

Fortens 500 Corp.  
Atomic Synergistics - Records  
Application Proposal



## About Us

Fortens 500 Corp. is a dynamic new startup tackling web development needs of today's ever-evolving industry. We are full stack developers with a variety of skillsets including:

.NET Core MVC Framework	Database Management
Full Stack Web Application Design	Unit Testing
C# Fundamentals	Software Development Cycle
Javascript (React)	Agile – Scrum Methodology

## Overview

Atomic Synergistics is a global leader in sales/distribution of electronic and technology products. Fortens 500 Corp. was sourced as a contractor to develop an in-house application to manage company records, like Employees, Products, Sales, etc. with an easy to use dashboard.

## Requirements

- Development of an administrative intranet application that will allow authorized staff to manage company records.
- Modern administrative dashboard with a clean, consistent look throughout.
- Functionality:
  - Display records in attractive tables.
    - Sort, page, & search functionality strongly preferred.
    - If time allows, data visualization options are welcome.
  - Ability to create, edit, and delete records.
  - Registration & management of user accounts, roles.
- Architecture:
  - .NET 6.0 (Long-Term Support)
  - .NET Core Web Application (Model-View-Controller Architecture)
  - Microsoft SQL Server Relational Database
  - User Account Registration, Login, Management

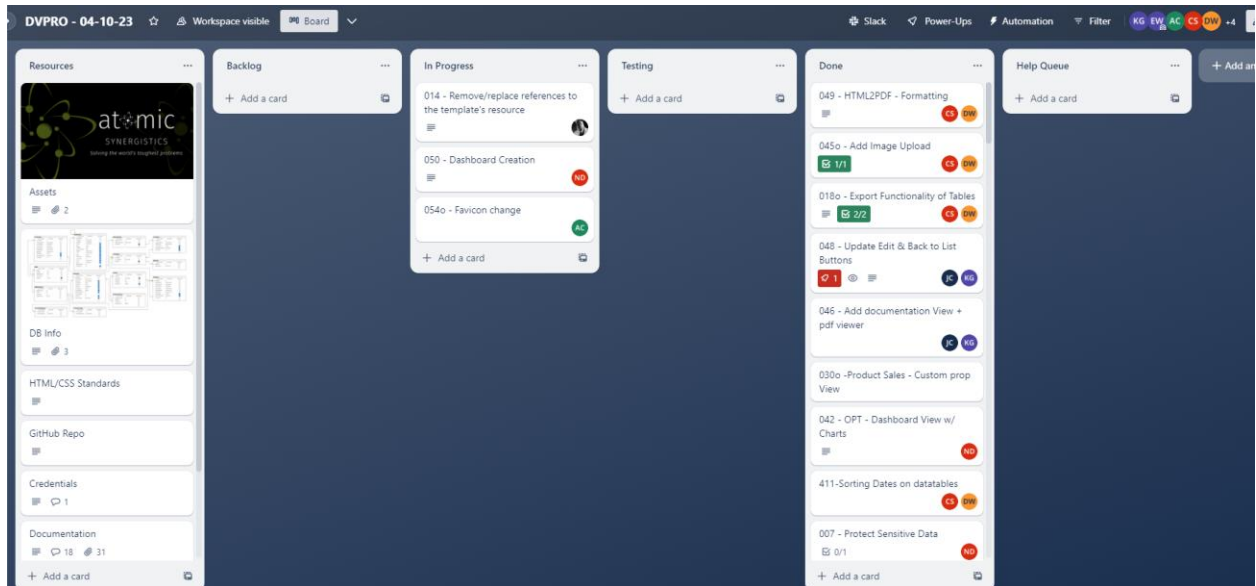
## Processes, Features, and Additional Functionality

- Tracked progress via Trello board with assigned tasks and cards
- Project history tracking and collaboration via GitHub
- Project documentation via screenshots
- Two-Tier, Data-Driven Web Application using .NET Core MVC
- Modern integrated styling consistent with client branding
- Integration with provided SQL database utilizing SSMS
- Metadata of classes for data annotation
- Utilize multiple roles for Administrators, HR, and Accounting for authentication
- CRUD (Create, Read, Update, Delete) functionality across all records for authorized users per Identity Matrix
- Visual data output for critical metrics
- Easily readable data outputs with search and pagination functions
- Export functionality for all data outputs
- Image upload utility

