A Project Report on

Student Registration Form

Submitted in partial fulfilment of the degree of
Bachelor of Computer Applications
V Semster

Submitted By
Saloni Thakur
Tanushree
Priya Mali

Under the Supervision of
Mr. Aman Mathur
Faculty of Computer Science

Roll No.
233501200
233501218
233501169

Course Code/Name:
CCBCA522
Adavance Web Design

Faculty of Computer Science
Lachoo Memorial College of Science & Technology(Autonomous)
Jodhpur
2025

Abstract

The Student Registration Form project aims to simplify and digitize the process of collecting and organizing student information for academic institutions. Traditionally, student data is gathered manually, leading to issues such as incomplete entries, misplaced records, slow processing, and difficulty in retrieving information when needed. To address these challenges, this project focuses on designing a structured, interactive, and user-friendly web-based registration system using modern web technologies.

The dataset for this project consists of essential student details, including name, age, gender, contact number, email address, course preferences, and residential information. These fields were chosen to ensure complete and accurate student profiling. The form was developed using **HTML** for layout, **CSS** for responsive styling, and optional **JavaScript** and **PHP** for validation and backend processing. Front-end validation ensures that users enter correct and complete information, while structured storage enables easier record management.

The results of this project show that the digital form significantly improves the efficiency of student data collection. It reduces manual errors, enhances data consistency, and makes record retrieval faster and more reliable. The final system provides a clean interface, easy navigation, and accurate data handling, making it suitable for integration into academic websites or admission portals. Overall, the project demonstrates how advanced web design techniques can create practical, real-world solutions for educational management.

Introduction

In today's digitally driven world, educational institutions increasingly rely on web-based systems to streamline their administrative processes. One of the most essential components in academic management is the student registration process, which involves collecting and organizing important personal and academic details from students. Traditionally, this process has been conducted through manual paperwork, which is often time-consuming, prone to errors, and difficult to maintain in the long run. To overcome these limitations, web-based registration forms have become a modern, efficient, and reliable alternative. This project, developed for the Advanced Web Design course, focuses on creating a professional and user-friendly Student Registration Form that simplifies data collection while ensuring accuracy and accessibility.

The primary purpose of this project is to design an interactive interface where students can easily input their necessary details. These may include their name, age, contact information, gender, course preferences, and address. By structuring these fields clearly and logically, the form not only guides users through the registration process but also helps maintain a complete and consistent dataset for institutional use. The project incorporates **HTML** to build the form structure, **CSS** to enhance visual appeal and responsiveness, and optional **JavaScript** or **PHP** to handle input validation and data processing.

A well-designed registration form is more than just a data-entry tool—it plays a crucial role in improving the overall user experience. It enhances operational efficiency by reducing manual workload, minimizing duplication of information, and eliminating common errors such as missing fields or incorrect data formats. Furthermore, the transition to a digital registration process ensures that student data remains securely stored and easily retrievable whenever needed.

This project demonstrates how advanced web design principles can be applied to create practical, real-world solutions. By blending aesthetics with functionality, the Student Registration Form provides a streamlined, organized, and modern approach to academic registration, highlighting the importance of digital transformation in education.

As the role of technology continues to expand across all fields, the adoption of web-based systems in education has become not just beneficial but essential. Institutions now require efficient, automated platforms that can handle routine tasks such as student registration, data management, and record maintenance without unnecessary delays. A well-structured online registration form plays a major part in fulfilling this need by simplifying the interaction between students and the institution. Instead of relying on physical forms that may get damaged, misplaced, or incorrectly filled, digital forms ensure accuracy by guiding users through a systematic and error-free process.

Another major advantage of a web-based registration system is the convenience it provides. Students can fill out the form anytime and from anywhere, reducing long queues and administrative burden during admission periods. This shift towards digital systems reflects the growing expectation for flexibility and accessibility in education. In addition, online forms allow institutions to quickly update fields, modify requirements, or add new courses without redesigning paper forms every time.

From a design perspective, this project emphasizes the importance of creating interfaces that are both visually appealing and functionally efficient. Using modern web design techniques such as responsive layouts, clean typography, and intuitive navigation, the form is built to ensure a smooth user experience across different devices. The use of input validation also plays a key role in ensuring that the information submitted by students is complete and consistent.

