# CHAPTER - 1 INTRODUCTION

**INTRODUCTION**

The **Student Fee Tracker** is a modern, web-based solution designed to streamline and enhance the management of student fee records. It provides a robust platform for educational institutions to efficiently track fee payments, manage pending balances, and ensure accurate financial records. With its user-friendly interface and comprehensive functionality, the application enhances accessibility and ease of use for administrators, students, and parents.

In today’s digital era, where institutions are increasingly adopting online systems for administrative tasks, the **Student Fee Tracker** bridges the gap between traditional fee management methods and modern technology. By automating fee tracking, generating reports, and providing real-time insights into payment statuses, the application simplifies financial management for educational institutions.

Whether it’s for managing tuition payments, installment-based fees, or overdue balances, this application caters to a wide range of financial tracking needs. It is designed to improve transparency and efficiency while ensuring accurate and hassle-free fee management.

**Key Features**

**for Staff in Student Fee Tracker**

* **Admin & Semester-Based Access**
  + **Admin users** can access fee records for all semesters.
  + **Semester-specific users** can only view and manage records for their assigned semester.
* **Student Fee Management**  
  Staff can view and track fee payments, ensuring all transactions are recorded accurately.
* **Comprehensive Payment Tracking**
  + View **Total Fee**, **Cumulative Fee Paid**, and **Pending Fee** for each student.
  + The system ensures correct calculations, even when students pay in **multiple installments**.
* **Search & Filter Student Records**
  + Staff can search for students using **Register Number, Name, Receipt Number, Date, Month, or Year**.
  + Filter students based on **Paid** or **Pending Fees** to manage follow-ups effectively.
* **Dynamic Fee Calculation**
  + The **Total Fee** is updated dynamically for each student, ensuring accurate pending fee calculations.
  + **Installments are tracked**, preventing duplicate or incorrect fee entries.
* **Unauthorized Access Prevention**
  + Admins can access all semesters, but semester-specific users are restricted to their assigned semester.

**General Features**

* **User-Friendly Interface**
  + A simple and well-structured UI for easy navigation.
* **Secure Session Management**
  + Uses Flask-Session to securely store login sessions and prevent unauthorized access.
* **Real-Time Updates**
  + Any changes in fee payments or student records are instantly reflected.
* **Error Handling & Validation**
  + Ensures correct data formatting and prevents incorrect calculations.
* **Scalability & Performance**
  + Handles multiple student records efficiently without performance issues.

**For Teachers :**

**Project Overview**

The **Online Quiz Application** is designed with a focus on clean architecture, responsiveness, and functionality. Below are the key components and their respective technologies:

### Frontend :

##### Technologies:

* + **HTML**: For structuring the web pages.
  + **CSS**: For styling the application to ensure a professional and user-friendly design.
  + **React.js**: For building a dynamic, responsive, and modular user interface.

##### Frontend Features:

* + **Landing Page**: A homepage with options to log in as a student or teacher.

##### Student Dashboard:

* + - Login form to enter name, register number, semester, and lesson.
    - Display quizzes based on the selected lesson.
    - Real-time feedback and score display.

##### Teacher Dashboard:

* + Create, edit, or delete quizzes.
  + View and analyze student performance.
* **Responsive Design**: Ensures usability across devices.

### Backend :

##### Technologies:

* + **Node.js**: For handling server-side logic.
  + **Express.js**: For setting up RESTful APIs to handle requests and responses efficiently.
  + **JWT Authentication**: For secure user login sessions.

##### Backend Features:

* **Authentication APIs**:
  + POST /api/auth/login: Authenticate users based on their role (student/teacher).

##### Quiz APIs:

* + GET /api/quiz/:lessonId: Fetch questions for a specific lesson.
  + POST /api/quiz/submit: Submit answers and calculate results.
  + POST /api/quiz/create: Allow teachers to add or modify quizzes.

##### Result APIs:

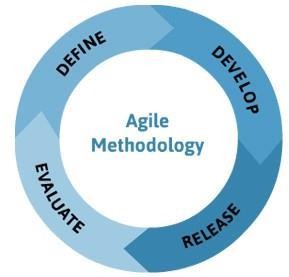
* + GET /api/results/student/:studentId: Fetch individual student results.
  + GET /api/results/lesson/:lessonId: Fetch results for a specific lesson (for teachers).

##### Data Validation:

* + Middleware to validate incoming data to ensure integrity and security.

### Methodology :

The methodology used in this project include repetition of design, code and implementation then testing and analyzing the result. Further when repeating the phases again, new functionalities are added to the project. This methodology is called agile methodology.



For example, while creating the basic structure and less of focus on creative interface first iteration of the phases was done. After evaluation in each cycle more focus on interface and debugging was done.

# CHAPTER - 2

## Home Page

### Home Page :

#### Code has been Provided for the Home.html page :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>East West Polytechnic</title>

<link rel="stylesheet" href="style.css"> </head>

<body>

<header>

<div id="logo-container">

<img id="logo" src="ewp.jpg" alt="East West Polytechnic Logo">

</div>

<h1 id="college-title">EAST-WEST POLYTECHNIC</h1>

</header>

<nav>

<ul>

<li><a href="home.html">Home</a></li>

<li><a href="login.html">Login</a></li>

<li><a href="#" id="quiz">Quiz</a></li>

<li><a href="#" id="teacher">Teacher</a></li>

<li><a href="#" id="report">Report</a></li>

<li><a href="Student.html">Student</a></li>

<li><a href="#" onclick="adminLogout()">Logout</a></li>

</ul>

</nav>

<main>

<div id="content">

<div id="text-container">

<h1>Welcome to East West Polytechnic</h1>

<p>East West Polytechnic, recognized by the Government of Karnataka, affiliated to the Department of Technical Education, Karnataka, and approved by AICTE, New Delhi, is a proud member of ISTE. Established in 2008, EWP is dedicated to providing high-quality technical education. Our campus is strategically located in the green city of Bengaluru and spans 20 acres, offering a serene and conducive learning environment.</p>

<p>East West Polytechnic, Anjananagar, Bengaluru, has always emphasized discipline and character building among its students. We take pride in providing a strong foundation for holistic education, ensuring our students are well-prepared for their future endeavors.</p>

<h2>Key Features</h2>

<ul>

<li><strong>Well-Equipped Laboratories:</strong> State-of-the-art laboratories for Computer Science, Electronics & Communication, Electrical & Electronics, Mechanical, and Civil Engineering.</li>

<li><strong>Full-Stacked Library:</strong> A comprehensive resource center with a vast collection of books, journals, and online resources.</li>

<li><strong>Sports Facilities:</strong> Indoor and outdoor facilities for various sports like table tennis, badminton, carom, chess, basketball, cricket, and more.</li>

<li><strong>Seminar Hall & Auditorium:</strong> Modern facilities for conducting seminars, workshops, and cultural events.</li>

<li><strong>Hostel Facilities:</strong> Comfortable and hygienic on-campus hostels for both boys and girls.</li>

<li><strong>Transport Facility:</strong> Safe and convenient transportation services with a fleet of buses.</li>

</ul>

</div>

<div id="image-container">

<img id="college-image" src="ewp-2.jpg" alt="East West Polytechnic Building">

</div>

</div>

</main>

<footer>

<p>&copy; 2024 East West Polytechnic. All rights reserved.</p>

</footer>

</body>

</html> .

#### Style.CSS :

/\* General Styles \*/ body {

font-family: sans-serif; margin: 0;

padding: 0;

background-color: #ececf5; /\* Light gray background \*/ color: #333; /\* Dark gray text \*/

}

/\* Header Styles \*/ header {

background-color: #003366; /\* Navy blue header \*/ display: flex;

align-items: center; /\* Vertically center items \*/

justify-content: center; /\* Space between left, center, and right \*/ padding: 1rem 2rem; /\* Padding for spacing \*/

position: fixed; top: 0;

width: 100%;

box-sizing: border-box;

z-index: 1000; /\* Ensure header stays on top \*/

}

/\* Logo Container \*/ #logo-container {

display: flex; align-items: left;

}

#logo {

max-height: 60px; /\* Adjust as needed for proper sizing \*/ height: auto;

width: auto;

}

/\* Title Container \*/ #title-container {

flex: 1;

text-align: center;

}

#college-title {

font-size: 3.5rem; /\* Adjust for emphasis \*/ font-weight: bold;

color: #fff; /\* White text for visibility \*/ margin: 0;

}

/\* Right Placeholder for Balance \*/ #right-placeholder {

width: 60px; /\* Same width as the logo to balance the layout \*/

}

/\* Navigation Styles \*/ nav {

background-color: #003366; /\* Navy blue navigation \*/ padding: 1rem 0;

margin-top: 80px; /\* Space below fixed header \*/

}

nav ul {

list-style: none; text-align: center; margin: 0;

padding: 0; display: flex;

justify-content: center;

}

nav li {

margin: 0 1.5rem;

}

nav a {

color: #fff;

text-decoration: none; font-weight: bold; font-size: 1rem;

}

nav a:hover {

text-decoration: underline;

}

/\* Main Content Styles \*/ main {

padding: 2rem;

margin-top: 0.2rem; /\* Adjust as needed \*/

}

/\* Content Alignment \*/ #content {

display: flex;

justify-content: left; max-width: 1500px; margin: auto;

}

#text-container { flex: 2;

padding-right: 2rem;

}

#image-container { flex: 1;

text-align: center;

}

#college-image { max-width: 100%; height: auto;

}

/\* Footer Styles \*/ footer {

background-color: #003366; color: #fff;

text-align: center; padding: 1rem 0; position: fixed; bottom: 0;

width: 100%;

}

/\* Login Page Styles \*/ #login-container {

background-color: rgba(255, 255, 255, 0.9); /\* Semi-transparent white \*/

padding: 2.5rem; border-radius: 12px; max-width: 500px; margin: 1.5rem auto; text-align: center;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);

}

#login-container h2 { color: #003366; margin-bottom: 2rem;

}

.form-group {

margin-bottom: 1.5rem; text-align: left;

}

.form-group label { display: block; font-weight: bold;

margin-bottom: 0.5rem;

}

.form-group input { width: 100%; padding: 0.8rem;

border: 1px solid #e8e4e4; border-radius: 5px;

}

button {

width: 100%; padding: 0.8rem;

background-color: #003366; color: white;

border: none;

border-radius: 5px; font-size: 1rem; cursor: pointer;

}

button[type="submit"]:hover {

background-color: #002b5c; /\* Slightly darker on hover \*/

}

#error-message { color: red;

margin-top: 1rem;

display: none; /\* Hidden initially \*/

}

.footer-note {

margin-top: 1.5rem; font-size: 0.9rem; color: #666;

}

.footer-note a { color: #003366;

text-decoration: none; font-weight: bold;

}

.footer-note a:hover {

text-decoration: underline;

}

/\* Responsive Adjustments \*/ @media (max-width: 768px) {

header {

flex-direction: column; align-items: flex-start;

}

#title-container { text-align: center; width: 100%;

margin-top: 0.5rem;

}

#right-placeholder {

display: none; /\* Hide placeholder on smaller screens \*/

}

nav ul {

flex-direction: column;

}

nav li {

margin: 0.5rem 0;

}

#login-container {

margin: 10rem auto 2rem; /\* Adjust spacing for smaller screens

\*/

}

}.

#### Code Explainition with applied Flow chart :

1. **Overview :-**

This code represents the **Home Page** of the East West Polytechnic website. It includes a **header, navigation menu, main content section, and footer**.

#### Key Features :-

* + **Header:** Displays the college logo and title.
  + **Navigation Menu:** Links to different pages like Home, Login, Quiz, Teacher, Report, Student, and Logout.

