**Aim:** To develop a JSP-based web application for managing student records in a database. The application will include four functionalities:

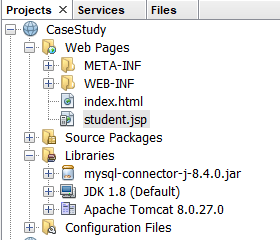
1. Add Student - Allows users to enter and save student details (ROLLNO, NAME, BRANCH) into the database.
2. View Students - Displays all student records in a tabular format.
3. Edit Student - Enables users to update existing student details in the database by providing the ROLLNO, NAME, and BRANCH.
4. Delete Student - Allows users to delete a student record from the database by specifying the ROLLNO.

The system facilitates seamless CRUD (Create, Read, Update, Delete) operations on the STUDENT table while ensuring an interactive and user-friendly interface for managing student data efficiently.

**Technologies Used:**

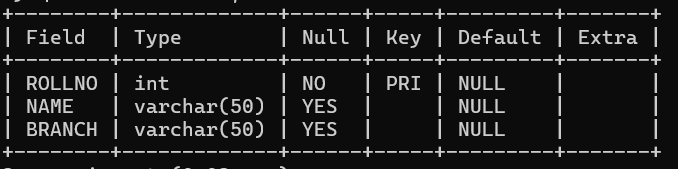
* **Frontend:** HTML and CSS for structuring and styling the user interface.
* **Backend:** JSP for server-side scripting and logic implementation.
* **Database:** MySQL for storing and managing student data.
* **Web Server:** Apache Tomcat for deploying and running the JSP application.

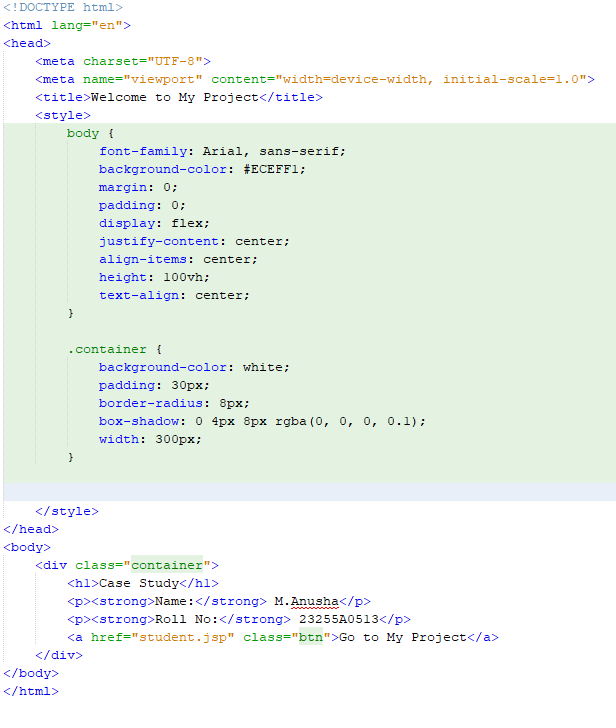
File Structure:



mysql> CREATE TABLE STUDENT (ROLLNO INT PRIMARY KEY,NAME VARCHAR(50),

BRANCH VARCHAR(50));



Index.html  


Student.jsp

<%@ page import="java.sql.\*" %>

<%@ page import="java.util.\*" %>

<!DOCTYPE html>

<html>

<head>

<title>Student Management System</title>

<style>

/\* General styles \*/

body {

font-family:'Oswald';

background-color: #ECEFF1; /\* Light grey background \*/

margin: 0;

padding: 0;

}

.container {

max-width: 800px;

margin: 50px auto;

background-color: white;

padding: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

border-radius: 8px;

}

header {

text-align: center;

font-family: 'Oswald';

font-size: 40px; /\* Increased font size for header \*/

color: #1A237E; /\* Dark indigo color \*/

margin-bottom: 30px;

letter-spacing: 2px; /\* Added letter spacing for more emphasis \*/

}

h1 {

text-align: center;

color: #1A237E; /\* Dark indigo for headings \*/

margin-bottom: 20px;

}

.actions {

text-align: center;

margin-bottom: 20px;

}

.actions a {

text-decoration: none;

margin: 0 10px;

padding: 10px 20px;

background-color: #283593; /\* Indigo shade for buttons \*/

color: white;

border-radius: 5px;

transition: background 0.3s ease;

}

.actions a:hover {

background-color: #1A237E; /\* Darker indigo for hover effect \*/

}

form {

margin: 20px 0;

}

form label {

display: block;

margin-bottom: 10px;

color: #1A237E; /\* Indigo text for labels \*/

font-weight: bold;

}

form input[type="text"],

form input[type="number"] {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #B0BEC5; /\* Light grey border \*/

border-radius: 5px;

}

form input[type="submit"] {

background-color: #283593; /\* Indigo shade for buttons \*/

color: white;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

font-size: 16px;

transition: background 0.3s ease;

