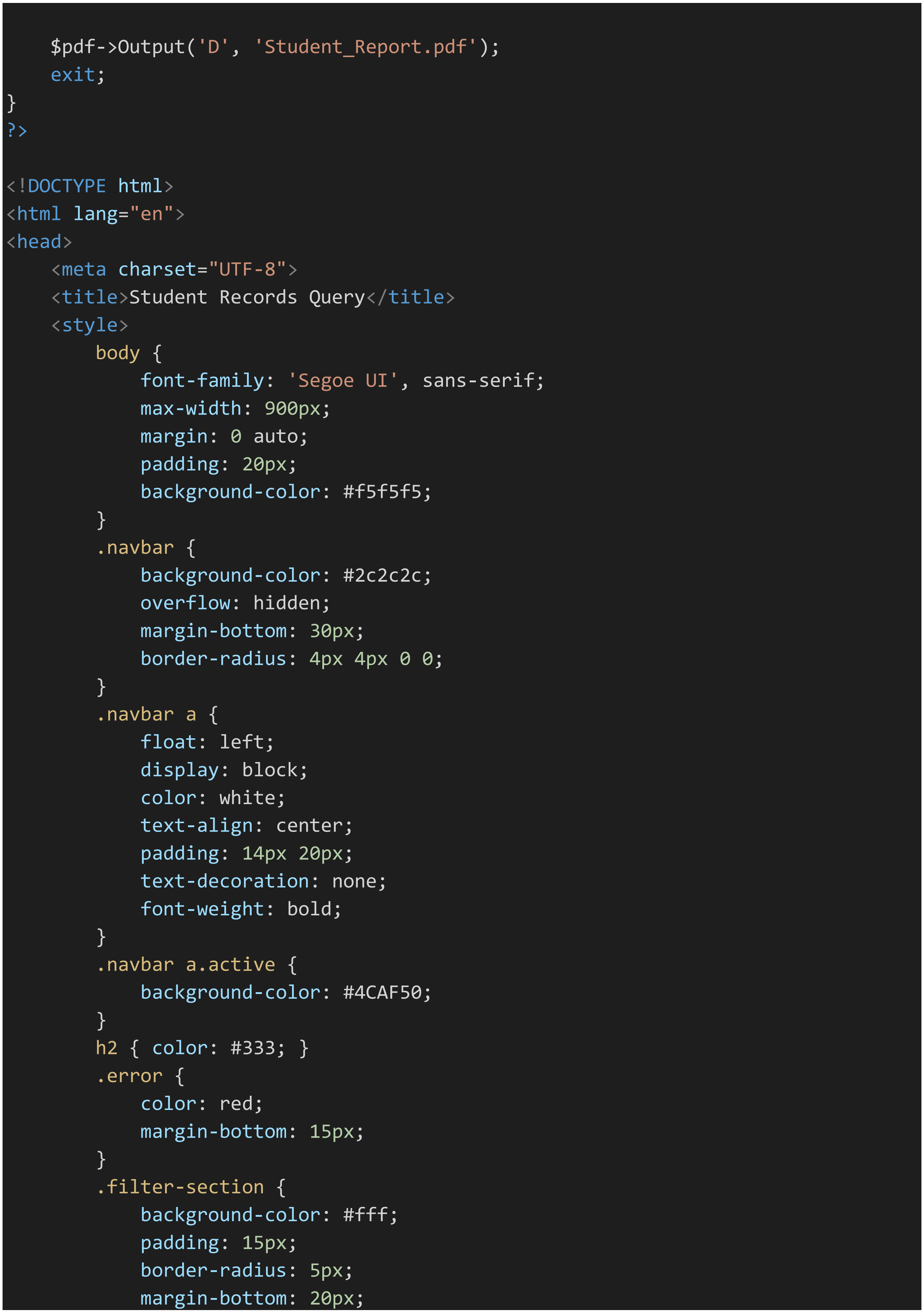
```php
<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);

$db_host = 'localhost';
$db_name = 'student_db';
$db_username = 'root';
$db_password = '';

$filter_type = $_GET['filter'] ?? '';
$rollno_value = $_GET['rollno'] ?? '';
$semester_value = $_GET['semester'] ?? '';
$result_value = $_GET['result'] ?? '';
$students = [];
$error = '';
$show_results = false;

$filter_value = '';
if ($filter_type === 'rollno') {
    $filter_value = $rollno_value;
} elseif ($filter_type === 'semester') {
    $filter_value = $semester_value;
} elseif ($filter_type === 'result') {
    $filter_value = $result_value;
} elseif (isset($_GET['show_all'])) {
    $show_results = true;
}

if ((isset($_GET['filter']) && $filter_value !== '') || isset($_GET['show_all']))
{
    $show_results = true;
} elseif (isset($_GET['query'])) {
    $error = "Please select a filter type and provide a corresponding value.";
}

try {
    $pdo = new PDO("mysql:host=$db_host;dbname=$db_name;charset=utf8mb4",
$db_username, $db_password, [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
    ]);

    if ($show_results) {
        $sql = "SELECT rollno, name, semester, result FROM students";
        $params = [];
```



