

Tribhuvan University

GREENFIELD NATIONAL COLLEGE

**Bafal- Sitapaila, Kathmandu**

**Lab Report**

**of**

**“Scripting Language”**

**Submitted By: Submitted To:**

Drona Paudel

Lecturer/ Faculty Member

BCA Department

Name: Manish karki

Semester: IV

Program: Bachelor of Computer Application

TU Reg. No: 6-2-717-6-2022

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **S.N** | **Topics** | **Signature** |
| 1 | Detailed history of Javascript:   1. Discuss its creation, evolution, and how it became a cornerstone of web development. 2. Include major milestones like the introduction of ECMAScript and advancements in frameworks/libraries. |  |
| 2 | Create an HTML form for user registration and write a JavaScript script to validate the form fields (e.g. email format, password length, required fields).Provide appropriate error messages when validation fails. |  |
| 3 | Create a web application to perform **CRUD (Create, Read, Update, Delete)** operations for a student database. |  |
| 4 | Create a **feedback form** in PHP that allows users to submit their name, email, and comments. Save the data in a MySQL database. Implement server-side validation and display a success message after submission. |  |
| 5 | Why AJAX and jquery is required in web development ? Write simple program to  sync data between browser and backend. |  |

**1. Write a detailed history of JavaScript.**

* Discuss its creation, evolution, and how it became a cornerstone of web development.
* Include major milestones like the introduction of ECMAScript and advancements in frameworks/libraries.

**Answer:**

* JavaScript, a cornerstone of modern web development, has a rich and fascinating history that spans over two decades.
* JavaScript was born in 1995, during the early days of the World Wide Web. Its creation is attributed to Brendan Eich, a developer at Netscape Communications Corporation. The language was conceived with a specific purpose: to enable interactivity on web pages, which were predominantly static at the time.
* The development of the JavaScript was rapid as Eich reportedly completed the first version of the language in just ten days during May 1995. Initially named Mocha, the language was briefly renamed LiveScript before finally being branded as JavaScript. This final name was chosen as a marketing strategy to capitalize on the popularity of Java, despite the two languages having little in common beyond some syntactical similarities.
* **Major Milestones and Version Releases**
  1. **ECMAScript 1 (ES1) - 1997**: The first standardized version of JavaScript, establishing a foundation for future development.
  2. **ECMAScript 3 (ES3) - 1999**: Introduced important features like regular expressions, better string handling, and structured error handling with try-catch statements.
  3. **Introduction of AJAX - 2005:** Asynchronous JavaScript and XML (AJAX) revolutionized web development by allowing web pages to be updated asynchronously without reloading the entire page. This technology enabled more dynamic and responsive web applications, akin to desktop applications.
  4. **Introduction of JSON - 2001**: JavaScript Object Notation (JSON) was developed as a lightweight data interchange format. The first JSON message was sent in April 2001, quickly becoming a popular alternative to XML for data exchange.
  5. **ECMAScript 5 (ES5) - 2009**: ES5 introduced "strict mode," JSON support, and higher-order functions, which enhanced code quality and expressiveness.
  6. **Introduction of Node.js - 2009**: Node.js extended JavaScript's capabilities to server-side development, allowing developers to use JavaScript for building scalable network applications.
  7. **ECMAScript 6 (ES6) - 2015**: Also known as ECMAScript 2015, ES6 was a significant milestone. It introduced arrow functions, classes, modules, promises, and template literals, modernizing JavaScript and making it more developer-friendly.
  8. **Subsequent Versions**: Since ES6, new versions of ECMAScript have been released annually, each bringing incremental improvements and new features. Notable additions include async/await in ES8 (2017) and optional chaining in ES11 (2020).
* **The Rise of Frameworks and Libraries**

As the javascript’s capabilities expanded and it is getting popular widely, many different frameworks and the libraries are designed to simplify development and improve efficiency.