Overall, the extended introduction highlights the increasing importance of digitization in academic environments and explains how a student registration form, though simple in concept, contributes significantly to improving accuracy, organization, and user convenience within educational systems.

Dataset Description

The dataset used in this project is generated entirely through the student registration form. Each submission represents a single student record, capturing all essential personal and academic information. The data includes student name, age, gender, email address, phone number, address, city, state, pincode, date of birth, and course selection. These fields are chosen to ensure that institutions receive a comprehensive profile of the student, enabling smooth communication and academic planning. The dataset is created in real-time as students enter their details into the form, making it dynamic and continuously expanding.

The collected data is clean and consistent because the form enforces validation rules that prevent incorrect or incomplete submissions. For example, the email field must follow the correct format, phone numbers must be numeric and fixed-length, and names cannot contain invalid characters. Once validated, this data can be stored as CSV files, JSON objects, or directly in a database such as MySQL. The structured nature of the dataset makes it suitable for record management, analysis, filtering, and classification. Although the dataset is simple in terms of size and structure, it serves an important administrative function by maintaining a digital record of all registered students. This ensures that information is not lost and can be accessed efficiently whenever required.

The dataset used in this project consists of basic student information. The data is collected directly from the registration form filled by students. Source of data: User input submitted through the web form.

Format: HTML Form Input stored as CSV or database entries. The dataset contains textual and numeric information. Each entry represents a single student.

Key variables collected in the dataset include:

Full Name

Email Address

Contact Number

Date of Birth

Residential Address

Course Selection

The dataset is structured so each field becomes a column. Each student record forms a new row. This dataset is small but meaningful for institutional use. It can be exported in various formats (CSV, JSON, SQL).

It is useful for admissions and internal record keeping. Proper validation ensures clean and accurate dataset collection. The structured data simplifies future analysis and filtering

Methodology

The methodology behind the Student Registration Form project involves several steps designed to produce a clean, functional, and user-friendly system. The first step is planning the structure of the form, identifying the required fields, and organizing them in a logical flow. HTML is then used to build the layout, ensuring that each input type—text boxes, dropdown menus, radio buttons, and date selectors—is appropriately used. CSS is applied to improve the visual design, making the form readable and aesthetically pleasing. Elements such as fonts, spacing, alignment, colors, and responsive layouts are carefully designed to ensure that the form works smoothly across different devices.

Data validation plays a major role in this methodology. JavaScript is implemented to check for empty fields, improper formats, or invalid characters. This prevents errors and reduces the burden on administrative staff who would otherwise need to manually verify the entries. Once the data is validated, it can be processed using PHP, which handles posting the form data into a backend file or database. The process ensures that the information is securely stored and retrievable whenever required. The combination of frontend and backend technologies results in a robust and efficient digital registration system.

A. Data Collection Process

1. Students access the registration form through a web browser.
2. They fill in personal and academic details in required fields.
3. The information is validated before submission.
4. Once submitted, data is stored in a file or database.

B. Data Cleaning Steps

1. Checking for missing fields.
2. Ensuring name field contains valid characters.
3. Age is checked to ensure it is numeric.
4. Email pattern is validated using regex.
5. Phone number length is validated.
6. Gender is selected from predefined options.
7. Address fields have minimum character requirements.

8. Course field is dropdown-based to avoid improper input.
9. Form rejects invalid submissions automatically.
10. Only clean data is stored in the dataset.

C. Feature Transformation

1. Converting string-based values to standardized formatting.
2. Trimming extra spaces from input fields.
3. Formatting names into Title Case.
4. Storing all email addresses in lowercase.
5. Numeric fields standardized for uniformity.

D. Tools & Technologies Used

1. HTML5
2. CSS3
3. JavaScript
4. PHP (optional for backend)
5. VS Code Editor
6. XAMPP/WAMP (if using database)
7. Google Chrome Browser

E. Visualization (If Applicable)

1. Basic charts (pie chart of gender distribution).
2. Bar graph of course selection counts.
3. Any visualization done using JS/chart libraries.

