**Admin.cshtml**

@model List<BankLoanProject.Models.RepaymentScheduleViewModel>

@{

ViewBag.Title = "Admin Loan Scheduling";

}

<link rel="stylesheet" href="~/css/repayment.css" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/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://code.jquery.com/jquery-3.6.4.min.js"></script>

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

<!-- Dashboard Header -->

<div class="dashboard-header">

<h2>

<i class="fas fa-calendar-check me-3"></i>

Loan Scheduling Dashboard

</h2>

<p>Manage and schedule loan repayments for approved applications</p>

</div>

<!-- Alert Messages -->

@if (TempData["Success"] != null)

{

<div class="alert alert-success">

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

@TempData["Success"]

</div>

}

@if (TempData["Error"] != null)

{

<div class="alert alert-danger">

<i class="fas fa-exclamation-triangle"></i>

@TempData["Error"]

</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="5" selected>5</option>

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

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

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

</select>

</div>

<div class="controls-right">

<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 loans..." oninput="searchLoans()">

</div>

</div>

</div>

<!-- Table -->

<div class="table-container">

<div class="table-wrapper">

<table class="table table-striped">

<thead>

<tr>

<th scope="col"><i class="fas fa-hashtag me-1"></i>Customer ID</th>

<th scope="col"><i class="fas fa-user me-1"></i>Name</th>

<th scope="col"><i class="fas fa-key me-1"></i>Loan ID</th>

<th scope="col"><i class="fas fa-tag me-1"></i>Product Type</th>

<th scope="col"><i class="fas fa-box me-1"></i>Product Name</th>

<th scope="col"><i class="fas fa-percentage me-1"></i>Interest Rate (%)</th>

<th scope="col"><i class="fas fa-rupee-sign me-1"></i>Loan Amount</th>

<th scope="col"><i class="fas fa-hourglass-half me-1"></i>Tenure (Months)</th>

<th scope="col"><i class="fas fa-cogs me-1"></i>Action</th>

</tr>

</thead>

<tbody id="table-body">

@foreach (var loan in Model)

{

<tr class="loan-row" data-loan-id="@loan.LoanId">

<td><strong>@loan.CustomerId</strong></td>

<td><strong>@loan.Name</strong></td>

<td><strong>@loan.LoanId</strong></td>

<td><strong>@loan.ProductType</strong></td>

<td><strong>@loan.ProductName</strong></td>

<td><span class="rate">@loan.InterestRate%</span></td>

<td><span class="amount">@loan.LoanAmount.ToString("C")</span></td>

<td><span class="tenure">@loan.LoanTenure months</span></td>

<td>

<form asp-action="ScheduleLoan" asp-controller="Repayment" method="post" class="schedule-form">

<input type="hidden" name="applicationId" value="@loan.ApplicationId" />

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

<i class="fas fa-calendar-plus me-1"></i>Schedule

</button>

</form>

</td>

</tr>

}

</tbody>

</table>

</div>

<!-- Pagination Section -->

<div class="pagination-section">

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

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

</div>

<nav aria-label="Loan pagination">

<ul class="pagination" id="paginationContainer">

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

</ul>

</nav>

</div>

</div>

@section Scripts {

<script>

// Global variables for pagination and search

let currentPage = 1;

let rowsPerPage = 5;

let allRows = [];

let filteredRows = [];

// Initialize on page load

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

initializeTable();

initializeFormHandlers();

});

function initializeTable() {

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

filteredRows = [...allRows];

updateTable();

}

function initializeFormHandlers() {

const forms = document.querySelectorAll('.schedule-form');

forms.forEach(form => {

form.addEventListener('submit', function(e) {

const button = form.querySelector('button[type="submit"]');

if (button && !button.disabled) {

button.disabled = true;

button.innerHTML = '<i class="fas fa-spinner fa-spin me-1"></i>Scheduling...';

}

});

});

}

function applyFilters() {

let rows = [...allRows];

// 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() {

// 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);

}

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

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

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

// Clear existing pagination

paginationContainer.innerHTML = '';

if (totalPages <= 1) {

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

paginationContainer.appendChild(singleBtn);

} else {

// First page button

const firstBtn = createBootstrapPaginationButton('«', 1, currentPage === 1);

paginationContainer.appendChild(firstBtn);

// Previous page button

const prevBtn = createBootstrapPaginationButton('‹', 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('›', currentPage + 1, currentPage === totalPages);

paginationContainer.appendChild(nextBtn);

