# ABCRetailers Web Application - Views Code Snapshots

# Views - Order View

## Index.cshtml

<!-- Views/Order/Index.cshtml -->

@model IEnumerable<ABCRetailers.Models.Order>

@{

ViewData["Title"] = "Orders";

}

<div class="d-flex justify-content-between align-items-center mb-4">

<h2><i class="fas fa-shopping-cart"></i> Orders</h2>

<**a** **asp-action**="Create" class="btn btn-primary">

<i class="fas fa-plus"></i> Add Order

</**a**>

</div>

@if (Model.Any())

{

<div class="table-responsive">

<table class="table table-striped table-hover">

<thead class="table-dark">

<tr>

<th>Order ID</th>

<th>Username</th>

<th>Product</th>

<th>Order Date</th>

<th>Quantity</th>

<th>Total Price</th>

<th>Status</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

@foreach (var order in Model.OrderByDescending(o => o.OrderDate))

{

<tr>

<td><code>@order.OrderId.Substring(0, 8)...</code></td>

<td>@order.Username</td>

<td>@order.ProductName</td>

<td>@order.OrderDate.ToString("MMM dd, yyyy")</td>

<td>@order.Quantity</td>

<td>@order.TotalPrice.ToString("C")</td>

<td>

<span class="badge bg-@(GetStatusBadgeClass(order.Status))">

@order.Status

</span>

</td>

<td>

<div class="btn-group" role="group">

<**a** **asp-action**="Details" **asp-route-id**="@order.RowKey" class="btn btn-sm btn-outline-info" title="Details">

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

</**a**>

<**a** **asp-action**="Edit" **asp-route-id**="@order.RowKey" class="btn btn-sm btn-outline-primary" title="Edit">

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

</**a**>

<!-- Status Update Dropdown -->

@if (order.Status != "Completed" && order.Status != "Cancelled")

{

<div class="btn-group" role="group">

<button type="button" class="btn btn-sm btn-outline-warning dropdown-toggle" data-bs-toggle="dropdown" title="Update Status">

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

</button>

<ul class="dropdown-menu">

@if (order.Status == "Submitted")

{

<li>

<a class="dropdown-item" href="#" onclick="updateOrderStatus('@order.RowKey', 'Processing')">

<i class="fas fa-cog text-info"></i> Mark as Processing

</a>

</li>

}

@if (order.Status == "Processing")

{

<li>

<a class="dropdown-item" href="#" onclick="updateOrderStatus('@order.RowKey', 'Completed')">

<i class="fas fa-check text-success"></i> Mark as Completed

</a>

</li>

}

<li>

<a class="dropdown-item" href="#" onclick="updateOrderStatus('@order.RowKey', 'Cancelled')">

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

</a>

</li>

</ul>

</div>

}

<button type="button" class="btn btn-sm btn-outline-danger" onclick="confirmDelete('@order.RowKey', '@order.OrderId')" title="Delete">

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

</button>

</div>

</td>

</tr>

}

</tbody>

</table>

</div>

}

else

{

<div class="alert alert-info">

<i class="fas fa-info-circle"></i> No orders found.

<**a** **asp-action**="Create" class="alert-link">Create your first order</**a**>!

</div>

}

<!-- Delete Confirmation Modal -->

<div class="modal fade" id="deleteModal" tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Confirm Delete</h5>

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

</div>

<div class="modal-body">

Are you sure you want to delete order <strong id="orderId"></strong>?

</div>

<div class="modal-footer">

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

<**form** id="deleteForm" method="post" style="display: inline;">

<button type="submit" class="btn btn-danger">Delete</button>

</**form**>

</div>

</div>

</div>

</div>

@{

string GetStatusBadgeClass(string status) => status?.ToLower() switch

{

"submitted" => "primary",

"processing" => "info",

"completed" => "success",

"cancelled" => "danger",

\_ => "secondary"

};

}

@section Scripts {

<script>

function confirmDelete(orderRowKey, orderId) {

document.getElementById('orderId').textContent = orderId.substring(0, 8) + '...';

document.getElementById('deleteForm').action = '/Order/Delete/' + orderRowKey;

new bootstrap.Modal(document.getElementById('deleteModal')).show();

}

function updateOrderStatus(orderRowKey, newStatus) {

if (confirm(`Are you sure you want to change the status to "${newStatus}"?`)) {

fetch('/Order/UpdateOrderStatus', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

id: orderRowKey,

newStatus: newStatus

})

})

.then(response => response.json())

.then(data => {

if (data.success) {

alert('Status updated successfully!');

location.reload();

} else {

alert('Error: ' + data.message);

}

})

.catch(error => {

alert('An error occurred while updating the status');

});

}

}

</script>

}

## Create.cshtml

<!-- Views/Order/Create.cshtml -->

@model ABCRetailers.Models.ViewModels.OrderCreateViewModel

@{

ViewData["Title"] = "Create Order";