# Documentation

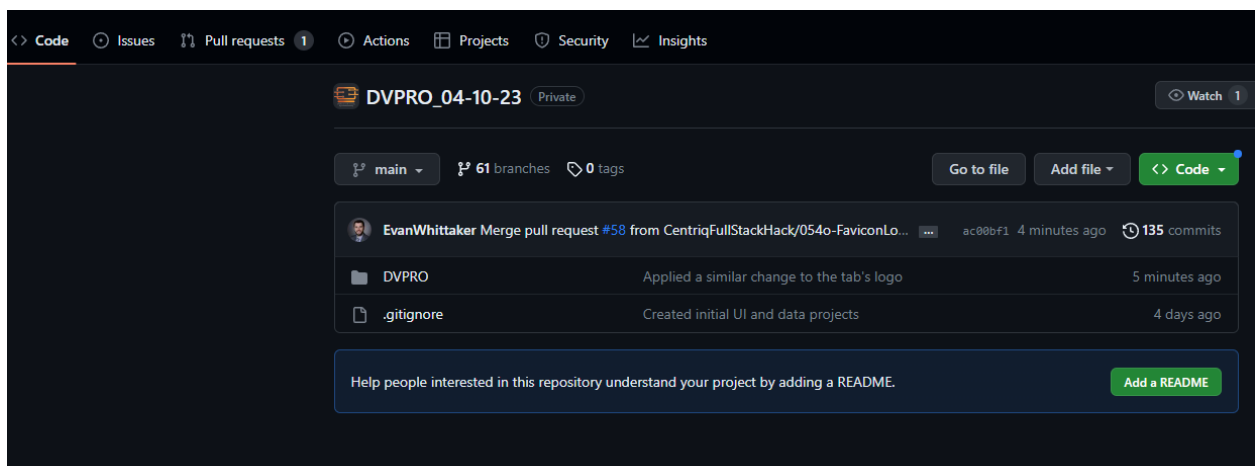
## Trello Board

Trello Boards were utilized to help track allocation of work, requirements, deadlines etc.



## GitHub – Tracking/Collaboration

GitHub was used to work collaboratively on the codebase, implementing branching techniques in line with current industry standards.



## Example Table Page View

Atomic Synergistics

Dashboard

Admin

Customers

Sales

Product Sales

Product Status

Employees

Products

Product Types

Departments

Countries

Orders

Order Products

Locations

Regions

Vendors

Roles

Users

Documentation

Help and Support

Products

Create New

Copy CSV Excel PDF Print

Search:

Product Name	Cost Per Unit	Price Per Unit	Unit Type	Units In Stock	Units on Order	Product Status	Product Type	Vendor	Actions
AeroJet Pro Gaming Laptop	\$11.00	\$13.20	-	720	1911	In Stock	Networking Device		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
AeroMax Pro Gaming Mouse	\$53.00	\$63.60	-	0	0	Discontinued	Productivity Software		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
AeroPodXtreme	\$299.99	\$359.99	-	100	5	In Stock	Wearable Tech		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
AeroStream Pro Dual-Band Wi-Fi Router	\$20.00	\$24.00	-	0	2013	Out of Stock	Networking Device		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
AeroStream Pro Wi-Fi Extender	\$35.00	\$42.00	-	17264	2651	In Stock	Productivity Software		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
BeamMaster Pro Laser Printer	\$91.00	\$109.20	-	0	591	Out of Stock	Networking Device		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
BioChargeMax	\$79.99	\$95.99	-	6	0	Discontinued	Wearable Tech		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
CyberShield Pro VPN Service	\$29.00	\$34.80	-	14557	2950	In Stock	Smart Home Accessory		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
HoloSyncPro	\$199.99	\$239.99	-	100	10	In Stock	Communications Device		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
iGlowTech	\$9.99	\$11.99	-	1500	100	In Stock	Smart Home Accessory		<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

