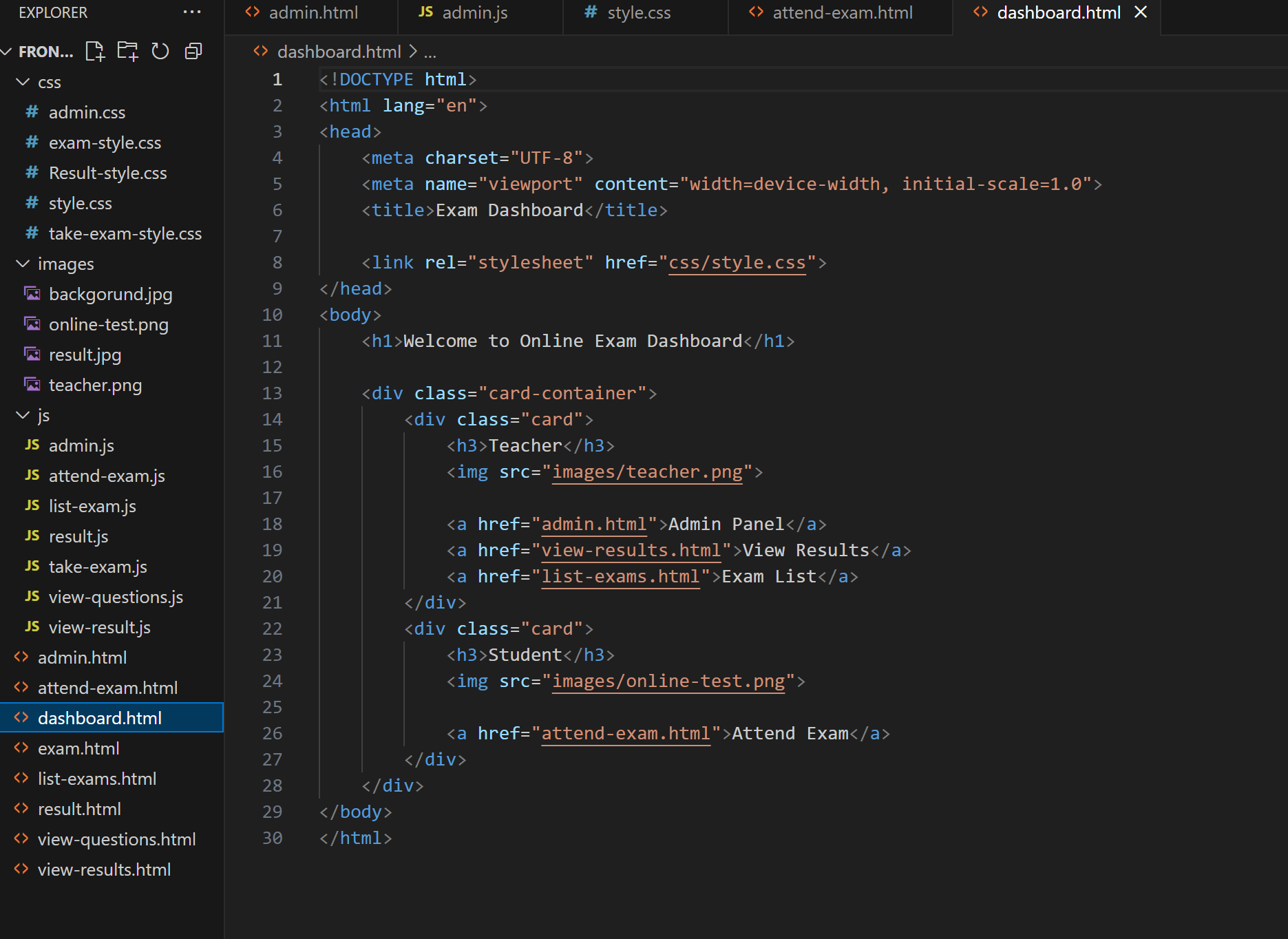
**ONLINE EXAM PORTAL – FRONT END**



**HTML**

dashboard.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Exam Dashboard</title>

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

</head>

<body>

    <h1>Welcome to Online Exam Dashboard</h1>

    <div class="card-container">

        <div class="card">

            <h3>Teacher</h3>

            <img src="images/teacher.png">

            <a href="admin.html">Admin Panel</a>

            <a href="view-results.html">View Results</a>

            <a href="list-exams.html">Exam List</a>

        </div>

        <div class="card">

            <h3>Student</h3>

            <img src="images/online-test.png">

            <a href="attend-exam.html">Attend Exam</a>

        </div>

    </div>

</body>

</html>

admin.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Online Exam Admin Panel</title>

