



LAB REPORT

05

CSE416: Web Engineering Lab

Experiment Title: PHP MVC Architecture with OOP

Submitted To

Mr. Monthasir Delwar Afnan
Lecturer
Department of CSE
Daffodil International University

Submitted By

Student ID: 221-15-5953
Student Name: Md. Fahimur Rahman
Section: 61_P2
Department of CSE
Daffodil International University

Date of Assignment Distribution: 06/08/2025
Date of Assignment Submission: 15/08/2025

1. Objective

- To understand the MVC (Model-View-Controller) architecture in PHP.
 - To apply Object-Oriented Programming (OOP) principles in PHP.
 - To build a structured, maintainable PHP application for CRUD operations.
-

2. Tools and Technologies Used

- **XAMPP** (Apache + MySQL) for running PHP and database.
 - **Visual Studio Code** as code editor.
 - **Google Chrome** or other web browsers for testing.
 - **MySQL Database** for storing student data.
-

3. Explanation

3.1 MVC Architecture

- **Model:** Handles data operations (CRUD) and communicates with the database.
- **View:** Displays the user interface (HTML/CSS forms and tables).
- **Controller:** Handles user requests, processes data, and interacts with Model and View.

3.2 OOP Concepts in PHP Used

- **Class & Object:** Defined classes (`Student`, `StudentController`) and created objects.
 - **Constructor:** Used to initialize database connection.
 - **Encapsulation:** Database connection and table name kept private in the Model.
 - **Methods:** Functions inside classes for CRUD operations.
-

4. SQL Code

Create the **new table students** inside existing database `user_info`:

```
USE user_info;
```

```
CREATE TABLE IF NOT EXISTS students (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,
```

```
        email VARCHAR(100) NOT NULL,  
        dept VARCHAR(50) NOT NULL  
    );
```

5. Folder Structure

```
Lab_Report_5/  
├── index.php                # Front Controller  
├── config/  
│   └── db.php              # Database Connection  
├── controllers/  
│   └── StudentController.php  
├── models/  
│   └── Student.php  
└── views/  
    ├── add.php  
    ├── list.php  
    ├── update.php  
    └── delete.php
```

6. Code Implementation

6.1 Database Connection (config/db.php)

```
<?php  
class Database {  
    private $host = "localhost";  
    private $user = "root";  
    private $pass = "";  
    private $dbname = "user_info"; // Existing database  
    public $conn;  
  
    public function __construct() {
```

```

        $this->conn = new mysqli($this->host, $this->user, $this->pass, $this->dbname);

        if ($this->conn->connect_error) {
            die("Connection failed: " . $this->conn->connect_error);
        }
    }
}
?>

```

6.2 Model (models/Student.php)

```

<?php
include_once __DIR__ . '/../config/db.php';

class Student {
    private $conn;

    private $table = "students"; // Use the new table name

    public function __construct() {
        $database = new Database();
        $this->conn = $database->conn;
    }

    // Create
    public function create($name, $email, $dept) {
        $sql = "INSERT INTO $this->table (name, email, dept) VALUES
('$name','$email','$dept')";
    }
}

```

```
        return $this->conn->query($sql);
    }

    // Read
    public function read() {
        $sql = "SELECT * FROM $this->table";
        return $this->conn->query($sql);
    }

    // Update
    public function update($id, $name, $email, $dept) {
        $sql = "UPDATE $this->table SET name='$name', email='$email',
dept='$dept' WHERE id=$id";
        return $this->conn->query($sql);
    }

    // Delete
    public function delete($id) {
        $sql = "DELETE FROM $this->table WHERE id=$id";
        return $this->conn->query($sql);
    }
}
?>
```

6.3 Controller (controllers/StudentController.php)

```
<?php
include_once __DIR__ . '/../models/Student.php';

class StudentController {
    private $student;

    public function __construct() {
        $this->student = new Student();
    }

    public function add($data) {
        return $this->student->create($data['name'], $data['email'],
$data['dept']);
    }

    public function list() {
        return $this->student->read();
    }

    public function update($data) {
        return $this->student->update($data['id'], $data['name'],
$data['email'], $data['dept']);
    }

    public function delete($id) {
```

```
        return $this->student->delete($id);
    }
}
?>
```