}

<h2><i class="fas fa-shopping-cart"></i> Add Order</h2>

<div class="row">

<div class="col-md-8">

<**form** **asp-action**="Create" method="post">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<div class="form-group mb-3">

<**label** **asp-for**="CustomerId" class="form-label">Select Customer</**label**>

<**select** **asp-for**="CustomerId" class="form-select" id="customerSelect">

<**option** **value**="">-- Select Customer --</**option**>

@foreach (var customer in Model.Customers)

{

<**option** **value**="@customer.RowKey">@customer.Name @customer.Surname (@customer.Username)</**option**>

}

</**select**>

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

</div>

<div class="form-group mb-3">

<**label** **asp-for**="ProductId" class="form-label">Select Product</**label**>

<**select** **asp-for**="ProductId" class="form-select" id="productSelect">

<**option** **value**="">-- Select Product --</**option**>

@foreach (var product in Model.Products)

{

<**option** **value**="@product.RowKey" data-price="@product.Price" data-stock="@product.StockAvailable">

@product.ProductName - @product.Price.ToString("C") (Stock: @product.StockAvailable)

</**option**>

}

</**select**>

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

</div>

<div class="row">

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

<div class="form-group mb-3">

<**label** **asp-for**="OrderDate" class="form-label">Order Date</**label**>

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

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

</div>

</div>

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

<div class="form-group mb-3">

<**label** **asp-for**="Quantity" class="form-label">Quantity</**label**>

<**input** **asp-for**="Quantity" class="form-control" **type**="number" min="1" id="quantityInput" />

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

<div id="stockWarning" class="form-text text-warning" style="display: none;"></div>

</div>

</div>

</div>

<div class="row">

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

<div class="form-group mb-3">

<label class="form-label">Unit Price</label>

<input type="text" class="form-control" id="unitPriceDisplay" readonly placeholder="Select product" />

</div>

</div>

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

<div class="form-group mb-3">

<label class="form-label">Total Price</label>

<input type="text" class="form-control" id="totalPriceDisplay" readonly placeholder="$0.00" />

</div>

</div>

</div>

<div class="form-group mb-3">

<**label** **asp-for**="Status" class="form-label">Status</**label**>

<**select** **asp-for**="Status" class="form-select" disabled>

<**option** **value**="Submitted" selected>Submitted (Default)</**option**>

</**select**>

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

<div class="form-text">Status will be set to "Submitted" when order is created</div>

</div>

<div class="form-group">

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

<i class="fas fa-save"></i> Create Order

</button>

<**a** **asp-action**="Index" class="btn btn-secondary">

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

</**a**>

</div>

</**form**>

</div>

<div class="col-md-4">

<div class="card">

<div class="card-header">

<h6 class="card-title mb-0"><i class="fas fa-info-circle"></i> Order Summary</h6>

</div>

<div class="card-body">

<div id="orderSummary">

<p class="text-muted">Select customer and product to see order summary</p>

</div>

</div>

</div>

</div>

</div>

@section Scripts {

@{

await Html.RenderPartialAsync("\_ValidationScriptsPartial");

}

<script>

let currentPrice = 0;

let availableStock = 0;

let selectedProductName = '';

document.getElementById('productSelect').addEventListener('change', function() {

const selectedOption = this.options[this.selectedIndex];

if (selectedOption.value) {

currentPrice = parseFloat(selectedOption.dataset.price);

availableStock = parseInt(selectedOption.dataset.stock);

selectedProductName = selectedOption.text.split(' - ')[0];

document.getElementById('unitPriceDisplay').value = '$' + currentPrice.toFixed(2);

document.getElementById('quantityInput').max = availableStock;

const currentQuantity = parseInt(document.getElementById('quantityInput').value) || 0;

if (currentQuantity > 0) {

updateTotalPrice();

}

updateOrderSummary();

validateQuantity();

} else {

currentPrice = 0;

availableStock = 0;

selectedProductName = '';

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

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

updateOrderSummary();

}

});

document.getElementById('quantityInput').addEventListener('input', function() {

validateQuantity();

updateTotalPrice();

updateOrderSummary();

});

document.getElementById('customerSelect').addEventListener('change', updateOrderSummary);

function validateQuantity() {

const quantity = parseInt(document.getElementById('quantityInput').value) || 0;

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

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

if (quantity > availableStock && availableStock > 0) {

stockWarning.textContent = `⚠️ Warning: Only ${availableStock} items in stock!`;

stockWarning.style.display = 'block';

stockWarning.className = 'form-text text-danger';

submitBtn.disabled = true;

} else if (quantity > 0 && quantity <= availableStock) {

stockWarning.textContent = `✅ ${quantity} items available`;

stockWarning.style.display = 'block';

stockWarning.className = 'form-text text-success';

submitBtn.disabled = false;