}

form input[type="submit"]:hover {

background-color: #1A237E; /\* Darker indigo for hover effect \*/

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

}

table th, table td {

padding: 10px;

text-align: left;

border: 1px solid #B0BEC5; /\* Light grey border \*/

}

table th {

background-color: #1A237E; /\* Dark indigo for table headers \*/

color: white;

}

table tr:nth-child(even) {

background-color: #F5F5F5; /\* Subtle striped rows \*/

}

table tr:hover {

background-color: #E3F2FD; /\* Light blue hover for rows \*/

}

</style>

</head>

<body>

<header>Student Management System</header>

<div class="container">

<div class="actions">

<a href="student.jsp?action=add">Add Student</a>

<a href="student.jsp?action=view">View Students</a>

<a href="student.jsp?action=edit">Edit Student</a>

<a href="student.jsp?action=delete">Delete Student</a>

</div>

<%

// Database connection details

String url = "jdbc:mysql://localhost:3306/casectudy";

String user = "root";

String password = "password";

Connection con = null;

PreparedStatement ps = null;

ResultSet rs = null;

String action = request.getParameter("action");

try {

Class.forName("com.mysql.cj.jdbc.Driver");

con = DriverManager.getConnection(url, user, password);

if ("add".equals(action) && request.getParameter("submit") != null) {

// Add Student

int rollno = Integer.parseInt(request.getParameter("rollno"));

String name = request.getParameter("name");

String branch = request.getParameter("branch");

ps = con.prepareStatement("INSERT INTO STUDENT (ROLLNO, NAME, BRANCH) VALUES (?, ?, ?)");

ps.setInt(1, rollno);

ps.setString(2, name);

ps.setString(3, branch);

ps.executeUpdate();

out.println("<p style='color:green; text-align:center;'>Student added successfully!</p>");

} else if ("view".equals(action)) {

// View Students

ps = con.prepareStatement("SELECT \* FROM STUDENT");

rs = ps.executeQuery();

%>

<div class="table-container">

<table>

<tr>

<th>Roll No</th>

<th>Name</th>

<th>Branch</th>

</tr>

<%

while (rs.next()) {

%>

<tr>

<td><%= rs.getInt("ROLLNO") %></td>

<td><%= rs.getString("NAME") %></td>

<td><%= rs.getString("BRANCH") %></td>

</tr>

<%

}

%>

</table>

</div>

<%

} else if ("edit".equals(action) && request.getParameter("submit") != null) {

// Edit Student

int rollno = Integer.parseInt(request.getParameter("rollno"));

String name = request.getParameter("name");

String branch = request.getParameter("branch");

ps = con.prepareStatement("UPDATE STUDENT SET NAME = ?, BRANCH = ? WHERE ROLLNO = ?");

ps.setString(1, name);

ps.setString(2, branch);

ps.setInt(3, rollno);

ps.executeUpdate();

out.println("<p style='color:green; text-align:center;'>Student updated successfully!</p>");

} else if ("delete".equals(action) && request.getParameter("submit") != null) {

// Delete Student

int rollno = Integer.parseInt(request.getParameter("rollno"));

ps = con.prepareStatement("DELETE FROM STUDENT WHERE ROLLNO = ?");

ps.setInt(1, rollno);

ps.executeUpdate();

out.println("<p style='color:green; text-align:center;'>Student deleted successfully!</p>");

}

if ("add".equals(action)) {

%>

<div class="form-container">

<form method="post">

<label>Roll No:</label>

<input type="number" name="rollno" required>

<label>Name:</label>

<input type="text" name="name" required>

<label>Branch:</label>

<input type="text" name="branch" required>

<input type="submit" name="submit" value="Add Student">

</form>

</div>

<%

} else if ("edit".equals(action)) {

%>

<div class="form-container">

<form method="post">

<label>Roll No (to update):</label>

<input type="number" name="rollno" required>

<label>New Name:</label>

<input type="text" name="name" required>

<label>New Branch:</label>

<input type="text" name="branch" required>

<input type="submit" name="submit" value="Update Student">

</form>

</div>

<%

} else if ("delete".equals(action)) {

%>

<div class="form-container">

<form method="post">

<label>Roll No (to delete):</label>

<input type="number" name="rollno" required>

<input type="submit" name="submit" value="Delete Student">

</form>

</div>

<%

}

} catch (Exception e) {

out.println("<p style='color:red; text-align:center;'>Error: " + e.getMessage() + "</p>");

} finally {

try {

if (rs != null) rs.close();

if (ps != null) ps.close();

if (con != null) con.close();

} catch (SQLException e) {

e.printStackTrace();