1. **jQuery (2006)**: Created by John Resig, jQuery quickly became a staple in web development due to its ability to resolve inconsistencies in JavaScript implementations across browsers.
2. **Angular (2010)** : Developed by Google, Angular was designed as a full-fledged framework to build complex Single Page Applications (SPAs) with features like two-way data binding and dependency injection.
3. **React (2013)** : Developed by Facebook, React introduced a component-based model that revolutionized UI development, allowing for reusable and maintainable code
4. **Vue.js (2014)** : Created by Evan You, Vue.js combined the best features of Angular and React into a lightweight and easy-to-learn framework.

* **JavaScript's Impact on Web Development**

1. **Dynamic Web Pages:** JavaScript's ability to manipulate the Document Object Model (DOM) allowed developers to dynamically update the content, structure, and style of web pages based on user interactions.
2. **Enhanced User Experience:** Technologies like AJAX and JSON, coupled with JavaScript's capabilities, have enabled the creation of more responsive and interactive web applications, significantly improving user experience.
3. **Full-Stack Development:** With the introduction of Node.js, JavaScript became a viable option for server-side development, enabling full-stack development with a single language.
4. **Cross-Platform Development:** JavaScript's versatility has led to its use in various environments beyond the web browser, including desktop and mobile app development through frameworks like Electron and React Native.
5. **Standardization and Consistency:** The standardization of JavaScript through ECMAScript has ensured consistent behavior across different browsers and platforms, resolving many of the early compatibility issues.

**2. Create an HTML form for user registration and write a JavaScript script to validate the form fields (e.g., email format, password length, required fields).Provide appropriate error messages when validation fails. Write code.**

**Code:**

//index.php

<?php

if ($\_SERVER["REQUEST\_METHOD"] == 'POST') {

    $full\_name = $\_POST['f\_name'];

    $email = $\_POST['email'];

    $username = $\_POST['u\_name'];

    $password = $\_POST['password'];

    $errors = [];

    if (strlen($full\_name) > 40) {

        $errors[] = "full name must not exceed 40 length";

    } else if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

        $errors[] = "Invalid email address";

    } else if (!preg\_match("/^[a-zA-Z]+\d+$/", $username)) {

        $errors[] = "username must start with a string followed by a number";

    } else if (strlen($password) < 8) {

        $errors[] = "Password must be at least 8 character long";

    }

    if (!empty($errors)) {

        foreach ($errors as $error) {

            echo "<p style='color:red;'> $error</p>";

        }

    }

    $conn = new mysqli('localhost', 'root', 'password', "validation");

    if ($conn->connect\_error) {

        die("connection failed");

    }

    $sql = "INSERT INTO registration(full\_name,email,username,password) VALUES ('$full\_name','$email','$username','$password')";

    if ($conn->query($sql)) {

        echo "registration successfully";

    } else {

        echo "Error: " . $conn->error;

    }

    $conn->close();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>form validation</title>

</head>

<body>

    <form action="index.php" method="post">

        <label for="f\_name">full name</label>

        <input type="text" name="f\_name" placeholder="enter your full name" maxlength="40" required><br><br>

        <label for="email">Email</label>

        <input type="email" name="email" placeholder="enter your email"><br><br>

        <label for="u\_name">User Name</label>

        <input type="text" name="u\_name" placeholder="enter your user name" required><br><br>

        <label for="password">password</label>

        <input type="password" name="password" placeholder="enter your password"><br><br>

        <button type="submit">Submit form</button>

    </form>

</body>

</html>

**Sql query:**

CREATE database validation;

USE validation;

CREATE TABLE registration(

id INT AUTO\_INCREMENT PRIMARY KEY,

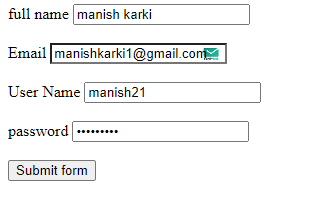
full\_name VARCHAR(40),

email VARCHAR(100),

username VARCHAR(50) UNIQUE,

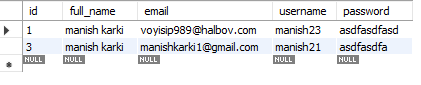
password VARCHAR(255)

);

**Output:**

**After submitting:**

SELECT \* FROM registration;

****

**3. Create a web application to perform CRUD (Create, Read, Update, Delete) operations for a student database.**

* The table should include fields like id, name, age, email, and course.