##### Main Content:

* + - Welcomes visitors with an introduction to East West Polytechnic.
    - Provides information about the college, including its history, affiliations, and key features.
    - Displays an image of the college.
  + **Footer:** Contains copyright information.

#### Functional Elements :-

##### Navigation Bar:

* + - Links to other pages: home.html, login.html, Student.html.
    - Some links (Quiz, Teacher, Report) have # as placeholders, meaning they don’t redirect yet.
    - Logout executes the adminLogout() function (assumed to be defined in a script).

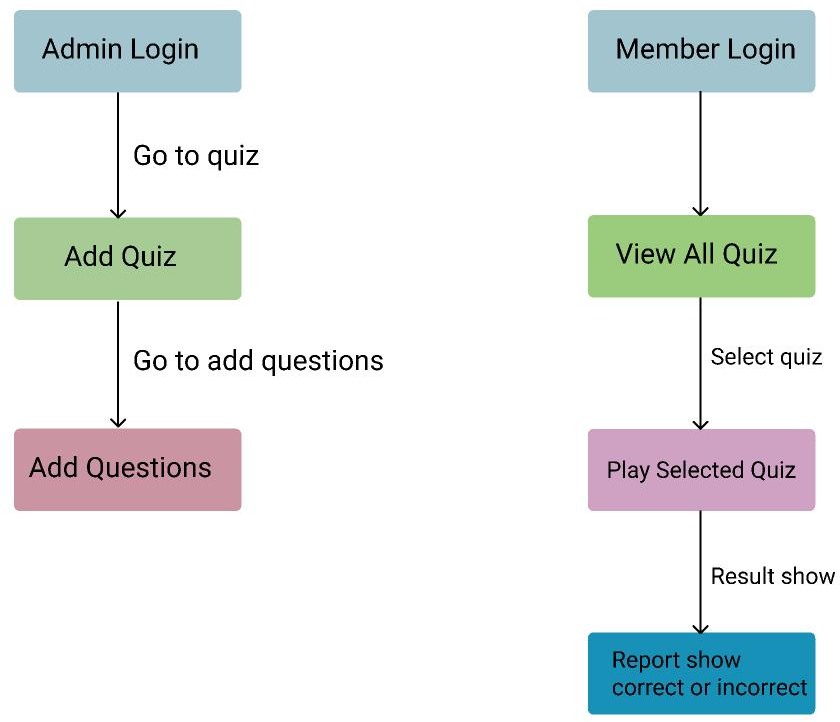
##### Main Content:

* + - Uses a structured layout with text and an image.

##### CSS Integration:

* + - Uses an external stylesheet (style.css) for design.

#### Flow Chart :

****

* **Admin Management:**
  + The admin logs in to manage quizzes by adding new quizzes and their respective questions.
* **Member Participation:**
  + Members log in to view all available quizzes and select one to attempt.
* **Quiz Process:**
  + After selecting a quiz, members play and submit their answers, which are recorded for evaluation.
* **Result Evaluation:**
  + A report is generated after submission, showing correct and incorrect answers for the quiz.
* **System Flow & Navigation:**
  + The system follows a structured flow, where admins manage quizzes and members participate, ensuring smooth user interaction and result feedback.

# CHAPTER - 3

## Login Page

### Login Page :

#### Code has been Provided for the Login.html page :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Admin Login - East West Polytechnic</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<div id="logo-container">

<img id="logo" src="ewp.jpg" alt="East West Polytechnic Logo">

</div>

<h1 id="college-title">EAST-WEST POLYTECHNIC</h1>

</header>

<nav>

<ul>

<li><a href="home.html">Home</a></li>

<li><a href="login.html">Login</a></li>

<li><a href="#" id="quiz">Quiz</a></li>

<li><a href="#" id="teacher">Teacher</a></li>

<li><a href="#" id="report">Report</a></li>

<li><a href="Student.html">Student</a></li>

<li><a href="#" onclick="adminLogout()">Logout</a></li>

</ul>

</nav>

<main>

<div id="login-container">

<h2>Administrator - Login</h2>

<form id="login-form">

<div class="form-group">

<label for="username">Username:</label>

<input type="text" id="username" name="username" placeholder="Enter your username">

</div>

<div class="form-group">

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

<input type="password" id="password" name="password" placeholder="Enter your password">

</div>

<p id="error-message" class="error" style="display: none; color: red; font-weight: bold;">

Incorrect username or password. Please try again.

</p>

<button type="submit">Login</button>

</form>

</div>

</main>

<footer>

<p>&copy; 2024 East West Polytechnic. All rights reserved.</p>

</footer>

<script src="script.js"></script>

</body>

</html> .

#### Code Explainition with applied Flow chart :

* **Form Handling:**

Ensure the login-form has an action attribute or JavaScript to handle login submission properly.

#### Navigation Issues:

<li><a href="#"> elements for **Quiz, Teacher, Report** don’t have proper links. You might want to replace # with valid URLs or handle them dynamically via JavaScript.

#### Logout Function:

Ensure adminLogout() function is defined in script.js to properly clear session data and redirect users.

#### Security Enhancements:

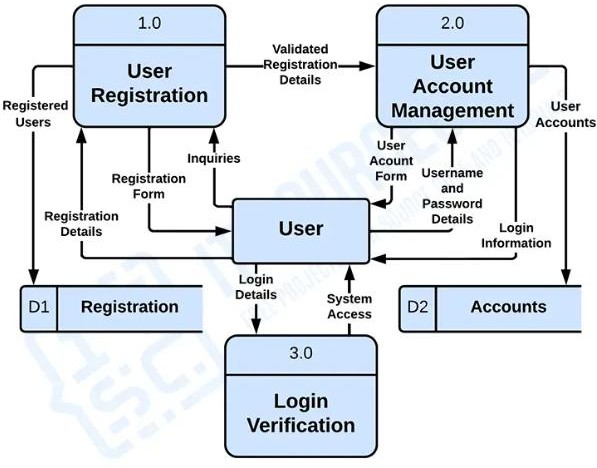
* + Add required attributes to input fields for better validation.
  + Implement password masking with type="password" already present, but consider hashing passwords before sending them.

#### CSS Adjustments:

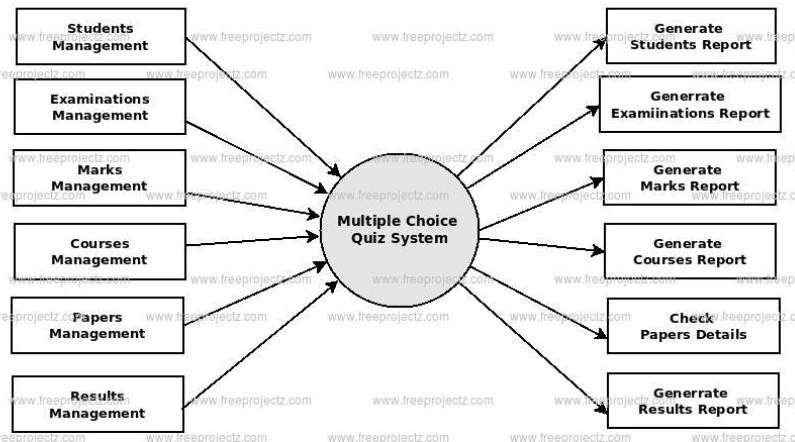
Ensure style.css has proper styling for the **login form alignment**, **button hover effects**, and **responsive layout** for mobile users.

#### Flow Chart :

It Display the Complete Work Flow all the Pages After Login it redirect for all the Page **Quiz, Teacher, Report**, Pages :



* **User Registration:** New users fill out a registration form, and the system validates and stores their details.
* **User Account Management:** Registered users’ account details (username & password) are managed and updated.
* **Login Verification:** Users enter login credentials, and the system verifies them for access.
* **User Interaction:** The user interacts with registration, account management, and login processes for authentication.
* **Data Storage:** Registration details (D1) and user account data (D2) are stored and managed securely.



* **Student & Exam Management:** The system handles student records, examination scheduling, and marks management.
* **Course & Paper Management:** It manages course details and exam papers efficiently.
* **Results & Reports Generation:** The system generates reports for students, examinations, marks, courses, and results.
* **Paper Checking & Validation:** Users can check paper details to ensure accuracy in assessments.
* **Comprehensive Quiz System:** Integrates all aspects of multiple-choice quiz management for smooth functionality.

# CHAPTER - 4

## Quiz Page

### Quiz Page :

#### Code has been Provided for the Quiz.html , Quiz.CSS, Quiz.js page :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Quiz - East West Polytechnic</title>

<link rel="stylesheet" href="Quiz.css">

</head>

<body>

<header>

<div id="logo-container">

<img id="logo" src="ewp.jpg" alt="East West Polytechnic Logo">

</div>

<h1 id="college-title">EAST-WEST POLYTECHNIC</h1>

</header>

<nav>

<ul>

<li><a href="home.html">Home</a></li>

<li><a href="login.html">Login</a></li>

<li><a href="Quiz.html">Quiz</a></li>

<li><a href="Teacher.html">Teacher</a></li>

<li><a href="Report.html">Report</a></li>

<li><a href="Student.html">Student</a></li>

<li><a href="#" onclick="adminLogout()">Logout</a></li>

</ul>

</nav>

<main>

<div id="quiz-container">

<h2>Add a Question</h2>

<form id="quiz-form">

<label for="course">Course Name:</label>

<input type="text" id="course" placeholder="Enter course name" required>

<label for="question">Question:</label>

<input type="text" id="question" placeholder="Enter question" required>

<label>Options:</label>

<input type="text" class="option" placeholder="Option 1" required>

<input type="text" class="option" placeholder="Option 2" required>

<input type="text" class="option" placeholder="Option 3" required>

<input type="text" class="option" placeholder="Option 4" required>

<label for="correct-answer">Correct Answer:</label>

<input type="text" id="correct-answer" placeholder="Correct Option Text" required>

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

</form>

<div id="success-message" style="display: none; color: green;">

Quiz successfully added!

</div>

</div>

</main>

<footer>

<p>&copy; 2024 East West Polytechnic. All rights reserved.</p>

</footer>

<script>

document.getElementById("quiz- form").addEventListener("submit", function (e) {

e.preventDefault();

const course = document.getElementById("course").value.trim();

const question = document.getElementById("question").value.trim();

const options = Array.from(document.querySelectorAll(".option")).map(opt => opt.value.trim());

const correctAnswer = document.getElementById("correct-answer").value.trim();

if (!course || !question || options.includes("")

|| !correctAnswer) {

alert("Please fill in all fields."); return;

}

const quizzes = JSON.parse(localStorage.getItem("quizzes")) || [];

let courseIndex = quizzes.findIndex(q => q.course === course);

if (courseIndex === -1) {

quizzes.push({ course, questions: [] }); courseIndex = quizzes.length - 1;

}

quizzes[courseIndex].questions.push({ question, options, correctAnswer });

localStorage.setItem("quizzes", JSON.stringify(quizzes)); document.getElementById("success-

message").style.display = "block"; setTimeout(() => {

window.location.href = "Teacher.html";

}, 1500);

document.getElementById("quiz-form").reset();

});

</script>

<script src="Quiz.js"></script>

</body>

</html> .