}

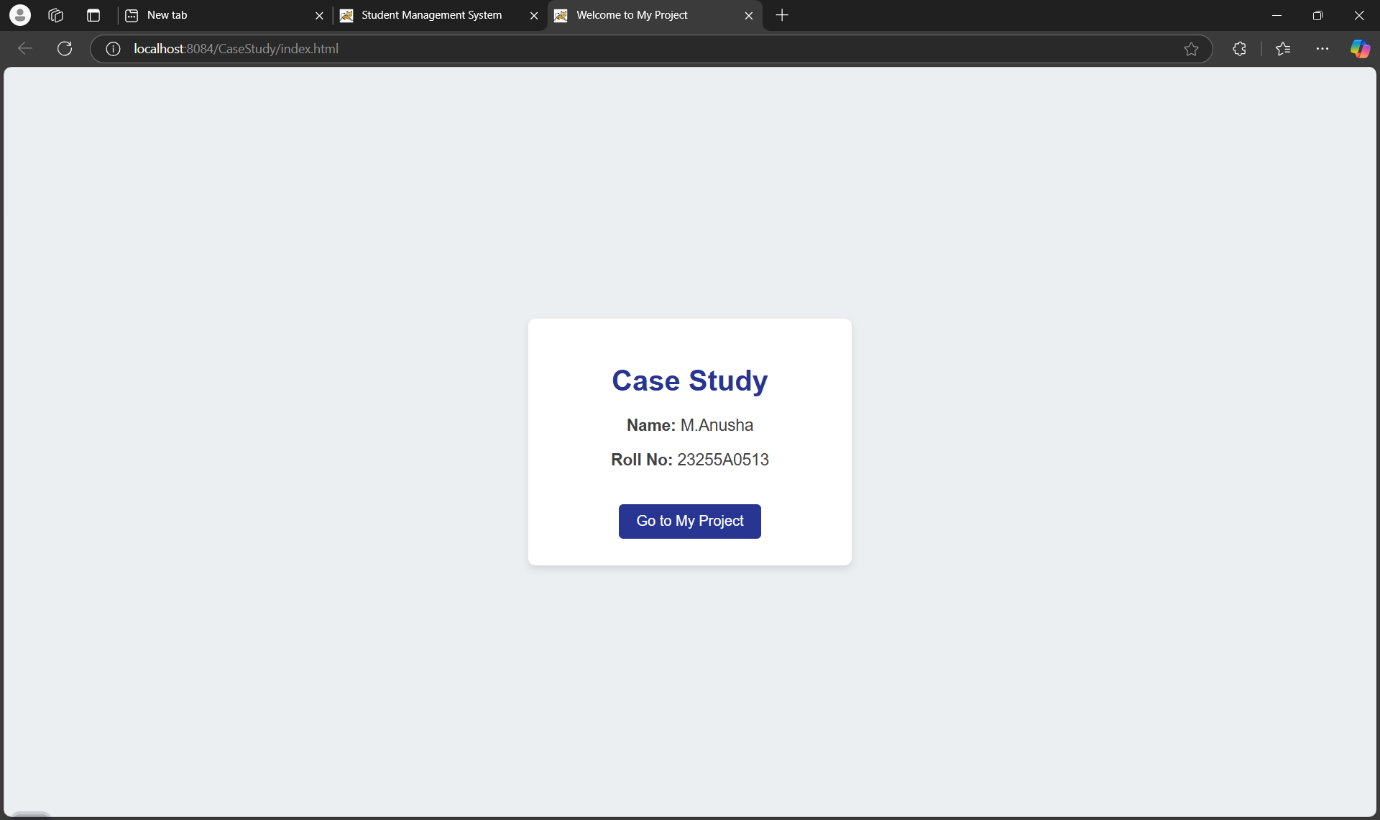
}

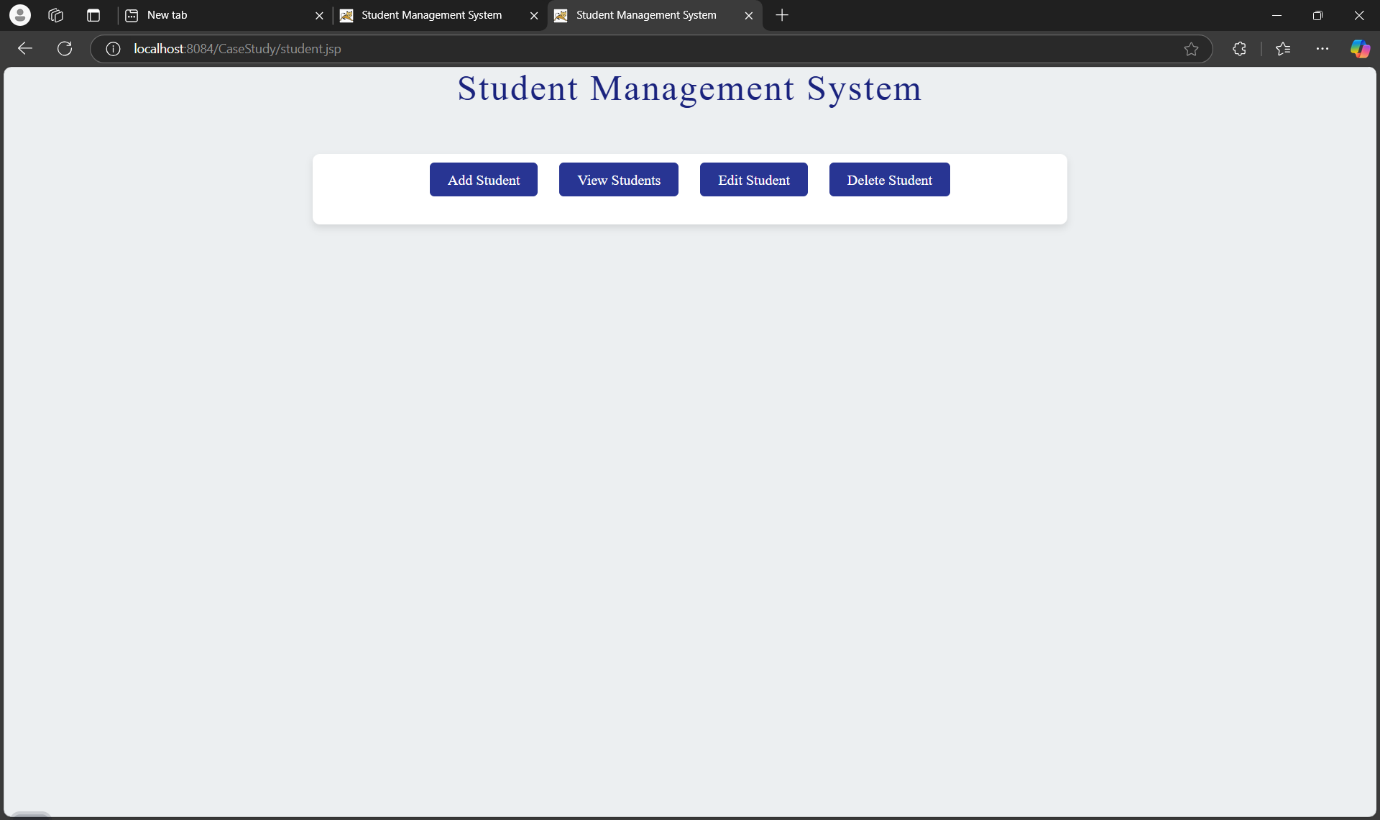
%>

</div>