Write code.

**Code:**

**Config.php**

<?php

$host = 'localhost';

$dbname = 'student\_db';

$username = 'root';

$password = '';

$conn = new mysqli($host, $username, $password, $dbname);

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

$sql = "CREATE TABLE IF NOT EXISTS students (

    id INT PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(100) NOT NULL,

    age INT NOT NULL,

    email VARCHAR(100) UNIQUE NOT NULL,

    course VARCHAR(100) NOT NULL,

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

)";

if (!$conn->query($sql)) {

    echo "Error creating table: " . $conn->error;

}

?>

**Index.php**

<?php

include 'config.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Student Database Management</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.min.css">

</head>

<body>

    <div class="container mt-5">

        <h2>Student Database Management</h2>

        <div class="card mb-4">

            <div class="card-header">Add New Student</div>

            <div class="card-body">

                <form action="process.php" method="POST">

                    <div class="mb-3">

                        <label for="name" class="form-label">Name</label>

                        <input type="text" class="form-control" id="name" name="name" required>

                    </div>

                    <div class="mb-3">

                        <label for="age" class="form-label">Age</label>

                        <input type="number" class="form-control" id="age" name="age" required>

                    </div>

                    <div class="mb-3">

                        <label for="email" class="form-label">Email</label>

                        <input type="email" class="form-control" id="email" name="email" required>

                    </div>

                    <div class="mb-3">

                        <label for="course" class="form-label">Course</label>

                        <input type="text" class="form-control" id="course" name="course" required>

                    </div>

                    <button type="submit" name="add" class="btn btn-primary">Add Student</button>

                </form>

            </div>

        </div>

        <div class="card">

            <div class="card-header">Student List</div>

            <div class="card-body">

                <table class="table">

                    <thead>

                        <tr>

                            <th>ID</th>

                            <th>Name</th>

                            <th>Age</th>

                            <th>Email</th>

                            <th>Course</th>

                            <th>Actions</th>

                        </tr>

                    </thead>

                    <tbody>

                        <?php

                        $result = $conn->query("SELECT \* FROM students ORDER BY id DESC");

                        while ($row = $result->fetch\_assoc()) {

                            echo "<tr>";

                            echo "<td>{$row['id']}</td>";

                            echo "<td>{$row['name']}</td>";

                            echo "<td>{$row['age']}</td>";

                            echo "<td>{$row['email']}</td>";

                            echo "<td>{$row['course']}</td>";

                            echo "<td>

                                    <a href='edit.php?id={$row['id']}' class='btn btn-sm btn-warning'>Edit</a>

                                    <a href='process.php?delete={$row['id']}' class='btn btn-sm btn-danger' onclick='return confirm(\"Are you sure?\")'>Delete</a>

                                  </td>";

                            echo "</tr>";

                        }

                        ?>

                    </tbody>

                </table>

            </div>

        </div>

    </div>

</body>

</html>

**Edit.php**

<?php

include 'config.php';

if (isset($\_GET['id'])) {

    $id = $conn->real\_escape\_string($\_GET['id']);

    $result = $conn->query("SELECT \* FROM students WHERE id = '$id'");

    $student = $result->fetch\_assoc();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Edit Student</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.min.css">

</head>

<body>

    <div class="container mt-5">

        <h2>Edit Student</h2>

        <div class="card">

            <div class="card-body">

                <form action="process.php" method="POST">

                    <input type="hidden" name="id" value="<?php echo $student['id']; ?>">

                    <div class="mb-3">

                        <label for="name" class="form-label">Name</label>

                        <input type="text" class="form-control" id="name" name="name"

                            value="<?php echo $student['name']; ?>" required>

                    </div>

                    <div class="mb-3">

                        <label for="age" class="form-label">Age</label>

                        <input type="number" class="form-control" id="age" name="age"