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

</head>

<body>

    <h2>Admin Panel - Online Exam Portal</h2>

    <form id="createExamForm">

        <h3>Create New Exam</h3>

        <label for="title">Exam Title</label>

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

        <button type="submit">Add Exam</button>

    </form>

    <form id="addQuestionForm">

        <h3>Add Questions to Exam</h3>

        <label for="examSelect">Select Exam:</label>

        <select id="examSelect" required>

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

        </select>

        <h4>Questions</h4>

        <div id="questionsContainer"></div>

        <button type="button" id="addQuestionBtn">Add Question</button>

        <button type="submit">Create Questions</button>

    </form>

    <div style="text-align: center;">

        <a href="dashboard.html" class="exam-button">Back to Home</a>

        <a href="view-questions.html"class="exam-button">View Questions</a>

    </div>

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

</body>

</html>

attend-exam.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Online Exam</title>

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

</head>

<body>

    <h1>Choose an Exam</h1>

    <div class="exam-container" id="examList">

    </div>

    <br>

    <center>

        <div>

            <a href="dashboard.html" class="exam-button">Back to Home</a>

        </div>

    </center>

    <script src="js/attend-exam.js"></script>

</body>

</html>

exam.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Take Exam</title>

    <link rel="stylesheet" href="css/take-exam-style.css">

</head>

<body>

    <h2 style="text-align: center;">Answer Questions</h2>

    <form id="examForm">

    </form>

    <div id="questions"></div>

    <script src="js/take-exam.js"></script>

</body>

</html>

list-exams.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Admin Dashboard - Online Exam</title>

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

</head>

<body>

    <h1>Exam List</h1>

    <div class="card">

        <table border="1">

            <tr>

                <th>Exam ID</th>

                <th>Title</th>

            </tr>

            <tbody id="examTable"></tbody>

        </table>

        <a href="dashboard.html">Back to Home</a>

    </div>

    <script src="js/list-exam.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>Result</title>

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

</head>

<body>

    <div class="card">

        <h1>Exam Result</h1>

        <img src="images/result.jpg">

        <p id="studentName"></p>

        <p id="score"></p>

        <p id="percentage"></p>

        <a href="dashboard.html">Back to Home</a>

    </div>

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

</body>

</html>

view-questions.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>View Questions - Online Exam Portal</title>

    <style>

        table{

            table-layout: fixed;

        }

        th,td{

            width: 90px;

            height:50px;

            text-align: center;

            overflow: hidden;

            text-overflow: ellipsis;

        }

        button.delete-btn:hover{

            background-color: red;

        }

    </style>

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

</head>

<body>

    <h2>View Questions - Online Exam Portal</h2>

    <form id="examForm">

        <label for="examSelect">

            <strong>Select Exam:</strong>

        </label>

        <select id="examSelect" required>

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

        </select>

    </form>

    <div id="questionsSection" style="max-width: 900px; margin:0 auto;">

        <table id="questionTable" style="display: none;">

            <thead>

                <tr>

                    <th>Q.No</th>

                    <th>Question Text</th>

                    <th>Option A</th>

                    <th>Option B</th>

                    <th>Option C</th>

                    <th>Option D</th>

                    <th>Correct Option</th>

                    <th>Action</th>

                </tr>

            </thead>

            <tbody id="questionsBody"></tbody>

        </table>

        <div id="noQuestionMsg" style="display: none;">

            No Questions Found for this Exam.

        </div>

        <div>

            <center>

                <a href="dashboard.html" class="exam-button">Back to Home</a>

            </center>

        </div>

    </div>

    <script src="js/view-questions.js"></script>

</body>

</html>

view-results.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Exam Results</title>

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

</head>

<body>

    <h2>Student Exam Results</h2>

    <div class="card">

        <table border="1">

            <tr>

                <th>ID</th>

                <th>Student Name</th>

                <th>Exam ID</th>

                <th>Score</th>

            </tr>

            <tbody id="resultTable"></tbody>

        </table>

        <a href="dashboard.html">Back to Home</a>

    </div>

    <script src="js/view-result.js"></script>

</body>

</html>

**Java Script**

admin.js

const baseUrl = 'http://localhost:8080/api';

async function loadExams() {

    try{

        const res = await fetch(`${baseUrl}/exams`);

        const exams = await res.json();

        const examSelect = document.getElementById('examSelect');

        examSelect.innerHTML= `<option value="">--Select Exam--</option>`;

        for(const exam of exams){

            examSelect.innerHTML += `<option value="${exam.id}">${exam.title}</option>`;