##### Quiz.CSS :

/\* General Styles \*/ body {

font-family: sans-serif; margin: 0;

padding: 0;

background-color: #f0f0f0; /\* Light gray background \*/ color: #333; /\* Dark gray text \*/

}

/\* Header Styles \*/ header {

background-color: #003366; /\* Navy blue header \*/ display: flex;

align-items: center; /\* Vertically center items \*/

justify-content: center; /\* Space between left, center, and right

\*/

padding: 1rem 2rem; /\* Padding for spacing \*/ position: fixed;

top: 0;

width: 100%;

box-sizing: border-box;

z-index: 1000; /\* Ensure header stays on top \*/

}

/\* Logo Container \*/ #logo-container {

display: flex; align-items: left;

}

#logo {

max-height: 60px; /\* Adjust as needed for proper sizing \*/ height: auto;

width: auto;

}

/\* Title Container \*/ #title-container {

flex: 1;

text-align: center;

}

#college-title {

font-size: 3.5rem; /\* Adjust for emphasis \*/ font-weight: bold;

color: #fff; /\* White text for visibility \*/ margin: 0;

}

/\* Right Placeholder for Balance \*/ #right-placeholder {

width: 60px; /\* Same width as the logo to balance the layout

\*/

}

/\* Navigation Styles \*/

nav {

background-color: #003366; /\* Navy blue navigation \*/ padding: 1rem 0;

margin-top: 80px; /\* Space below fixed header \*/

}

nav ul {

list-style: none; text-align: center; margin: 0;

padding: 0; display: flex;

justify-content: center;

}

nav li {

margin: 0 1.5rem;

}

nav a {

color: #fff;

text-decoration: none; font-weight: bold; font-size: 1rem;

}

nav a:hover {

text-decoration: underline;

}

main {

max-width: 900px; margin: 13px auto; background: white; padding: 30px; border-radius: 30px;

box-shadow: 0 5px 8px rgba(0, 0, 0, 0.1);

}

h2 {

text-align: center; color: #004d99;

}

.form-group {

margin-bottom: 15px;

}

label {

font-weight: bold; display: block; margin-bottom: 5px;

}

input, textarea { width: 100%; padding: 8px;

border: 1px solid #ccc; border-radius: 10px;

}

textarea { resize: none;

}

.options-group label { display: flex;

align-items: center; margin-bottom: 10px;

}

.options-group input[type="radio"] { margin-right: 10px;

}

button {

background-color: #004d99; color: white;

border: none; padding: 10px 15px; border-radius: 10px; cursor: auto;

font-size: 20px;

}

button:hover {

background-color: #003366;

}

#success-message { text-align: center; color: green;

font-weight: bold; margin-top: 10px;

}

/\* Footer Styles \*/ footer {

background-color: #003366; color: #fff;

text-align: center; padding: 1.2rem 0; position: fixed; bottom: 0;

width: 100%;

} .

##### Quiz.js :

document.addEventListener("DOMContentLoaded", () => { const quizForm = document.getElementById("quiz-form"); const successMessage = document.getElementById("success-

message");

quizForm.addEventListener("submit", (e) => { e.preventDefault();

const questionNumber = document.getElementById("question-number").value.trim();

const courseName = document.getElementById("course- name").value.trim();

const questionText = document.getElementById("question- text").value.trim();

const marks = document.getElementById("marks").value.trim();

const options = { option1:

document.querySelector("input[name='option1']").value.trim(), option2:

document.querySelector("input[name='option2']").value.trim(), option3:

document.querySelector("input[name='option3']").value.trim(), option4:

document.querySelector("input[name='option4']").value.trim(),

};

const correctOption = document.querySelector("input[name='correct- option']:checked").value;

// Save the question in localStorage const quizData =

JSON.parse(localStorage.getItem("quizData")) || [];

quizData.push({ questionNumber, courseName, questionText, marks, options, correctOption });

localStorage.setItem("quizData", JSON.stringify(quizData));

// Show success message and reset the form successMessage.style.display = "block"; setTimeout(() => {

successMessage.style.display = "none";

}, 2000);

quizForm.reset();

});

});

// Logout function function adminLogout() {

localStorage.removeItem("adminLoggedIn"); alert("You have been logged out.");

window.location.href = "home.html"; // Redirect to login page

}

#### Code Explainition :

##### Event Listener for Page Load

* + Ensures the script runs only after the page fully loads.
  + Retrieves the quiz form and success message elements.

##### Retrieving Input Values

* + Captures **question number, course name, question text, and marks** from the form.
  + Uses .trim() to remove extra spaces.

##### Retrieving Answer Options

* + Uses a loop to **collect all option values** dynamically.
  + Checks if all options are filled.

##### Retrieving the Correct Answer

* + Identifies the selected radio button for the correct option.

##### Saving Quiz Data in localStorage

* + Retrieves existing quiz data or initializes an empty array.
  + Adds the new question to the array.
  + Saves the updated quiz data in localStorage.

##### Displaying Success Message & Resetting Form

* + Shows a success message for 2 seconds.
  + Clears all input fields after submission.

##### Handling Admin Logout

* + Removes adminLoggedIn from localStorage.
  + Displays a logout confirmation alert.
  + Redirects to "home.html" for login.

# CHAPTER - 5

## Teacher Page

### Teacher Page :

#### Code has been Provided Teacher.html,Teacher.CSS, Teacher.js page :

##### Teacher.html :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Quiz - East West Polytechnic</title>

<link rel="stylesheet" href="Teacher.css"> </head>

</head>

<body>

<header>

<div id="logo-container">

<img id="logo" src="ewp.jpg" alt="East West Polytechnic Logo">

</div>

<h1 id="college-title">EAST-WEST POLYTECHNIC</h1>

</header>

<nav>

<ul>

<li><a href="home.html">Home</a></li>

<li><a href="login.html">Login</a></li>

<li><a href="Quiz.html">Quiz</a></li>

<li><a href="Teacher.html">Teacher</a></li>

<li><a href="Report.html">Report</a></li>

<li><a href="Student.html">Student</a></li>

<li><a href="#" onclick="adminLogout()">Logout</a></li>

</ul>

</nav>

<main>

<div id="course-list">

<h2>Courses</h2>

<div id="courses">

<!-- Dynamic Course List -->

</div>

</div>

<div id="quiz-sheet">

<div id="rules">

<h3>Quiz Rules</h3>

<ul>

<li>Each question has only one correct answer.</li>

<li>Select the correct option and proceed to the next question.</li>

<li>Correct answers will be highlighted in green, and incorrect ones in red.</li>

</ul>

</div>

<div id="questions">

<p>Select a course to view the quiz.</p>

</div>

<div>

<button class="applyButton">Apply</button>

<button class="cancel-btn">Cancel</button>

</div>

</div>

</main>

<footer>

<p>&copy; 2024 East West Polytechnic. All rights reserved.</p>

</footer>

<script src="Teacher.js"></script>

</body>

</html>

##### Teacher.js :

document.addEventListener("DOMContentLoaded", () => { const coursesDiv = document.getElementById("courses"); const questionsDiv = document.getElementById("questions");

let quizzes = JSON.parse(localStorage.getItem("quizzes")) ||

[];

function renderCourses() { coursesDiv.innerHTML = ""; quizzes.forEach((quiz, index) => {

const courseItem = document.createElement("div"); courseItem.className = "course-item"; courseItem.innerHTML = `

<span>${quiz.course}</span>

<button onclick="viewQuiz(${index})">View</button>

<button class="delete-btn" onclick="deleteCourse(${index})">&#128465;</button>

`; coursesDiv.appendChild(courseItem);

});

}

window.viewQuiz = (index) => { const selectedQuiz = quizzes[index];

if (selectedQuiz && selectedQuiz.questions.length > 0) { questionsDiv.innerHTML = selectedQuiz.questions

.map(

(q, i) => `

<div class="question">

<p><strong>Q${i + 1}:</strong> <span id="q- text-${index}-${i}">${q.question}</span></p>

${q.options

.map(

(option, j) => `

<label>

<input type="radio" name="question${i}">

<span id="opt-${index}-${i}-

${j}">${option}</span>

</label>

`

)

.join("")}

<br>

<button onclick="updateQuestion(${index},

${i})"> Update</button>

<button onclick="deleteQuestion(${index},

${i})"> Delete</button>

</div>

<hr>

`

)

.join("");

} else {

questionsDiv.innerHTML = "<p>No questions available for this course.</p>";

}

};

window.deleteCourse = (index) => {

if (confirm("Are you sure you want to delete this course?"))

{

quizzes.splice(index, 1); // Remove course from the

array

localStorage.setItem("quizzes", JSON.stringify(quizzes));

// Update local storage

renderCourses(); // Re-render courses after deletion questionsDiv.innerHTML = "<p>Select a course to view

the quiz.</p>"; // Clear the questions section

}

};

window.deleteQuestion = (courseIndex, questionIndex) => { if (confirm("Are you sure you want to delete this

question?")) {

quizzes[courseIndex].questions.splice(questionIndex, 1); localStorage.setItem("quizzes", JSON.stringify(quizzes)); viewQuiz(courseIndex);

}

};

window.updateQuestion = (courseIndex, questionIndex) => { let newQuestion = prompt("Enter the updated question:",

quizzes[courseIndex].questions[questionIndex].question); if (newQuestion) {

quizzes[courseIndex].questions[questionIndex].question

= newQuestion;

for (let i = 0; i < quizzes[courseIndex].questions[questionIndex].options.length; i++) {

let newOption = prompt(`Update option ${i + 1}:`, quizzes[courseIndex].questions[questionIndex].options[i]);

if (newOption) {

quizzes[courseIndex].questions[questionIndex].options[i] = newOption;

}

}

localStorage.setItem("quizzes", JSON.stringify(quizzes)); viewQuiz(courseIndex);

}

};

document.querySelector(".applyButton").addEventListener("clic k", () => {

if (quizzes.length === 0) {

alert("No MCQs available to apply."); return;

}

localStorage.setItem("appliedQuizzes", JSON.stringify(quizzes));

alert("Quiz applied successfully! Students can now access

it.");

});

renderCourses();

});

// Logout function function adminLogout() {

localStorage.removeItem("adminLoggedIn"); alert("You have been logged out.");

window.location.href = "home.html"; // Redirect to login page

}

##### Teacher.CSS :

/\* General Styles \*/ body {

font-family: sans-serif; margin: 0;

padding: 0;

background-color: #f0f0f0; /\* Light gray background \*/ color: #333; /\* Dark gray text \*/

}

/\* Header Styles \*/ header {

background-color: #003366; /\* Navy blue header \*/ display: flex;

align-items: center; /\* Vertically center items \*/

justify-content: center; /\* Space between left, center, and right

\*/

padding: 1rem 2rem; /\* Padding for spacing \*/ position: fixed;

top: 0;

width: 100%;

box-sizing: border-box;

z-index: 1000; /\* Ensure header stays on top \*/

}

/\* Logo Container \*/ #logo-container {

display: flex; align-items: left;

}

#logo {

max-height: 60px; /\* Adjust as needed for proper sizing \*/ height: auto;

width: auto;

}

/\* Title Container \*/ #title-container {

flex: 1;

text-align: center;

}

#college-title {

font-size: 3.5rem; /\* Adjust for emphasis \*/ font-weight: bold;

color: #fff; /\* White text for visibility \*/ margin: 0;

}

/\* Right Placeholder for Balance \*/ #right-placeholder {

width: 60px; /\* Same width as the logo to balance the layout

\*/

}

/\* Navigation Styles \*/

nav {

background-color: #003366; /\* Navy blue navigation \*/ padding: 1rem 0;

margin-top: 80px; /\* Space below fixed header \*/

}

nav ul {

list-style: none; text-align: center; margin: 0;

padding: 0; display: flex;

justify-content: center;

}

nav li {

margin: 0 1.5rem;

}

nav a {

color: #fff;

text-decoration: none; font-weight: bold; font-size: 1rem;

}

nav a:hover {

text-decoration: underline;

}

main {

display: flex; flex: 1;

}

#course-list { width: 25%;

background-color: #f4f4f4; padding: 20px;

border-right: 1px solid #ccc; overflow-y: auto;

}

#quiz-sheet { flex: 1; padding: 20px;

overflow-y: auto; position: relative;