                            value="<?php echo $student['age']; ?>" required>

                    </div>

                    <div class="mb-3">

                        <label for="email" class="form-label">Email</label>

                        <input type="email" class="form-control" id="email" name="email"

                            value="<?php echo $student['email']; ?>" required>

                    </div>

                    <div class="mb-3">

                        <label for="course" class="form-label">Course</label>

                        <input type="text" class="form-control" id="course" name="course"

                            value="<?php echo $student['course']; ?>" required>

                    </div>

                    <button type="submit" name="update" class="btn btn-primary">Update Student</button>

                    <a href="index.php" class="btn btn-secondary">Cancel</a>

                </form>

            </div>

        </div>

    </div>

</body>

</html>

**Process.php**

<?php

include 'config.php';

if (isset($\_POST['add'])) {

    $name = $conn->real\_escape\_string($\_POST['name']);

    $age = $conn->real\_escape\_string($\_POST['age']);

    $email = $conn->real\_escape\_string($\_POST['email']);

    $course = $conn->real\_escape\_string($\_POST['course']);

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

    if ($conn->query($sql)) {

        header("Location: index.php");

    } else {

        echo "Error: " . $conn->error;

    }

}

if (isset($\_POST['update'])) {

    $id = $conn->real\_escape\_string($\_POST['id']);

    $name = $conn->real\_escape\_string($\_POST['name']);

    $age = $conn->real\_escape\_string($\_POST['age']);

    $email = $conn->real\_escape\_string($\_POST['email']);

    $course = $conn->real\_escape\_string($\_POST['course']);

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

    if ($conn->query($sql)) {

        header("Location: index.php");

    } else {

        echo "Error: " . $conn->error;

    }

}

if (isset($\_GET['delete'])) {

    $id = $conn->real\_escape\_string($\_GET['delete']);

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

    if ($conn->query($sql)) {

        header("Location: index.php");

    } else {

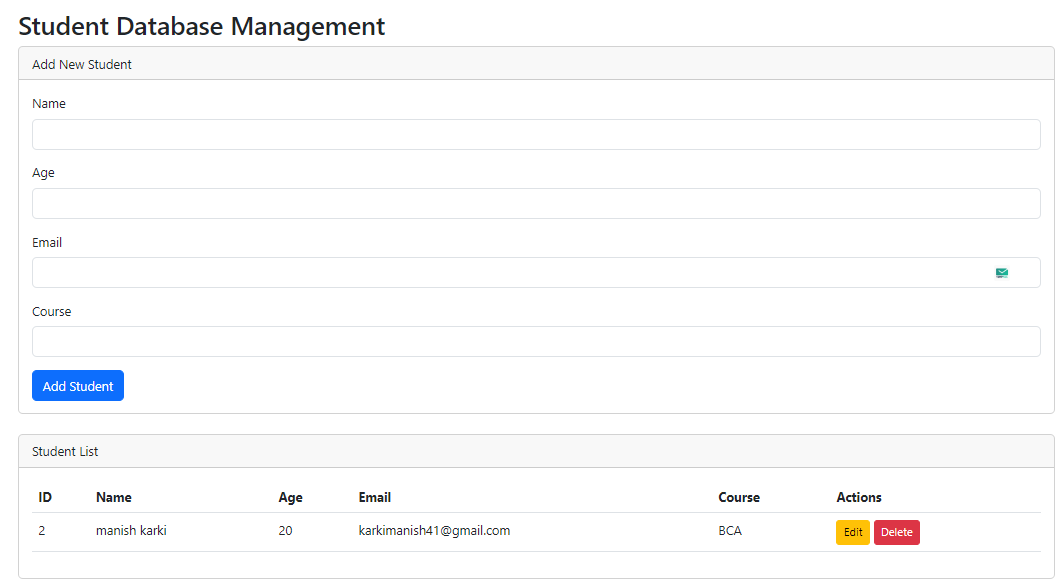
        echo "Error: " . $conn->error;

    }

}

$conn->close();