        }

    }catch(err){

        alert("❌ Could not Load exams");

    }

}

// Exam

document.getElementById('createExamForm').addEventListener('submit' ,async function (e) {

    e.preventDefault();

    const title = document.getElementById('title').value.trim();

    if(!title) return alert('Please Enter exam title');

    const res = await fetch(`${baseUrl}/exam`,{

        method : 'POST',

        headers:{'Content-Type':'application/json'},

        body : JSON.stringify({title:title})

    });

    if(!res.ok) return alert('❌Failed to create Exam');

    alert('✅ Exam Created!');

    this.reset();

    await loadExams();

});

function createQuestionElement(idx){

    const div = document.createElement('div');

    div.className='question-block';

    div.innerHTML = `

    <label>Question ${idx+1}:</label>

    <input type="text" name="questionText" placeholder="Enter question text" required/> <br>

    <label>Option A : </label>

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

    <label>Option B : </label>

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

    <label>Option C : </label>

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

    <label>Option D : </label>

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

    <label>Correct Option : </label>

    <select name="correctOption" required>

        <option values="">Select Correct Option</option>

        <option values="A">A</option>

        <option values="B">B</option>

        <option values="C">C</option>

        <option values="D">D</option>

    </select>

    <button type="button" class="remove-btn">Remove Question</button>

    `;

    div.querySelector('.remove-btn').addEventListener('click',()=>{

        div.remove();

        updateQuestionLabels();

    });

    return div;

}

function updateQuestionLabels(){

    document.querySelectorAll('.question-block').forEach((block,i) => {

        block.querySelector('label').textContent = `Question ${i+1} : `

    });

}

document.getElementById('addQuestionBtn').addEventListener('click',()=>{

    const questionsContainer = document.getElementById('questionsContainer');

    questionsContainer.appendChild(createQuestionElement(questionsContainer.children.length));

});

// Submit

document.getElementById('addQuestionForm').addEventListener('submit', async function(e){

    e.preventDefault();

    const examId = document.getElementById('examSelect').value;

    if(!examId) return alert('Please select an Exam');

    const questionBlock = document.querySelectorAll('.question-block');

    if(questionBlock.length===0) return alert('Please add at least one question');

    const questions = [];

    for(const block of questionBlock){

        const questionText = block.querySelector('input[name="questionText"]').value.trim();

        const optionA = block.querySelector('input[name="optionA"]').value.trim();

        const optionB = block.querySelector('input[name="optionB"]').value.trim();

        const optionC = block.querySelector('input[name="optionC"]').value.trim();

        const optionD = block.querySelector('input[name="optionD"]').value.trim();

        const correctOption = block.querySelector('select[name="correctOption"]').value;

        if(!questionText || !optionA || !optionB || !optionC || !optionD || !correctOption){

            alert('❌ Please fill all fields for each question.')

            return;

        }

        questions.push(

            {

                questionText,

                optionA,

                optionB,

                optionC,

                optionD,

                correctOption,

                examId:parseInt(examId)

            }

        );

    }

    const res = await fetch(`${baseUrl}/questions`,{

        method : 'POST',

        headers : {'Content-Type':'application/json'},

        body:JSON.stringify(questions)

    });

    if(!res.ok) return alert('❌ Failed to Add Questions');

    alert('✅ Question Added');

    document.getElementById('questionsContainer').innerHTML = '';

})

// load exam

window.addEventListener('DOMContentLoaded',loadExams)

attend-exam.js

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

      const baseUrl = 'http://localhost:8080/api';

      fetch(`${baseUrl}/exams`)

        .then(res => {

          if (!res.ok) throw new Error("Network response was not ok");

          return res.json();

        })

        .then(data => {

          const container = document.getElementById("examList");

          if (!data.length) {

            container.innerHTML = "<p style='text-align:center;'>No exams found</p>";

            return;

          }

          data.forEach(e => {

            const card = document.createElement("div");

            card.className = "exam-card";

            card.innerHTML = `

              <div class="exam-title">${e.title}</div>

              <a class="exam-button" href="exam.html?examId=${e.id}">Start Exam</a>

            `;

            container.appendChild(card);