## Styling

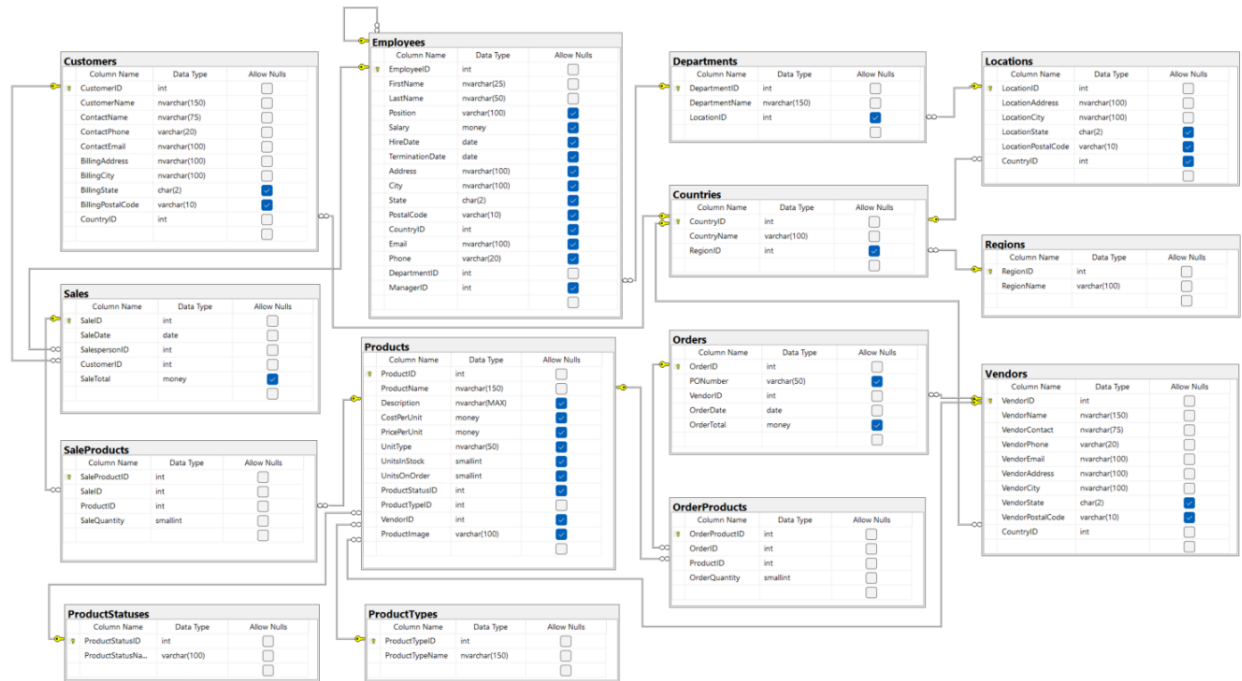
Site provides a modern, dark-mode type display throughout, utilizing the main scheme below:

```
:root {
  scroll-behavior: smooth;
  --atomic-secondary: #455515;
  --atomic-primary: #61771d;
  --atomic-highlight: #aed635;
  --atomic-light: #acacac;
  --atomic-dark: #404040;
  --atomic-black: #131313;
}

body {
  /*font-family: "Open Sans", sans-serif;
  background: #f6f9ff;
  color: #444444;*/
  background-color: var(--atomic-black);
  color: var(--atomic-light);
}
```

# SQL Database Schema

DB schema provided by Atomic Synergistics



## Authorization – Identity Matrix

<b>Home</b>	<b>Login</b>
Anonymous	x
Scheduling	x
Admin	x

<b>Customers</b>	<b>Index</b>	<b>Detail</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting	x	x	x	x	
HR					
Admin	x	x	x	x	x

<b>Sales</b>	<b>Index</b>	<b>Details</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting	x	x	x	x	
HR					
Admin	x	x	x	x	x

<b>SaleProducts</b>	<b>Index</b>	<b>Details</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting	x	x	x	x	
HR					
Admin	x	x	x	x	x

<b>Products</b>	<b>Index</b>	<b>Details</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting	x	x	x	x	
HR					
Admin	x	x	x	x	x

<b>Orders</b>	<b>Index</b>	<b>Details</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting	x	x	x	x	
HR					
Admin	x	x	x	x	x

<b>OrderProducts</b>	<b>Index</b>	<b>Details</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting	x	x	x	x	
HR					
Admin	x	x	x	x	x

<b>Vendors</b>	<b>Index</b>	<b>Details</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting	x	x	x	x	
HR					
Admin	x	x	x	x	x

<b>Employees</b>	<b>Index</b>	<b>Details</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting	x	x	x	x	
HR	x	x	x	x	
Admin	x	x	x	x	x