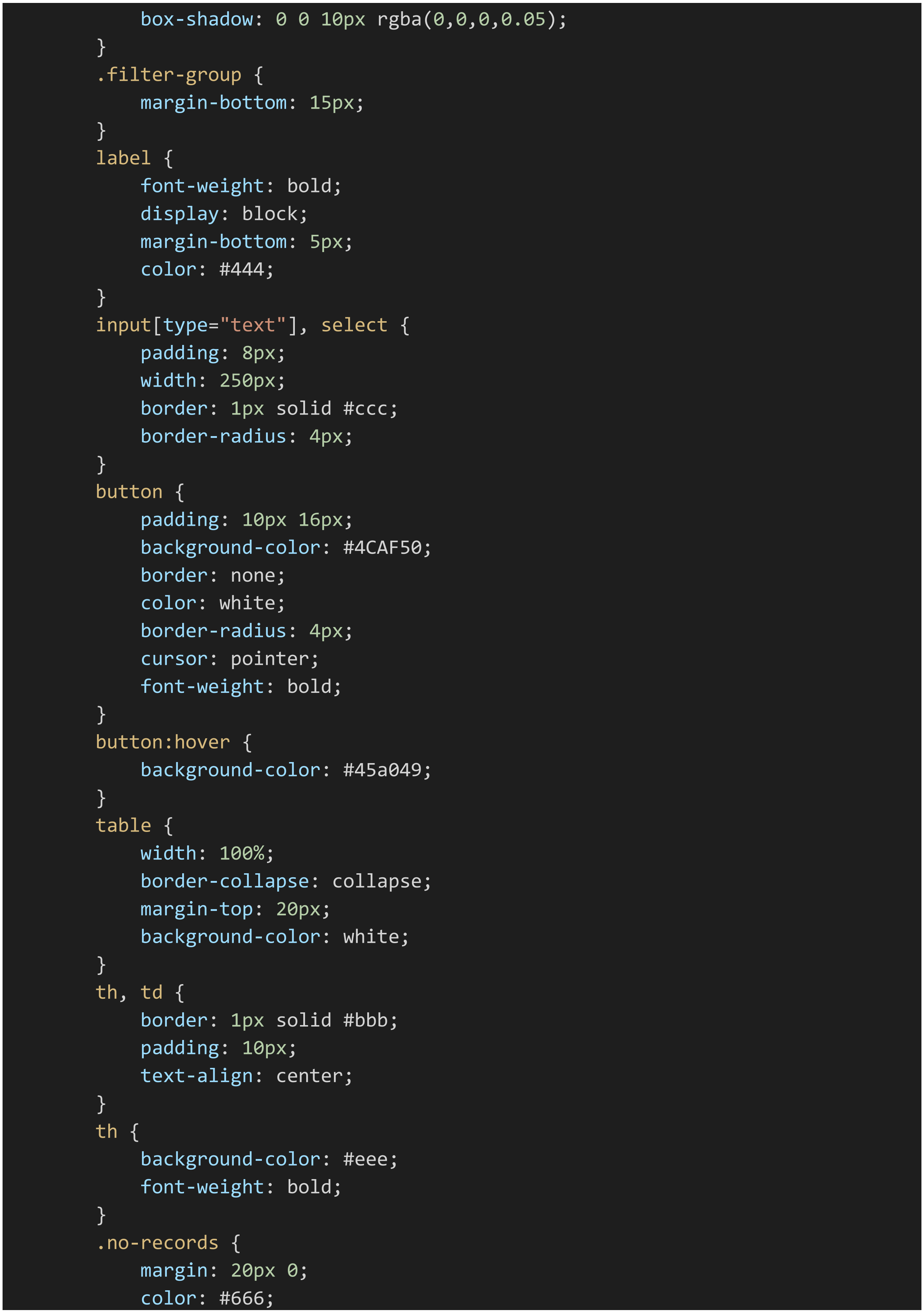
```
        if (!isset($_GET['show_all'])) {
            switch ($filter_type) {
                case 'rollno':
                    $sql .= " WHERE rollno = ?";
                    $params[] = (int)$filter_value;
                    break;
                case 'semester':
                    $sql .= " WHERE semester = ?";
                    $params[] = (int)$filter_value;
                    break;
                case 'result':
                    $sql .= " WHERE result = ?";
                    $params[] = ucfirst(strtolower($filter_value));
                    break;
            }
        }

        $sql .= " ORDER BY rollno";
        $stmt = $pdo->prepare($sql);
        $stmt->execute($params);
        $students = $stmt->fetchAll();
    }

} catch (PDOException $e) {
    $error = "Database error: " . $e->getMessage();
}

// ------------------ PDF HANDLER ------------------
require_once 'lib/fpdf/fpdf.php';

if (isset($_GET['download_pdf']) && $show_results && !empty($students)) {

    class PDF extends FPDF {
        public $filterText = 'Report includes all records';

        function Header() {
            $this->SetFont('Arial', 'B', 14);
            $this->Cell(0, 10, 'STUDENT RECORDS REPORT', 0, 1, 'C');
            $this->Ln(3);
            $this->SetFont('Arial', '', 11);
            $this->Cell(0, 8, $this->filterText, 0, 1, 'C');
            $this->Ln(5);
        }

        function Footer() {
            $this->SetY(-15);
```



