**Dashboard.cshtml**

@model IEnumerable<CustomerManagement.Models.Entities.Customer>

<link rel="stylesheet" href="~/css/admin-dashboard.css" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

<!-- Dashboard Header -->

<div class="dashboard-header">

<h1>Customer Management Dashboard</h1>

<p>Manage customer registrations and KYC verification</p>

</div>

<!-- Refresh Button -->

<div class="refresh-container">

<button type="button" class="btn-refresh" onclick="refreshData()">

<i class="fas fa-sync-alt"></i>

Refresh

</button>

</div>

<!-- Table Controls -->

<div class="table-controls">

<div class="rows-control">

<label for="rowsPerPage">

<i class="fas fa-list-ol"></i>

Rows per page:

</label>

<select id="rowsPerPage" class="form-select" onchange="changeRowsPerPage()">

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

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

<option value="10">10</option>

<option value="15">15</option>

<option value="20">20</option>

</select>

</div>

<div class="controls-right">

<!-- Filter Dropdown -->

<div class="dropdown me-3">

<button class="btn btn-outline-primary dropdown-toggle" type="button" id="filterDropdown" data-bs-toggle="dropdown" aria-expanded="false">

<i class="fas fa-filter"></i> Filter by Status

</button>

<ul class="dropdown-menu" aria-labelledby="filterDropdown">

<li>

<a class="dropdown-item" href="#" onclick="filterByStatus('all')">

<i class="fas fa-list"></i> All Status

</a>

</li>

<li>

<a class="dropdown-item" href="#" onclick="filterByStatus('pending')">

<i class="fas fa-clock"></i> Pending

</a>

</li>

<li>

<a class="dropdown-item" href="#" onclick="filterByStatus('verified')">

<i class="fas fa-check-circle"></i> Verified

</a>

</li>

</ul>

</div>

<!-- Bootstrap Search -->

<div class="input-group search-input-group">

<span class="input-group-text">

<i class="fas fa-search"></i>

</span>

<input type="text" class="form-control" id="searchInput" placeholder="Search customers..." oninput="searchCustomers()">

</div>

</div>

</div>

<!-- Table Container -->

<div class="table-container">

<table class="table">

<thead>

<tr>

<th><i class="fas fa-id-badge"></i> Customer ID</th>

<th><i class="fas fa-user"></i> Name</th>

<th><i class="fas fa-envelope"></i> Email</th>

<th><i class="fas fa-map-marker-alt"></i> Address</th>

<th><i class="fas fa-file-pdf"></i> Documents</th>

<th><i class="fas fa-check-circle"></i> KYC Status</th>

<th class="actions-column"><i class="fas fa-cogs"></i> Actions</th>

</tr>

</thead>

<tbody id="customerTableBody">

@foreach (var customer in Model)

{

<tr class="customer-row" data-customer-id="@customer.Id" data-original-id="@customer.Id" data-kyc-status="@customer.KycStatus.ToLower()">

<td><span class="customer-id">#@customer.Id</span></td>

<td><strong>@customer.FullName</strong></td>

<td>@customer.Email</td>

<td>@customer.Address</td>

<td>

<a href="@customer.DocumentPath" target="\_blank" class="document-link">

documents/@(customer.FullName.ToLower().Replace(" ", "\_"))\_kyc.pdf

</a>

</td>

<td>

@if (customer.KycStatus == "Approved" || customer.KycStatus == "Verified")

{

<span class="status-badge status-verified">

<i class="fas fa-check-circle"></i> @customer.KycStatus

</span>

}

else

{

<span class="status-badge status-pending">

<i class="fas fa-clock"></i> @customer.KycStatus

</span>

}

</td>

<td class="actions-cell">

<div class="action-buttons">

<button class="btn-action btn-edit" onclick="openEditModal(@customer.Id)">

<i class="fas fa-edit"></i>

<span>Edit</span>

</button>

<form asp-action="Delete" method="post" style="display:inline;" onsubmit="handleFormSubmit(event)">

<input type="hidden" name="id" value="@customer.Id" />

<button type="submit" class="btn-action btn-delete"

onclick="return confirm('Are you sure you want to delete this customer?');">

<i class="fas fa-trash-alt"></i>

<span>Delete</span>

</button>

</form>

<form asp-action="Verify" method="post" style="display:inline;" onsubmit="handleFormSubmit(event)">