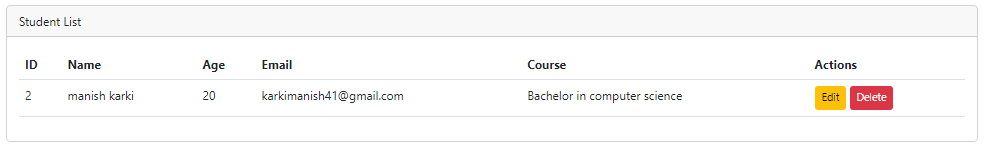
?>

**Output :**

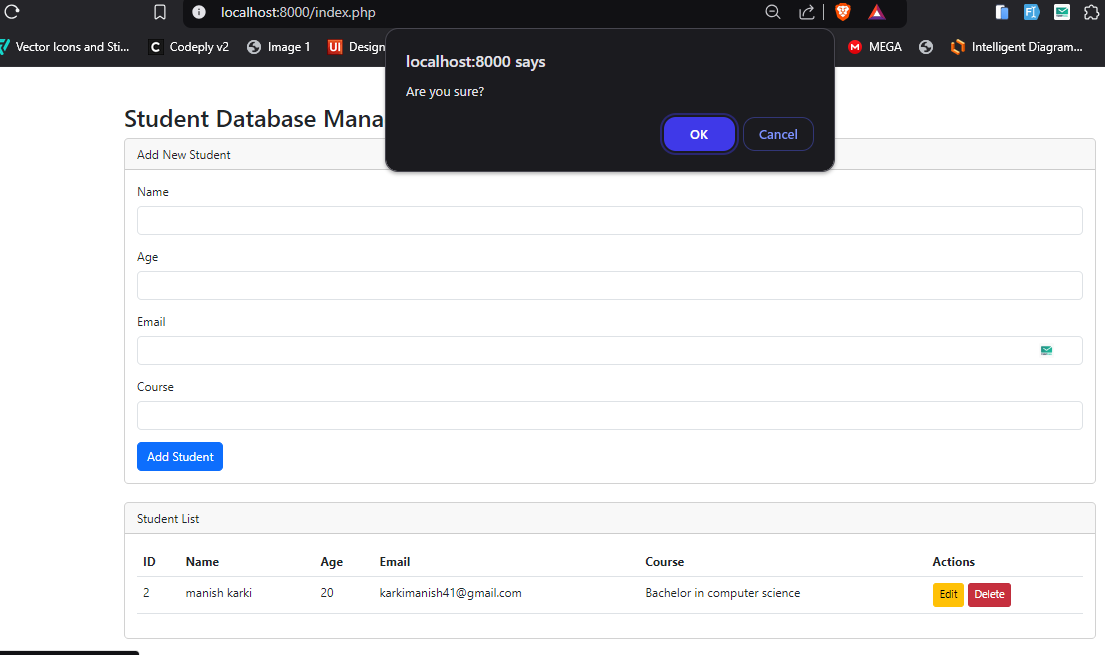
* **when click on the Edit button:**

****

* **After updating the course from BCA to Bachelor in computer science:**

****

* **After clicking on the delete button:**

****

ID 2 has been deleted and the table is now empty:

****

**4.  Create a feedback form in PHP that allows users to submit their name, email, and comments. Save the data in a MySQL database. Implement server-side validation and display a success message after submission.**

**Code:**

<!-- index.php -->

<?php

$host = 'localhost';

$dbname = 'feedback';

$username = 'root';

$password = 'D@t@base123';

$conn = new mysqli($host, $username, $password, $dbname);

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

$sql = "CREATE TABLE IF NOT EXISTS feedback (

    id INT PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(100) NOT NULL,

    email VARCHAR(100) NOT NULL,

    comments TEXT NOT NULL,

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

)";

if (!$conn->query($sql)) {

    echo "Error creating table: " . $conn->error;

}

$name = $email = $comments = "";

$nameErr = $emailErr = $commentsErr = "";

$successMsg = "";

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    if (empty($\_POST["name"])) {

        $nameErr = "Name is required";