// Last page button

const lastBtn = createBootstrapPaginationButton('»', 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 searchLoans() {

currentPage = 1;

applyFilters();

updateTable();

}

function refreshData() {

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

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

// Clear search

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

// Reset to original data

currentPage = 1;

applyFilters();

setTimeout(() => {

updateTable();

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

}, 800);

}

</script>

}

**Customer.cshtml**

@model List<BankLoanProject.Models.RepaymentScheduleViewModel>

<link rel="stylesheet" href="~/css/repayment.css" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/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://code.jquery.com/jquery-3.6.4.min.js"></script>

<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>

<i class="fas fa-credit-card me-3"></i>

Loan Repayment Portal

</h1>

<p>Manage your loan repayments and check outstanding balances</p>

</div>

<!-- Outstanding Balance Section -->

<div class="balance-section">

<h3>

<i class="fas fa-search-dollar me-2"></i>

Check Outstanding Balance

</h3>

<form asp-action="GetOutstandingBalance" asp-controller="Repayment" method="get" class="balance-form">

<div class="form-group">

<label for="loanId">

<i class="fas fa-key me-2"></i>

Loan ID

</label>

<input type="number" class="form-control" name="loanId" placeholder="Enter Loan ID" required />

</div>

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

<i class="fas fa-search me-2"></i>

Get Balance

</button>

</form>

@if (ViewBag.OutstandingBalance != null)

{

<div class="alert alert-info mt-3">

<i class="fas fa-info-circle me-2"></i>

<div>

<strong>Loan ID:</strong> @ViewBag.LoanId<br />

<strong>Outstanding Balance:</strong> ₹@((decimal)ViewBag.OutstandingBalance).ToString("N2")

</div>

</div>

}

</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="5" selected>5</option>

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

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

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

</select>

</div>

<div class="controls-right">

<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 repayments..." oninput="searchRepayments()">

</div>

</div>

</div>

<!-- Table -->

<div class="table-container">

<div class="table-wrapper">

<table class="table table-striped">

<thead>

<tr>

<th scope="col"><i class="fas fa-hashtag me-1"></i>Repayment ID</th>

<th scope="col"><i class="fas fa-key me-1"></i>Loan ID</th>

<th scope="col"><i class="fas fa-tag me-1"></i>Product Type</th>

<th scope="col"><i class="fas fa-box me-1"></i>Product Name</th>

<th scope="col"><i class="fas fa-calendar-alt me-1"></i>Due Date</th>

<th scope="col"><i class="fas fa-rupee-sign me-1"></i>Amount Due</th>

<th scope="col"><i class="fas fa-calendar-check me-1"></i>Payment Date</th>

<th scope="col"><i class="fas fa-info-circle me-1"></i>Payment Status</th>

<th scope="col"><i class="fas fa-cogs me-1"></i>Action</th>

</tr>

</thead>

<tbody id="table-body">

@foreach (var item in Model)

{

<tr class="repayment-row" data-repayment-id="@item.RepaymentId">

<td><strong>@item.RepaymentId</strong></td>

<td><strong>@item.LoanId</strong></td>

<td><strong>@item.ProductType</strong></td>

<td><strong>@item.ProductName</strong></td>

<td><strong>@item.DueDate.ToString("dd-MM-yyyy")</strong></td>

<td><span class="amount">₹@($"{item.AmountDue:N2}")</span></td>

<td><strong>@(item.PaymentDate?.ToString("dd-MM-yyyy") ?? "-")</strong></td>

<td>

@{

string statusClass = "";

string statusIcon = "";

switch (item.PaymentStatus?.ToUpper())

{

case "PAID":

case "COMPLETED":

statusClass = "status-paid";

statusIcon = "fas fa-check-circle";

break;

case "PENDING":

statusClass = "status-pending";

statusIcon = "fas fa-clock";

break;

case "OVERDUE":

statusClass = "status-overdue";

statusIcon = "fas fa-exclamation-triangle";

break;

default:

statusClass = "status-pending";

statusIcon = "fas fa-clock";

break;

}

}

<span class="status-badge @statusClass" data-status="@item.PaymentStatus">

<i class="@statusIcon me-1"></i>@item.PaymentStatus

</span>

</td>

<td>

@if (item.PaymentStatus?.ToUpper() != "PAID" && item.PaymentStatus?.ToUpper() != "COMPLETED")