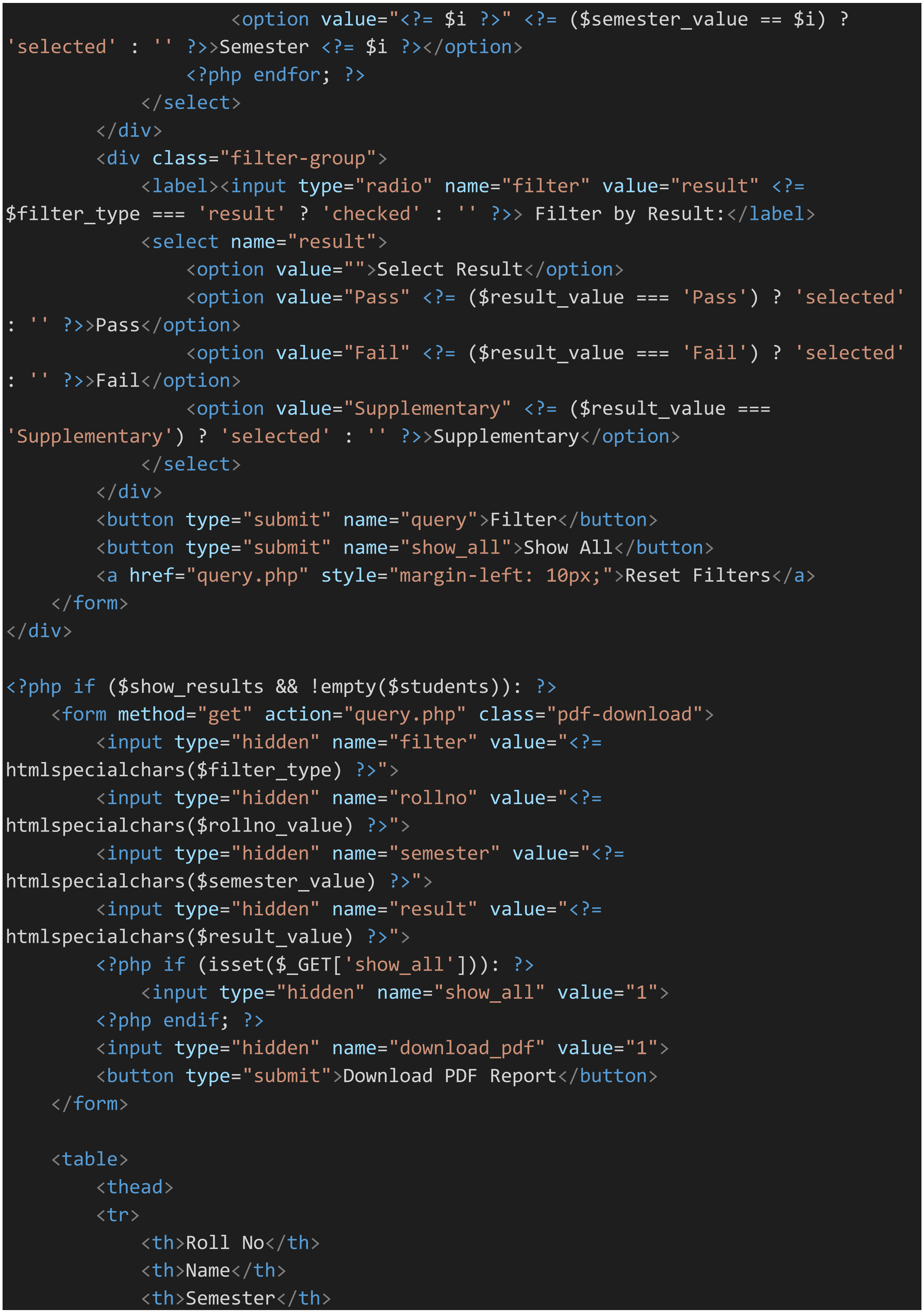
```php
        $this->SetFont('Arial', 'I', 8);
        $this->Cell(0, 10, 'Page ' . $this->PageNo(), 0, 0, 'C');
    }
}

$filterText = 'Report includes all student records';
if (!empty($filter_type) && !empty($filter_value)) {
    switch ($filter_type) {
        case 'rollno':
            $filterText = "Report based on Roll Number = $filter_value";
            break;
        case 'semester':
            $filterText = "Report based on Semester = $filter_value";
            break;
        case 'result':
            $filterText = "Report based on Result = $filter_value";
            break;
    }
}

$pdf = new PDF();
$pdf->filterText = $filterText;
$pdf->AddPage();
$pdf->SetFont('Arial', '', 12);

// Centering logic
$tableWidth = 30 + 70 + 30 + 40; // = 170
$marginX = ($pdf->GetPageWidth() - $tableWidth) / 2;

// Table Header
$pdf->SetX($marginX);
$pdf->SetFillColor(200, 200, 200);
$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(30, 10, 'Roll No', 1, 0, 'C', true);
$pdf->Cell(70, 10, 'Name', 1, 0, 'C', true);
$pdf->Cell(30, 10, 'Semester', 1, 0, 'C', true);
$pdf->Cell(40, 10, 'Result', 1, 1, 'C', true);

// Table Body
$pdf->SetFont('Arial', '', 12);
foreach ($students as $student) {
    $pdf->SetX($marginX);
    $pdf->Cell(30, 10, $student['rollno'], 1, 0, 'C');
    $pdf->Cell(70, 10, $student['name'], 1, 0, 'C');
    $pdf->Cell(30, 10, $student['semester'], 1, 0, 'C');
    $pdf->Cell(40, 10, $student['result'], 1, 1, 'C');
}
```