<b>Departments</b>	<b>Index</b>	<b>Details</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting					
HR	x	x	x	x	
Admin	x	x	x	x	x

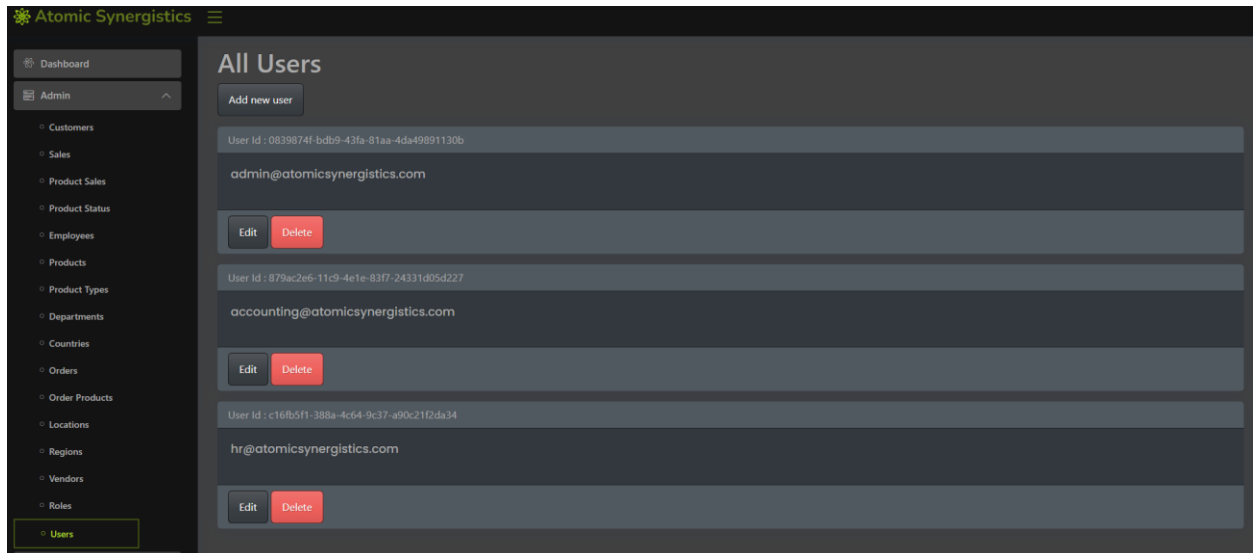
<b>Locations</b>	<b>Index</b>	<b>Details</b>	<b>Create</b>	<b>Edit</b>	<b>Delete</b>
Accounting					
HR	x	x	x	x	
Admin	x	x	x	x	x

<b>AdminOnly</b>	<b>Countries</b>	<b>Region</b>	<b>Region</b>	<b>ProductTypes</b>	<b>ProductStatuses</b>
Admin	x	x	x		x