Result & Analysis

The results of the Student Registration Form project show significant improvement in the registration process. Students can complete the form quickly, and validation ensures that the information they provide is accurate. The interface is user-friendly and visually organized, which reduces confusion and enhances user experience. The digital form eliminates manual paperwork and significantly reduces human error. The collected dataset is clear, structured, and easily manageable. Administrators can review records, analyze data, and retrieve information in seconds. The system demonstrates how even a simple web form can produce meaningful results that improve overall administrative efficiency.

- The student registration form successfully collects data.
- The interface supports error-free data entry.
- Validation ensures complete and correct inputs.
- Users find the form easy to use and navigate.
- Data stored is consistent and well-organized.
- The system reduces the time taken for manual registration.
- It eliminates data duplication and misplaced forms.
- Overall accuracy is improved compared to handwritten forms.
- Data retrieval becomes instant using backend storage.
- The collected dataset helps the institution maintain digital records.
- Pie charts show gender distribution.
- Bar charts show which courses students prefer most.
- Analysis helps in planning class size and faculty allocation.
- Administrators can track the number of new registrations easily.
- Results show that digital registration is faster and more reliable.
- The system can be extended for full admission automation.

Conclusion

The Student Registration Form project successfully demonstrates how traditional registration processes can be digitalized using advanced web design techniques. It offers a reliable, accurate, and efficient solution that benefits both students and administrators. The reduction of manual effort, improvement in data accuracy, and enhanced accessibility show the value of integrating technology into educational systems. Future enhancements may include integrating SMS or email confirmations, connecting the form to a larger academic database, enabling admin dashboards, or applying security features such as encryption and authentication. The project can also be expanded into a complete student management system.

- **Increased Efficiency and Accessibility:** The system significantly streamlines the registration process, allowing students and parents to complete forms anytime, anywhere, thus saving time and effort for both users and administrative staff. This eliminates geographical and time constraints associated with manual, in-person registration.
- **Improved Data Management and Accuracy:** By automating data collection and storage in an organized digital database, the risk of human errors, such as manual recording mistakes or data duplication, is minimized. This results in more accurate records and faster data retrieval and report generation.
- **Enhanced User Experience:** A well-designed, user-friendly, and intuitive interface is a key outcome of advanced web design. This positive user experience contributes to high user satisfaction and makes the application easy to navigate, even for those less familiar with online platforms.
- **Better Communication and Interaction:** The system fosters improved interactions between prospective students, parents, and school officials by providing online channels for communication, inquiries, and clarification of information related to the registration process.
- **Data Security and Reliability:** Advanced web design also incorporates robust data security measures, giving users confidence that their personal data is well-protected. The system proves reliable in carrying out its tasks, reducing the risk of failure that often plagues manual or outdated systems.
- **Scalability and Adaptability:** The web-based platform is designed to handle a large volume of data and records efficiently and can be adapted to new

requirements or modified in response to a constantly changing environment and evolving technologies.

References

Mozilla Developer Network (MDN). *HTML Forms Guide*. Available at: https://developer.mozilla.org/en-US/docs/Learn/Forms

Mozilla Developer Network (MDN). *CSS: Cascading Style Sheets Documentation*. Available at: https://developer.mozilla.org/en-US/docs/Web/CSS

Mozilla Developer Network (MDN). *JavaScript Form Validation*. Available at: https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation

W3Schools. *HTML Form Elements*. Available at: https://www.w3schools.com/html/html_forms.asp

W3Schools. *CSS Form Styling Techniques*. Available at: https://www.w3schools.com/css/css_form.asp

W3Schools. *JavaScript Input Validation*. Available at: https://www.w3schools.com/js/js_validation.asp

World Wide Web Consortium (W3C). *HTML5 Standard Specifications*. Available at: https://www.w3.org/TR/html52/

World Wide Web Consortium (W3C). *CSS3 Standard Specifications*. Available at: https://www.w3.org/Style/CSS/

TutorialsPoint. *Web Technologies Basics*. Available at: https://www.tutorialspoint.com/internet_technologies/index.htm