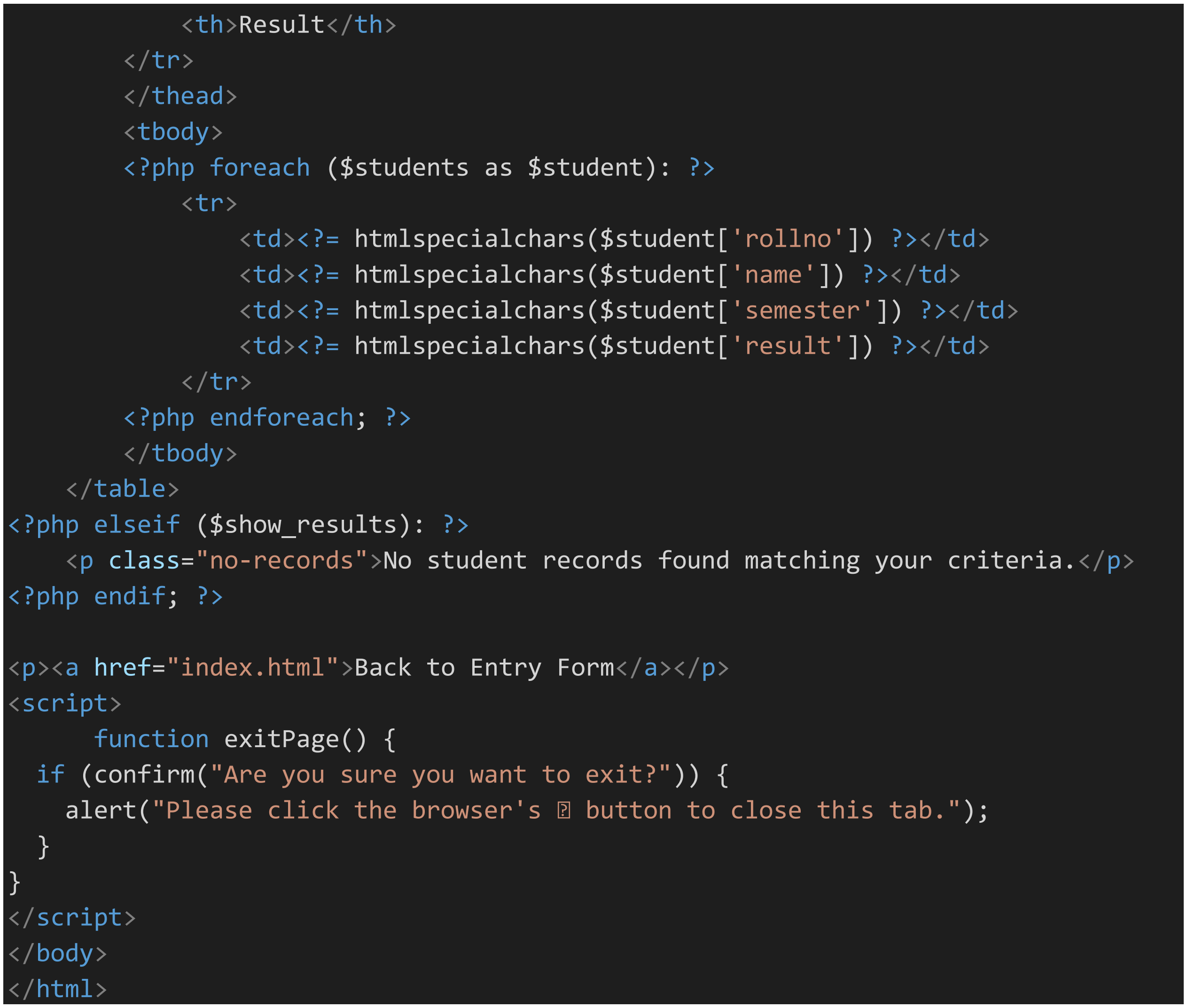
```
    $pdf->Output('D', 'Student_Report.pdf');
    exit;
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Student Records Query</title>
    <style>
        body {
            font-family: 'Segoe UI', sans-serif;
            max-width: 900px;
            margin: 0 auto;
            padding: 20px;
            background-color: #f5f5f5;
        }
        .navbar {
            background-color: #2c2c2c;
            overflow: hidden;
            margin-bottom: 30px;
            border-radius: 4px 4px 0 0;
        }
        .navbar a {
            float: left;
            display: block;
            color: white;
            text-align: center;
            padding: 14px 20px;
            text-decoration: none;
            font-weight: bold;
        }
        .navbar a.active {
            background-color: #4CAF50;
        }
        h2 { color: #333; }
        .error {
            color: red;
            margin-bottom: 15px;
        }
        .filter-section {
            background-color: #fff;
            padding: 15px;
            border-radius: 5px;
            margin-bottom: 20px;
```



```
            box-shadow: 0 0 10px rgba(0,0,0,0.05);
        }
        .filter-group {
            margin-bottom: 15px;
        }
        label {
            font-weight: bold;
            display: block;
            margin-bottom: 5px;
            color: #444;
        }
        input[type="text"], select {
            padding: 8px;
            width: 250px;
            border: 1px solid #ccc;
            border-radius: 4px;
        }
        button {
            padding: 10px 16px;
            background-color: #4CAF50;
            border: none;
            color: white;
            border-radius: 4px;
            cursor: pointer;
            font-weight: bold;
        }
        button:hover {
            background-color: #45a049;
        }
        table {
            width: 100%;
            border-collapse: collapse;
            margin-top: 20px;
            background-color: white;
        }
        th, td {
            border: 1px solid #bbb;
            padding: 10px;
            text-align: center;
        }
        th {
            background-color: #eee;
            font-weight: bold;
        }
        .no-records {
            margin: 20px 0;
            color: #666;
            font-style: italic;
        }
        form.pdf-download {
            margin-bottom: 20px;
        }
        form.pdf-download button {
            background-color: #2196F3;
            padding: 10px 16px;
            border: none;
            border-radius: 4px;
            color: white;
            cursor: pointer;
            font-weight: bold;
        }
        form.pdf-download button:hover {
            background-color: #1976D2;
        }
    </style>
</head>
<body>
<div class="navbar">
    <a href="index.html">Entry</a>
    <a href="query.php" class="active">Query</a>
    <a href="#" onclick="exitPage()" style="float: right; background-color:
#f44336;">Exit</a>
</div>

<h2>Student Records</h2>

<?php if (!empty($error)): ?>
    <div class="error"><?= htmlspecialchars($error) ?></div>
<?php endif; ?>

<div class="filter-section">
    <form method="get" action="query.php">
        <div class="filter-group">
            <label><input type="radio" name="filter" value="rollno" <?=