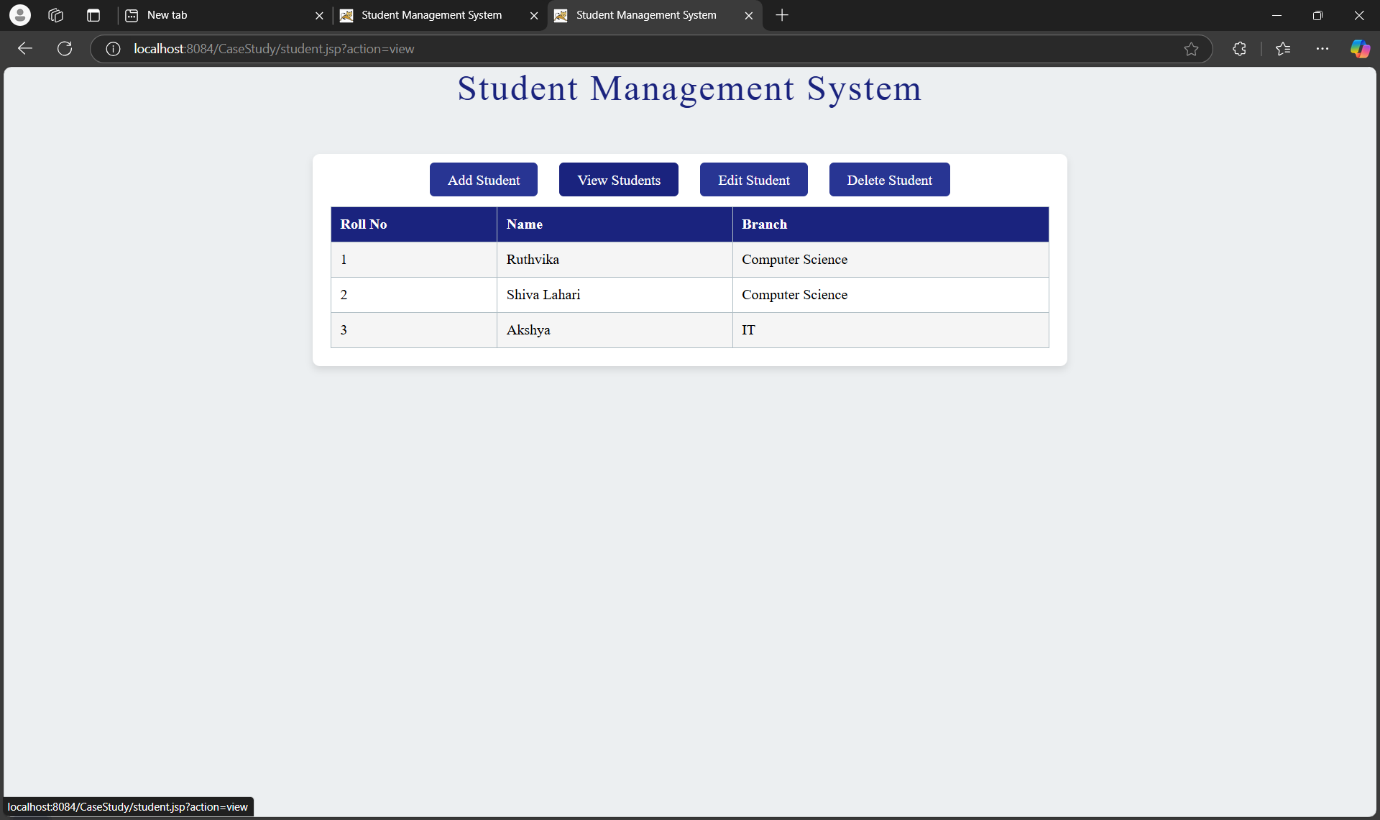
</body>

</html>

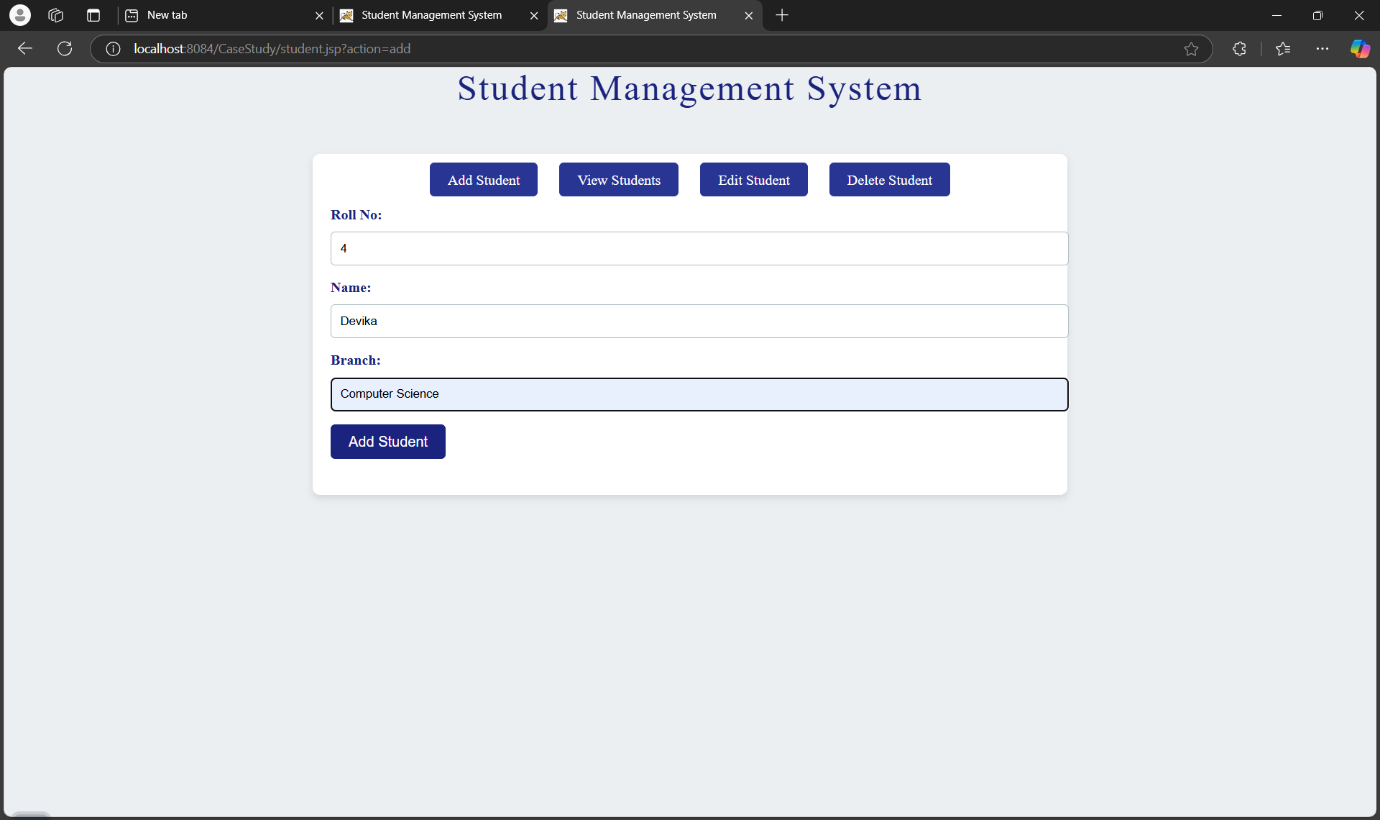