GeeksforGeeks. *Form Handling in PHP*. Available at: https://www.geeksforgeeks.org/php-form-handling/

IBM Documentation. *Importance of Digital Data Management*. Available at: https://www.ibm.com/docs/en

ResearchGate. *Digital Transformation in Educational Institutions*. Available at: https://www.researchgate.net/

Appendix

Full Code:

Index.php

```
<?php
include "db.php";
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href="style.css">
    <meta charset="UTF-8">
    <title>Student Records</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/
css/bootstrap.min.css">
</head>

<body class="bg-light">

<div class="container mt-5">
    <div class="d-flex justify-content-between mb-3">
        <h2>Student Records</h2>
        <a href="create.php" class="btn btn-primary">Add New Student</a>
    </div>

    <?php if(isset($_GET['msg'])): ?>
        <div class="alert alert-success">
            <?= $_GET['msg']; ?>
        </div>
    <?php endif; ?>

    <table class="table table-bordered table-hover shadow">
        <thead class="table-dark">
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>Email</th>
                <th>Course</th>
                <th>Actions</th>
            </tr>
```

```
        </thead>

        <tbody>
            <?php
            $sql = "SELECT * FROM students";
            $result = $conn->query($sql);

            while($row = $result->fetch_assoc()):
            ?>
            <tr>
                <td><?= $row['id']; ?></td>
                <td><?= $row['name']; ?></td>
                <td><?= $row['email']; ?></td>
                <td><?= $row['course']; ?></td>

                <td>
                    <a href="edit.php?id=<?= $row['id']; ?>" class="btn btn-warning btn-
sm">Edit</a>
                    <a href="delete.php?id=<?= $row['id']; ?>" class="btn btn-danger
btn-sm"
                       onclick="return confirm('Are you sure?');">
                       Delete
                    </a>
                </td>
            </tr>
            <?php endwhile; ?>
        </tbody>
    </table>
</div>

</body>
</html>
```

Create.php

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

include "config.php";

if (isset($_POST['submit'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $course = $_POST['course'];

    $sql = "INSERT INTO students (name, email, course) VALUES ('$name', '$email',
'$course')";
```

```
    if ($conn->query($sql) === TRUE) {
        header("Location: index.php?msg=Student Added Successfully");
        exit;
    } else {
        echo "Error: " . $conn->error;
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href="style.css">
    <meta charset="UTF-8">
    <title>Add Student</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/
css/bootstrap.min.css">
</head>

<body>
<div class="container mt-5">
    <h2>Add New Student</h2>
    <a href="index.php" class="btn btn-secondary mb-3">Back</a>

    <form action="" method="POST" class="p-4 shadow bg-white rounded">
        <div class="mb-3">
            <label>Name</label>
            <input type="text" name="name" class="form-control" required>
        </div>

        <div class="mb-3">
            <label>Email</label>
            <input type="email" name="email" class="form-control" required>
        </div>

        <div class="mb-3">
            <label>Course</label>
            <input type="text" name="course" class="form-control" required>
        </div>

        <button type="submit" name="submit" class="btn btn-success">Submit</
button>
    </form>
</div>
</body>
</html>
```

Edit.php

```
<?php
include "db.php";

$id = $_GET['id'];

$sql = "SELECT * FROM students WHERE id = $id";
$result = $conn->query($sql);
$row = $result->fetch_assoc();

if (isset($_POST['update'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $course = $_POST['course'];

    $update = "UPDATE students SET name='$name', email='$email',
course='$course' WHERE id=$id";

    if ($conn->query($update) === TRUE) {
        header("Location: index.php?msg=Record updated successfully");
        exit();
    } else {
        echo "Error updating record: " . $conn->error;
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href="style.css">
    <meta charset="UTF-8">
    <title>Edit Student</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/
css/bootstrap.min.css">
</head>

<body class="bg-light">
<div class="container mt-5">
    <div class="card shadow p-4">
        <h2 class="text-center mb-4">Edit Student</h2>

        <form method="POST">
            <div class="mb-3">
                <label class="form-label">Student Name</label>
                <input type="text" class="form-control" name="name" value="<?=
$row['name']; ?>" required>
            </div>
```

