

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" />
```

```
<!-- 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>
```

```
<!-- 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">2</option>
```

```
        <option value="5">5</option>
        <option value="10">10</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">Customer ID</th>
                    <th scope="col">Name</th>
                    <th scope="col">Loan ID</th>
                    <th scope="col">Product Type</th>
                    <th scope="col">Product Name</th>
                    <th scope="col">Interest Rate (%)</th>
```

```

        <th scope="col">Loan Amount</th>

        <th scope="col">Tenure (Months)</th>

        <th scope="col">Action</th>

    </tr>

</thead>

<tbody id="table-body">

    @foreach (var loan in Model)
    {
        <tr class="loan-row" data-loan-id="@loan.LoanId">

            <td class="text-center"><strong>@loan.CustomerId</strong></td>

            <td class="text-center"><strong>@loan.Name</strong></td>

            <td class="text-center"><strong>@loan.LoanId</strong></td>

            <td class="text-center"><strong>@loan.ProductType</strong></td>

            <td class="text-center"><strong>@loan.ProductName</strong></td>

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

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

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

            <td class="text-center">

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

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

                    <!-- Hidden inputs to preserve pagination state -->

                    <input type="hidden" name="currentPage" class="current-page-input"
value="1" />

                    <input type="hidden" name="rowsPerPage" class="rows-per-page-input"
value="2" />

```

```
        <input type="hidden" name="searchTerm" class="search-term-input"
value="" />
```

```
        <button type="submit" class="btn btn-schedule" data-application-
id="@loan.ApplicationId">
```

```
            <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 = 2;
```

```
    let allRows = [];
```

```
    let filteredRows = [];
```

```
    let searchTerm = "";
```

```
    // Initialize on page load
```

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

```
        restorePaginationState();
```

```
        initializeTable();
```

```
        initializeFormHandlers();
```

```
    });
```

```
    function restorePaginationState() {
```

```
        // Restore pagination state from URL parameters or localStorage
```

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

```
        const savedPage = urlParams.get('currentPage') || localStorage.getItem('admin-currentPage') || '1';
```

```
        const savedRowsPerPage = urlParams.get('rowsPerPage') || localStorage.getItem('admin-rowsPerPage') || '2';
```

```
        const savedSearchTerm = urlParams.get('searchTerm') || localStorage.getItem('admin-searchTerm') || "";
```

```
        currentPage = parseInt(savedPage);
```

```
        rowsPerPage = parseInt(savedRowsPerPage);
```

```
        searchTerm = savedSearchTerm;
```

```

// Update UI elements

document.getElementById('rowsPerPage').value = rowsPerPage;
document.getElementById('searchInput').value = searchTerm;
}

function savePaginationState() {
    // Save current state to localStorage

    localStorage.setItem('admin-currentPage', currentPage);
    localStorage.setItem('admin-rowsPerPage', rowsPerPage);
    localStorage.setItem('admin-searchTerm', searchTerm);
}

function initializeTable() {
    allRows = Array.from(document.querySelectorAll('.loan-row'));
    filteredRows = [...allRows];
    applyFilters();
    updateTable();
}

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

    forms.forEach(form => {
        form.addEventListener('submit', function(e) {
            const button = form.querySelector('button[type="submit"]');
            const applicationId = button.getAttribute('data-application-id');

            if (button && !button.disabled) {

```

```

// Save current pagination state before submission
savePaginationState();

// Update hidden form inputs with current state
form.querySelector('.current-page-input').value = currentPage;
form.querySelector('.rows-per-page-input').value = rowsPerPage;
form.querySelector('.search-term-input').value = searchTerm;

// Disable button and change text
button.disabled = true;

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

// Store the scheduled state in localStorage for this specific application
localStorage.setItem(`scheduled-${applicationId}`, 'true');
}
});
});

// Check for previously scheduled items and update their button states
forms.forEach(form => {
  const button = form.querySelector('button[type="submit"]');
  const applicationId = button.getAttribute('data-application-id');

  if (localStorage.getItem(`scheduled-${applicationId}`) === 'true') {
    button.disabled = true;

    button.innerHTML = '<i class="fas fa-check-circle me-1"></i>Done';
    button.classList.add('btn-scheduled');
  }
});

```