<input type="hidden" name="id" value="@customer.Id" />

<button type="submit" class="btn-action btn-verify">

<i class="fas fa-user-check"></i>

<span>Verify</span>

</button>

</form>

</div>

</td>

</tr>

}

</tbody>

</table>

<!-- Bootstrap Pagination -->

<nav aria-label="Customer pagination">

<ul class="pagination justify-content-center" id="paginationContainer">

<!-- Pagination buttons will be generated by JavaScript -->

</ul>

</nav>

<!-- Pagination Info -->

<div class="pagination-info" id="paginationInfo">

<!-- Pagination info will be generated by JavaScript -->

</div>

</div>

<!-- Edit Modal -->

<div class="modal fade" id="editModal" tabindex="-1" aria-labelledby="editModalLabel" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="editModalLabel">

<i class="fas fa-edit"></i> Edit Customer

</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body" id="editModalBody">

<!-- Form will be loaded here -->

</div>

</div>

</div>

</div>

@section Scripts {

<script>

// Global variables for pagination and search

let currentPage = 1;

let rowsPerPage = 2; // Default

let allRows = [];

let filteredRows = [];

let currentScrollPosition = 0;

let currentFilter = 'all';

// Initialize on page load

document.addEventListener('DOMContentLoaded', function() {

// Restore saved state first

restoreSavedState();

initializeTable();

});

function restoreSavedState() {

const savedRowsPerPage = sessionStorage.getItem('rowsPerPage') || localStorage.getItem('rowsPerPage');

const savedCurrentPage = sessionStorage.getItem('currentPage') || localStorage.getItem('currentPage');

const savedFilter = sessionStorage.getItem('currentFilter') || localStorage.getItem('currentFilter');

if (savedRowsPerPage) {

rowsPerPage = parseInt(savedRowsPerPage);

document.getElementById('rowsPerPage').value = rowsPerPage;

}

if (savedCurrentPage) {

currentPage = parseInt(savedCurrentPage);

}

if (savedFilter) {

currentFilter = savedFilter;

}

}

function saveCurrentState() {

// Save to both sessionStorage and localStorage for persistence

sessionStorage.setItem('rowsPerPage', rowsPerPage);

sessionStorage.setItem('currentPage', currentPage);

sessionStorage.setItem('currentFilter', currentFilter);

localStorage.setItem('rowsPerPage', rowsPerPage);

localStorage.setItem('currentPage', currentPage);

localStorage.setItem('currentFilter', currentFilter);

}

function initializeTable() {

allRows = Array.from(document.querySelectorAll('.customer-row'));

applyFilters();

updateTable();

}

function applyFilters() {

let rows = [...allRows];

// Apply status filter

if (currentFilter !== 'all') {

rows = rows.filter(row => {

const status = row.getAttribute('data-kyc-status');

return status === currentFilter;

});

}

// Apply search filter

const searchTerm = document.getElementById('searchInput').value.toLowerCase();

if (searchTerm !== '') {

rows = rows.filter(row => {

const text = row.textContent.toLowerCase();

return text.includes(searchTerm);

});

}

filteredRows = rows;

}

function updateTable() {

// Save current scroll position

currentScrollPosition = window.pageYOffset || document.documentElement.scrollTop;

// Hide all rows first

allRows.forEach(row => row.style.display = 'none');

// Calculate pagination

const totalRows = filteredRows.length;

const totalPages = Math.ceil(totalRows / rowsPerPage) || 1;

const startIndex = (currentPage - 1) \* rowsPerPage;

const endIndex = Math.min(startIndex + rowsPerPage, totalRows);

// Show current page rows

for (let i = startIndex; i < endIndex; i++) {

if (filteredRows[i]) {

filteredRows[i].style.display = '';

}

}

// Update pagination controls

updatePagination(totalPages, totalRows, startIndex, endIndex);

// Save current state

saveCurrentState();

// Restore scroll position

setTimeout(() => {

window.scrollTo(0, currentScrollPosition);

}, 10);