{

<button class="btn btn-pay" data-bs-toggle="modal" data-bs-target="#paymentModal"

data-repayment-id="@item.RepaymentId"

data-loan-id="@item.LoanId"

data-product-type="@item.ProductType"

data-product-name="@item.ProductName"

data-amount="@item.AmountDue"

data-due-date="@item.DueDate.ToString("dd-MM-yyyy")">

<i class="fas fa-credit-card me-1"></i>Pay

</button>

}

else

{

<span class="text-muted">

<i class="fas fa-check-circle me-1"></i>Paid

</span>

}

</td>

</tr>

}

</tbody>

</table>

</div>

<!-- Pagination Section -->

<div class="pagination-section">

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

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

</div>

<nav aria-label="Repayment pagination">

<ul class="pagination" id="paginationContainer">

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

</ul>

</nav>

</div>

</div>

<!-- Enhanced Payment Modal -->

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

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

<div class="modal-content">

<div class="modal-header">

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

<i class="fas fa-credit-card me-2"></i>Process Payment

</h5>

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

</div>

<div class="modal-body">

<!-- Payment Details Section -->

<div class="payment-details">

<h6><i class="fas fa-info-circle me-2"></i>Payment Details</h6>

<div class="payment-info">

<div class="payment-info-item">

<span class="payment-info-label">

<i class="fas fa-hashtag me-2"></i>Repayment ID:

</span>

<span class="payment-info-value" id="modalRepaymentId"></span>

</div>

<div class="payment-info-item">

<span class="payment-info-label">

<i class="fas fa-key me-2"></i>Loan ID:

</span>

<span class="payment-info-value" id="modalLoanId"></span>

</div>

<div class="payment-info-item">

<span class="payment-info-label">

<i class="fas fa-tag me-2"></i>Product Type:

</span>

<span class="payment-info-value" id="modalProductType"></span>

</div>

<div class="payment-info-item">

<span class="payment-info-label">

<i class="fas fa-box me-2"></i>Product Name:

</span>

<span class="payment-info-value" id="modalProductName"></span>

</div>

<div class="payment-info-item">

<span class="payment-info-label">

<i class="fas fa-calendar-alt me-2"></i>Due Date:

</span>

<span class="payment-info-value" id="modalDueDate"></span>

</div>

<div class="payment-info-item">

<span class="payment-info-label">

<i class="fas fa-rupee-sign me-2"></i>Amount Due:

</span>

<span class="payment-info-value amount" id="modalAmount"></span>

</div>

</div>

</div>

<!-- Payment Form -->

<form id="paymentForm" method="post" asp-action="Pay" asp-controller="Repayment">

@Html.AntiForgeryToken()

<input type="hidden" name="repaymentId" id="repaymentId" />

<div class="form-group">

<label for="paymentMethod">

<i class="fas fa-credit-card me-2"></i>Payment Method

</label>

<select id="paymentMethod" name="paymentMethod" class="form-select" required>

<option value="">-- Select Payment Method --</option>

<option value="UPI">UPI</option>

<option value="NetBanking">Net Banking</option>

<option value="Card">Credit/Debit Card</option>

</select>

</div>

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

<i class="fas fa-check-circle me-2"></i>Confirm Payment

</button>

</form>

</div>

</div>

</div>

</div>

<script>

// Global variables for pagination and search

let currentPage = 1;

let rowsPerPage = 5;

let allRows = [];

let filteredRows = [];

let currentRepaymentId = null;

// Initialize on page load

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

initializeTable();

initializeModal();

initializeFormHandlers();

});

function initializeTable() {

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

filteredRows = [...allRows];

updateTable();

}

function initializeModal() {

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

// Handle modal show event

paymentModal.addEventListener('show.bs.modal', function(event) {

const button = event.relatedTarget;

// Get data from button attributes

currentRepaymentId = button.getAttribute('data-repayment-id');

const loanId = button.getAttribute('data-loan-id');

const productType = button.getAttribute('data-product-type');

const productName = button.getAttribute('data-product-name');

const amount = button.getAttribute('data-amount');

const dueDate = button.getAttribute('data-due-date');

// Populate modal with data

document.getElementById('repaymentId').value = currentRepaymentId;

document.getElementById('modalRepaymentId').textContent = currentRepaymentId;

document.getElementById('modalLoanId').textContent = loanId;

document.getElementById('modalProductType').textContent = productType;

document.getElementById('modalProductName').textContent = productName;

document.getElementById('modalDueDate').textContent = dueDate;

document.getElementById('modalAmount').textContent = '₹' + parseFloat(amount).toLocaleString('en-IN', {minimumFractionDigits: 2});