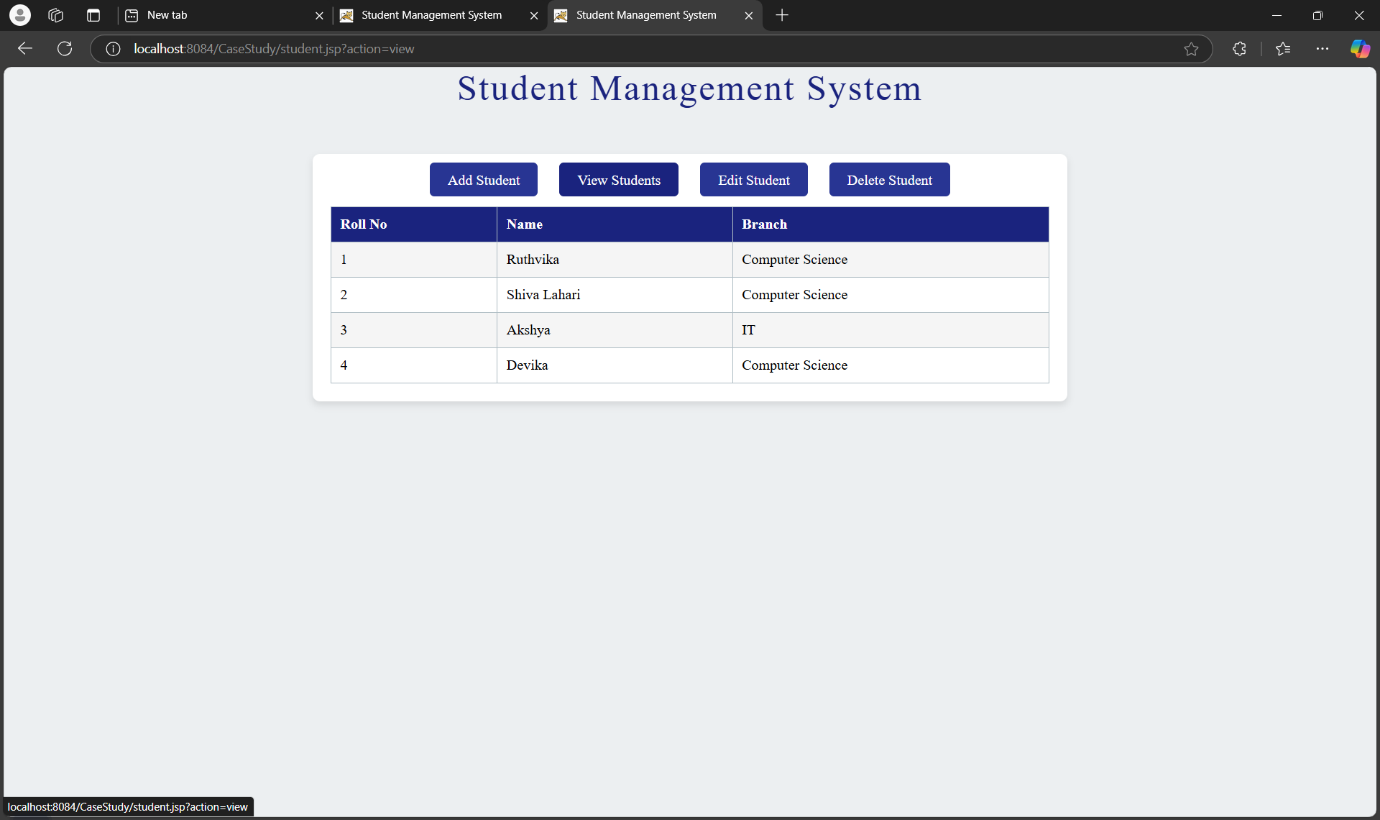
1. View Students information



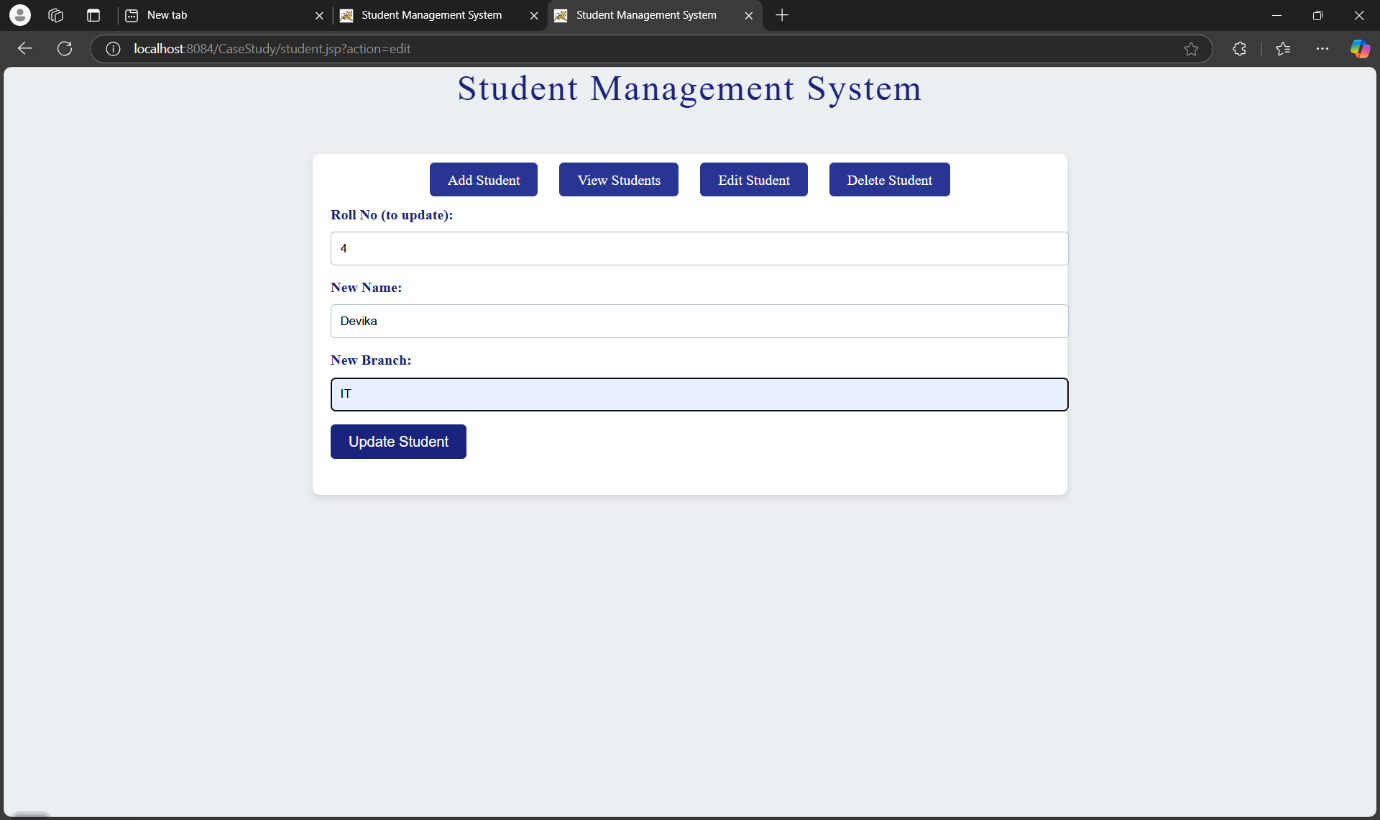