}

function updatePagination(totalPages, totalRows, startIndex, endIndex) {

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

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

// Clear existing pagination

paginationContainer.innerHTML = '';

if (totalPages <= 1) {

// Show single page button when only one page

const singleBtn = createBootstrapPaginationButton('1', 1, false, true);

paginationContainer.appendChild(singleBtn);

} else {

// First page button

const firstBtn = createBootstrapPaginationButton('&laquo;&laquo;', 1, currentPage === 1);

paginationContainer.appendChild(firstBtn);

// Previous page button

const prevBtn = createBootstrapPaginationButton('&laquo;', currentPage - 1, currentPage === 1);

paginationContainer.appendChild(prevBtn);

// Page number buttons

const maxVisiblePages = 5;

let startPage = Math.max(1, currentPage - Math.floor(maxVisiblePages / 2));

let endPage = Math.min(totalPages, startPage + maxVisiblePages - 1);

if (endPage - startPage + 1 < maxVisiblePages) {

startPage = Math.max(1, endPage - maxVisiblePages + 1);

}

for (let i = startPage; i <= endPage; i++) {

const pageBtn = createBootstrapPaginationButton(i, i, false, i === currentPage);

paginationContainer.appendChild(pageBtn);

}

// Next page button

const nextBtn = createBootstrapPaginationButton('&raquo;', currentPage + 1, currentPage === totalPages);

paginationContainer.appendChild(nextBtn);

// Last page button

const lastBtn = createBootstrapPaginationButton('&raquo;&raquo;', totalPages, currentPage === totalPages);

paginationContainer.appendChild(lastBtn);

}

// Update pagination info

if (totalRows > 0) {

paginationInfo.innerHTML = `

<i class="fas fa-info-circle"></i>

Showing ${startIndex + 1} to ${endIndex} of ${totalRows} entries

`;

} else {

paginationInfo.innerHTML = `

<i class="fas fa-info-circle"></i>

No matching records found

`;

}

}

function createBootstrapPaginationButton(text, page, disabled, active = false) {

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

li.className = `page-item ${active ? 'active' : ''} ${disabled ? 'disabled' : ''}`;

const button = document.createElement('a');

button.className = 'page-link';

button.innerHTML = text;

button.href = '#';

if (!disabled) {

button.onclick = (e) => {

e.preventDefault();

goToPage(page);

};

}

li.appendChild(button);

return li;

}

function goToPage(page) {

currentPage = page;

updateTable();

}

function changeRowsPerPage() {

rowsPerPage = parseInt(document.getElementById('rowsPerPage').value);

currentPage = 1;

updateTable();

}

function searchCustomers() {

currentPage = 1;

applyFilters();

updateTable();

}

function filterByStatus(status) {

currentFilter = status;

currentPage = 1;

// Update dropdown button text

const filterBtn = document.getElementById('filterDropdown');

if (status === 'all') {

filterBtn.innerHTML = '<i class="fas fa-filter"></i> Filter by Status';

} else if (status === 'pending') {

filterBtn.innerHTML = '<i class="fas fa-clock"></i> Pending';

} else if (status === 'verified') {

filterBtn.innerHTML = '<i class="fas fa-check-circle"></i> Verified';

}

applyFilters();

updateTable();

}

function refreshData() {

const refreshBtn = document.querySelector('.btn-refresh i');

refreshBtn.classList.add('fa-spin');

// Clear search and filters

document.getElementById('searchInput').value = '';

currentFilter = 'all';

document.getElementById('filterDropdown').innerHTML = '<i class="fas fa-filter"></i> Filter by Status';

// Reset to original data

currentPage = 1;

applyFilters();

setTimeout(() => {

updateTable();

refreshBtn.classList.remove('fa-spin');

}, 800);

}

function openEditModal(id) {

$.get('/Admin/Edit/' + id, function (data) {

$('#editModalBody').html(data);

$('#editModal').modal('show');

});

}

// Handle form submissions to prevent page scroll and maintain state

function handleFormSubmit(event) {

// Save current scroll position and state before form submission

currentScrollPosition = window.pageYOffset || document.documentElement.scrollTop;

saveCurrentState();

sessionStorage.setItem('scrollPosition', currentScrollPosition);