});

// Handle modal hide event to ensure cleanup

paymentModal.addEventListener('hidden.bs.modal', function(event) {

// Reset form

const form = document.getElementById('paymentForm');

if (form) {

form.reset();

}

// Reset button

const submitBtn = form.querySelector('.btn-confirm');

if (submitBtn) {

submitBtn.innerHTML = '<i class="fas fa-check-circle me-2"></i>Confirm Payment';

submitBtn.disabled = false;

}

// Ensure backdrop is removed

const backdrops = document.querySelectorAll('.modal-backdrop');

backdrops.forEach(backdrop => backdrop.remove());

document.body.classList.remove('modal-open');

document.body.style.removeProperty('padding-right');

});

}

function initializeFormHandlers() {

// Payment form submission - Fixed to prevent freezing

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

e.preventDefault();

const submitBtn = this.querySelector('.btn-confirm');

const originalText = submitBtn.innerHTML;

const paymentMethod = document.getElementById('paymentMethod').value;

// Validate payment method

if (!paymentMethod) {

alert('Please select a payment method');

return;

}

// Disable button and show loading

submitBtn.innerHTML = '<i class="fas fa-spinner fa-spin me-2"></i>Processing...';

submitBtn.disabled = true;

// Simulate payment processing

setTimeout(() => {

try {

// Update the status in the table

updatePaymentStatusInTable(currentRepaymentId, 'COMPLETED');

// Reset button first

submitBtn.innerHTML = originalText;

submitBtn.disabled = false;

// Reset and close modal properly

this.reset();

const modalElement = document.getElementById('paymentModal');

const modal = bootstrap.Modal.getInstance(modalElement);

if (modal) {

modal.hide();

}

// Remove any lingering backdrop

setTimeout(() => {

const backdrops = document.querySelectorAll('.modal-backdrop');

backdrops.forEach(backdrop => backdrop.remove());

document.body.classList.remove('modal-open');

document.body.style.removeProperty('padding-right');

}, 300);

// Show success message

showSuccessMessage('Payment processed successfully!');

} catch (error) {

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

// Reset button on error

submitBtn.innerHTML = originalText;

submitBtn.disabled = false;

alert('Error processing payment. Please try again.');

}

}, 2000);

});

}

function updatePaymentStatusInTable(repaymentId, newStatus) {

const row = document.querySelector(`[data-repayment-id="${repaymentId}"]`);

if (row) {

// Update status badge

const statusCell = row.querySelector('.status-badge');

if (statusCell) {

statusCell.classList.remove('status-pending', 'status-overdue', 'status-paid', 'status-completed');

statusCell.classList.add('status-completed');

statusCell.innerHTML = '<i class="fas fa-check-circle me-1"></i>COMPLETED';

statusCell.setAttribute('data-status', newStatus);

}

// Update action cell

const actionCell = row.querySelector('td:last-child');

if (actionCell) {

actionCell.innerHTML = '<span class="text-muted"><i class="fas fa-check-circle me-1"></i>Paid</span>';

}

// Update payment date (current date)

const paymentDateCell = row.querySelector('td:nth-child(7)');

if (paymentDateCell) {

const today = new Date().toLocaleDateString('en-GB');

paymentDateCell.innerHTML = `<strong>${today}</strong>`;

}

}

}

function showSuccessMessage(message) {

// Remove any existing success messages first

const existingAlerts = document.querySelectorAll('.alert-success');

existingAlerts.forEach(alert => alert.remove());

// Create and show success alert

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

alertDiv.className = 'alert alert-success';

alertDiv.innerHTML = `<i class="fas fa-check-circle"></i> ${message}`;

alertDiv.style.opacity = '0';

alertDiv.style.transition = 'opacity 0.3s ease-in-out';

// Insert after dashboard header

const header = document.querySelector('.dashboard-header');

header.parentNode.insertBefore(alertDiv, header.nextSibling);

// Fade in

setTimeout(() => {

alertDiv.style.opacity = '1';

}, 100);

// Auto-hide after 5 seconds with fade out

setTimeout(() => {

alertDiv.style.opacity = '0';

setTimeout(() => {

if (alertDiv.parentNode) {

alertDiv.remove();

}

}, 300);

}, 5000);

}

function applyFilters() {

let rows = [...allRows];