    } else {

        $name = test\_input($\_POST["name"]);

        if (!preg\_match("/^[a-zA-Z ]\*$/", $name)) {

            $nameErr = "Only letters and white space allowed";

        }

    }

    if (empty($\_POST["email"])) {

        $emailErr = "Email is required";

    } else {

        $email = test\_input($\_POST["email"]);

        if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

            $emailErr = "Invalid email format";

        }

    }

    if (empty($\_POST["comments"])) {

        $commentsErr = "Comments are required";

    } else {

        $comments = test\_input($\_POST["comments"]);

        if (strlen($comments) < 10) {

            $commentsErr = "Comments must be at least 10 characters long";

        }

    }

    if (empty($nameErr) && empty($emailErr) && empty($commentsErr)) {

        $sql = "INSERT INTO feedback (name, email, comments) VALUES (?, ?, ?)";

        $stmt = $conn->prepare($sql);

        $stmt->bind\_param("sss", $name, $email, $comments);

        if ($stmt->execute()) {

            $successMsg = "Thank you for your feedback!";

            $name = $email = $comments = "";

        } else {

            echo "Error: " . $stmt->error;

        }

        $stmt->close();

    }

}

function test\_input($data)

{

    $data = trim($data);

    $data = stripslashes($data);

    $data = htmlspecialchars($data);

    return $data;

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Feedback Form</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.min.css">

    <style>

        .error {

            color: red;

        }

        .success {

            color: green;

            font-weight: bold;

        }

    </style>

</head>

<body>

    <div class="container mt-5">

        <div class="row justify-content-center">

            <div class="col-md-8">

                <div class="card">

                    <div class="card-header">

                        <h2 class="text-center">Feedback Form</h2>

                    </div>

                    <div class="card-body">

                        <?php if (!empty($successMsg)): ?>

                            <div class="alert alert-success">

                                <?php echo $successMsg; ?>

                            </div>

                        <?php endif; ?>

                        <form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

                            <div class="mb-3">

                                <label for="name" class="form-label">Name:</label>

                                <input type="text" class="form-control" id="name" name="name"

                                    value="<?php echo $name; ?>">

                                <span class="error"><?php echo $nameErr; ?></span>

                            </div>

                            <div class="mb-3">

                                <label for="email" class="form-label">Email:</label>

                                <input type="email" class="form-control" id="email" name="email"

                                    value="<?php echo $email; ?>">

                                <span class="error"><?php echo $emailErr; ?></span>

                            </div>

                            <div class="mb-3">

                                <label for="comments" class="form-label">Comments:</label>

                                <textarea class="form-control" id="comments" name="comments"