$filter_type === 'rollno' ? 'checked' : '' ?>> Filter by Roll No:</label>
            <input type="text" name="rollno" placeholder="Enter Roll No"
value="<?= htmlspecialchars($rollno_value) ?>">
        </div>
        <div class="filter-group">
            <label><input type="radio" name="filter" value="semester" <?=
$filter_type === 'semester' ? 'checked' : '' ?>> Filter by Semester:</label>
            <select name="semester">
                <option value="">Select Semester</option>
                <?php for ($i = 1; $i <= 8; $i++): ?>
```



```
                    <option value="<?= $i ?>" <?= ($semester_value == $i) ?
'selected' : '' ?>>Semester <?= $i ?></option>
                <?php endfor; ?>
            </select>
        </div>
        <div class="filter-group">
            <label><input type="radio" name="filter" value="result" <?=
$filter_type === 'result' ? 'checked' : '' ?>> Filter by Result:</label>
            <select name="result">
                <option value="">Select Result</option>
                <option value="Pass" <?= ($result_value === 'Pass') ? 'selected'
: '' ?>>Pass</option>
                <option value="Fail" <?= ($result_value === 'Fail') ? 'selected'
: '' ?>>Fail</option>
                <option value="Supplementary" <?= ($result_value ===
'Supplementary') ? 'selected' : '' ?>>Supplementary</option>
            </select>
        </div>
        <button type="submit" name="query">Filter</button>
        <button type="submit" name="show_all">Show All</button>
        <a href="query.php" style="margin-left: 10px;">Reset Filters</a>
    </form>
</div>

<?php if ($show_results && !empty($students)): ?>
    <form method="get" action="query.php" class="pdf-download">
        <input type="hidden" name="filter" value="<?=
htmlspecialchars($filter_type) ?>">
        <input type="hidden" name="rollno" value="<?=
htmlspecialchars($rollno_value) ?>">
        <input type="hidden" name="semester" value="<?=
htmlspecialchars($semester_value) ?>">
        <input type="hidden" name="result" value="<?=
htmlspecialchars($result_value) ?>">
        <?php if (isset($_GET['show_all'])): ?>
            <input type="hidden" name="show_all" value="1">
        <?php endif; ?>
        <input type="hidden" name="download_pdf" value="1">
        <button type="submit">Download PDF Report</button>
    </form>

    <table>
        <thead>
        <tr>
            <th>Roll No</th>
            <th>Name</th>
            <th>Semester</th>
```



```
            <th>Result</th>
        </tr>
        </thead>
        <tbody>
        <?php foreach ($students as $student): ?>
            <tr>
                <td><?= htmlspecialchars($student['rollno']) ?></td>
                <td><?= htmlspecialchars($student['name']) ?></td>
                <td><?= htmlspecialchars($student['semester']) ?></td>
                <td><?= htmlspecialchars($student['result']) ?></td>
            </tr>
        <?php endforeach; ?>
        </tbody>
    </table>
<?php elseif ($show_results): ?>
    <p class="no-records">No student records found matching your criteria.</p>
<?php endif; ?>

<p><a href="index.html">Back to Entry Form</a></p>
<script>
      function exitPage() {
  if (confirm("Are you sure you want to exit?")) {
    alert("Please click the browser's ⍰ button to close this tab.");
  }
}
</script>
</body>
</html>
```