// 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() {

// 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);

}

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

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

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

// Clear existing pagination

paginationContainer.innerHTML = '';

if (totalPages <= 1) {

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

paginationContainer.appendChild(singleBtn);

} else {

// First page button

const firstBtn = createBootstrapPaginationButton('«', 1, currentPage === 1);

paginationContainer.appendChild(firstBtn);

// Previous page button

const prevBtn = createBootstrapPaginationButton('‹', 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('›', currentPage + 1, currentPage === totalPages);

paginationContainer.appendChild(nextBtn);

// Last page button

const lastBtn = createBootstrapPaginationButton('»', 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 searchRepayments() {

currentPage = 1;

applyFilters();

updateTable();

}

function refreshData() {

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

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

// Clear search

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

// Reset to original data

currentPage = 1;

applyFilters();

setTimeout(() => {

updateTable();

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

}, 800);

}

</script>  
  
  
**repayment.css**

/\* Loan Repayment Management Styling - Based on Reference Design \*/

:root {

--primary: #2c3e50;

--secondary: #3498db;

--accent: #16a085;

--light: #f8f9fa;

--dark: #343a40;

--success: #27ae60;

--warning: #f39c12;

--danger: #e74c3c;

--info: #1abc9c;

--table-header-bg: linear-gradient(to right, #2c3e50, #3498db);

--pagination-border: #495057;

--pagination-bg: #f8f9fa;

--pagination-active-bg: #3498db;

}

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 h2 {

margin: 0;

font-size: 1.8rem;

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 p {

margin: 10px 0 0 0;

opacity: 0.9;

font-size: 1.1rem;

}

/\* Alert Messages \*/

.alert {

padding: 15px 20px;

margin: 20px auto;

border-radius: 12px;

font-weight: 500;

display: flex;

align-items: center;

gap: 10px;

backdrop-filter: blur(10px);

max-width: 96%;

border: none;

}

.alert-success {

background: linear-gradient(135deg, rgba(39, 174, 96, 0.9), rgba(26, 188, 156, 0.9));

color: white;

box-shadow: 0 10px 30px rgba(39, 174, 96, 0.3);

}

.alert-danger {

background: linear-gradient(135deg, rgba(231, 76, 60, 0.9), rgba(192, 57, 43, 0.9));

color: white;

box-shadow: 0 10px 30px rgba(231, 76, 60, 0.3);

}

.alert-info {

background: linear-gradient(135deg, rgba(26, 188, 156, 0.9), rgba(52, 152, 219, 0.9));

color: white;

box-shadow: 0 10px 30px rgba(26, 188, 156, 0.3);

border-radius: 12px;

padding: 20px;

margin: 20px auto;

}

/\* Outstanding Balance Section \*/

/\* Outstanding Balance Section - Compact \*/

.balance-section {

background-color: white;

border-radius: 12px;

padding: 25px;

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

margin: 20px auto;

max-width: 600px; /\* Changed from 96% to 600px for compact size \*/

width: 90%; /\* Added width constraint \*/

}

.balance-section h3 {

color: var(--primary);

font-weight: 600;

margin-bottom: 20px;

display: flex;

align-items: center;

gap: 10px;

}

.balance-form {

display: flex;

gap: 15px;

align-items: end;

flex-wrap: wrap;

justify-content: center; /\* Center the form elements \*/

}

.balance-form .form-group {

flex: 0 0 auto; /\* Changed from flex: 1 to fixed width \*/

min-width: 200px;

max-width: 250px; /\* Added max-width constraint \*/

}

.balance-form label {

color: var(--primary);

font-weight: 600;

margin-bottom: 8px;

display: flex;

align-items: center;

gap: 8px;

}

.balance-form .form-control {

border: 2px solid #e0e0e0;

border-radius: 8px;

padding: 12px 15px;

transition: all 0.3s;

font-size: 0.95rem;

}

.balance-form .form-control:focus {

border-color: var(--secondary);

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

outline: none;

}

.btn-balance {

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

color: white;

border: none;

border-radius: 8px;

padding: 12px 25px;

font-weight: 600;

transition: all 0.3s;

box-shadow: 0 4px 6px rgba(26, 188, 156, 0.2);

display: flex;

align-items: center;

gap: 8px;

white-space: nowrap;

}

.btn-balance:hover {

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

transform: translateY(-2px);

color: white;

box-shadow: 0 6px 12px rgba(26, 188, 156, 0.3);

}

/\* 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);