          });

        })

        .catch(err => {

          console.error("Failed to fetch exams:", err);

          document.getElementById("examList").innerHTML = "<p>Error loading exams</p>";

        });

    });

list-exam.js

fetch('http://localhost:8080/api/exams')

    .then(res=>res.json())

    .then(data => {

        const table = document.getElementById('examTable');

        data.forEach(r=>{

            table.innerHTML += `

            <tr>

             <td>${r.id}</td>

             <td>${r.title}</td>

            </tr>

            `;

        });

    })

    .catch(()=>{

        document.getElementById("examTable").innerHTML = "<tr><td colspan='4'>Failed to load Exam List</td></tr>";

    })

result.js

const urlParams = new URLSearchParams(window.location.search);

const studentName = urlParams.get("studentName");

const score = urlParams.get("score");

const total = Number(urlParams.get("total"));

const percentage = (score/total)\*100;

document.getElementById("studentName").textContent = `Student : ${studentName}`;

document.getElementById("score").textContent = `Correct Answer : ${score}`;

document.getElementById("percentage").textContent = `Percentage : ${percentage}%`;

take-exam.js

const baseUrl = 'http://localhost:8080/api'

const examId = new URLSearchParams(window.location.search).get("examId");

fetch(`${baseUrl}/questions/${examId}`)

.then(res => res.json())

.then(questions =>{

    const form = document.getElementById("examForm");

    questions.forEach((q,i)=>{

        const questionHtml =`

        <p><strong>${i+1}. ${q.questionText}</strong></p>

        <label><input type="radio" name="q${q.id}" value="A">${q.optionA}</label>

        <label><input type="radio" name="q${q.id}" value="B">${q.optionB}</label>

        <label><input type="radio" name="q${q.id}" value="C">${q.optionC}</label>

        <label><input type="radio" name="q${q.id}" value="D">${q.optionD}</label><br><br>

        `;

        form.innerHTML +=questionHtml;

    })

    const btn = document.createElement("button");

    btn.type = "button";

    btn.textContent = "Submit";

    btn.onclick = () => submitExam(questions);

    form.appendChild(btn);

});

function submitExam(questions){

    let score=0;

    let unanswered = false;

    questions.forEach(q =>{

        const selected = document.querySelector(`input[name="q${q.id}"]:checked`);

        if(!selected){

            unanswered = true;

        }else if(selected.value === q.correctOption){

            score++;

        }

    });

    if(unanswered){

        alert("Please answer all questions before submitting.")

        return;

    }

    const student = prompt("Enter your Name");

    if(!student) return;

    fetch(`${baseUrl}/submit`,{

        method: "POST",

        headers : {'Content-Type' :'application/json'},

        body : JSON.stringify({

            examId:Number(examId),

            studentName : student,

            score : score

        })

    }).then(res=>{

        if(res.ok){

            alert(`✅ Submitted successfully. Score: ${score}/${questions.length}`);

            window.location.href = `result.html?studentName=${encodeURIComponent(student)}&score=${score}&total=${questions.length}`;

        }else{

            alert("❌ Failed to Submit result.");

        }

    });

}

View-questions.js

const baseUrl = 'http://localhost:8080/api';

async function loadExams(){

    try{

        const res = await fetch(`${baseUrl}/exams`);

        if(!res.ok) throw new Error('Failed fetching exams');

        const exams = await res.json();

        const select = document.getElementById('examSelect');

        select.innerHTML = '<option value="">-- Select Exam --</option>';

        exams.forEach(exam =>{

            const opt = document.createElement('option');

            opt.value = exam.id;

            opt.textContent = exam.title;

            select.appendChild(opt);

        });

    }catch(e){

        alert('Error Loading Exam : '+e.message);

    }

}

async function loadQuestionForExam(examId){

    const table = document.getElementById('questionTable')

    const tbody = document.getElementById('questionsBody')

    const noQMsg = document.getElementById('noQuestionMsg');

    tbody.innerHTML = '';

    table.style.display = 'none';

    noQMsg.style.display = 'none';

    if(!examId) return;

    try{

        const res = await fetch(`${baseUrl}/questions/${examId}`);

        if(!res.ok) throw new Error('Failed fetching questions');

        const questions = await res.json();

        if(questions.length ===0){

            noQMsg.style.display ='block';

            return;