}

// Restore state after page reload

window.addEventListener('load', function() {

const savedScrollPosition = sessionStorage.getItem('scrollPosition');

if (savedScrollPosition) {

setTimeout(() => {

window.scrollTo(0, parseInt(savedScrollPosition));

// Clear saved scroll position

sessionStorage.removeItem('scrollPosition');

}, 100);

}

});

</script>

}

**Edit.cshtml**

@model CustomerManagement.Models.EditCustomerViewModel

<!-- Remove any layout or navigation references -->

@{

Layout = null;

}

<form asp-action="Edit" method="post" id="editForm">

@Html.AntiForgeryToken()

<input type="hidden" asp-for="Id" />

<div class="row">

<div class="col-md-6">

<div class="form-group">

<label asp-for="FullName">

<i class="fas fa-user"></i> Full Name

</label>

<input asp-for="FullName" class="form-control" placeholder="Enter full name" />

<span asp-validation-for="FullName" class="text-danger"></span>

</div>

</div>

<div class="col-md-6">

<div class="form-group">

<label asp-for="Email">

<i class="fas fa-envelope"></i> Email

</label>

<input asp-for="Email" class="form-control" placeholder="Enter email address" />

<span asp-validation-for="Email" class="text-danger"></span>

</div>

</div>

</div>

<div class="row">

<div class="col-md-6">

<div class="form-group">

<label asp-for="Phone">

<i class="fas fa-phone"></i> Phone

</label>

<input asp-for="Phone" class="form-control" placeholder="Enter phone number" />

<span asp-validation-for="Phone" class="text-danger"></span>

</div>

</div>

<div class="col-md-6">

<div class="form-group">

<label asp-for="DateOfBirth">

<i class="fas fa-calendar"></i> Date of Birth

</label>

<input asp-for="DateOfBirth" class="form-control" type="date" />

<span asp-validation-for="DateOfBirth" class="text-danger"></span>

</div>

</div>

</div>

<div class="form-group">

<label asp-for="Address">

<i class="fas fa-map-marker-alt"></i> Address

</label>

<textarea asp-for="Address" class="form-control" rows="3" placeholder="Enter complete address"></textarea>

<span asp-validation-for="Address" class="text-danger"></span>

</div>

<div class="row">

<div class="col-md-6">

<div class="form-group">

<label asp-for="IdentityType">

<i class="fas fa-id-card"></i> Identity Type

</label>

<input asp-for="IdentityType" class="form-control" placeholder="Enter identity type" />

<span asp-validation-for="IdentityType" class="text-danger"></span>

</div>

</div>

<div class="col-md-6">

<div class="form-group">

<label asp-for="KycStatus">

<i class="fas fa-check-circle"></i> KYC Status

</label>

<select asp-for="KycStatus" class="form-control">

<option value="Pending">Pending</option>

<option value="Verified">Verified</option>

</select>

<span asp-validation-for="KycStatus" class="text-danger"></span>

</div>

</div>

</div>

<div class="d-flex justify-content-end gap-2 mt-4">

<button type="button" class="btn-secondary" data-bs-dismiss="modal">

<i class="fas fa-times"></i> Cancel

</button>

<button type="submit" class="btn-primary">

<i class="fas fa-save"></i> Save Changes

</button>

</div>

</form>

<script>

$('#editForm').submit(function (e) {

e.preventDefault();

const submitBtn = $(this).find('.btn-primary');

const originalText = submitBtn.html();

submitBtn.html('<i class="fas fa-spinner fa-spin"></i> Saving...');

submitBtn.prop('disabled', true);

$.ajax({

type: 'POST',

url: '/Admin/Edit',

data: $(this).serialize(),

success: function () {

$('#editModal').modal('hide');

location.reload(); // Refresh dashboard to reflect changes

},

error: function() {

alert('Error updating customer. Please try again.');

submitBtn.html(originalText);

submitBtn.prop('disabled', false);

}

});

});

</script>

**Admin-dashboard.css**

/\* Admin Dashboard Styling \*/

:root {

--primary: #2c3e50;

--secondary: #3498db;

--accent: #16a085;

--light: #f8f9fa;