6.4 Views

Add Student (views/add.php)

```
<style>
    body {
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        background: #f7f7f7;
        margin: 0;
        padding: 0;
    }
    .form-container {
        background: #fff;
        max-width: 400px;
        margin: 40px auto;
        padding: 30px 25px;
        border-radius: 8px;
        box-shadow: 0 2px 8px rgba(0,0,0,0.08);
    }
    h2 {
        text-align: center;
        color: #333;
    }
```

```
    margin-bottom: 20px;
}
label {
    display: block;
    margin-bottom: 6px;
    color: #555;
    font-weight: 500;
}
input[type="text"], input[type="email"] {
    width: 100%;
    padding: 8px 10px;
    margin-bottom: 16px;
    border: 1px solid #ccc;
    border-radius: 4px;
    font-size: 15px;
}
input[type="submit"] {
    width: 100%;
    padding: 10px;
    background: #0078d4;
    color: #fff;
    border: none;
    border-radius: 4px;
    font-size: 16px;
    cursor: pointer;
    transition: background 0.2s;
}
```



```

        input[type="submit"]:hover {
            background: #005fa3;
        }
        hr {
            margin-top: 30px;
        }
    </style>
    <div class="form-container">
        <h2>Add Student</h2>
        <form method="POST" action="index.php?action=add">
            <label for="name">Name:</label>
            <input type="text" name="name" id="name" required>
            <label for="email">Email:</label>
            <input type="email" name="email" id="email" required>
            <label for="dept">Department:</label>
            <input type="text" name="dept" id="dept" required>
            <input type="submit" value="Add">
        </form>
    </div>
    <hr>

```

List Students (views/list.php)

```

<?php
$result = $controller->list();
?>
<style>

```

```
body {  
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
    background: #f7f7f7;  
    margin: 0;  
    padding: 0;  
}  
  
.list-container {  
    background: #fff;  
    max-width: 700px;  
    margin: 40px auto;  
    padding: 30px 25px;  
    border-radius: 8px;  
    box-shadow: 0 2px 8px rgba(0,0,0,0.08);  
}  
  
h2 {  
    text-align: center;  
    color: #333;  
    margin-bottom: 20px;  
}  
  
table {  
    width: 100%;  
    border-collapse: collapse;  
    margin-bottom: 20px;  
}  
  
th, td {  
    padding: 10px 12px;  
    border-bottom: 1px solid #eee;
```

```

        text-align: left;
    }
    th {
        background: #f0f4f8;
        color: #0078d4;
        font-weight: 600;
    }
    tr:hover {
        background: #f9fafb;
    }
    hr {
        margin-top: 30px;
    }
</style>
<div class="list-container">
    <h2>Student List</h2>
    <?php
    if ($result->num_rows > 0) {
        echo "<table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Email</th>
                    <th>Department</th>
                </tr>
            </thead>

```

```

        <tbody>";
        while ($row = $result->fetch_assoc()) {
            echo "<tr>
                <td>".$row['id']. "</td>
                <td>".$row['name']. "</td>
                <td>".$row['email']. "</td>
                <td>".$row['dept']. "</td>
            </tr>";
        }
        echo "    </tbody>
        </table>";
    } else {
        echo "No records found.";
    }
?>
</div>
<hr>

```

Update Student (views/update.php)

```

<style>
    body {
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        background: #f7f7f7;
        margin: 0;
        padding: 0;
    }