OVERVIEEW OF THE FILES

process_form.php ; Handles Student Data Submission

Purpose : To receive**,** validate**,** and insert multiple student records (submitted from **index.html)** into a MySQL database **(student_db.students** table).

Step-by-Step Breakdown

1 Enable Error Reporting for Debugging

```
error_reporting(E_ALL);
ini_set('display_errors', 1);
```

Ensures any warnings or errors are shown in the browser — helpful during development.

---

2. **Database Connection via PDO**

```
$pdo = new PDO(...);
```

- Connects to the `student_b` database using **PDO** (PHP Data Objects).
- Uses UTF-8 encoding.
- Enables **exceptions** for error handling.
- Fetches results as associative arrays by default.

---

3. **Initial Setup**

```
$errors = [];
$success = '';
```

Creates empty containers to store validation errors and success messages.

4.Handles POST Request from index.html

```
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $allData = json_decode($_POST['allData'], true);
```

- This is triggered when the form is submitted.
- Retrieves `allData` — a JSON string from the hidden input.
- Converts it into a PHP array of students.

5.Validation

Validation Rules

- Roll No: must be between 1000 and 9999
- Name: must not contain numbers
- Semester: 1 to 8
- Result: only `Pass`, `Fail`, `Supplementary`

```
if ($rollno < 1000 || $rollno > 9999) { ... }
if (preg_match('/[0-9]/', $name)) { ... }
if ($semester < 1 || $semester > 8) { ... }
if (!in_array(strtolower($result), ['pass', 'fail', 'supplementary'])) { ... }
```

Duplicate Check

```
$checkStmt->execute([$rollno]);
if ($checkStmt->rowCount() > 0) {
    $errors[] = "Duplicate Roll No: $rollno";
    continue;
}
```

Skips inserting if the roll number already exists.

6.Insert into Database

```
$insertStmt->execute([$rollno, $name, $semester, $result]);
```

If all validations pass and it's not a duplicate, inserts the record into the `students` table.

7.After Loop: Display Status

```
if (empty($errors)) {
    $success = "Student record(s) added successfully!";
}
```

If no errors, shows a success message. Otherwise, it lists the validation errors.

HTML Output (User Feedback)

At the end of the script:

- Displays:
  - Success message (green)
  - List of errors (red)
- Shows a link to return to the form (`index.html`)

Security & Best Practices

- Uses **prepared statements** → protects from SQL injection
- Sanitizes and normalizes inputs (trim, `strtoupper`, etc.)
- Handles all errors gracefully
- Uses **POST** method for submission
- Returns meaningful feedback

query.php : Student Records Search + PDF Report Generator

Purpose : To allow users to search student records in the database using filters and download a PDF report of the results.

2. Setup Conditions to Run the Query

```
if ((isset($_GET['filter']) && $filter_value !== '') ||
isset($_GET['show_all'])) {
    $show_results = true;
}
```

- If a filter is applied or "Show All" is clicked, it prepares to query the database.

3. Connect to MySQL Database (PDO)

```
<?php

// Basic PDO connection

$host = 'localhost';

$dbname = 'your_database';

$username = 'root';

$password = '';

try {

    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username,
$password);

    // Set error mode to exception

    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    echo "Connected successfully!";

} catch (PDOException $e) {

    echo "Connection failed: " . $e->getMessage();

}

?>
```

*In PHP, $e is just a **variable name** commonly used to refer to the **exception object** caught by a catch block.*

4. Build SQL Query Based on Filter

```
$sql = "SELECT rollno, name, semester, result FROM students";
```

- If a filter is selected:
  - Adds a `WHERE` clause (`rollno = ?`, `semester = ?`, or `result = ?`)
- If "Show All" is selected, it fetches everything.
- Uses `ORDER BY rollno` to sort neatly.

```
$stmt = $pdo->prepare($sql);
$stmt->execute($params);
$students = $stmt->fetchAll();
```

- Uses placeholders (`?`) to prevent SQL injection.
- Results are saved in the `$students` array.

---

6. Generate PDF Report

```
if (isset($_GET['download_pdf']) && $show_results && !empty($students)) {
    require_once 'lib/fpdf/fpdf.php';
    ...
}
```

- If "Download PDF Report" is clicked:
  - Uses **FPDF library** to generate a well-formatted PDF:
    - Header with title
    - Optional filter note (e.g., "Semester = 4")
    - Table with student data
    - Footer with page number
- Automatically downloads as `Student_Report.pdf`

7. Display the Web Page (HTML)

Navbar:

```
<a href="index.html">Entry</a>
<a href="query.php" class="active">Query</a>
```

Filter Form:

- Radio buttons to choose filter type
- Input/select box to enter filter value
- Buttons:
  - Filter
  - Show All
  - Reset

Result Table:

- Shown only if results are available
- If not: shows "No student records found"

PDF Download Button:

```
<form method="get" action="query.php" class="pdf-download">
    <button type="submit">Download PDF Report</button>
</form>
```