        }

        questions.forEach((q,i)=>{

            const tr = document.createElement('tr');

            tr.innerHTML = `

            <td>${i+1}</td>

            <td>${q.questionText}</td>

            <td>${q.optionA}</td>

            <td>${q.optionB}</td>

            <td>${q.optionC}</td>

            <td>${q.optionD}</td>

            <td>${q.correctOption}</td>

            <td><button class="delete-btn" data-qid="${q.id}">Delete</button></td>

            `;

            tbody.appendChild(tr);

        });

        table.style.display = 'table';

        document.querySelectorAll('button.delete-btn').forEach(btn=>{

            btn.addEventListener('click' , async function(){

                const questionId = this.getAttribute('data-qid');

                if(confirm('Are you sure want to delete this question ? ')){

                    try{

                        const delRes = await fetch(`${baseUrl}/question/${questionId}`,{

                            method:'DELETE'

                        });

                        if(!delRes.ok) throw new Error('Delete Failed');

                        alert('✅ Question deleted successfully.')

                        loadQuestionForExam(examId);

                    }catch(e){

                        alert('Error Deleting question'+e.message)

                    }

                }

            });

        });

    }catch(e){

        alert('ERROR  Loading Questions : '+e.message);

    }

}

document.getElementById('examSelect').addEventListener('change',e=>{

    loadQuestionForExam(e.target.value);

})

window.addEventListener('DOMContentLoaded',loadExams);

view-result.js

fetch('http://localhost:8080/api/result')

    .then(res=>res.json())

    .then(data => {

        const table = document.getElementById('resultTable');

        data.forEach(r=>{

            table.innerHTML += `

            <tr>

             <td>${r.id}</td>

             <td>${r.studentName}</td>

             <td>${r.examId}</td>

             <td>${r.score}</td>

            </tr>

            `;

        });

    })

    .catch(()=>{

        document.getElementById("resultTable").innerHTML = "<tr><td colspan='4'>Failed to load result</td></tr>";

    })

CSS

admin.css

body {

    font-family: Arial, sans-serif;

    background: #f4f4f4;

    margin: 0;

    padding: 20px;

    background-image: url('../images/backgorund.jpg');

    background-repeat: no-repeat;

    background-size: cover;

    background-position: center;

}

h2 {

    color: #333;

    text-align: center;

}

form {

    background: #fff;

    padding: 20px;

    margin: 20px auto;

    border-radius: 10px;

    width: 500px;

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

}

label {

    display: block;

    margin: 10px 0 5px;

}

select, textarea, button {

    width: 100%;

    padding: 8px;

    margin-bottom: 15px;

}

input{

    width: 90%;

    padding: 8px;

    margin-bottom: 15px;

}

button {

    background-color: #007bff;

    color: white;

    border: none;

    cursor: pointer;

    transition: 0.3s;

    border-radius: 10px;

}

button:hover {

    background-color: #0056b3;

}

.section {

    margin-top: 40px;

}

.question-block {

    border: 1px solid #ccc;

    padding: 10px;

    margin-bottom: 10px;

    border-radius: 6px;

    background: #fafcff;

}

.remove-btn {

    width: auto !important;

    background: #d9534f;

    margin-top: 0;

    margin-bottom: 0;

    border-radius: 10px;

}

.remove-btn:hover {

    background: #c9302c;

}

a{

  background: #007aff;

  color: white;

  padding: 10px 16px;

  border-radius: 10px;

  text-decoration: none;

  display: inline-block;

  margin-top: 1rem;

}

a:hover {

  background: #005fcc;

}

exam-style.css

body {

    font-family: -apple-system, BlinkMacSystemFont, sans-serif;

    background: #f2f4f8;

    margin: 0;

    padding: 2rem;

    background-image: url('../images/backgorund.jpg');

    background-repeat: no-repeat;

    background-size: cover;

    background-position: center;

}

h1 {

    text-align: center;

    font-size: 2rem;

    color: #007aff;

    margin-bottom: 2rem;

}

.exam-container {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

    gap: 1.5rem;

    max-width: 1200px;

    margin: 0 auto;

}

.exam-card {

    background: #ffffff;

    padding: 1.5rem;

    border-radius: 16px;

    box-shadow: 0 6px 18px rgba(0, 0, 0, 0.1);

    transition: transform 0.3s ease, box-shadow 0.3s ease;

    display: flex;

    flex-direction: column;

    align-items: center;