}

/\* Container for quiz questions \*/

.quiz-container { width: 80%;

max-width: 300px; margin: 0 auto; padding: 20px;

background-color: white; border-radius: 8px;

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

}

/\* Styling for each question \*/

.question {

background-color: #fafafa; padding: 15px;

margin-bottom: 10px; border: 1px solid #ddd; border-radius: 2px;

}

#rules {

background-color: rgba(173, 216, 230, 0.3); padding: 15px;

border: 1px solid #ADD8E6;

border-radius: 5px; margin-bottom: 20px;

}

.course-item { display: flex;

justify-content: space-between; align-items: center;

margin-bottom: 15px; padding: 10px;

border: 1px solid #ccc; border-radius: 5px; background-color: #fff;

}

.course-item button { background-color: #1712af; color: white;

border: none; padding: 5px 10px; cursor: pointer; border-radius: 5px;

}

.course-item button:hover { background-color: #150674;

}

/\* Delete Button Styling \*/

.delete-btn {

background-color: #ff4d4d; color: rgb(251, 248, 248); border: none;

padding: 10px 15px; border-radius: 5px; cursor: pointer;

transition: background-color 0.3s ease;

}

.delete-btn:hover { background-color: #e60000;

}

footer {

background-color: #950c0c; color: rgb(227, 219, 219); text-align: center;

padding: 10px 0;

}

.delete-question-btn { background-color: #ff4d4d; color: white;

border: none; padding: 5px 10px; border-radius: 5px; cursor: pointer;

transition: background-color 0.3s ease;

}

.delete-question-btn:hover { background-color: #e60000;

}

.question p {

font-weight: bold; margin-bottom: 9px;

}

/\* Styling for MCQ options \*/

.question label {

display: block; /\* Ensures each label appears on a new line \*/ margin: 7px 0; /\* Adds space between each option \*/

cursor: pointer; /\* Adds a pointer cursor for better UX \*/ font-size: 1rem; /\* Standard font size for options \*/

}

.question input[type="radio"] {

margin-right: 8px; /\* Space between radio button and text \*/ vertical-align: middle; /\* Aligns radio buttons with text \*/

}

.question label.correct { background-color: lightgreen;

}

.question label.wrong { background-color: lightcoral;

}

.applyButton {

background-color: #007bff; /\* Blue for Update \*/ color: white;

border: none; padding: 8px ; margin: 45px; cursor: pointer; border-radius: 15px; font-size: 14px; transition: 0.3s;

}

.cancel-btn {

background-color: #dc3545; /\* Red for Cancel \*/ color: white;

border: none; padding: 8px ; margin: 45px; cursor: pointer; border-radius: 15px; font-size: 14px; transition: 0.3s;

}

.applyButton:hover { background-color: #0056b3;

}

.cancel-btn:hover { background-color: #a71d2a;

}

/\* Footer Styles \*/ footer {

background-color: #003366; color: #fff;

text-align: center; padding: 0.2rem 0; position: fixed; bottom: 0;

width: 100%;

} .

#### Code Explainition :

##### Teacher.html (Structure & Layout) Header Section:

* + Displays the **East West Polytechnic** logo and title.
  + Uses a <header> with #logo-container and #college- title.

##### Navigation Bar:

* + Contains links to **Home, Login, Quiz, Teacher, Report, Student**, and a **Logout** button.
  + onclick="adminLogout()" function is used to log out.

##### Main Content:

* + **Course List (**#course-list**)**: Dynamically displays courses from localStorage.
  + **Quiz Rules (**#rules**)**: Instructions on how the quiz works.
  + **Quiz Display (**#questions**)**: Displays questions when a course is selected.
  + **Buttons (**Apply **and** Cancel**)**: To apply the quiz for students.

##### Footer:

* + Displays **copyright information**. **External Files:**
  + Links **Teacher.css** for styling.
  + Links **Teacher.js** for functionality.

##### Teacher.js (Functionality) Loading Courses & Quizzes

* localStorage.getItem("quizzes"): Retrieves stored quizzes.
* renderCourses(): Displays the list of stored courses dynamically.

##### Displaying Quizzes

* viewQuiz(index):
  + Retrieves selected course questions.
  + Dynamically generates **MCQs with radio buttons**.
  + Displays **Update** and **Delete** buttons for each question.

##### Deleting Data

* deleteCourse(index): Removes a course and updates localStorage.
* deleteQuestion(courseIndex, questionIndex): Deletes a specific question from a course.

##### Updating Questions

* updateQuestion(courseIndex, questionIndex):
  + Prompts the user to edit the question text and options.
  + Updates localStorage with new values.

##### Applying Quiz

* .applyButton:
  + Saves quizzes to localStorage.
  + Displays an **alert message** when successfully applied.

##### Admin Logout

* adminLogout(): Clears login data from localStorage and redirects to home.html.

##### Teacher.css (Styling) General Styles

* body: Light gray background with dark gray text.

##### Fixed Header:

* + **Navy blue** background.
  + Logo aligned to the left.
  + Title centered.

##### Navigation Bar

* **Background:** #003366 (Navy blue).
* **Hover Effect:** Underlines text when hovered.

##### Main Content

**Course List (**#course-list**)**:

* + **Left-side column** with a scrollable section.
  + Displays courses dynamically.

**Quiz Display (**#quiz-sheet**)**:

* + **Question Container:** White background, box- shadow.
  + **MCQ Options (**label**)**:
    - Correct answers highlighted in **green**.
    - Wrong answers in **red**.

##### Buttons

* **Apply & Cancel:**
  + Apply button → **Blue background**.
  + Cancel button → **Red background**.
  + Rounded corners with a hover effect.

##### Delete Button

* **Red Background** (#ff4d4d).
* **Hover Effect:** Darker red on hover.

# CHAPTER - 6

## Report Page

### Report Page :

#### Code has been Provided Report.html, Report.CSS, page :

##### Report.html :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Student Report - East West Polytechnic</title>

<script src="https://cdn.sheetjs.com/xlsx- latest/package/dist/xlsx.full.min.js"></script>

<link rel="stylesheet" href="Report.css">

</head>

<body>

<header>

<div id="logo-container">

<img id="logo" src="ewp.jpg" alt="East West Polytechnic Logo">

</div>

<h1 id="college-title">EAST-WEST POLYTECHNIC</h1>

</header>

<nav>

<ul>

<li><a href="home.html">Home</a></li>

<li><a href="login.html">Login</a></li>

<li><a href="Quiz.html">Quiz</a></li>

<li><a href="Teacher.html">Teacher</a></li>

<li><a href="#" onclick="showTeacherLogin()">Report</a></li>

<li><a href="Student.html">Student</a></li>

<li><a href="#" onclick="adminLogout()">Logout</a></li>

</ul>

</nav>

<main id="login-section">

<div class="login-container">

<h2>Teacher Login</h2>

<label for="teacherName">Teacher Name:</label>

<input type="text" id="teacherName" placeholder="Enter your name" required>

<label for="subject">Select Subject:</label>

<select id="subject" required>

<option value="">-- Select Subject --</option>

<option value="FSD">FSD</option>

<option value="ES">ES</option>

<option value="IC">IC</option>

<option value="Kannada">Kannada</option>

</select>

<button onclick="showReport()">Submit</button>

</div>

</main>

<main id="report-section" style="display:none;">

<div class="report-container">

<table>

<thead>

<tr>

<th>Sl. No</th>

<th>Name</th>

<th>Registration No.</th>

<th>Subject</th>

<th>Marks</th>

<th>Percentage</th>

<th>Action</th>

</tr>

</thead>

<tbody id="reportTableBody"></tbody>

</table>

<button class="button" onclick="exportToExcel()">Preview (Edit in Excel)</button>

<input type="file" id="fileInput" style="display:none;" onchange="importFromExcel()">

<button class="button" onclick="document.getElementById('fileInput').click()">Upload Updated Excel</button>

</div>

</main>

<footer>

<p>&copy; 2024 East West Polytechnic. All rights reserved.</p>

</footer>

<script>

function showTeacherLogin() { document.getElementById("login-section").style.display

= "block";

document.getElementById("report- section").style.display = "none";

}

function showReport() { let selectedSubject =

document.getElementById("subject").value; if (!selectedSubject) {

alert("Please select a subject."); return;

}

let studentResults = JSON.parse(localStorage.getItem("studentResults")) || [];

let filteredResults = studentResults.filter(student => student.subject === selectedSubject);

let tableBody = document.getElementById("reportTableBody");

tableBody.innerHTML = filteredResults.map((student, index) => `

<tr>

<td>${index + 1}</td>

<td>${student.name}</td>

<td>${student.regNo}</td>

<td>${student.subject}</td>

<td>${student.score}/${student.totalQuestions}</td>

<td>${student.percentage}%</td>

<td>

<button class="delete-btn" onclick="deleteRecord(${index})">Delete</button>

</td>

</tr>

`).join("");

document.getElementById("login-section").style.display

= "none";

document.getElementById("report- section").style.display = "block";

}

function deleteRecord(index) { let studentResults =

JSON.parse(localStorage.getItem("studentResults")) || []; if (confirm("Are you sure you want to delete this

record?")) {

studentResults.splice(index, 1); localStorage.setItem("studentResults",

JSON.stringify(studentResults));

showReport();

}

}

function exportToExcel() { let studentResults =

JSON.parse(localStorage.getItem("studentResults")) || []; if (studentResults.length === 0) {

alert("No records to export."); return;

}

let ws = XLSX.utils.json\_to\_sheet(studentResults); let wb = XLSX.utils.book\_new(); XLSX.utils.book\_append\_sheet(wb, ws, "Student

Report");

XLSX.writeFile(wb, "Student\_Report.xlsx");

}

function importFromExcel() {

let file = document.getElementById("fileInput").files[0]; if (!file) {

alert("Please select an Excel file."); return;

}

let reader = new FileReader(); reader.readAsArrayBuffer(file); reader.onload = function (e) {

let data = new Uint8Array(e.target.result);

let workbook = XLSX.read(data, { type: 'array' }); let sheetName = workbook.SheetNames[0];

let sheet = workbook.Sheets[sheetName];

let studentResults = XLSX.utils.sheet\_to\_json(sheet); localStorage.setItem("studentResults",

JSON.stringify(studentResults));

showReport();

alert("Student records updated successfully!");

};

}

function adminLogout() { localStorage.removeItem("adminLoggedIn");

alert("You have been logged out."); window.location.href = "home.html";

}

</script>

</body>

</html>

##### Report.CSS :

/\* General Styles \*/ body {

font-family: sans-serif; margin: 0;

padding: 0;

background-color: #f0f0f0; /\* Light gray background \*/ color: #333; /\* Dark gray text \*/

}

/\* Header Styles \*/ header {

background-color: #003366; /\* Navy blue header \*/ display: flex;

align-items: center; /\* Vertically center items \*/

justify-content: center; /\* Space between left, center, and right

\*/

padding: 1rem 2rem; /\* Padding for spacing \*/ position: fixed;

top: 0;

width: 100%;

box-sizing: border-box;

z-index: 1000; /\* Ensure header stays on top \*/

}

/\* Logo Container \*/ #logo-container {

display: flex; align-items: left;

}

#logo {

max-height: 60px; /\* Adjust as needed for proper sizing \*/ height: auto;

width: auto;

}

/\* Title Container \*/ #title-container {

flex: 1;

text-align: center;

}

#college-title {

font-size: 3.5rem; /\* Adjust for emphasis \*/ font-weight: bold;

color: #fff; /\* White text for visibility \*/ margin: 0;