                                    rows="5"><?php echo $comments; ?></textarea>

                                <span class="error"><?php echo $commentsErr; ?></span>

                            </div>

                            <div class="text-center">

                                <button type="submit" class="btn btn-primary">Submit Feedback</button>

                            </div>

                        </form>

                    </div>

                </div>

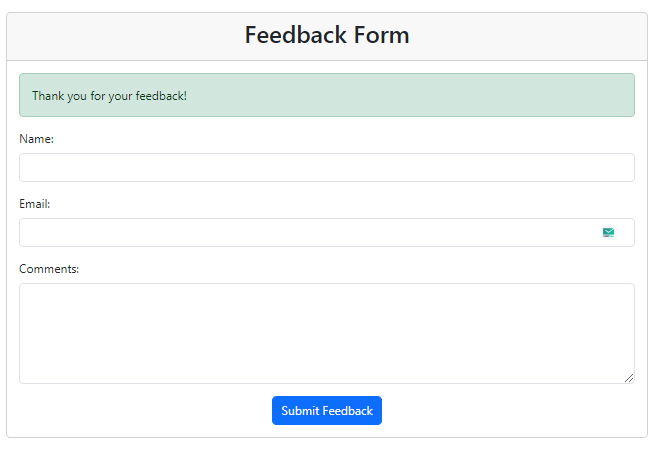
            </div>

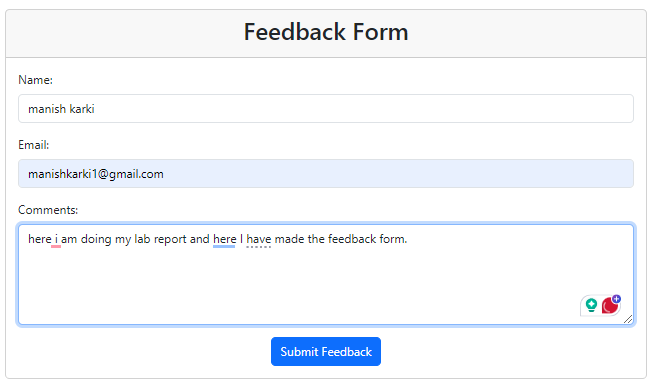
        </div>

    </div>

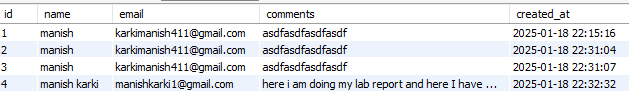
</body>

</html>

**Output:**

**After submission:**

Let’s see, the database feedback, whether we have the submission data or not.

****SELECT \* FROM feedback;

**5. Why AJAX and jquery is required in web development ? Write simple program to  sync data between browser and backend.**

* **Importance of AJAX in Web Development**

AJAX (Asynchronous JavaScript and XML) is a crucial technique in modern web development that enhances user experience by allowing web applications to communicate with the server asynchronously. This means that web pages can update content dynamically without requiring a full page reload. For instance, when a user submits a form, AJAX can validate the input in real-time, checking if a username is available before the user proceeds to the next step.

The benefits of using AJAX include:

1. **Improved User Interaction**: AJAX enables seamless content integration, which leads to a more engaging experience for users. This responsiveness can significantly increase customer satisfaction.
2. **Efficiency**: By only updating parts of a web page, AJAX reduces the amount of data transferred between the client and server, leading to faster load times and a more efficient use of resources.

* **Role of jQuery in AJAX**

jQuery is a popular JavaScript library that simplifies the process of making AJAX requests. It provides a clean and straightforward API for handling these requests, making it easier for developers to implement dynamic content updates. With jQuery, developers can easily manage AJAX calls, handle responses, and manipulate the DOM to reflect changes without extensive coding.

Key advantages of using jQuery with AJAX include:

1. **Simplified Syntax:** jQuery's syntax is more concise and easier to understand compared to vanilla JavaScript, which can make development faster and less error-prone.
2. **Built-in Animation Capabilities**: jQuery also offers built-in functions for animations, allowing developers to create visually appealing transitions when content is updated dynamically.

* **program to sync data between browser and backend:**

**Index.php :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>AJAX Data Sync Example</title>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.min.css">

</head>

<body>

    <div class="container mt-5">

        <div class="row">

            <div class="col-md-6">

                <div class="card">

                    <div class="card-header">

                        <h4>Add Note</h4>

                    </div>

                    <div class="card-body">

                        <form id="noteForm">

                            <div class="mb-3">

                                <label for="note" class="form-label">Note:</label>

                                <textarea class="form-control" id="note" rows="3"></textarea>

                            </div>

                            <button type="submit" class="btn btn-primary">Save Note</button>

                        </form>

                    </div>

                </div>

            </div>

            <div class="col-md-6">

                <div class="card">

                    <div class="card-header">

                        <h4>Notes <small>(Auto-updates every 5 seconds)</small></h4>

                    </div>

                    <div class="card-body">

                        <div id="notesList"></div>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <script>

        $(document).ready(function () {

            function loadNotes() {

                $.ajax({

                    url: 'get\_notes.php',

                    type: 'GET',

                    success: function (response) {

                        $('#notesList').html(response);

                    },

                    error: function (xhr, status, error) {

                        console.error('Error:', error);

                    }

                });

            }

            loadNotes();

            setInterval(loadNotes, 5000);

            $('#noteForm').on('submit', function (e) {

                e.preventDefault();

                $.ajax({

                    url: 'save\_note.php',

                    type: 'POST',

                    data: {

                        note: $('#note').val()