}

.exam-card:hover {

    transform: translateY(-5px);

    box-shadow: 0 10px 20px rgba(0, 0, 0, 0.15);

}

.exam-title {

    font-size: 1.3rem;

    font-weight: 600;

    color: #007aff;

    margin-bottom: 1rem;

    text-align: center;

}

.exam-button {

    background: #007aff;

    color: white;

    padding: 10px 18px;

    border-radius: 10px;

    text-decoration: none;

    font-size: 1rem;

    transition: background 0.3s;

}

.exam-button:hover {

    background: #005fcc;

}

Result-style.css

body {

    font-family: sans-serif;

    text-align: center;

    background-color: #f4f4f9;

    padding-top: 50px;

    background-image: url('../images/backgorund.jpg');

    background-repeat: no-repeat;

    background-size: cover;

    background-position: center;

}

.card {

    display: inline-block;

    background: white;

    padding: 2rem;

    border-radius: 10px;

    box-shadow: 0 4px 15px rgba(0,0,0,0.1);

}

h1{

    color: #007aff;

}

a{

  background: #007aff;

  color: white;

  padding: 10px 16px;

  border-radius: 10px;

  text-decoration: none;

  display: inline-block;

}

a:hover{

    background: #005fcc

}

img{

    width: 90%;

    height: 100px;

}

style.css

body {

  font-family: -apple-system, BlinkMacSystemFont, sans-serif;

  background: #f2f4f8;

  margin: 0;

  padding: 2rem;

  color: #333;

  background-image: url('../images/backgorund.jpg');

  background-repeat: no-repeat;

  background-size: cover;

  background-position: center;

}

h1, h2, h3 {

  text-align: center;

  color: #007aff;

  margin-bottom: 1rem;

}

table {

  width: 100%;

  border-collapse: collapse;

  margin: 20px auto;

  background: #fff;

  border-radius: 10px;

  overflow: hidden;

  box-shadow: 0 6px 20px rgba(0, 0, 0, 0.05);

}

th, td {

  padding: 12px 20px;

  text-align: left;

  border-bottom: 1px solid #eee;

}

th {

  background-color: #f0f4f9;

}

form {

  background: #fff;

  max-width: 600px;

  margin: auto;

  padding: 2rem;

  border-radius: 15px;

  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);

}

input, textarea, select, button {

  width: 100%;

  padding: 12px;

  margin-bottom: 1rem;

  font-size: 1rem;

  border: 1px solid #ccc;

  border-radius: 10px;

  box-sizing: border-box;

}

button {

  background-color: #007aff;

  color: white;

  border: none;

  cursor: pointer;

  transition: background 0.3s ease;

}

button:hover {

  background-color: #005fcc;

}

.exam-container, .card-container {

  display: grid;

  grid-template-columns: repeat(auto-fit, minmax(260px, 1fr));

  gap: 1.5rem;

  max-width: 1200px;

  margin: 0 auto;

}

.card, .exam-card {

  background: #ffffff;

  padding: 1.5rem;

  border-radius: 16px;

  box-shadow: 0 6px 18px rgba(0, 0, 0, 0.1);

  transition: 0.3s ease;

  text-align: center;

}

.card:hover, .exam-card:hover {

  transform: translateY(-5px);

  box-shadow: 0 10px 20px rgba(0, 0, 0, 0.15);

}

.card a, .exam-button ,.button{

  background: #007aff;

  color: white;

  padding: 10px 16px;

  border-radius: 10px;

  text-decoration: none;

  display: inline-block;

  margin-top: 1rem;

}

.card a:hover, .exam-button:hover {

  background: #005fcc;

}

img{

  width: 80%;

}

take-exam-style.css

body {

    font-family: -apple-system, BlinkMacSystemFont, sans-serif;

    background-color: #f0f0f5;

    padding: 2rem;

    background-image: url('../images/backgorund.jpg');

    background-repeat: no-repeat;

    background-size: cover;

    background-position: center;

}

form {

    max-width: 600px;

    margin: auto;

    background: #fff;

    padding: 2rem;

    border-radius: 15px;

    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);

}

p {

    margin-bottom: 0.5rem;

}

label {

    display: block;

    margin-bottom: 0.5rem;

}

button {

    background: #007aff;

    color: white;

    padding: 10px 20px;

    border: none;

    border-radius: 10px;

    font-size: 1rem;

    margin-top: 1rem;

    cursor: pointer;

}