}

/\* Table Container - Optimized for visibility \*/

.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 wrapper \*/

.table-wrapper {

flex: 1;

min-height: 400px;

display: flex;

flex-direction: column;

padding: 20px;

overflow-x: auto;

}

/\* Table Styling - Improved visibility \*/

.table {

margin-bottom: 0;

border-collapse: separate;

border-spacing: 0;

width: 100%;

flex: 1;

font-size: 0.9rem;

table-layout: auto;

}

.table thead {

background: var(--table-header-bg);

}

.table thead th {

background: var(--table-header-bg);

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: 12px 10px;

text-transform: uppercase;

font-size: 0.8rem;

letter-spacing: 0.5px;

vertical-align: middle;

position: relative;

text-align: left;

white-space: nowrap;

}

.table thead th:first-child {

border-left: none;

}

.table thead th:last-child {

border-right: none;

}

.table thead th i {

margin-right: 6px;

font-size: 0.9rem;

}

.table tbody td {

padding: 12px 10px;

border: 1px solid #e0e0e0;

border-top: none;

vertical-align: middle;

color: #2c3e50;

font-weight: 500;

font-size: 0.85rem;

white-space: normal;

word-break: normal;

}

.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, 0.08);

transition: background-color 0.2s ease;

}

.table tbody tr:last-child td {

border-bottom: none;

}

.table-striped > tbody > tr:nth-of-type(odd) > \* {

background-color: rgba(248, 249, 250, 0.8);

}

/\* Status Badge - Compact and Optimized \*/

.status-badge {

padding: 5px 10px;

border-radius: 20px;

font-weight: 600;

display: inline-flex;

align-items: center;

gap: 4px;

font-size: 0.75rem;

text-align: center;

min-width: 80px;

justify-content: center;

text-transform: uppercase;

letter-spacing: 0.5px;

white-space: nowrap;

}

.status-paid {

background: linear-gradient(135deg, #27ae60, #2ecc71);

color: white;

box-shadow: 0 2px 5px rgba(39, 174, 96, 0.3);

}

.status-pending {

background: linear-gradient(135deg, #f39c12, #e67e22);

color: white;

box-shadow: 0 2px 5px rgba(243, 156, 18, 0.3);

}

.status-overdue {

background: linear-gradient(135deg, #e74c3c, #c0392b);

color: white;

box-shadow: 0 2px 5px rgba(231, 76, 60, 0.3);

}

.status-completed {

background: linear-gradient(135deg, #27ae60, #2ecc71);

color: white;

box-shadow: 0 2px 5px rgba(39, 174, 96, 0.3);

}

/\* Amount Styling \*/

.amount {

font-weight: 700;

color: var(--success);

font-size: 0.9rem;

white-space: nowrap;

}

/\* Interest Rate Styling \*/

.rate {

font-weight: 600;

color: var(--warning);

background: rgba(243, 156, 18, 0.1);

padding: 4px 8px;

border-radius: 4px;

white-space: nowrap;

}

/\* Tenure Styling \*/

.tenure {

font-weight: 500;

color: var(--info);

white-space: nowrap;

}

/\* Action Buttons \*/

.btn-schedule {

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

color: white;

border: none;

border-radius: 6px;

padding: 6px 12px;

font-size: 0.75rem;

font-weight: 600;

transition: all 0.3s;

cursor: pointer;

display: inline-flex;

align-items: center;

gap: 4px;

text-decoration: none;

box-shadow: 0 2px 5px rgba(52, 152, 219, 0.3);

text-transform: uppercase;

letter-spacing: 0.5px;

white-space: nowrap;

}

.btn-schedule:hover {

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

color: white;

transform: translateY(-2px);

box-shadow: 0 3px 8px rgba(52, 152, 219, 0.4);

}

.btn-schedule:disabled {

background: #6c757d;

cursor: not-allowed;

transform: none;

box-shadow: 0 1px 3px rgba(108, 117, 125, 0.3);

}

.btn-pay {

background: linear-gradient(135deg, var(--success), #1e8449);

color: white;

border: none;

border-radius: 6px;

padding: 6px 12px;

font-size: 0.75rem;

font-weight: 600;

transition: all 0.3s;

cursor: pointer;

display: inline-flex;

align-items: center;

gap: 4px;

text-decoration: none;

box-shadow: 0 2px 5px rgba(39, 174, 96, 0.3);

text-transform: uppercase;

letter-spacing: 0.5px;

white-space: nowrap;