2.Add a Student to the database



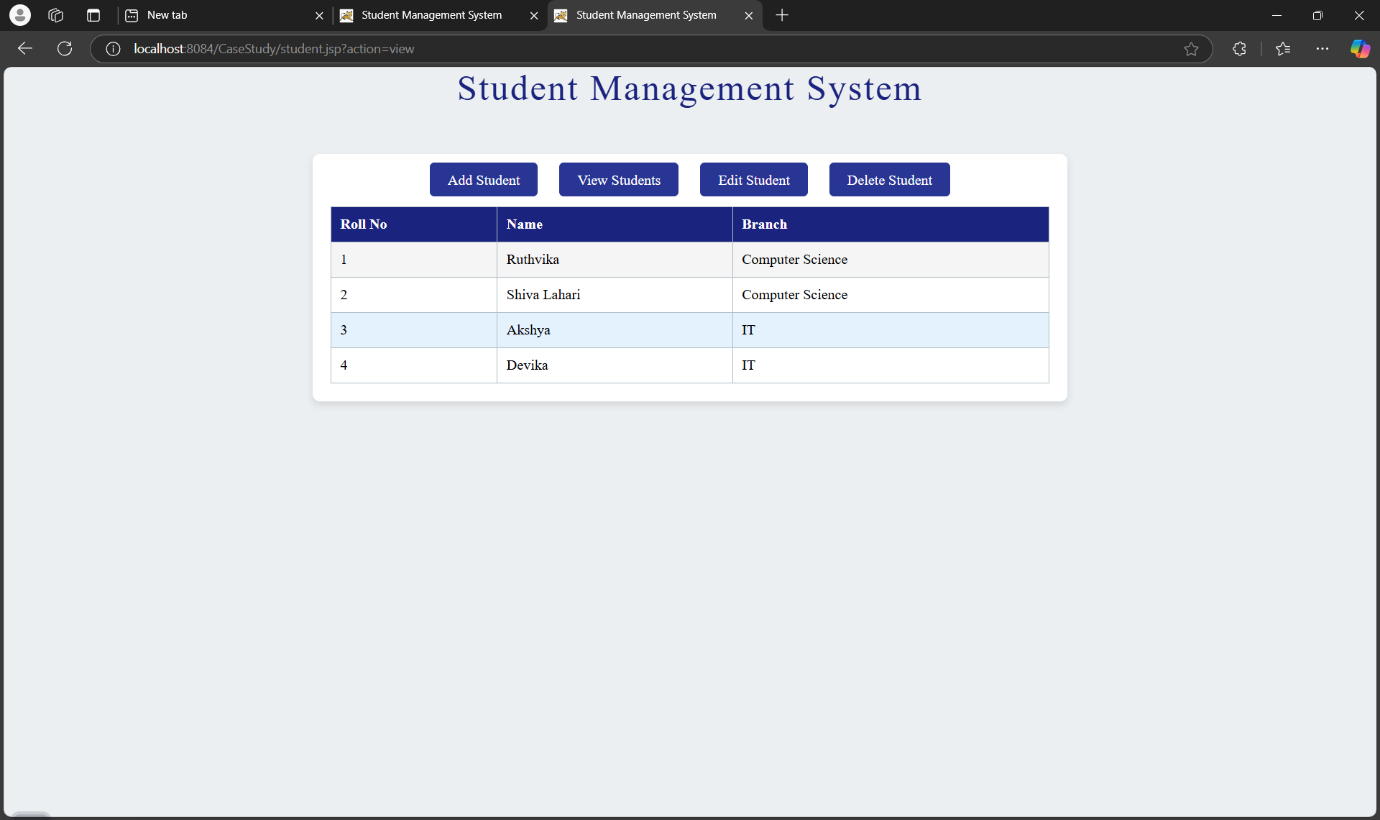
* Student information added