```

```
}  
  
.form-container {  
    background: #fff;  
    max-width: 400px;  
    margin: 40px auto;  
    padding: 30px 25px;  
    border-radius: 8px;  
    box-shadow: 0 2px 8px rgba(0,0,0,0.08);  
}  
  
h2 {  
    text-align: center;  
    color: #333;  
    margin-bottom: 20px;  
}  
  
label {  
    display: block;  
    margin-bottom: 6px;  
    color: #555;  
    font-weight: 500;  
}  
  
input[type="text"], input[type="email"] {  
    width: 100%;  
    padding: 8px 10px;  
    margin-bottom: 16px;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
    font-size: 15px;
```

```
}  
  
input[type="submit"] {  
    width: 100%;  
    padding: 10px;  
    background: #0078d4;  
    color: #fff;  
    border: none;  
    border-radius: 4px;  
    font-size: 16px;  
    cursor: pointer;  
    transition: background 0.2s;  
}  
  
input[type="submit"]:hover {  
    background: #005fa3;  
}  
  
hr {  
    margin-top: 30px;  
}  
  
</style>  
  
<div class="form-container">  
    <h2>Update Student</h2>  
    <form method="POST" action="index.php?action=update">  
        <label for="id">ID:</label>  
        <input type="text" name="id" id="id" required>  
        <label for="name">Name:</label>  
        <input type="text" name="name" id="name" required>  
        <label for="email">Email:</label>
```

```
<input type="email" name="email" id="email" required>
<label for="dept">Department:</label>
<input type="text" name="dept" id="dept" required>
<input type="submit" value="Update">
</form>
</div>
<hr>
```

Delete Student (views/delete.php)

```
<style>
  body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background: #f7f7f7;
    margin: 0;
    padding: 0;
  }
  .form-container {
    background: #fff;
    max-width: 400px;
    margin: 40px auto;
    padding: 30px 25px;
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(0,0,0,0.08);
  }
  h2 {
```

```
    text-align: center;
    color: #333;
    margin-bottom: 20px;
}
label {
    display: block;
    margin-bottom: 6px;
    color: #555;
    font-weight: 500;
}
input[type="text"] {
    width: 100%;
    padding: 8px 10px;
    margin-bottom: 16px;
    border: 1px solid #ccc;
    border-radius: 4px;
    font-size: 15px;
}
input[type="submit"] {
    width: 100%;
    padding: 10px;
    background: #0078d4;
    color: #fff;
    border: none;
    border-radius: 4px;
    font-size: 16px;
    cursor: pointer;
```



```
        transition: background 0.2s;
    }
    input[type="submit"]:hover {
        background: #005fa3;
    }
    hr {
        margin-top: 30px;
    }
</style>

<div class="form-container">
    <h2>Delete Student</h2>
    <form method="POST" action="index.php?action=delete">
        <label for="id">Enter Student ID to Delete:</label>
        <input type="text" name="id" id="id" required>
        <input type="submit" value="Delete">
    </form>
</div>
<hr>
```

6.5 Front Controller (index.php)

```
<?php
include 'controllers/StudentController.php';
$controller = new StudentController();
```

```
$action = $_GET['action'] ?? '';

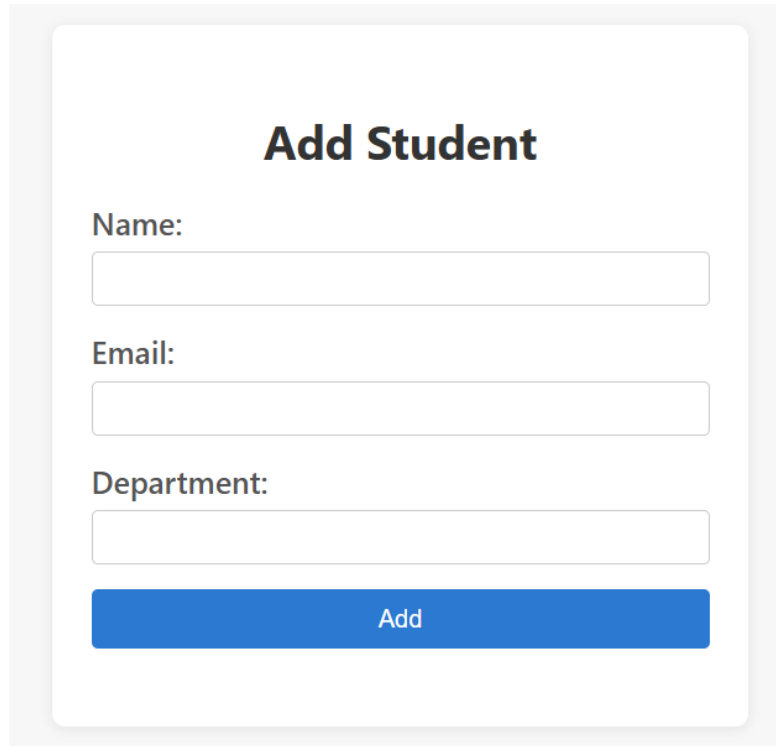
if ($action == 'add' && $_SERVER['REQUEST_METHOD'] == 'POST') {
    $controller->add($_POST);
    header("Location: index.php");
}

elseif ($action == 'update' && $_SERVER['REQUEST_METHOD'] == 'POST') {
    $controller->update($_POST);
    header("Location: index.php");
}

elseif ($action == 'delete' && $_SERVER['REQUEST_METHOD'] == 'POST') {
    $controller->delete($_POST['id']);
    header("Location: index.php");
}

// Include views
include 'views/add.php';
include 'views/list.php';
include 'views/update.php';
include 'views/delete.php';
```