                    },

                    success: function (response) {

                        $('#note').val('');

                        loadNotes();

                    },

                    error: function (xhr, status, error) {

                        console.error('Error:', error);

                    }

                });

            });

        });

    </script>

</body>

</html>

**save\_note.php :**

<?php

$host = 'localhost';

$dbname = 'notes\_db';

$username = 'root';

$password = 'D@t@base123';

$conn = new mysqli($host, $username, $password, $dbname);

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

$sql = "CREATE TABLE IF NOT EXISTS notes (

    id INT PRIMARY KEY AUTO\_INCREMENT,

    note TEXT NOT NULL,

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

)";

$conn->query($sql);

if (isset($\_POST['note']) && !empty($\_POST['note'])) {

    $note = $conn->real\_escape\_string($\_POST['note']);

    $sql = "INSERT INTO notes (note) VALUES ('$note')";

    if ($conn->query($sql)) {

        echo "Note saved successfully";

    } else {

        echo "Error: " . $conn->error;

    }

}

$conn->close();

?>

**get\_notes.php**

<?php

$host = 'localhost';

$dbname = 'notes\_db';

$username = 'root';

$password = 'D@t@base123';

$conn = new mysqli($host, $username, $password, $dbname);

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

$sql = "SELECT \* FROM notes ORDER BY created\_at DESC";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

    while ($row = $result->fetch\_assoc()) {

        echo '<div class="card mb-2">';

        echo '<div class="card-body">';

        echo '<p class="card-text">' . htmlspecialchars($row['note']) . '</p>';

        echo '<small class="text-muted">' . $row['created\_at'] . '</small>';

        echo '</div>';

        echo '</div>';

    }

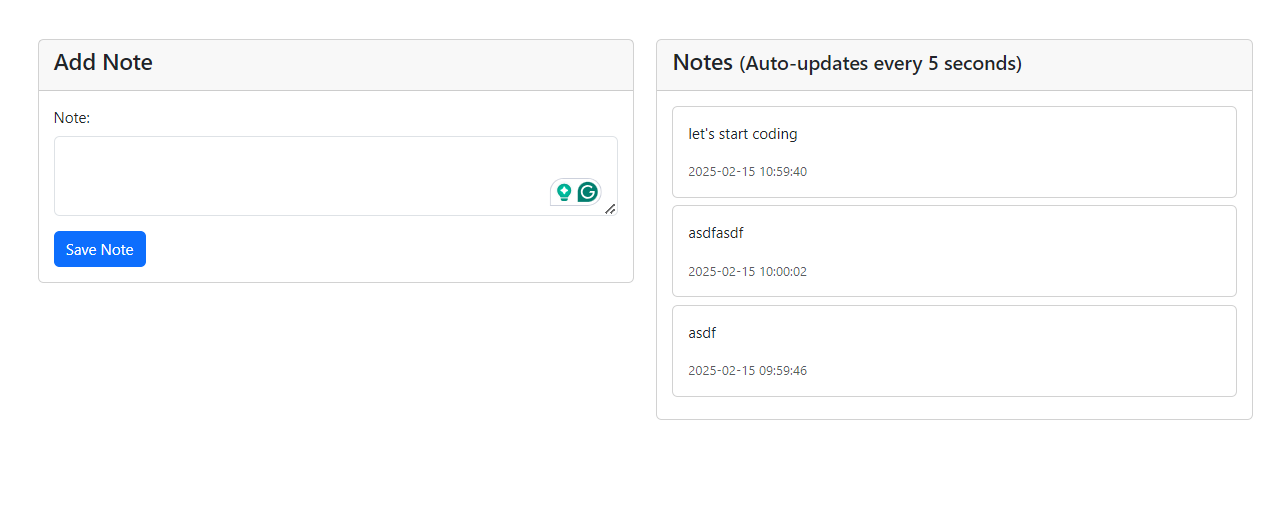
} else {

    echo '<p class="text-center">No notes found</p>';

}

$conn->close();

?>

**Output:**