```
    }  
  });  
}
```

```
function applyFilters() {  
  let rows = [...allRows];  
  
  // Apply search filter  
  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;  
  
  // Ensure currentPage is within valid range
```



```
    if (currentPage > totalPages) {  
        currentPage = totalPages;  
    }  
    if (currentPage < 1) {  
        currentPage = 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  
    savePaginationState();  
}  
  
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();
}

// Handle browser back/forward buttons
window.addEventListener('popstate', function(event) {
    restorePaginationState();
    applyFilters();
    updateTable();
});
</script>
}

```

Customer.cshtml

```

@model List<BankLoanProject.Models.RepaymentScheduleViewModel>

@{
    ViewBag.Title = "Customer Repayment Portal";
}

<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-credit-card me-3"></i>
```

```
    Loan Repayment Portal
```

```
  </h2>
```

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

```
</div>
```

```
<h3 class="mt-5">Check Outstanding Balance</h3>
```

```
<form asp-action="GetOutstandingBalance" asp-controller="Repayment"
method="get" class="row g-3">
```

```
  <div class="col-auto">
```

```
    <label for="loanId" class="visually-hidden">Loan ID</label>
```

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

```
  </div>
```

```
  <div class="col-auto">
```

```
    <button type="submit" class="btn btn-info mb-3">Get Balance</button>
```

```
  </div>
```

```
</form>
```

```
@if (ViewBag.OutstandingBalance != null)
```

```
{
```

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

```
<strong>Loan ID:</strong> @ViewBag.LoanId<br />
<strong>Outstanding Balance:</strong> @((decimal)ViewBag.OutstandingBalance)
</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">2</option>
      <option value="5">5</option>
      <option value="10">10</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>
```

```
<!-- Repayment Schedule Table -->
```

```
<div class="table-container">
```

```
  <div class="table-wrapper">
```

```
    <table class="table table-bordered">
```

```
      <thead>
```

```
        <tr>
```

```
          <th scope="col" class="col-repayment-id">Repayment ID</th>
```

```
          <th scope="col" class="col-loan-id">Loan ID</th>
```

```
          <th scope="col" class="col-product-type">Product Type</th>
```

```
          <th scope="col" class="col-product-name">Product Name</th>
```

```
          <th scope="col" class="col-due-date">Due Date</th>
```

```
          <th scope="col" class="col-amount-due">Amount Due</th>
```

```
          <th scope="col" class="col-payment-date">Payment Date</th>
```

```
          <th scope="col" class="col-status">Status</th>
```

```
          <th scope="col" class="col-action">Action</th>
```

```
        </tr>
```

```
      </thead>
```

```
      <tbody id="table-body">
```

```
        @foreach (var item in Model)
```

```
        {
```

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

```
            <td class="text-center">
```

```
              <span class="repayment-id">@item.RepaymentId</span>
```

```
            </td>
```

```
            <td class="text-center">
```

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



```

</td>

<td class="text-center">@item.ProductType</td>

<td class="text-center">@item.ProductName</td>

<td class="text-center">@item.DueDate.ToString("dd-MM-yyyy")</td>

<td class="text-center">

    <span class="amount">₹@item.AmountDue?.ToString("N2") ??
"0.00")</span>

</td>

<td class="text-center">

    @if (item.PaymentDate.HasValue)

    {

        @item.PaymentDate.Value.ToString("dd-MM-yyyy")

    }

    else

    {

        <span class="text-muted">-</span>

    }

</td>

<td class="text-center">

    @{

        string statusClass = "";

        string statusIcon = "";

        switch (item.PaymentStatus?.ToUpper())

        {

            case "PAID":

            case "COMPLETED":

                statusClass = "status-completed";

                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">
    <i class="@statusIcon"></i>@item.PaymentStatus
</span>
</td>

<td class="text-center">
    @if (item.PaymentStatus != "PAID" && item.PaymentStatus !=
"COMPLETED")
    {
        <button class="btn-pay" data-bs-toggle="modal" data-bs-
target="#paymentModal"

            data-repayment-id="@item.RepaymentId"
            data-amount="@item.AmountDue"
            data-loan-id="@item.LoanId"
            data-due-date="@item.DueDate.ToString("dd-MM-yyyy")"

```