7. Output / Screenshots

A screenshot of a web form titled "Add Student". The form is contained within a white rounded rectangle with a light gray border. It features three input fields: "Name:", "Email:", and "Department:", each with a corresponding text input box. Below the input fields is a blue button with the text "Add".

Add Student

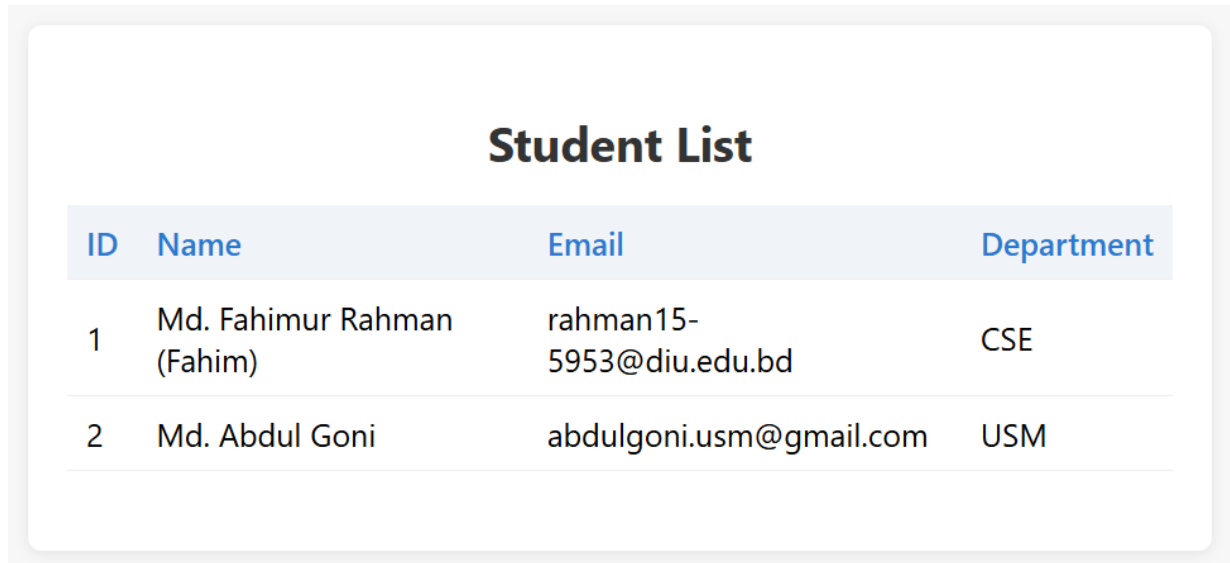
Name:

Email:

Department:

Add

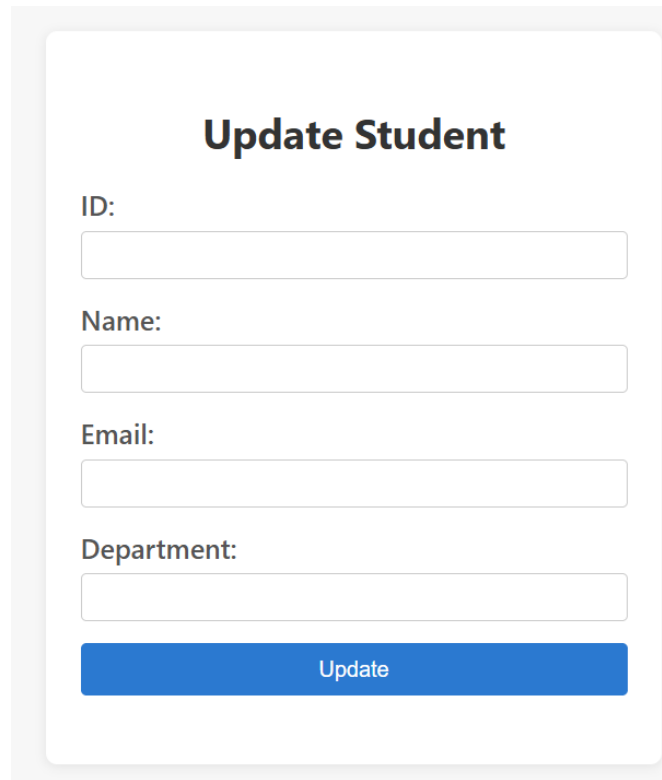
Fig 7.1: Add Student form

A screenshot of a web page showing a table titled "Student List". The table has four columns: ID, Name, Email, and Department. It contains two rows of student data.

Student List

ID	Name	Email	Department
1	Md. Fahimur Rahman (Fahim)	rahman15-5953@diu.edu.bd	CSE
2	Md. Abdul Goni	abdulgoni.usm@gmail.com	USM

Fig 7.2: List Students



Update Student

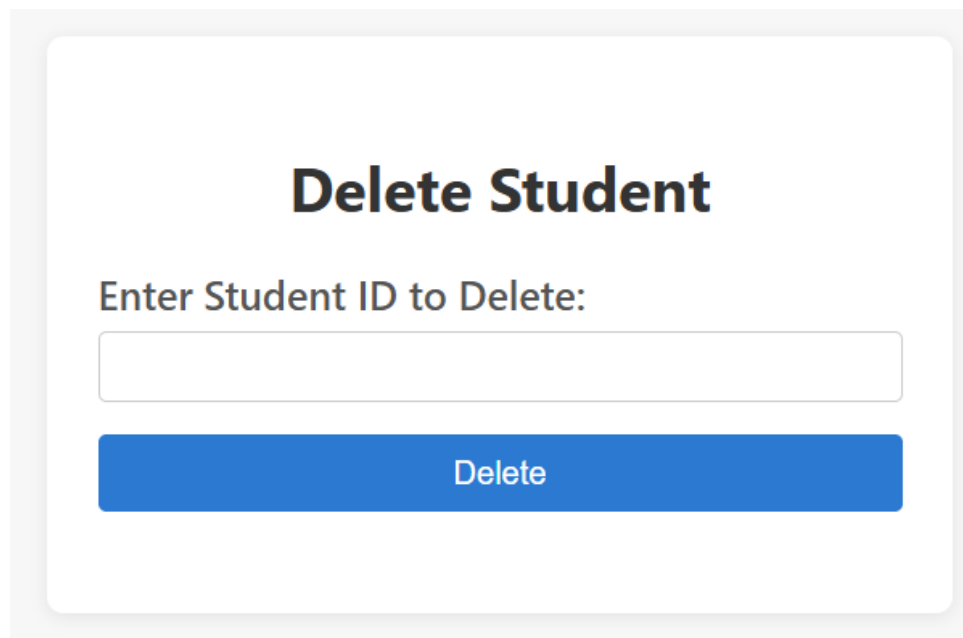
ID:

Name:

Email:

Department:

Fig 5.3: Update Student



Delete Student

Enter Student ID to Delete:

Fig 5.4: Delete Student

Add Student

Name:

Email:

Department:

Add

Student List

ID	Name	Email	Department
1	Md. Fahimur Rahman (Fahim)	rahman15-5953@diu.edu.bd	CSE

Update Student

ID:

Name:

Email:

Department:

Update

Delete Student

Enter Student ID to Delete:

Delete

Fig 5.5: The whole site web view

8. Conclusion

- Successfully implemented PHP MVC architecture with OOP.
- Learned how to separate concerns: Models for data, Views for UI, Controllers for logic.
- Applied CRUD operations using MySQL.
- Achieved a maintainable and structured PHP web application.