} else {

stockWarning.style.display = 'none';

submitBtn.disabled = false;

}

}

function updateTotalPrice() {

const quantity = parseInt(document.getElementById('quantityInput').value) || 0;

if (currentPrice > 0 && quantity > 0) {

const total = currentPrice \* quantity;

document.getElementById('totalPriceDisplay').value = '$' + total.toFixed(2);

document.getElementById('totalPriceDisplay').style.fontWeight = 'bold';

document.getElementById('totalPriceDisplay').style.color = '#28a745';

} else {

document.getElementById('totalPriceDisplay').value = '$0.00';

document.getElementById('totalPriceDisplay').style.fontWeight = 'normal';

document.getElementById('totalPriceDisplay').style.color = '#6c757d';

}

}

function updateOrderSummary() {

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

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

const quantity = parseInt(document.getElementById('quantityInput').value) || 0;

const summaryDiv = document.getElementById('orderSummary');

if (customerSelect.value && productSelect.value && quantity > 0) {

const customerText = customerSelect.options[customerSelect.selectedIndex].text;

const total = currentPrice \* quantity;

summaryDiv.innerHTML = `

<div class="alert alert-info">

<h6><i class="fas fa-shopping-cart"></i> Order Preview</h6>

<p><strong>Customer:</strong><br/><small>${customerText}</small></p>

<p><strong>Product:</strong><br/><small>${selectedProductName}</small></p>

<hr>

<div class="row">

<div class="col-6"><strong>Quantity:</strong></div>

<div class="col-6 text-end">${quantity}</div>

</div>

<div class="row">

<div class="col-6"><strong>Unit Price:</strong></div>

<div class="col-6 text-end">$${currentPrice.toFixed(2)}</div>

</div>

<hr>

<div class="row">

<div class="col-6"><strong>Total:</strong></div>

<div class="col-6 text-end"><span class="text-success h5">$${total.toFixed(2)}</span></div>

</div>

</div>

`;

} else if (customerSelect.value && productSelect.value) {

summaryDiv.innerHTML = `

<div class="alert alert-warning">

<h6><i class="fas fa-exclamation-triangle"></i> Almost Ready</h6>

<p>Selected product: <strong>${selectedProductName}</strong></p>

<p>Unit price: <strong>$${currentPrice.toFixed(2)}</strong></p>

<p class="mb-0"><small>Enter quantity to complete order</small></p>

</div>

`;

} else {

summaryDiv.innerHTML = `

<div class="alert alert-secondary">

<h6><i class="fas fa-clipboard-list"></i> Order Summary</h6>

<p class="text-muted mb-0">Select customer and product to see order preview</p>

</div>

`;

}

}

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

updateOrderSummary();

});

</script>

}

## Edit.cshtml

<!-- Views/Order/Edit.cshtml -->

@model ABCRetailers.Models.Order

@{

ViewData["Title"] = "Edit Order";

}

<h2><i class="fas fa-edit"></i> Edit Order</h2>

<div class="row">

<div class="col-md-8">

<**form** **asp-action**="Edit" method="post">

<**input** **type**="hidden" **asp-for**="RowKey" />

<**input** **type**="hidden" **asp-for**="PartitionKey" />

<**input** **type**="hidden" **asp-for**="ETag" />

<**input** **type**="hidden" **asp-for**="Timestamp" />

<**input** **type**="hidden" **asp-for**="CustomerId" />

<**input** **type**="hidden" **asp-for**="ProductId" />

<**input** **type**="hidden" **asp-for**="UnitPrice" />

<**input** **type**="hidden" **asp-for**="TotalPrice" />

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<div class="row">

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

<div class="form-group mb-3">

<label class="form-label">Customer</label>

<input type="text" class="form-control" value="@Model.Username" readonly />

</div>

</div>

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

<div class="form-group mb-3">

<label class="form-label">Product</label>

<input type="text" class="form-control" value="@Model.ProductName" readonly />

</div>

</div>

</div>

<div class="row">

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

<div class="form-group mb-3">

<**label** **asp-for**="OrderDate" class="form-label"></**label**>

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

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

</div>

</div>

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

<div class="form-group mb-3">

<**label** **asp-for**="Quantity" class="form-label"></**label**>

<**input** **asp-for**="Quantity" class="form-control" **type**="number" min="1" readonly />

<div class="form-text">Quantity cannot be changed after order creation</div>

</div>

</div>

</div>

<div class="row">

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

<div class="form-group mb-3">

<label class="form-label">Unit Price</label>

<input type="text" class="form-control" value="@Model.UnitPrice.ToString("C")" readonly />