```

        data-product-type="@item.ProductType"
        data-product-name="@item.ProductName">
        <i class="fas fa-credit-card"></i>
        <span>Pay Now</span>
    </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-lg 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>

                    Make 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-1"></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-1"></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-1"></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-1"></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-1"></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-1"></i>Amount Due:

    </span>

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

    <!-- Hidden inputs to preserve pagination state -->

    <input type="hidden" name="currentPage" class="current-page-input"
value="1" />

    <input type="hidden" name="rowsPerPage" class="rows-per-page-input"
value="2" />

    <input type="hidden" name="searchTerm" class="search-term-input" value=""
/>

    <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">

```

```

        <i class="fas fa-mobile-alt"></i> UPI Payment
    </option>
    <option value="NetBanking">
        <i class="fas fa-university"></i> Net Banking
    </option>
    <option value="Card">
        <i class="fas fa-credit-card"></i> Credit/Debit Card
    </option>
</select>

</div>

</form>

</div>

<div class="modal-footer">
    <button type="button" class="btn-secondary" data-bs-dismiss="modal">
        <i class="fas fa-times me-2"></i>Cancel
    </button>
    <button type="submit" form="paymentForm" class="btn-confirm">
        <i class="fas fa-check-circle me-2"></i>Confirm Payment
    </button>
</div>

</div>

</div>

</div>

```

```
@section Scripts {
```

```
<script>
```

```
// Global variables for pagination and search
```

```
let currentPage = 1;
```

```

let rowsPerPage = 2;

let allRows = [];

let filteredRows = [];

let searchTerm = "";


// Initialize on page load

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

    restorePaginationState();

    initializeTable();

    initializeModal();

    initializePaymentFormHandlers();

});


function restorePaginationState() {

    // Restore pagination state from URL parameters or localStorage

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

    const savedPage = urlParams.get('currentPage') ||
localStorage.getItem('customer-currentPage') || '1';

    const savedRowsPerPage = urlParams.get('rowsPerPage') ||
localStorage.getItem('customer-rowsPerPage') || '2';

    const savedSearchTerm = urlParams.get('searchTerm') ||
localStorage.getItem('customer-searchTerm') || "";

    currentPage = parseInt(savedPage);

    rowsPerPage = parseInt(savedRowsPerPage);

    searchTerm = savedSearchTerm;


    // Update UI elements

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

```



```
document.getElementById('searchInput').value = searchTerm;
}
```

```
function savePaginationState() {
    // Save current state to localStorage
    localStorage.setItem('customer-currentPage', currentPage);
    localStorage.setItem('customer-rowsPerPage', rowsPerPage);
    localStorage.setItem('customer-searchTerm', searchTerm);
}
```

```
function initializeTable() {
    allRows = Array.from(document.querySelectorAll('.repayment-row'));
    filteredRows = [...allRows];
    applyFilters();
    updateTable();
}
```

```
function initializeModal() {
    const paymentModal = document.getElementById('paymentModal');
    paymentModal.addEventListener('show.bs.modal', function (event) {
        const button = event.relatedTarget;
        const repaymentId = button.getAttribute('data-repayment-id');
        const amount = button.getAttribute('data-amount');
        const loanId = button.getAttribute('data-loan-id');
        const dueDate = button.getAttribute('data-due-date');
        const productType = button.getAttribute('data-product-type');
        const productName = button.getAttribute('data-product-name');
```

```

        // Update modal content

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

        document.getElementById('modalRepaymentId').textContent = '#' +
repaymentId;

        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});

    });
}

```

```

function initializePaymentFormHandlers() {

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

    if (paymentForm) {

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

            // Save current pagination state before submission

            savePaginationState();

            // Update hidden form inputs with current state

            paymentForm.querySelector('.current-page-input').value = currentPage;

            paymentForm.querySelector('.rows-per-page-input').value = rowsPerPage;

            paymentForm.querySelector('.search-term-input').value = searchTerm;

        });

    }

}

```

```
function applyFilters() {  
    let rows = [...allRows];  
  
    // Apply search filter  
    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;  
  