```
            <div class="mb-3">
                <label class="form-label">Email</label>
                <input type="email" class="form-control" name="email" value="<?=
$row['email']; ?>" required>
            </div>

            <div class="mb-3">
                <label class="form-label">Course</label>
                <input type="text" class="form-control" name="course" value="<?=
$row['course']; ?>" required>
            </div>

            <button name="update" class="btn btn-success w-100">Update</button>
            <a href="index.php" class="btn btn-secondary w-100 mt-2">Back</a>
        </form>
    </div>
</div>
</body>
</html>
```

Delete.php

```
<?php
include "db.php";

$id = $_GET['id'];

$sql = "DELETE FROM students WHERE id = $id";

if ($conn->query($sql) === TRUE) {
    header("Location: index.php?msg=Record deleted successfully");
    exit();
} else {
    echo "Error deleting record: " . $conn->error;
}
?>
```

Config.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "CRUD_db";

$conn = new mysqli($servername, $username, $password, $database);

if ($conn->connect_error) {
```

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

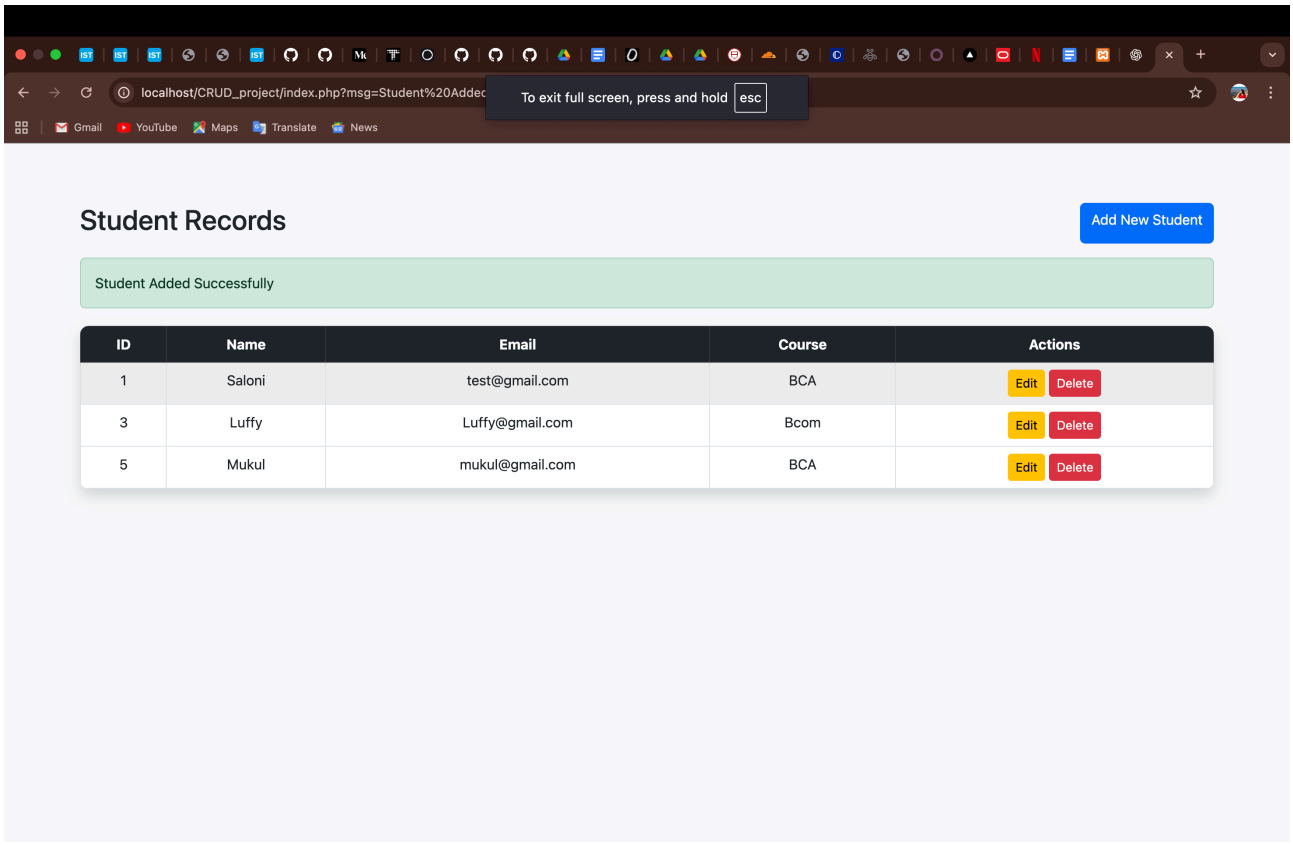
Style.css

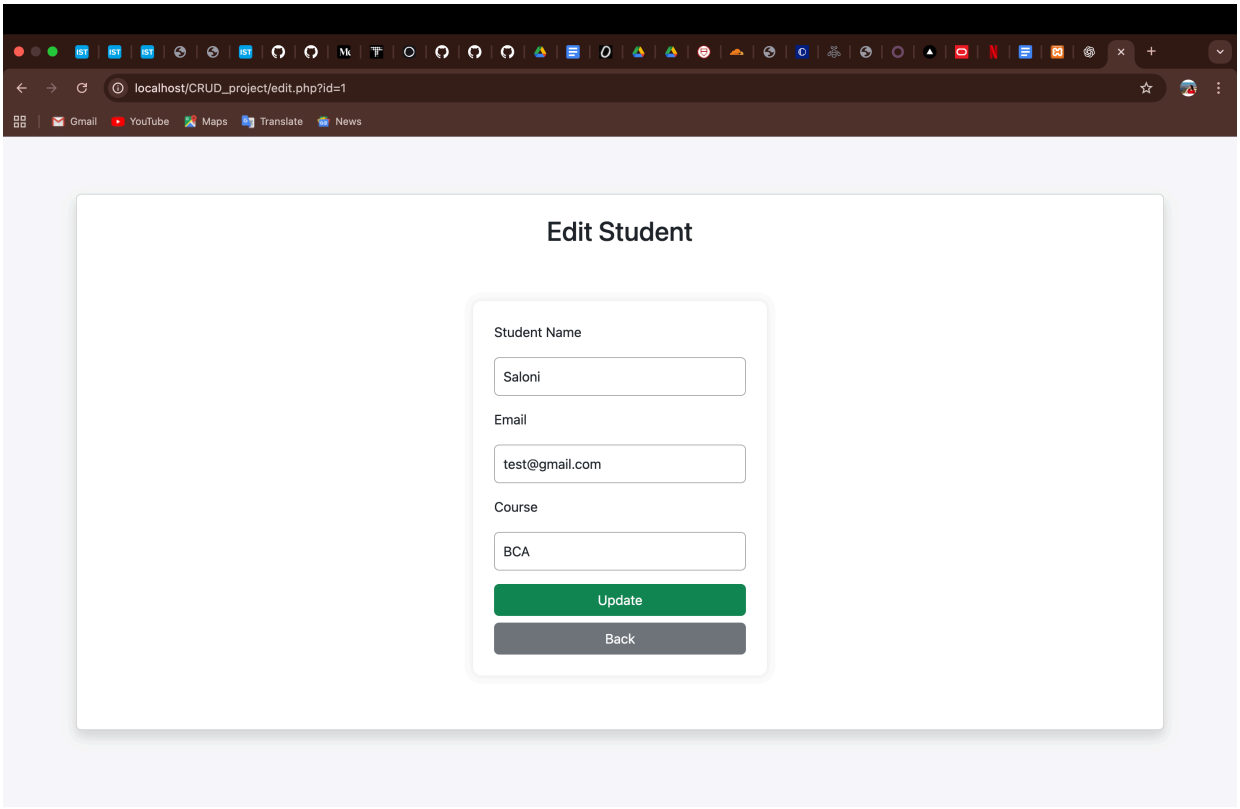
```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 20px;
    background: #f8f5ff;
}

h2 {
    color: #5a3e8c;
    text-align: center;
}

table {
    width: 80%;
    margin: 20px auto;
    border-collapse: collapse;
    background: white;
    border-radius: 10px;
    overflow: hidden;
}

table th, table td {
    padding: 12px 15px;
    border-bottom: 1px solid #ddd;
    text-align: center;
}

table th {
    background: #7b4bc4;
    color: white;
}

button, input[type="submit"] {
    background: #7b4bc4;
    color: white;
    padding: 10px 15px;
    border: none;
    border-radius: 6px;
    cursor: pointer;
}

button:hover, input[type='submit']:hover {
    background: #5a3ea9;
```

```
}

form {
    width: 350px;
    margin: 40px auto;
    background: white;
    padding: 25px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

input[type="text"], input[type="email"] {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border-radius: 6px;
    border: 1px solid #aaa;
}
```