}

/\* Right Placeholder for Balance \*/ #right-placeholder {

width: 60px; /\* Same width as the logo to balance the layout

\*/

}

/\* Navigation Styles \*/ nav {

background-color: #003366; /\* Navy blue navigation \*/ padding: 1rem 0;

margin-top: 80px; /\* Space below fixed header \*/

}

nav ul {

list-style: none; text-align: center; margin: 0;

padding: 0; display: flex;

justify-content: center;

}

nav li {

margin: 0 1.5rem;

}

nav a {

color: #fff;

text-decoration: none; font-weight: bold; font-size: 1rem;

}

nav a:hover {

text-decoration: underline;

}

/\* Report Container \*/

.report-container { width: 97%; padding: 20px;

background-color: white; border-radius: 10px;

box-shadow: 0px 10px 20px rgba(0, 0, 0, 0.2); text-align: center;

margin: 10px auto 60px;

max-height: 100vh; /\* Ensures proper display \*/

overflow: hidden; /\* Prevents unwanted scrolling outside \*/

}

/\* Scrollable Table Container \*/

.table-container {

max-height: 100vh; /\* Ensures it fits well \*/ overflow-y: auto; /\* Enables scrolling \*/ border-radius: 10px;

border: 1px solid #ddd;

}

/\* Table Styles \*/ table {

width: 100%;

border-collapse: 36;

}

/\* Fixed Table Header \*/ thead {

position: sticky; top: 0;

background-color: #222093; color: white;

z-index: 10;

}

/\* Table Row Styling \*/ th, td {

padding: 10px;

border: 1px solid #ddd; text-align: center; white-space: nowrap;

}

th {

background-color: #222093; color: white;

}

/\* No Data Message \*/

.no-data { color: gray;

font-size: 18px; margin-top: 20px;

}

/\* Delete Button \*/

.delete-btn {

background-color: red; color: white;

border: none; padding: 5px 10px; cursor: pointer; border-radius: 5px;

}

.delete-btn:hover { background-color: darkred;

}

/\* General Button Styles \*/

.button {

margin-top: 10px; padding: 10px 15px; font-size: 16px;

background-color: rgb(56, 19, 148); color: white;

border: none; border-radius: 5px; cursor: pointer;

}

.button:hover {

background-color: darkgray;

}

/\* Teacher Login Container \*/

.login-container { width: 100%;

max-width: 400px; background: #fff; padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); text-align: center;

margin: 108px auto; /\* Push it down to avoid overlap with fixed header \*/

display: flex;

flex-direction: column; align-items: center;

}

/\* Teacher Login Inputs \*/

.login-container label { display: block;

text-align: left; margin-bottom: 6px; font-weight: bold; width: 100%;

}

.login-container input,

.login-container select { width: 100%; padding: 10px; margin-bottom: 15px;

border: 1px solid #ccc; border-radius: 5px; font-size: 16px;

}

/\* Login Button \*/

.login-container button { width: 100%; padding: 10px; background: #0a3a6d; color: white;

border: none; border-radius: 5px;

font-size: 16px; cursor: pointer;

transition: background 0.3s ease;

}

.login-container button:hover { background: #0e3d70;

}

/\* Footer Styles \*/ footer {

background-color: #003366; color: #fff;

text-align: center; padding: 0.8rem 0; position: fixed; bottom: 0;

width: 100%;

} .

#### Code Explainition :

##### Report.html - Structure & Functionality Header Section

* + Displays the **East West Polytechnic** logo and title.
  + The header is **fixed** at the top.

##### Navigation Bar

* + Contains links to different pages: Home, Login, Quiz, Teacher, Report, Student, and Logout.
  + Report and Logout use JavaScript functions:
    - showTeacherLogin() → Displays the teacher login form.
    - adminLogout() → Logs out the admin and redirects to home.html.

##### Teacher Login Section

* + **Fields:**
    - Teacher Name (Text input)
    - Select Subject (Dropdown: FSD, ES, IC, Kannada)
  + **Submit Button:** Calls showReport() to display the student quiz report.

##### Report Section

* + Initially **hidden** (display: none).
  + Displays student quiz results **filtered by subject**.
  + Contains:
    - **Table** listing student details: Sl. No, Name, Reg No, Subject, Marks, Percentage, and Delete Button.

##### Buttons:

* + - * Preview (Edit in Excel): Exports the report to an **Excel file**.
      * Upload Updated Excel: Allows importing an updated **Excel file**.

showTeacherLogin()

1. Shows the **teacher login form**.
2. Hides the **report section**. showReport()
3. Fetches student records from localStorage.
4. Filters the records based on the **selected subject**.
5. Populates the table with the filtered data.
6. Displays the **Report Section** and hides the **Login Section**.

deleteRecord(index)

1. Deletes a student record from localStorage.
2. Updates the displayed report. exportToExcel()
3. Converts student data from localStorage to an **Excel file**.
4. Downloads it as Student\_Report.xlsx. importFromExcel()
5. Reads an **uploaded Excel file**.
6. Updates localStorage with new student records.
7. Refreshes the displayed report. adminLogout()
8. Clears **admin login data**.
9. Redirects to home.html.

##### Report.css - Styling

**Body & Layout**

* + Uses **sans-serif font**.
  + Background color: #f0f0f0 (light gray).
  + Text color: #333 (dark gray).

##### Header & Navigation Header:

* + - background-color: #003366 (navy blue).
    - display: flex for logo and title alignment.
    - position: fixed to remain at the top.

##### Navigation Bar:

* + - Navy blue background.
    - Links (<a>) are **white** and bold.
    - Uses flexbox for proper spacing.

##### Report Table

**Table Container:**

* + - max-height: 100vh → Ensures it fits well.
    - overflow-y: auto → Enables scrolling.
    - border-radius: 10px → Rounded corners.

**Table Header (**thead**)**

* + - **Fixed at the top** (position: sticky).
    - background-color: #222093 (dark blue).
    - **White text**. **Table Cells (**th, td**)**

##### Centered text.

* + - Borders: 1px solid #ddd (light gray).
    - white-space: nowrap to prevent text wrapping.

##### Buttons

**General Buttons:**

* + - background-color: rgb(56, 19, 148); (dark blue).
    - color: white;
    - Rounded corners.
    - **Hover effect** → Changes to darkgray.

##### Delete Button:

* + - **Red background** (background-color: red).
    - Turns **dark red** on hover.

##### Teacher Login Form Centered Container

* + - White background.
    - Rounded corners.
    - **Box-shadow** for a professional look.
    - max-width: 400px (limits form width).

##### Input Fields

* + - **Full width**, **bordered**, and **rounded**.
    - Proper spacing.

**Footer**

* + **Fixed at the bottom**.
  + **Navy blue background**.
  + **Centered text**.

# CHAPTER - 7

## Student Page

### Student Page :

#### Code has been Provided Student.html, Test.html, Result.html page :

##### Student.html :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Student Login - East West Polytechnic</title>

<link rel="stylesheet" href="Student.css">

</head>

<body>

<header>

<div id="logo-container">

<img id="logo" src="ewp.jpg" alt="East West Polytechnic Logo">

</div>

<h1 id="college-title">EAST-WEST POLYTECHNIC</h1>

</header>

<nav>

<ul>

<li><a href="home.html">Home</a></li>

<li><a href="login.html">Login</a></li>

<li><a href="#" id="quiz">Quiz</a></li>

<li><a href="#" id="teacher">Teacher</a></li>

<li><a href="#" id="report">Report</a></li>

<li><a href="Student.html">Student</a></li>

</ul>

</nav>

<main>

<div class="login-container">

<h2>Student Login</h2>

<form id="studentForm">

<label for="name">Name</label>

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

<label for="regNo">Registration Number</label>

<input type="text" id="regNo" required>

<label for="semester">Semester</label>

<select id="semester" required>

<option value="" disabled selected>Select your semester</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

</select>

<label for="branch">Branch</label>

<select id="branch" required>

<option value="" disabled selected>Select your branch</option>

<option value="CSE">Computer Science</option>

<option value="ECE">Electronics</option>

<option value="EEE">Electrical</option>

<option value="MECH">Mechanical</option>

<option value="CIVIL">Civil</option>

</select>

<label for="subject">Subject</label>

<select id="subject" required>

<option value="" disabled selected>Select your subject</option>

<option value="FSD">Full Stack Development</option>

<option value="ES">Embedded Systems</option>

<option value="IC">Instrumentation</option>

<option value="Kannada">Kannada</option>

</select>

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

</form>

</div>

</main>

<footer>

<p>&copy; 2024 East West Polytechnic. All rights reserved.</p>

</footer>

<script>

document.getElementById("studentForm").addEventListener("s ubmit", function(event) {

event.preventDefault();

let name = document.getElementById("name").value; let regNo = document.getElementById("regNo").value;

let semester = document.getElementById("semester").value;

let branch = document.getElementById("branch").value; let subject = document.getElementById("subject").value;

if (!name || !regNo || !semester || !branch || !subject) { alert("Please fill all fields.");

return;

}

// Store student details in localStorage localStorage.setItem("studentName", name); localStorage.setItem("studentRegNo", regNo); localStorage.setItem("studentSemester", semester);

localStorage.setItem("studentBranch", branch); localStorage.setItem("studentSubject", subject);

page

alert("Login successful! Redirecting to test page..."); window.location.href = "test.html"; // Redirect to test

});

</script>

</body>

</html>

##### Test.html :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Online Test</title>

<link rel="stylesheet" href="test.css">

</head>

<body>

<div class="test-container">

<h1>Online Test</h1>

<h2 id="student-info"></h2>

<div id="questions-container"></div>

<button id="submitTest">Submit</button>

</div>

<script src="test.js"></script>

</body>

</html>

##### Result.html :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Quiz Result</title>

<link rel="stylesheet" href="result.css">

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<style>

body {

font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0;

height: 100vh; display: flex;

align-items: center; justify-content: center;

}

.result-container { width: 50%; padding: 30px;

background-color: white; border-radius: 10px;

box-shadow: 0px 10px 20px rgba(0, 0, 0, 0.2); text-align: center;

position: absolute; top: 33%;

left: 50%;

transform: translate(-50%, -50%);

}

.circle-box { position: relative; width: 120px; height: 120px; margin: 20px auto;

}

.circle-box canvas { position: absolute; top: 0;

left: 0;

}

.circle-text { position: absolute; top: 50%;

left: 50%;

transform: translate(-50%, -50%); font-size: 20px;

font-weight: bold; color: black;

}

.score-labels { margin-top: 10px; font-size: 16px; font-weight: bold; text-align: center;

}

.back-button { margin-top: 20px;

padding: 10px 20px; font-size: 16px;

background-color: #111895; /\* Green \*/ color: white;

border: none; border-radius: 5px; cursor: pointer;

}

.back-button:hover { background-color: #222093;

}

</style>

</head>

<body>

<div class="result-container">

<h1>Quiz Result</h1>

<div class="info-container">

<p><strong>Name:</strong> <span id="studentName"></span></p>

<p><strong>Registration No:</strong> <span id="studentRegNo"></span></p>

<p><strong>Subject:</strong> <span id="studentSubject"></span></p>

</div>

<!-- Circular Score Display -->

<div class="circle-box">

<canvas id="scoreChart" width="120" height="120"></canvas>

<div class="circle-text" id="resultText"></div>

</div>

<div class="score-labels">

<p><strong>Marks:</strong> <span id="studentMarks"></span></p>

<p><strong>Percentage:</strong> <span id="studentPercentage"></span>%</p>

</div>

<button class="back-button"

onclick="window.location.href='Student.html'">Back to Home</button>

</div>

<script>

document.addEventListener("DOMContentLoaded", () =>

{

let studentName = localStorage.getItem("studentName"); let studentRegNo =

localStorage.getItem("studentRegNo");

let studentSubject = localStorage.getItem("studentSubject");

let studentScore = parseInt(localStorage.getItem("studentScore"));

let totalQuestions = parseInt(localStorage.getItem("totalQuestions"));

document.getElementById("studentName").textContent

= studentName;

document.getElementById("studentRegNo").textContent

= studentRegNo;

document.getElementById("studentSubject").textContent = studentSubject;

let scorePercentage = ((studentScore / totalQuestions) \* 100).toFixed(1);

document.getElementById("studentMarks").textContent

= `${studentScore}/${totalQuestions}`;

document.getElementById("studentPercentage").textContent = scorePercentage;

// Fetch previous results or create a new array

let studentResults = JSON.parse(localStorage.getItem("studentResults")) || [];

// Add new student result to array studentResults.push({

name: studentName, regNo: studentRegNo, subject: studentSubject, score: studentScore,

totalQuestions: totalQuestions, percentage: scorePercentage

});

// Save updated results to localStorage localStorage.setItem("studentResults",

JSON.stringify(studentResults));

createCircularChart("scoreChart", studentScore, totalQuestions, scorePercentage);

});

function createCircularChart(canvasId, score, total, percentage) {

let ctx = document.getElementById(canvasId).getContext("2d");

new Chart(ctx, { type: "doughnut", data: {

datasets: [{

data: [score, total - score],

backgroundColor: ["#28a745", "#E0E0E0"], /\* Green Circle \*/

borderWidth: 0

}]

},

options: { responsive: false,

maintainAspectRatio: false, cutout: "75%",

plugins: {

tooltip: { enabled: false }, legend: { display: false }

}}

});

document.getElementById("resultText").innerHTML =

`<b>${percentage}%</b>`; // Display only percentage inside the circle

}

</script>

</body>

</html> .