Exit Button (JS Confirmation):

```
function exitPage() {
    if (confirm("Are you sure you want to exit?")) {
        alert("Please click the browser's □ button to close this tab.");
    }
}
```

Goal of the Project:

To collect, validate, store, search, and export student records (roll number, name, semester, result) using:

index.html for entry

process_form.php for saving

query.php for searching + PDF export

MySQL database using **PDO**

Components Used:

HTML + CSS + JS (Frontend UI)

PHP with PDO (Backend logic)

MySQL (student_db database, students table)

FPDF (PDF report generator)

WORKFLOW OVERVIEW

1. Student Entry via index.html

User Action:

- Fills form: Roll No, Name, Semester, Result
- Clicks **"Add"** → JS adds row to the table
- Clicks **"FINAL SUBMIT"**

Behind the Scenes:

- JavaScript collects all entries into an array
- Converts array to JSON
- Sends it via POST to `process_form.php`

---

2. Data Processing via process_form.php

PHP Script Does:

- Receives JSON data **(`$_POST['allData']`)**
- Decodes into PHP array
- Validates each student:
  - Roll No: 1000–9999
  - Name: no numbers
  - Semester: 1–8
  - Result: pass/fail/supplementary
  - Checks for duplicate roll numbers
- Valid data is inserted into MySQL (`students` table)

User Sees:

- Success message if all okay
- Error messages for invalid or duplicate entries

---

3. Search & Report via query.php

User Can:

- Filter records by:
  - Roll Number
  - Semester
  - Result
- Or click **"Show All"** to display all data

PHP Script Does:

- Checks GET parameters
- Builds a SELECT query based on filter
- Fetches and displays results as HTML table

---

4. Export PDF Report (query.php)

User Can Click:

- **Download PDF Report** (by the help of fpdf package)

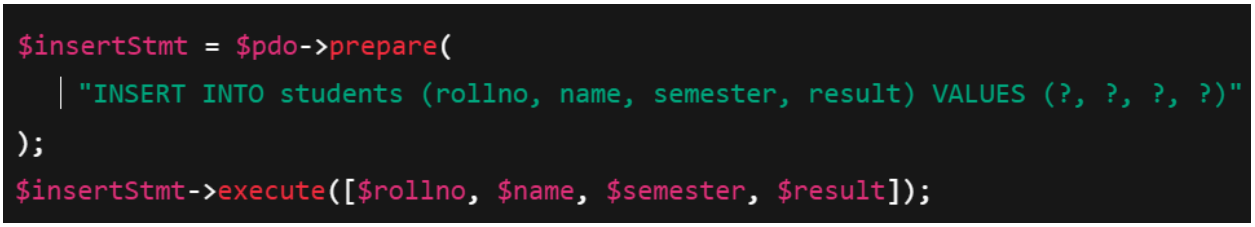
Behind the Scenes:

- FPDF library generates a PDF:
- By clicking on the "**Download Report** " button Automatically downloads as `Student_Report.pdf`

WHY MY PROJECT IS SAFE FROM SQL INJECTION HACKING ?

Project **is protected from SQL injection attacks** because of using **PDO with prepared statements**, which is the a secure method.

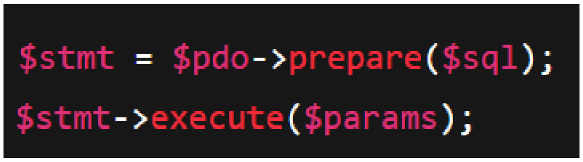
In `process_form.php`:



```
$insertStmt = $pdo->prepare(
    "INSERT INTO students (rollno, name, semester, result) VALUES (?, ?, ?, ?)"
);
$insertStmt->execute([$rollno, $name, $semester, $result]);
```

In line 33

In `query.php`:



```
$stmt = $pdo->prepare($sql);
$stmt->execute($params);
```

In line 63

- I **never directly include user input in SQL queries**.
- I use **placeholders (`?`)** and bind data separately using `execute()`.
- Even if a hacker tries to inject , it will be treated as normal text —

  not as SQL.

Since the validation is done in PHP too at process_form.php I have also add it in the index.html too with Javascript Because :

JS/HTML Validation = User Experience (Nice-to-Have)

- It's for **convenience**, not security.
- It gives users **instant feedback** without needing to submit the form.
- Example: alerts when fields are empty, name has numbers, roll number is out of range.
- Prevents **frustration** from submitting and then seeing errors from PHP.