--dark: #343a40;

--success: #27ae60;

--warning: #f39c12;

--danger: #e74c3c;

--info: #1abc9c;

}

body {

background-color: #f5f7fa;

font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;

padding: 20px 0;

margin: 0;

}

/\* Header Container \*/

.dashboard-header {

background: linear-gradient(135deg, var(--primary), var(--secondary));

color: white;

border-radius: 15px;

padding: 25px;

margin: 25px auto;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

text-align: center;

max-width: 96%;

}

.dashboard-header h1 {

margin: 0;

font-size: 2rem;

font-weight: 700;

text-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);

display: flex;

align-items: center;

justify-content: center;

gap: 15px;

}

.dashboard-header h1::before {

content: "👥🛡️";

font-size: 1.8rem;

}

.dashboard-header p {

margin: 10px 0 0 0;

opacity: 0.9;

font-size: 1.1rem;

}

/\* Refresh Button \*/

.refresh-container {

max-width: 96%;

margin: 0 auto;

display: flex;

justify-content: flex-end;

margin-bottom: 20px;

}

.btn-refresh {

background: linear-gradient(135deg, var(--accent), var(--info));

color: white;

border: none;

border-radius: 8px;

padding: 10px 20px;

font-weight: 600;

transition: all 0.3s;

box-shadow: 0 4px 6px rgba(22, 160, 133, 0.2);

display: flex;

align-items: center;

gap: 8px;

}

.btn-refresh:hover {

background: linear-gradient(135deg, var(--info), var(--accent));

transform: translateY(-2px);

color: white;

box-shadow: 0 6px 12px rgba(22, 160, 133, 0.3);

}

.btn-refresh i {

font-size: 1rem;

}

/\* Table Controls \*/

.table-controls {

background-color: white;

border-radius: 12px;

padding: 20px;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05);

margin: 20px auto;

max-width: 96%;

display: flex;

justify-content: space-between;

align-items: center;

flex-wrap: wrap;

gap: 15px;

min-height: 70px;

}

.rows-control {

display: flex;

align-items: center;

gap: 10px;

}

.rows-control label {

color: var(--primary);

font-weight: 600;

font-size: 0.95rem;

display: flex;

align-items: center;

gap: 8px;

white-space: nowrap;

}

.rows-control .form-select {

min-width: 80px;

border: 2px solid #e0e0e0;

border-radius: 6px;

font-size: 0.9rem;

}

.rows-control .form-select:focus {

border-color: var(--secondary);

box-shadow: 0 0 0 0.25rem rgba(52, 152, 219, 0.25);

}

/\* Controls Right Side \*/

.controls-right {

display: flex;

align-items: center;

gap: 15px;

}

/\* Bootstrap Search Input Group \*/

.search-input-group {

width: 300px;

}

.search-input-group .input-group-text {

background-color: var(--secondary);

color: white;

border: 2px solid var(--secondary);

}

.search-input-group .form-control {

border: 2px solid #e0e0e0;

border-left: none;

}

.search-input-group .form-control:focus {

border-color: var(--secondary);

box-shadow: 0 0 0 0.25rem rgba(52, 152, 219, 0.25);

}

/\* Filter Dropdown \*/

.dropdown .btn-outline-primary {

border: 2px solid var(--secondary);

color: var(--secondary);

font-weight: 600;

}

.dropdown .btn-outline-primary:hover {

background-color: var(--secondary);

border-color: var(--secondary);

}

/\* Table Container \*/

.table-container {

background-color: white;

border-radius: 12px;

padding: 0;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05);

margin: 20px auto;

max-width: 96%;

overflow: hidden;

border: 4px solid var(--primary);

min-height: 600px;

display: flex;

flex-direction: column;

}

/\* Table Styling \*/

.table {

margin-bottom: 0;

border-collapse: separate;

border-spacing: 0;

width: 100%;

flex: 1;

}

.table thead {

background: linear-gradient(to right, var(--primary), var(--secondary));

}

.table thead th {

background: linear-gradient(135deg, var(--primary), var(--secondary));

color: white;

font-weight: 600;

border: 1px solid rgba(255, 255, 255, 0.2);

border-bottom: 2px solid rgba(255, 255, 255, 0.3);

padding: 18px 15px;

text-transform: uppercase;

font-size: 0.85rem;

letter-spacing: 1px;

vertical-align: middle;

position: relative;