#### Code Explainition with applied Flow chart :

##### Student.html (Student Login Page) Features:

* + Displays a **header** with a logo and college name.
  + A **navigation bar** with links (Home, Login, Quiz, Teacher, Report, Student).
  + A **form** for students to enter:
    - Name
    - Registration Number
    - Semester (Dropdown)
    - Branch (Dropdown)
    - Subject (Dropdown)

##### JavaScript functionality:

* + - Prevents form submission if fields are empty.
    - Stores form data in localStorage.
    - Redirects to test.html.

##### Test.html (Test Page) Features:

* + Displays student details fetched from localStorage.
  + Displays quiz questions dynamically inside #questions- container.
  + Includes a **Submit** button.

##### Key JavaScript Code (in test.js - Not provided but should be implemented):

* + Fetches student details from localStorage and displays them.
  + Loads quiz questions dynamically.
  + Submits the quiz and calculates the score.
  + Stores the score in localStorage and redirects to result.html.

##### Result.html (Result Page) Features:

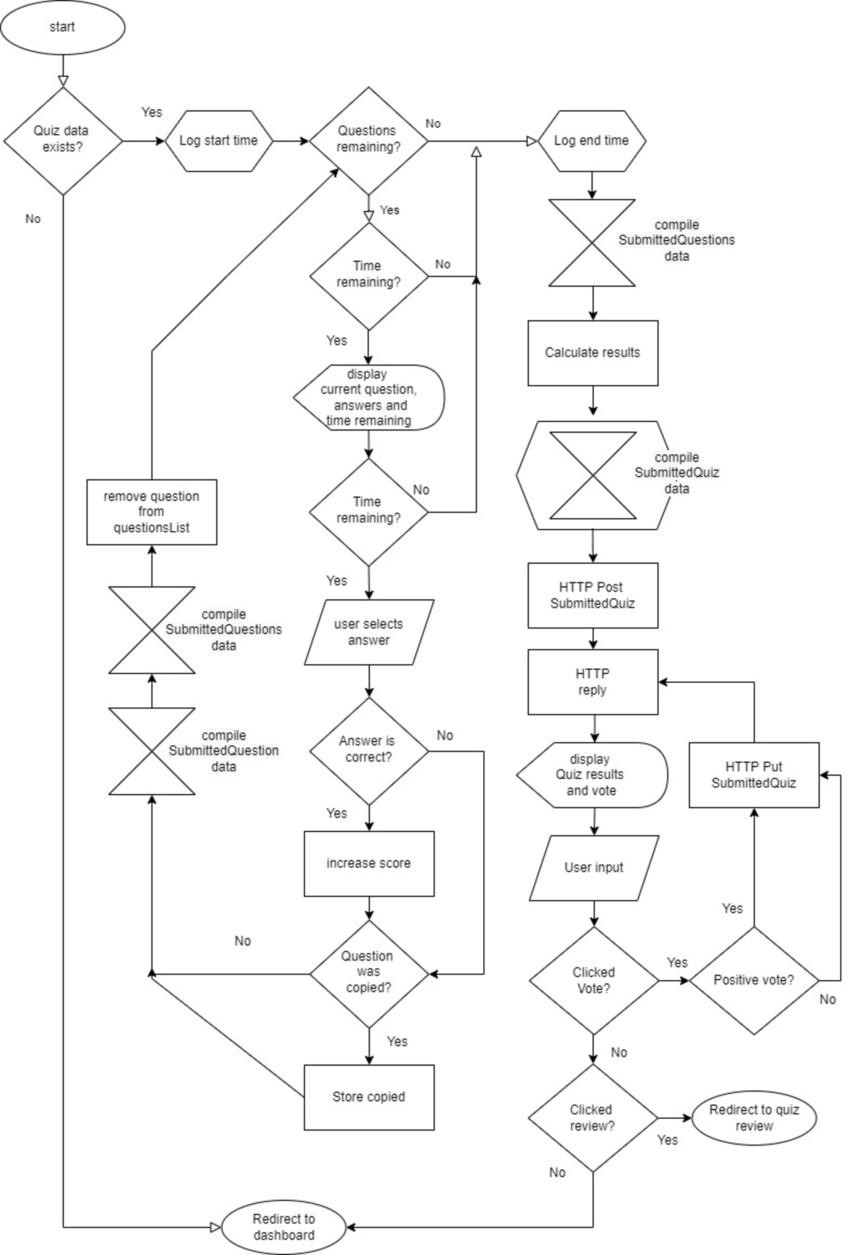
* + Displays student details.
  + Shows quiz results with:

##### Marks

* + - **Percentage**
  + Uses **Chart.js** to create a **circular progress chart** for score visualization.
  + Stores student results in localStorage for future retrieval.
  + Provides a **Back to Home** button.

##### Flow Chart :

* The quiz starts only if quiz data exists; otherwise, the process stops.
* The system logs the start time and checks if there are remaining questions.
* If there is time left, the current question, answers, and remaining time are displayed.
* The user selects an answer, which is then evaluated for correctness.
* If the answer is correct, the score is increased; otherwise, the process continues.
* Each answered question is removed from the question list and stored in submitted data.
* If all questions are answered or time runs out, the end time is logged.
* The system compiles submitted quiz data and calculates results.
* The quiz results are displayed, and the user can vote or review the quiz.
* After voting or reviewing, the user is redirected to the dashboard.



# CHAPTER - 8 CONCLUSION

#### Conclusion :

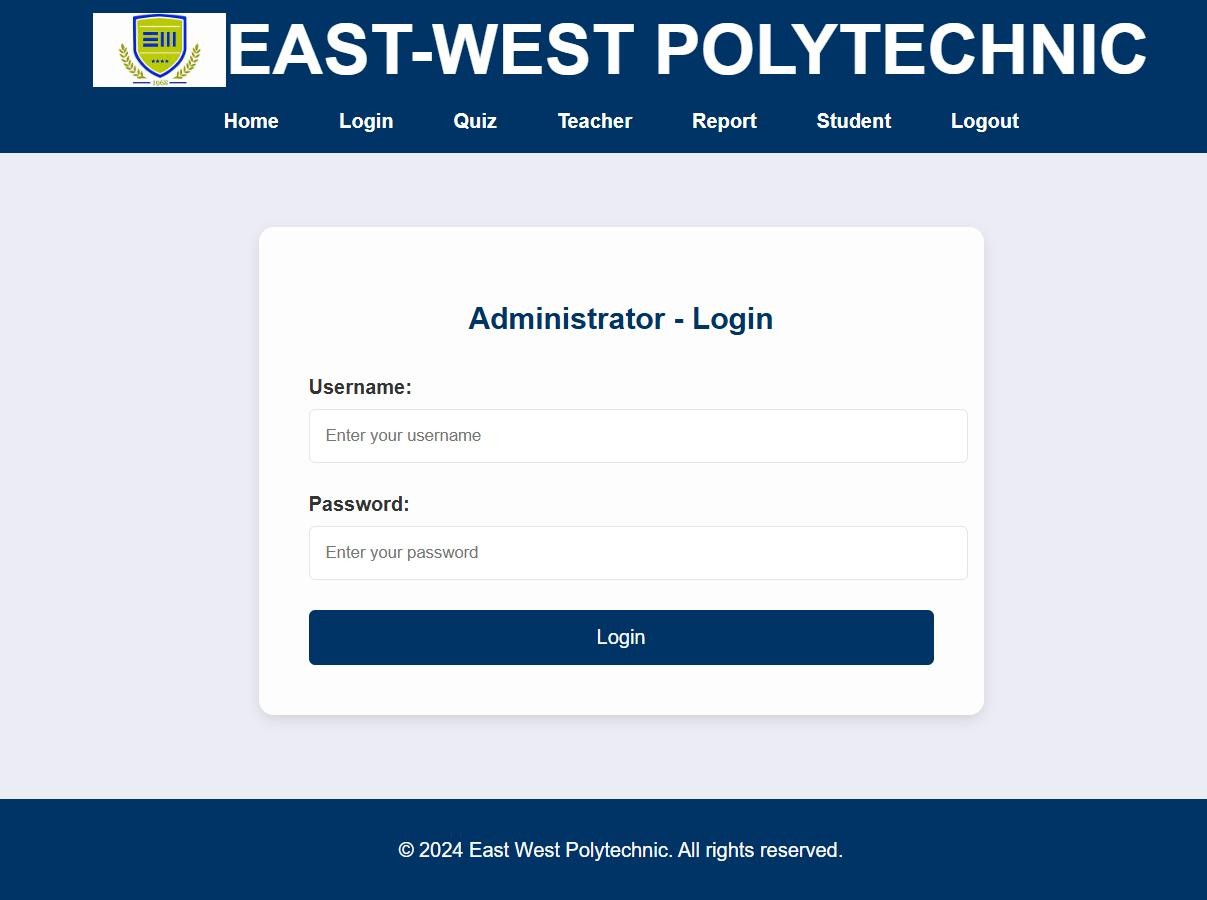
The **Online Quiz Application** provides an efficient, user- friendly, and interactive platform for conducting quizzes. With seamless student login, quiz selection, automated result calculation, and PDF generation, the application enhances the learning and evaluation process. Key features such as real-time question updates, quiz result previews, and a structured report page ensure a smooth experience for both students and teachers.