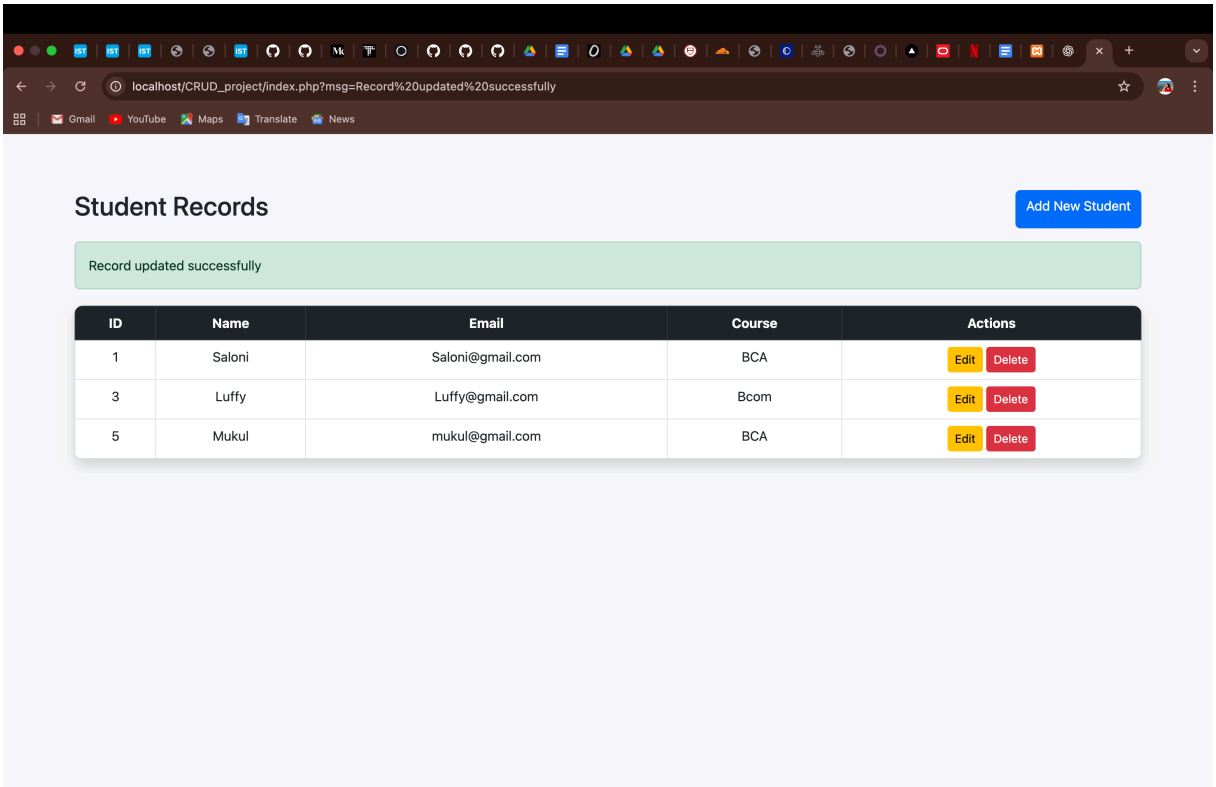
Screenshots [Output]

Adding Student in the form:





Edited a student record



Updated a record successfully:

THANK YOU!!