    // Ensure currentPage is within valid range  
    if (currentPage > totalPages) {  
        currentPage = totalPages;  
    }  
  
    if (currentPage < 1) {
```

```
    currentPage = 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
savePaginationState();
}

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();
    }

    // Handle browser back/forward buttons
    window.addEventListener('popstate', function(event) {
        restorePaginationState();
        applyFilters();
        updateTable();
    });
</script>
}

```

repayment.css

```

/* Loan Repayment Management Styling - Enhanced for Customer Portal */
: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 - Reduced height, increased width with margins */
.dashboard-header {
  background: linear-gradient(135deg, var(--primary), var(--secondary));
  color: white;
  border-radius: 15px;
  padding: 10px 25px; /* Reduced padding from 25px to 15px */
  margin: 70px auto 30px auto; /* Reduced margin from 25px to 15px */
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
  text-align: center;
  max-width: 92%; /* Increased from 96% to 92% for more width with margins */
}
```

```
.dashboard-header h1 {
  margin: 0;
  font-size: 1.6rem; /* Reduced from 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: 1.5rem; /* Reduced from 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: 8px 0 0 0; /* Reduced from 10px to 8px */
opacity: 0.9;
font-size: 1rem; /* Reduced from 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: 92%;
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;
```

```
}
```

```
/* Table Controls - Moved closer to header */
```

```
.table-controls {  
  background-color: white;  
  border-radius: 12px;  
  padding: 15px 20px; /* Reduced padding from 20px to 15px */  
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05);  
  margin: 10px auto; /* Reduced margin from 20px to 10px */  
  max-width: 92%; /* Match header width */  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  flex-wrap: wrap;  
  gap: 15px;  
  min-height: 60px; /* Reduced from 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 - Moved closer to controls */
```

```
.table-container {  
  background-color: white;  
  border-radius: 12px;  
  padding: 0;  
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05);  
  margin: 10px auto; /* Reduced margin from 20px to 10px */  
  max-width: 92%; /* Match header width */  
  border: 4px solid var(--primary);  
  display: flex;
```

```
flex-direction: column;

overflow: hidden;

}
```

```
/* Table wrapper - Compact height */
```

```
.table-wrapper {

  flex: 1;

  display: flex;

  flex-direction: column;

  padding: 0;

  overflow: visible;

}
```

```
/* Table Styling - All content centered */
```

```
.table {

  margin-bottom: 0;

  border-collapse: separate;

  border-spacing: 0;

  width: 100%;

  table-layout: auto;

  font-size: 0.85rem;

}
```

```
/* Responsive Column Widths - Percentage Based */
```

```
.col-repayment-id {

  width: 10%;

  min-width: 80px;

}
```

```
.col-loan-id {  
  width: 8%;  
  min-width: 70px;  
}
```

```
.col-product-type {  
  width: 12%;  
  min-width: 100px;  
}
```

```
.col-product-name {  
  width: 15%;  
  min-width: 120px;  
}
```

```
.col-due-date {  
  width: 10%;  
  min-width: 90px;  
}
```

```
.col-amount-due {  
  width: 12%;  
  min-width: 100px;  
}
```

```
.col-payment-date {  
  width: 10%;
```



```
    min-width: 90px;
}
```

```
.col-status {
    width: 12%;
    min-width: 100px;
}
```

```
.col-action {
    width: 11%;
    min-width: 90px;
}
```

```
/* Table Header - Compact Design */
```

```
.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: 8px 6px;
    text-transform: uppercase;
    font-size: 0.7rem;
    letter-spacing: 0.3px;
```

```
vertical-align: middle;
text-align: center;
white-space: nowrap;
line-height: 1.2;
}
```

```
.table thead th:first-child {
    border-left: none;
}
```

```
.table thead th:last-child {
    border-right: none;
}
```

```
.table thead th i {
    font-size: 0.8rem;
    display: block;
    margin-bottom: 2px;
}
```

```
.table tbody td {
    padding: 8px 6px;
    border: 1px solid #e0e0e0;
    border-top: none;
    vertical-align: middle;
    font-size: 0.8rem;
    word-wrap: break-word;
    overflow: hidden;
```