}

.btn-pay:hover {

background: linear-gradient(135deg, #1e8449, #196f3d);

color: white;

transform: translateY(-2px);

box-shadow: 0 3px 8px rgba(39, 174, 96, 0.4);

}

.text-muted {

color: #6c757d !important;

font-style: italic;

font-weight: 500;

}

/\* Modal Styling - Enhanced \*/

.modal {

display: flex;

align-items: center;

justify-content: center;

}

.modal-dialog {

margin: 0;

max-width: 600px;

width: 100%;

}

.modal-dialog-centered {

display: flex;

align-items: center;

min-height: calc(100% - 1rem);

}

.modal-content {

border-radius: 15px;

border: none;

box-shadow: 0 15px 35px rgba(0, 0, 0, 0.1);

overflow: hidden;

}

.modal-header {

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

color: white;

border-bottom: none;

padding: 25px 30px;

}

.modal-title {

font-weight: 600;

letter-spacing: 0.5px;

font-size: 1.4rem;

display: flex;

align-items: center;

gap: 12px;

}

.modal-body {

padding: 30px;

background-color: #fafafa;

}

/\* Payment Modal Specific Styling \*/

.payment-details {

background: white;

border-radius: 10px;

padding: 20px;

margin-bottom: 20px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);

}

.payment-details h6 {

color: var(--primary);

font-weight: 600;

margin-bottom: 15px;

padding-bottom: 8px;

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

}

.payment-info {

display: grid;

grid-template-columns: 1fr 1fr;

gap: 15px;

margin-bottom: 20px;

}

.payment-info-item {

display: flex;

justify-content: space-between;

align-items: center;

padding: 10px 0;

border-bottom: 1px solid #e0e0e0;

}

.payment-info-label {

font-weight: 600;

color: var(--primary);

display: flex;

align-items: center;

gap: 8px;

}

.payment-info-value {

font-weight: 500;

color: var(--dark);

}

/\* Form Styling - Enhanced \*/

.form-group {

margin-bottom: 25px;

position: relative;

}

.form-group label {

font-weight: 600;

color: var(--primary);

margin-bottom: 10px;

display: flex;

align-items: center;

gap: 10px;

font-size: 1rem;

}

.form-control,

.form-select {

border: 2px solid #e0e0e0;

border-radius: 8px;

padding: 15px 18px;

transition: all 0.3s;

font-size: 1rem;

width: 100%;

box-sizing: border-box;

color: var(--dark);

font-weight: 500;

}

.form-control:focus,

.form-select:focus {

border-color: var(--secondary);

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

outline: none;

}

/\* Form Buttons - Enhanced \*/

.btn-confirm {

background: linear-gradient(135deg, var(--success), #1e8449);

color: white;

border: none;

border-radius: 8px;

padding: 15px 30px;

font-weight: 600;

transition: all 0.3s;

cursor: pointer;

display: inline-flex;

align-items: center;

gap: 10px;

font-size: 1rem;

text-transform: uppercase;

letter-spacing: 0.5px;

width: 100%;

justify-content: center;

}

.btn-confirm:hover {

background: linear-gradient(135deg, #1e8449, #196f3d);

color: white;

transform: translateY(-2px);

box-shadow: 0 5px 15px rgba(39, 174, 96, 0.3);

}

.btn-secondary {

background-color: #6c757d;

color: white;

border: none;

border-radius: 8px;

padding: 15px 30px;

font-weight: 600;

transition: all 0.3s;

cursor: pointer;

display: inline-flex;

align-items: center;

gap: 10px;

font-size: 1rem;

text-transform: uppercase;

letter-spacing: 0.5px;

}

.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;

}

/\* Modal Footer - Enhanced \*/

.modal-footer {

padding: 25px 30px;

border-top: 1px solid #e0e0e0;

background-color: #f8f9fa;

display: flex;

justify-content: flex-end;

gap: 15px;

}

/\* Pagination section \*/

.pagination-section {

background-color: #f8f9fa;

border-top: 1px solid #e0e0e0;

padding: 15px 20px;

margin-top: auto;

}

.pagination {

margin: 8px 0 0 0;

gap: 8px;

justify-content: center;

}

.pagination-info {

color: var(--primary);

font-weight: 500;

font-size: 0.9rem;

display: flex;

align-items: center;

justify-content: center;

gap: 8px;

margin-bottom: 8px;