}

.table thead th:first-child {

border-left: none;

}

.table thead th:last-child {

border-right: none;

}

.table thead th i {

margin-right: 8px;

font-size: 1rem;

}

.table thead th.actions-column {

width: 220px;

min-width: 220px;

}

.table tbody td {

padding: 18px 15px;

border: 1px solid #e0e0e0;

border-top: none;

vertical-align: middle;

}

.table tbody td:first-child {

border-left: none;

}

.table tbody td:last-child {

border-right: none;

}

.table tbody tr:hover {

background-color: rgba(52, 152, 219, 0.05);

transition: background-color 0.2s ease;

}

.table tbody tr:last-child td {

border-bottom: none;

}

/\* Customer ID Styling \*/

.customer-id {

font-weight: 700;

color: var(--primary);

font-size: 1rem;

}

/\* Status Badge \*/

.status-badge {

padding: 6px 12px;

border-radius: 20px;

font-weight: 500;

display: inline-flex;

align-items: center;

gap: 5px;

font-size: 0.8rem;

text-align: center;

min-width: 90px;

justify-content: center;

}

.status-verified {

background-color: rgba(39, 174, 96, 0.2);

color: var(--success);

border: 1px solid var(--success);

}

.status-pending {

background-color: rgba(243, 156, 18, 0.2);

color: var(--warning);

border: 1px solid var(--warning);

}

.status-approved {

background-color: rgba(39, 174, 96, 0.2);

color: var(--success);

border: 1px solid var(--success);

}

/\* Document Link \*/

.document-link {

color: var(--secondary);

text-decoration: none;

font-weight: 500;

transition: all 0.3s ease;

}

.document-link:hover {

color: var(--primary);

text-decoration: underline;

}

/\* Action Buttons - Modern Card Style \*/

.actions-cell {

width: 220px;

min-width: 220px;

padding: 12px 8px;

}

.action-buttons {

display: flex;

gap: 6px;

flex-wrap: nowrap;

justify-content: center;

align-items: center;

}

.btn-action {

border-radius: 8px;

padding: 8px 12px;

font-size: 0.75rem;

font-weight: 600;

transition: all 0.3s;

border: none;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

min-width: 50px;

height: 50px;

cursor: pointer;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

gap: 2px;

}

.btn-action i {

font-size: 1rem;

margin-bottom: 2px;

}

.btn-action span {

font-size: 0.7rem;

font-weight: 600;

text-transform: uppercase;

letter-spacing: 0.5px;

}

.btn-edit {

background-color: #ff9500;

color: white;

}

.btn-edit:hover {

background-color: #e6850e;

color: white;

transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(255, 149, 0, 0.3);

}

.btn-delete {

background-color: #ff4757;

color: white;

}

.btn-delete:hover {

background-color: #ff3742;

color: white;

transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(255, 71, 87, 0.3);

}

.btn-verify {

background-color: #2ed573;

color: white;

}

.btn-verify:hover {

background-color: #26d065;

color: white;

transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(46, 213, 115, 0.3);

}

/\* Bootstrap Pagination Styling \*/

.pagination {

margin: 20px 0;

}

.pagination .page-link {

color: var(--primary);

border: 2px solid #e0e0e0;

padding: 8px 12px;

font-weight: 500;

transition: all 0.3s;

}

.pagination .page-item.active .page-link {

background-color: var(--secondary);

border-color: var(--secondary);

color: white;

}

.pagination .page-link:hover {

background-color: var(--secondary);

border-color: var(--secondary);

color: white;

transform: translateY(-1px);

}

.pagination .page-item.disabled .page-link {

opacity: 0.5;

cursor: not-allowed;

}

/\* Pagination Info \*/

.pagination-info {

padding: 0 20px 20px;

color: var(--primary);

font-weight: 500;

font-size: 0.9rem;

display: flex;

align-items: center;

justify-content: center;

gap: 8px;

background-color: #f8f9fa;

min-height: 40px;