#### Visual Representation Of Online Quiz Application :

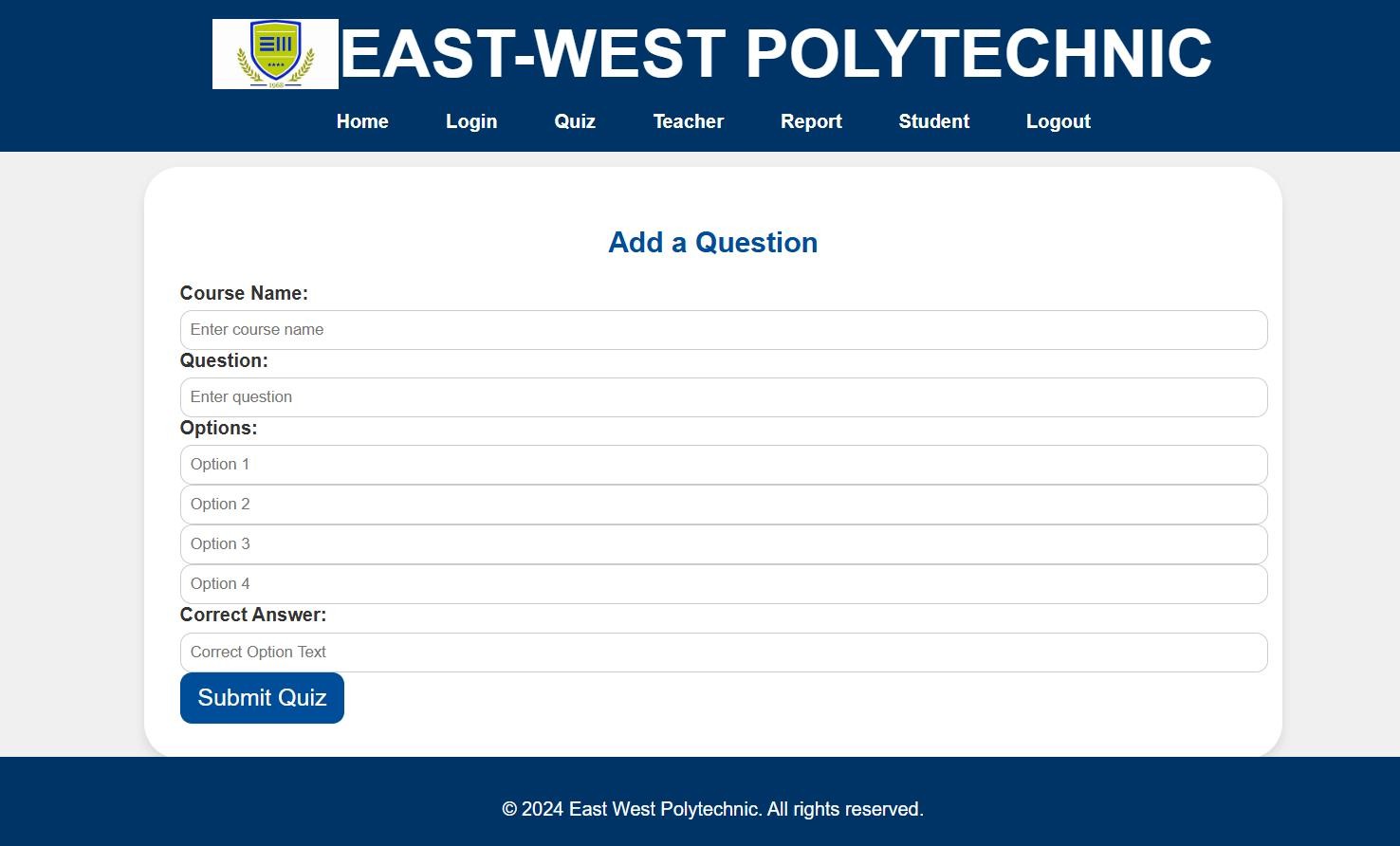
**Home Page :**

****

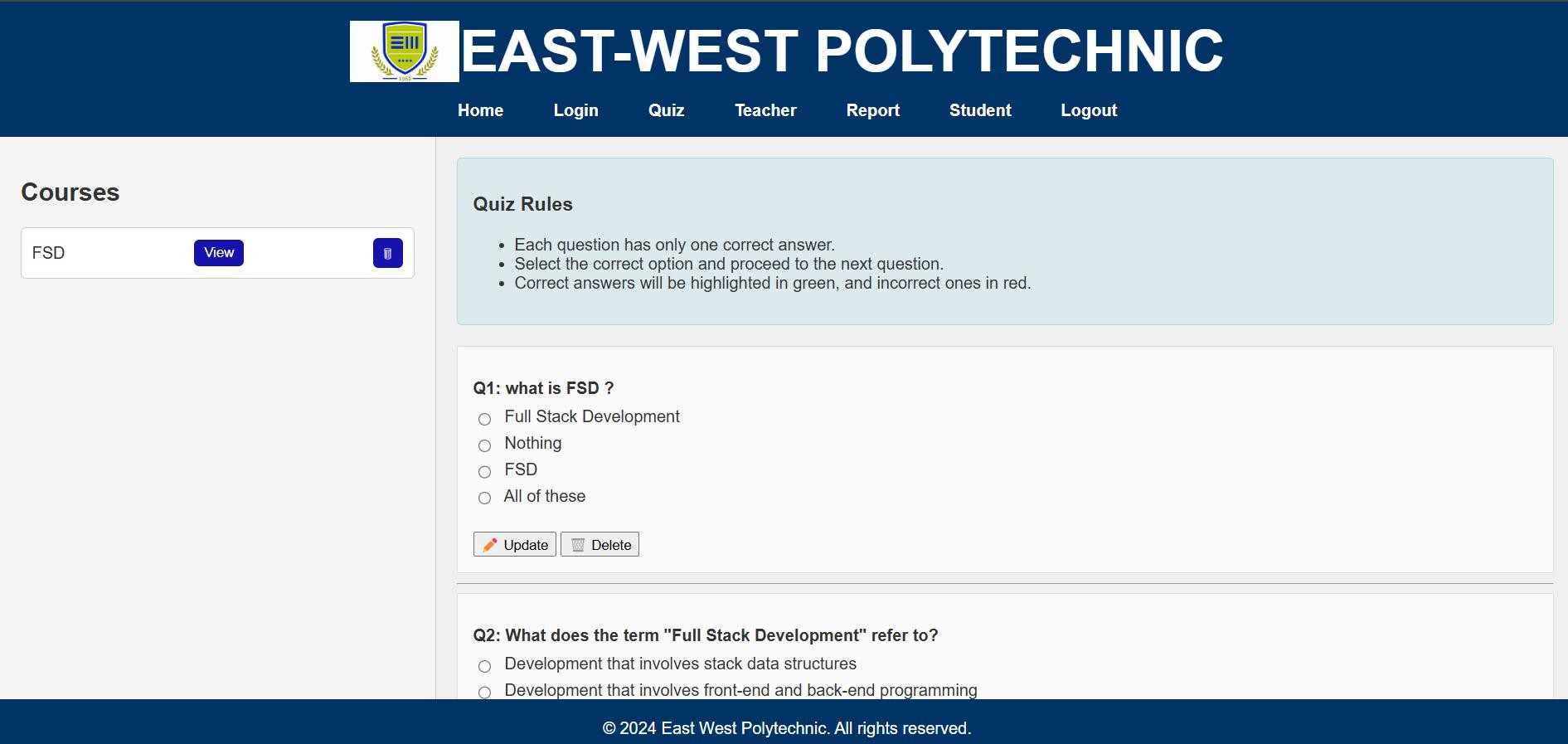
**Login Page :**

****

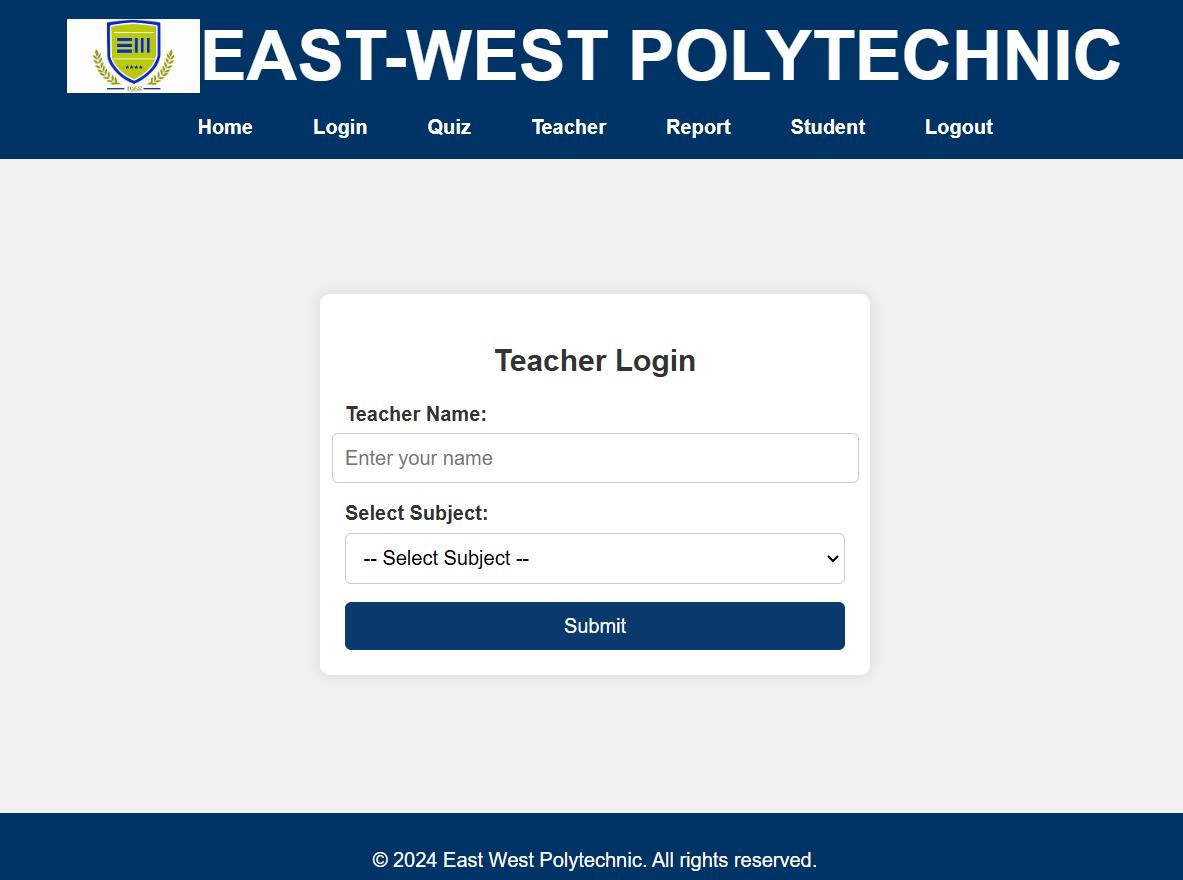
**Quiz Page :**

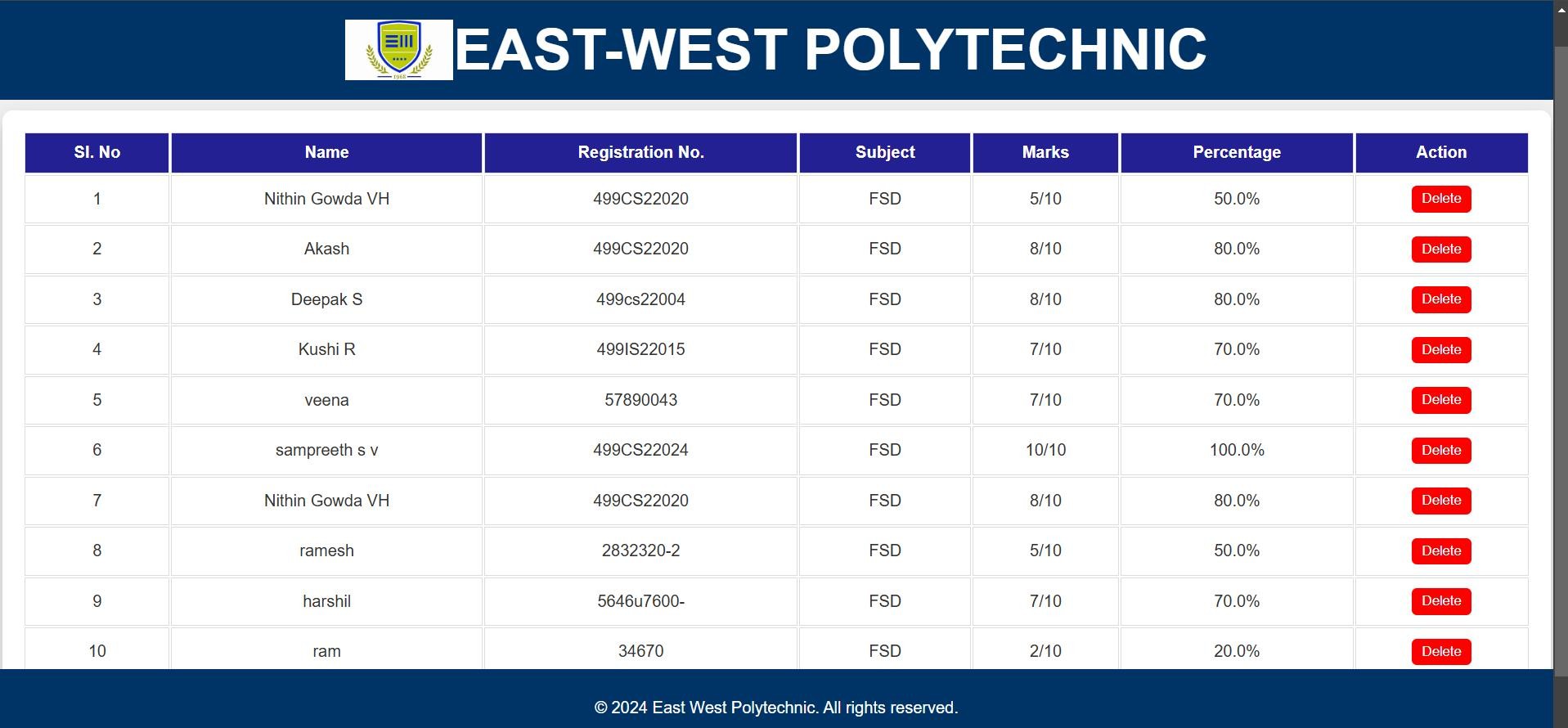
****

**Teacher Page :**

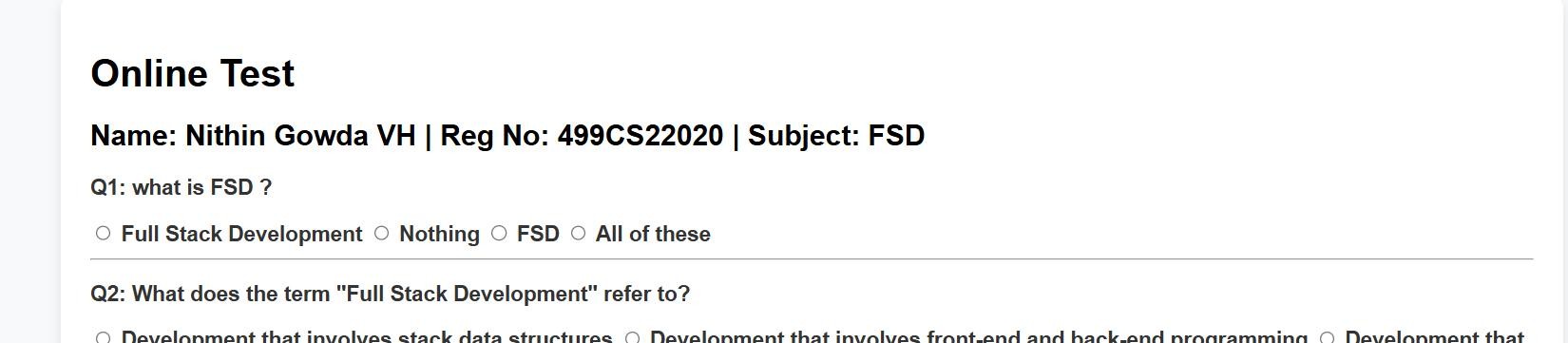
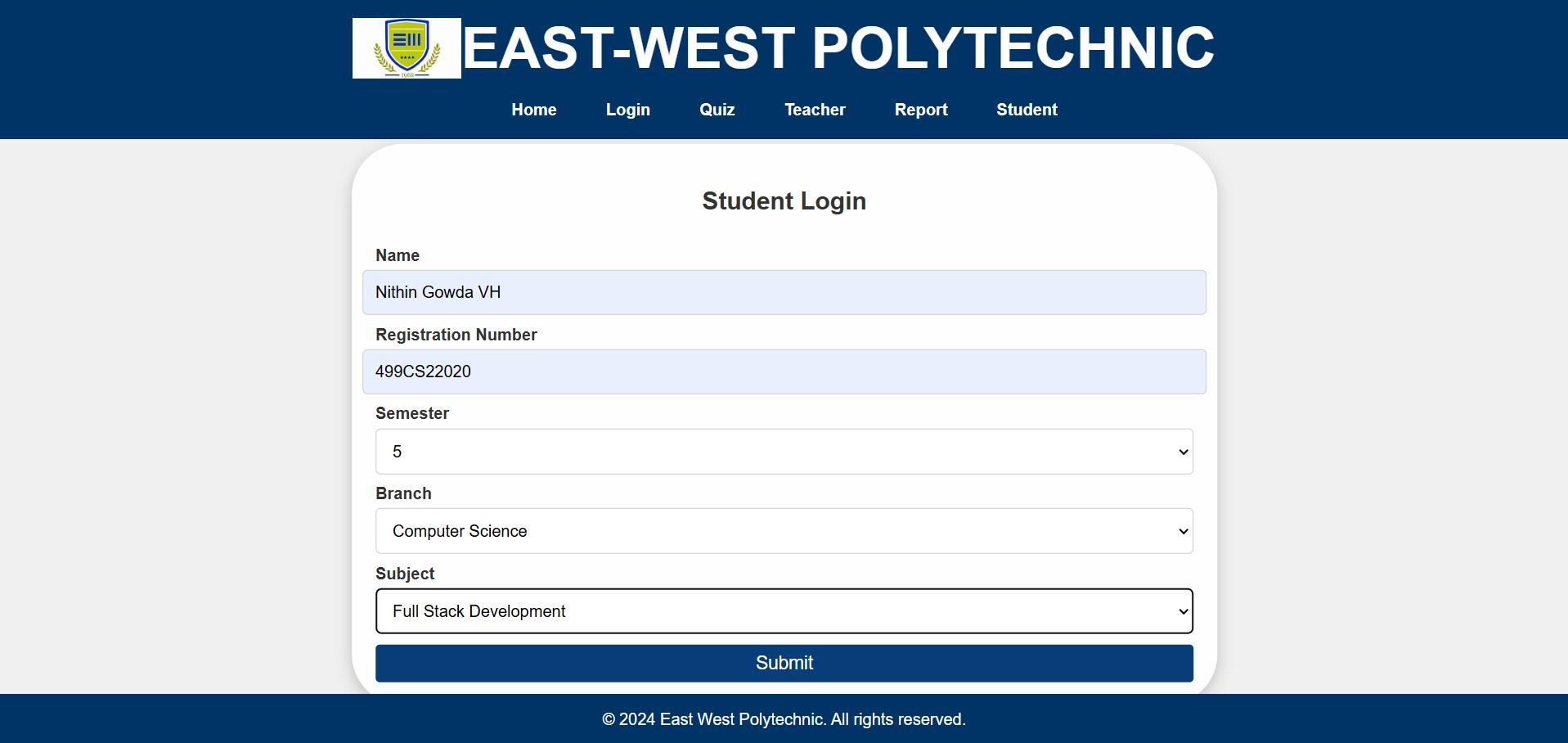
****

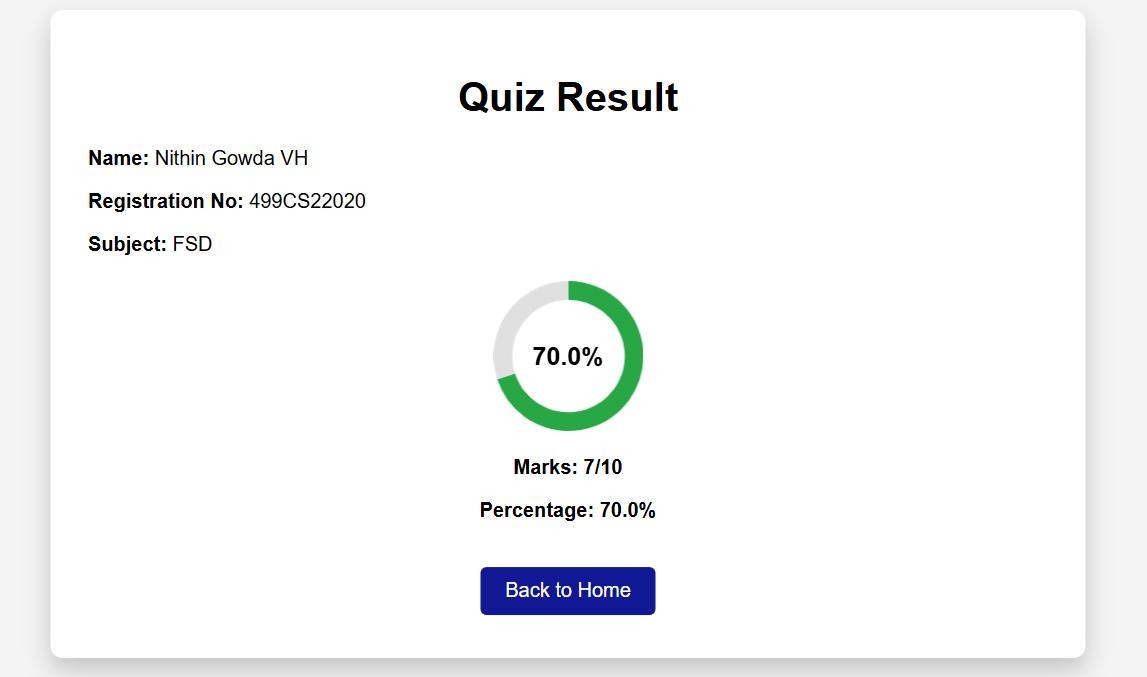
**Report Page :**

****

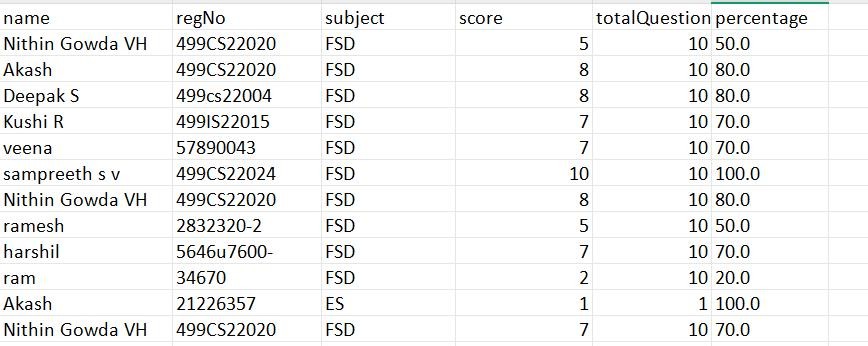


**Student Page :**

****



**Database in Excel :**

****

Additionally, the integration of features like voting, quiz reviews, and question filtering makes the application dynamic and engaging. The use of proper alignment, professional design, and an intuitive navigation system ensures a clean and structured layout. Overall, this application simplifies online assessments and serves as a valuable tool for academic evaluation and knowledge enhancement.