```
text-overflow: ellipsis;

line-height: 1.3;

text-align: center; /* Center all table content */
}
```

```
.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;

}
```

```
/* Repayment ID Styling */
```

```
.repayment-id {

    font-weight: 700;

    color: var(--primary);

    font-size: 0.85rem;

}
```

```
/* Status Badge - Updated for completed status */
```

```
.status-badge {  
    padding: 2px 6px;  
    border-radius: 12px;  
    font-weight: 500;  
    display: inline-flex;  
    align-items: center;  
    gap: 2px;  
    font-size: 0.65rem;  
    text-align: center;  
    min-width: 60px;  
    justify-content: center;  
    white-space: nowrap;  
}
```

```
.status-paid,  
.status-completed {  
    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-overdue {  
    background-color: rgba(231, 76, 60, 0.2);  
    color: var(--danger);  
    border: 1px solid var(--danger);  
}
```

```
/* Amount Styling */
```

```
.amount {  
    font-weight: 700;  
    color: var(--success);  
    font-size: 0.85rem;  
    white-space: nowrap;  
}
```

```
/* Action Buttons - Centered */
```

```
.btn-pay {  
    border-radius: 4px;  
    padding: 4px 6px;  
    font-size: 0.6rem;  
    font-weight: 600;  
    transition: all 0.3s;  
    border: 2px solid;  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    justify-content: center;  
    min-width: 35px;
```

```
height: 35px;
cursor: pointer;
box-shadow: 0 1px 3px rgba(0, 0, 0, 0.1);
gap: 1px;
background-color: #2ed573;
color: white;
border-color: #2ed573;
margin: 0 auto; /* Center the button */
}
```

```
.btn-pay i {
  font-size: 0.7rem;
  margin-bottom: 1px;
}
```

```
.btn-pay span {
  font-size: 0.55rem;
  font-weight: 600;
  text-transform: uppercase;
  letter-spacing: 0.2px;
}
```

```
.btn-pay:hover {
  background-color: #26d065;
  color: white;
  transform: translateY(-1px);
  box-shadow: 0 2px 6px rgba(46, 213, 115, 0.3);
  border-color: #26d065;
```

```
}
```

```
.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;  
    margin: 0 auto; /* Center the button */  
}
```

```
.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);  
}
```

```
/* New class for scheduled buttons */
```

```
.btn-scheduled {  
    background: linear-gradient(135deg, var(--success), #1e8449) !important;  
    color: white !important;  
}
```

```
.btn-scheduled:hover {  
    background: linear-gradient(135deg, #1e8449, #196f3d) !important;  
    color: white !important;  
}
```

```
.text-muted {  
    color: #6c757d !important;  
    font-style: italic;  
    font-weight: 500;  
}
```

```
/* Enhanced Payment Modal */
```

```
.modal-xl {
```



```
    max-width: 800px;
}
```

```
.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 30px;
}
```

```
.modal-title {
    font-weight: 600;
    letter-spacing: 0.5px;
    font-size: 1.4rem;
    display: flex;
    align-items: center;
    gap: 10px;
}
```

```
.modal-body {
    padding: 30px;
```

```
background-color: #fafafa;
}
```

```
/* Payment Details Section */
```

```
.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 */

```
.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,
.form-select {
    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,
.form-select:focus {
    border-color: var(--secondary);
    box-shadow: 0 0 0 0.25rem rgba(52, 152, 219, 0.25);
    outline: none;
}
```

```
/* Form Buttons */
```

```
.btn-confirm {
```

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

/* Modal Footer */

.modal-footer {

    padding: 20px 30px;
```

```
border-top: 1px solid #e0e0e0;
background-color: #f8f9fa;
display: flex;
justify-content: flex-end;
gap: 15px;
}
```

```
/* Pagination section - Minimal Gap */
```

```
.pagination-section {
  background-color: #f8f9fa;
  border-top: 1px solid #e0e0e0;
  padding: 8px 15px;
  margin-top: 0;
}
```

```
/* Enhanced Bootstrap Pagination Styling */
```

```
.pagination {
  margin: 4px 0 0 0;
  gap: 6px;
  justify-content: center;
}
```