}

.pagination .page-item .page-link {

color: #2c3e50;

background-color: var(--pagination-bg);

border: 3px solid #2c3e50;

padding: 12px 18px;

font-weight: 700;

font-size: 1rem;

min-width: 55px;

min-height: 50px;

text-align: center;

display: flex;

align-items: center;

justify-content: center;

border-radius: 8px;

transition: all 0.3s ease;

text-decoration: none;

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

}

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

opacity: 0.5;

cursor: not-allowed;

background-color: #e9ecef;

border-color: #6c757d;

color: #495057;

transform: none;

box-shadow: 0 1px 2px rgba(0, 0, 0, 0.05);

}

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

background-color: #e9ecef;

border-color: #6c757d;

color: #495057;

transform: none;

box-shadow: 0 1px 2px rgba(0, 0, 0, 0.05);

}

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

background-color: var(--pagination-active-bg);

color: white;

border-color: var(--secondary);

transform: translateY(-3px);

box-shadow: 0 4px 8px rgba(52, 152, 219, 0.4);

font-weight: 800;

}

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

background-color: var(--secondary);

color: white;

border-color: var(--primary);

}

/\* Form styling for inline forms \*/

.schedule-form {

margin: 0;

display: inline-block;

}

/\* Responsive Design \*/

@media (max-width: 1200px) {

.table {

font-size: 0.85rem;

}

.table thead th {

padding: 10px 8px;

font-size: 0.75rem;

}

.table tbody td {

padding: 10px 8px;

}

.payment-info {

grid-template-columns: 1fr;

}

}

@media (max-width: 992px) {

.table-wrapper {

overflow-x: auto;

}

.table {

min-width: 900px;

}

}

@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 {

max-width: 98%;

}

.dashboard-header {

margin: 15px auto;

padding: 20px;

max-width: 98%;

}

.dashboard-header h1,

.dashboard-header h2 {

font-size: 1.5rem;

}

.balance-form {

flex-direction: column;

gap: 15px;

align-items: center; /\* Center align for mobile \*/

}

.balance-form .form-group {

min-width: 100%;

max-width: 100%;

}

.btn-balance {

width: 100%; /\* Full width on mobile \*/

}

.refresh-container {

max-width: 98%;

}

.pagination {

gap: 4px;

}

.pagination .page-item .page-link {

padding: 8px 12px;

min-width: 40px;

min-height: 40px;

font-size: 0.9rem;

}

/\* Modal adjustments for mobile \*/

.modal-dialog {

max-width: 95vw;

}

.modal-body {

padding: 20px;

}

.form-control,

.form-select {

padding: 12px 15px;

font-size: 0.95rem;

}

.btn-confirm,

.btn-secondary {

padding: 12px 25px;

font-size: 0.95rem;

}

}

@media (max-width: 576px) {

.dashboard-header h1,

.dashboard-header h2 {

font-size: 1.3rem;

flex-direction: column;

gap: 10px;

}

.dashboard-header p {

font-size: 1rem;

}

.table thead th {

padding: 8px 6px;

font-size: 0.7rem;

}

.table tbody td {

padding: 8px 6px;

font-size: 0.75rem;

}

.btn-schedule,

.btn-pay {

font-size: 0.65rem;

padding: 4px 8px;

}

.pagination .page-item .page-link {

padding: 6px 10px;

min-width: 35px;

min-height: 35px;

font-size: 0.8rem;

}

.status-badge {

font-size: 0.65rem;

padding: 4px 8px;

min-width: 70px;

}

}

/\* Loading states \*/

.loading {

opacity: 0.6;

pointer-events: none;

}

/\* Smooth transitions \*/

\* {

box-sizing: border-box;

}

/\* Utility classes \*/

.d-flex {

display: flex;

}

.gap-2 {

gap: 0.5rem;

}

.mt-auto {

margin-top: auto;

}

.me-1 {

margin-right: 0.25rem;

}

.me-2 {

margin-right: 0.5rem;

}

.me-3 {

margin-right: 1rem;

}

.mb-3 {

margin-bottom: 1rem;

}

.w-50 {

width: 50%;

}

.text-center {

text-align: center;

}

.justify-content-end {

justify-content: flex-end;

}

.visually-hidden {

position: absolute !important;

width: 1px !important;

height: 1px !important;

padding: 0 !important;

margin: -1px !important;

overflow: hidden !important;

clip: rect(0, 0, 0, 0) !important;

white-space: nowrap !important;

border: 0 !important;

}