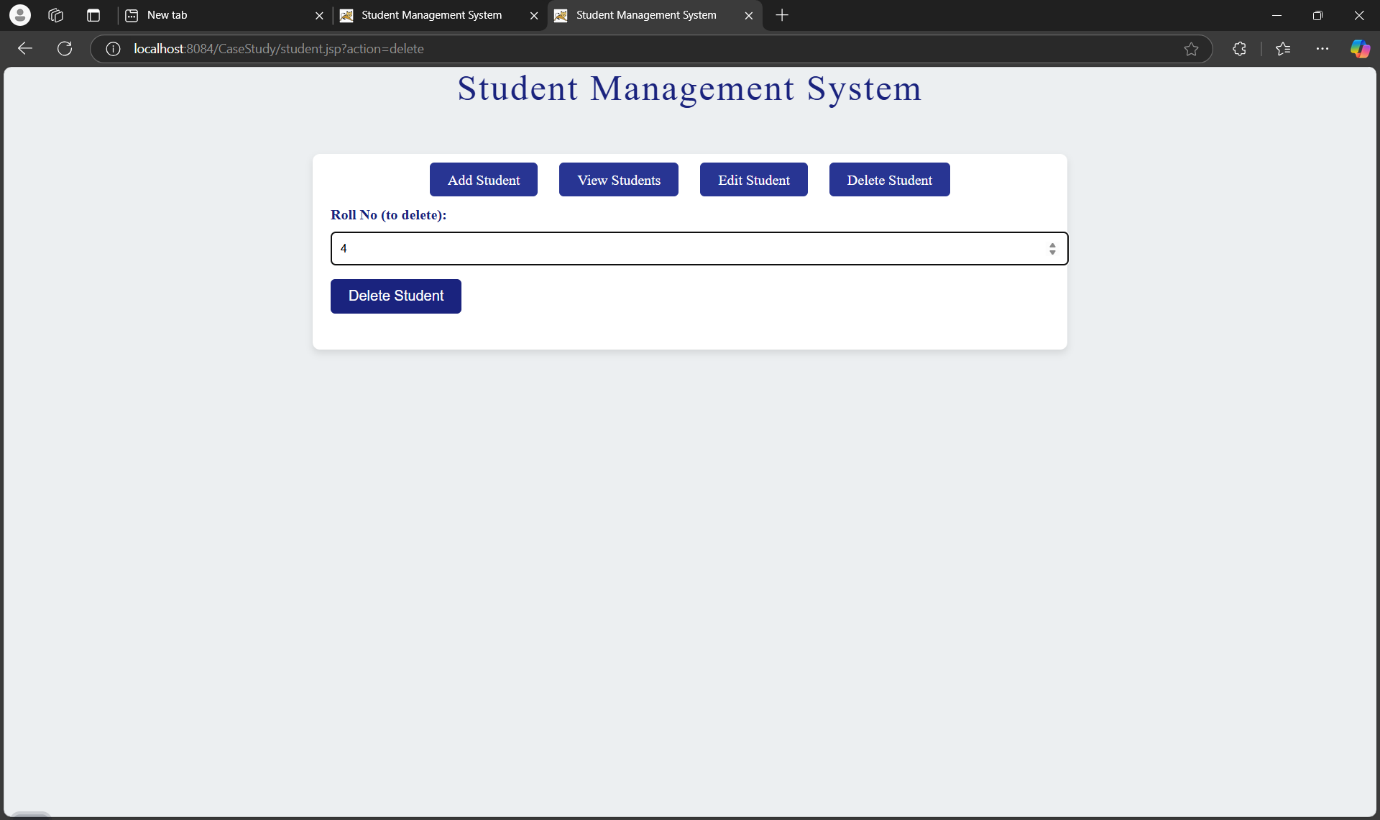
3.Update Student nformation



* Student information updated in the table.



4.Delete a Student‘ s information from the database



* Student information deleted.