}

/\* Modal Styling \*/

.modal-content {

border-radius: 15px;

border: none;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);

overflow: hidden;

}

.modal-header {

background: linear-gradient(to right, var(--primary), var(--secondary));

color: white;

border-bottom: none;

padding: 20px 25px;

}

.modal-title {

font-weight: 600;

letter-spacing: 0.5px;

font-size: 1.3rem;

display: flex;

align-items: center;

gap: 10px;

}

.modal-body {

padding: 25px;

background-color: #fafafa;

}

/\* Form Styling \*/

.form-group {

margin-bottom: 20px;

position: relative;

}

.form-group label {

font-weight: 600;

color: var(--primary);

margin-bottom: 8px;

display: flex;

align-items: center;

gap: 8px;

font-size: 0.95rem;

}

.form-control {

border: 2px solid #e0e0e0;

border-radius: 8px;

padding: 12px 15px;

transition: all 0.3s;

font-size: 0.95rem;

width: 100%;

box-sizing: border-box;

}

.form-control:focus {

border-color: var(--secondary);

box-shadow: 0 0 0 0.25rem rgba(52, 152, 219, 0.25);

outline: none;

}

/\* Form Buttons \*/

.btn-primary {

background: linear-gradient(135deg, var(--secondary), var(--primary));

color: white;

border: none;

border-radius: 8px;

padding: 12px 25px;

font-weight: 600;

transition: all 0.3s;

cursor: pointer;

display: flex;

align-items: center;

gap: 8px;

}

.btn-primary:hover {

background: linear-gradient(135deg, var(--primary), var(--secondary));

color: white;

transform: translateY(-2px);

box-shadow: 0 5px 15px rgba(52, 152, 219, 0.3);

}

.btn-secondary {

background-color: #6c757d;

color: white;

border: none;

border-radius: 8px;

padding: 12px 25px;

font-weight: 600;

transition: all 0.3s;

cursor: pointer;

display: flex;

align-items: center;

gap: 8px;

}

.btn-secondary:hover {

background-color: #5a6268;

color: white;

transform: translateY(-2px);

}

/\* Close button styling \*/

.btn-close {

background: none;

border: none;

font-size: 1.5rem;

color: white;

opacity: 0.8;

}

.btn-close:hover {

opacity: 1;

}

/\* Row styling \*/

.row {

display: flex;

margin: 0 -10px;

flex-wrap: wrap;

}

.col-md-6 {

flex: 0 0 50%;

max-width: 50%;

padding: 0 10px;

box-sizing: border-box;

}

/\* Responsive Design \*/

@media (max-width: 768px) {

.table-controls {

flex-direction: column;

align-items: stretch;

gap: 15px;

min-height: 140px;

}

.controls-right {

flex-direction: column;

gap: 10px;

}

.search-input-group {

width: 100%;

}

.table-container {

overflow-x: auto;

max-width: 98%;

}

.table {

min-width: 900px;

}

.dashboard-header {

margin: 15px auto;

padding: 20px;

max-width: 98%;

}

.dashboard-header h1 {

font-size: 1.5rem;

}

.action-buttons {

flex-direction: column;

gap: 2px;

}

.btn-action {

font-size: 0.65rem;

padding: 4px 6px;

min-width: 28px;

height: 28px;

}

.col-md-6 {

flex: 0 0 100%;

max-width: 100%;

}

.refresh-container {

max-width: 98%;

}

}

@media (max-width: 576px) {

.dashboard-header h1 {

font-size: 1.3rem;

flex-direction: column;

gap: 10px;

}

.dashboard-header p {

font-size: 1rem;

}

.table thead th {

padding: 12px 8px;

font-size: 0.75rem;

}

.table tbody td {

padding: 12px 8px;

}

.actions-cell {

width: 120px;

min-width: 120px;

}

.btn-action {

font-size: 0.6rem;

padding: 3px 5px;

min-width: 24px;

height: 24px;

}

}

/\* Loading states \*/

.loading {

opacity: 0.6;

pointer-events: none;

}

/\* Smooth transitions \*/

\* {

box-sizing: border-box;

}

/\* Prevent layout shifts during transitions \*/

.table-container,

.table-controls {

transition: none !important;

}