```
/* Pagination Info */
```

```
.pagination-info {
  color: var(--primary);
  font-weight: 500;
  font-size: 0.85rem;
  display: flex;
```

```
align-items: center;
justify-content: center;
gap: 6px;
margin-bottom: 4px;
}
```

```
.pagination .page-item .page-link {
  color: #2c3e50;
  background-color: var(--pagination-bg);
  border: 3px solid #2c3e50;
  padding: 8px 12px;
  font-weight: 700;
  font-size: 0.9rem;
  min-width: 40px;
  min-height: 40px;
  text-align: center;
  display: flex;
  align-items: center;
  justify-content: center;
  border-radius: 6px;
  transition: all 0.3s ease;
  text-decoration: none;
  box-shadow: 0 1px 3px 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(-2px);
    box-shadow: 0 3px 6px 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: 0.8rem;
```

```
    }
```

```
        .table thead th {
```

```
            padding: 6px 4px;
```

```
            font-size: 0.65rem;
```

```
        }
```

```
        .table tbody td {
```

```
            padding: 6px 4px;
```

```
            font-size: 0.75rem;
```

```
        }
```

```
.payment-info {
```

```
    grid-template-columns: 1fr;
```

```
}
```

```
.modal-xl {  
    max-width: 95%;  
}  
}
```

```
@media (max-width: 768px) {  
    .table-controls {  
        flex-direction: column;  
        align-items: stretch;  
        gap: 15px;  
        min-height: 120px;  
    }  
}
```

```
.controls-right {  
    flex-direction: column;  
    gap: 10px;  
}
```

```
.search-input-group {  
    width: 100%;  
}
```

```
.table-container {  
    max-width: 98%;  
}
```

```
.dashboard-header {  
    margin: 15px auto;
```

```
padding: 15px; /* Reduced padding for mobile */  
max-width: 98%;  
}
```

```
.dashboard-header h1,  
.dashboard-header h2 {  
    font-size: 1.3rem; /* Reduced for mobile */  
}
```

```
.balance-check-container {  
    max-width: 98%;  
}
```

```
.balance-input-group {  
    flex-direction: column;  
    align-items: stretch;  
    gap: 10px;  
}
```

```
.balance-input {  
    min-width: 100%;  
    max-width: 100%;  
}
```

```
.btn-balance-check {  
    width: 100%;  
    justify-content: center;  
}
```

```
.pagination {  
  gap: 4px;  
}
```

```
.pagination .page-item .page-link {  
  padding: 6px 10px;  
  min-width: 35px;  
  min-height: 35px;  
  font-size: 0.8rem;  
}
```

```
.modal-xl {  
  max-width: 98%;  
}
```

```
.modal-body {  
  padding: 20px;  
}  
}
```

```
@media (max-width: 576px) {  
  .dashboard-header h1,  
  .dashboard-header h2 {  
    font-size: 1.2rem; /* Further reduced for small screens */  
    flex-direction: column;  
    gap: 10px;  
  }
```

```
.dashboard-header p {  
    font-size: 0.9rem;  
}
```

```
.table thead th {  
    padding: 4px 2px;  
    font-size: 0.6rem;  
}
```

```
.table tbody td {  
    padding: 4px 2px;  
    font-size: 0.7rem;  
}
```

```
.btn-pay {  
    font-size: 0.55rem;  
    padding: 2px 4px;  
    min-width: 30px;  
    height: 30px;  
}
```

```
.pagination .page-item .page-link {  
    padding: 4px 8px;  
    min-width: 30px;  
    min-height: 30px;  
    font-size: 0.75rem;  
}
```

```
}
```

```
/* Loading states */
```

```
.loading{  
    opacity: 0.6;  
    pointer-events: none;  
}
```

```
/* Smooth transitions */
```

```
*{  
    box-sizing: border-box;  
}
```

```
/* Utility classes */
```

```
.me-1 {  
    margin-right: 0.25rem;  
}
```

```
.me-2 {  
    margin-right: 0.5rem;  
}
```

```
.me-3 {  
    margin-right: 1rem;  
}
```

```
.text-center {  
    text-align: center;
```

```
}
```

```
.text-end {
```

```
  text-align: 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;
```

```
}
```