</div>

</div>

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

<div class="form-group mb-3">

<label class="form-label">Total Price</label>

<input type="text" class="form-control" value="@Model.TotalPrice.ToString("C")" readonly />

</div>

</div>

</div>

<div class="form-group mb-3">

<**label** **asp-for**="Status" class="form-label"></**label**>

<**select** **asp-for**="Status" class="form-select">

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

<**option** **value**="Processing">Processing</**option**>

<**option** **value**="Completed">Completed</**option**>

<**option** **value**="Cancelled">Cancelled</**option**>

</**select**>

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

</div>

<div class="form-group">

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

<i class="fas fa-save"></i> Update Order

</button>

<**a** **asp-action**="Details" **asp-route-id**="@Model.RowKey" class="btn btn-info">

<i class="fas fa-eye"></i> View Details

</**a**>

<**a** **asp-action**="Index" class="btn btn-secondary">

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

</**a**>

</div>

</**form**>

</div>

<div class="col-md-4">

<div class="card">

<div class="card-header">

<h6 class="card-title mb-0">Order Summary</h6>

</div>

<div class="card-body">

<p>

<strong>Order ID:</strong><br />

<small class="text-muted">@Model.OrderId</small>

</p>

<p>

<strong>Created:</strong><br />

<small class="text-muted">@Model.Timestamp?.ToString("MMM dd, yyyy HH:mm")</small>

</p>

<p>

<strong>Current Status:</strong><br />

<span class="badge bg-@(GetStatusBadgeClass(Model.Status))">@Model.Status</span>

</p>

</div>

</div>

</div>

</div>

@functions {

string GetStatusBadgeClass(string status)

{

return status?.ToLower() switch

{

"submitted" => "primary",

"processing" => "info",

"completed" => "success",

"cancelled" => "danger",

\_ => "secondary"

};

}

}

@section Scripts {

@{

await Html.RenderPartialAsync("\_ValidationScriptsPartial");

}

}

## Details.cshtml

<!-- Views/Order/Details.cshtml -->

@model ABCRetailers.Models.Order

@{

ViewData["Title"] = "Order Details";

}

<div class="d-flex justify-content-between align-items-center mb-4">

<h2><i class="fas fa-receipt"></i> Order Details</h2>

<div>

<**a** **asp-action**="Edit" **asp-route-id**="@Model.RowKey" class="btn btn-primary">

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

</**a**>

<**a** **asp-action**="Index" class="btn btn-secondary">

<i class="fas fa-list"></i> Back to List

</**a**>

</div>

</div>

<div class="row">

<div class="col-md-8">

<div class="card">

<div class="card-header">

<h5 class="card-title mb-0">Order Information</h5>

</div>

<div class="card-body">

<div class="row">

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

<p><strong>Order ID:</strong></p>

<p class="text-muted">@Model.OrderId</p>

<p><strong>Customer:</strong></p>

<p class="text-muted">@Model.Username</p>

<p><strong>Product:</strong></p>

<p class="text-muted">@Model.ProductName</p>

</div>

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

<p><strong>Order Date:</strong></p>

<p class="text-muted">@Model.OrderDate.ToString("MMMM dd, yyyy")</p>

<p><strong>Quantity:</strong></p>

<p class="text-muted">@Model.Quantity</p>

<p><strong>Status:</strong></p>

<p>

@{

string GetStatusBadgeClass(string status) => status?.ToLower() switch

{

"submitted" => "primary",

"processing" => "info",

"completed" => "success",

"cancelled" => "danger",

\_ => "secondary"

};

}

<span class="badge bg-@(GetStatusBadgeClass(Model.Status)) fs-6">

@Model.Status

</span>

</p>

</div>

</div>

</div>

</div>

</div>

<div class="col-md-4">

<div class="card">

<div class="card-header">

<h6 class="card-title mb-0">Pricing Details</h6>

</div>

<div class="card-body">

<div class="d-flex justify-content-between">

<span>Unit Price:</span>

<span>@Model.UnitPrice.ToString("C")</span>

</div>

<div class="d-flex justify-content-between">

<span>Quantity:</span>

<span>@Model.Quantity</span>

</div>

<hr>

<div class="d-flex justify-content-between">

<strong>Total Price:</strong>

<strong class="text-success">@Model.TotalPrice.ToString("C")</strong>

</div>

</div>

</div>

<div class="card mt-3">

<div class="card-header">

<h6 class="card-title mb-0">System Information</h6>

</div>

<div class="card-body">

<p><strong>Created:</strong></p>

<p class="text-muted small">@Model.Timestamp?.ToString("MMM dd, yyyy HH:mm")</p>

<p><strong>Customer ID:</strong></p>

<p class="text-muted small">@Model.CustomerId</p>

<p><strong>Product ID:</strong></p>

<p class="text-muted small">@Model.ProductId</p>

</div>

</div>

</div>

</div>