## Identity Management - Code Example

Below shows the admin functionality to manage the users and roles within the site



```
<h1>All Users</h1>

@if (Model.Any())
{
    <a asp-area="Identity" asp-page="/Account/Register" class="btn btn-primary mb-3" style="width:auto">
        Add new user
    </a>

    foreach (var user in Model)
    {
        <div class="card mb-3">
            <div class="card-header">
                User Id : @user.Id
            </div>
            <div class="card-body">
                <h5 class="card-title">@user.UserName</h5>
            </div>
            <div class="card-footer">
                <form method="post" asp-action="DeleteUser" asp-route-id="@user.Id">
                    <a asp-action="EditUser" asp-controller="Administration" asp-route-id="@user.Id" class="btn btn-primary">Edit</a>

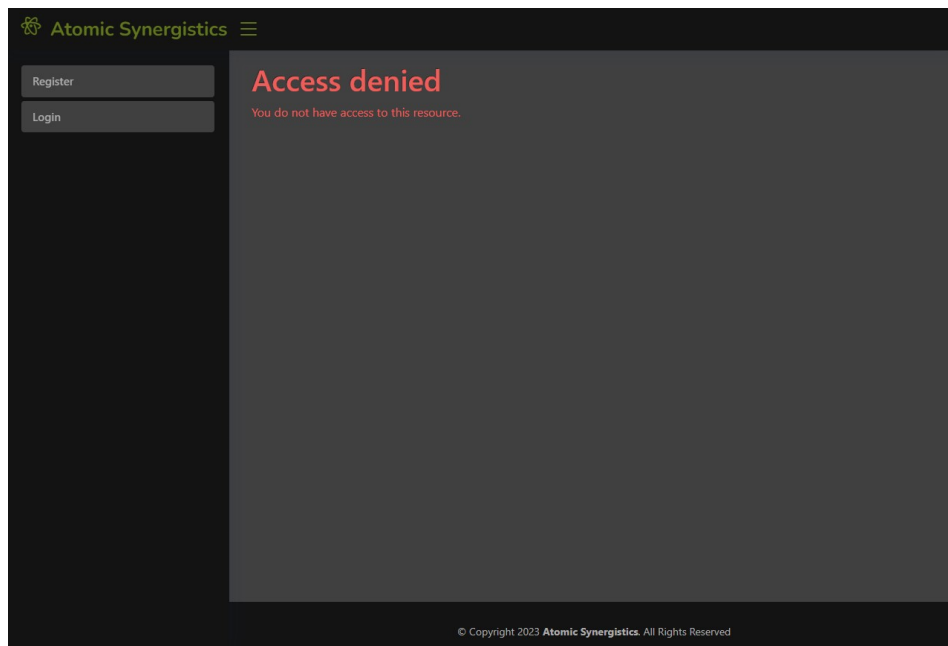
                    <span id="confirmDeleteSpan_@user.Id" style="display:none">
                        <span>Are you sure you want to delete?</span>
                        <button type="submit" class="btn btn-danger">Yes</button>
                        <a href="#" class="btn btn-primary" onclick="confirmDelete('@user.Id', false)">No</a>
                    </span>

                    <span id="deleteSpan_@user.Id">
                        <a href="#" class="btn btn-danger" onclick="confirmDelete('@user.Id', true)">Delete</a>
                    </span>
                </form>
            </div>
        </div>
    }
}
```



## Authorization – Navigation Example

Below is a snip of how the navigation pages are limited to certain views and are locked for other uses.

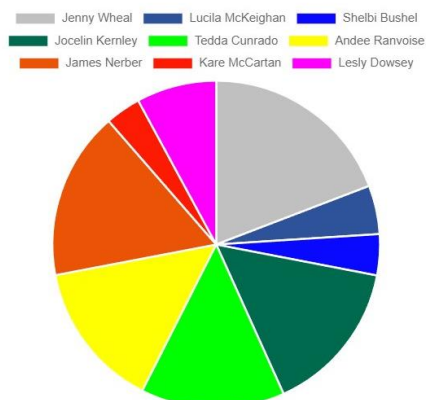


```
namespace DVPRO.UI.MVC.Controllers
{
    [Authorize(Roles = "Admin, Accounting")]
    1 reference
    public class ProductsController : Controller
    {
    }
```

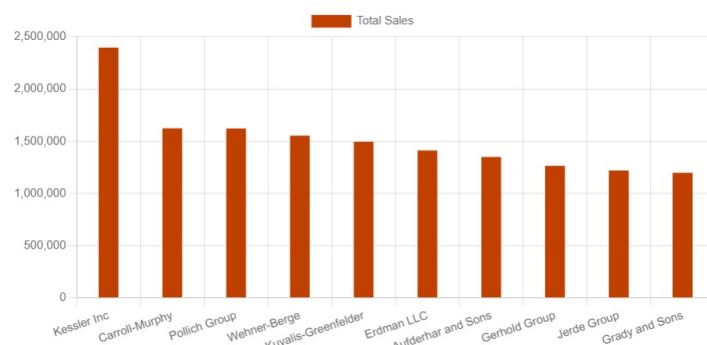
## Dashboard – Data Visualization

Below is a snapshot of the data in graph form with code examples on how this was achieved. This functionality was developed using [Chart.js](#) plugins.

Sales by Rep - 2023



Top 10 Customers by Revenue



## Pie Chart Code

```
#region Top SalesPerson Pie Chart
#region Collect Totals by Month and sales person
//Totals the sales by Year and sales person, saves as a list
List<decimal?> monthlyTotalSalesPerson = salesData
    .GroupBy(sale => new {sale.Salesperson})
    .Select(group => group.Sum(sale => sale.SaleTotal))
    .ToList();

//Converts list to doubles for use in ChartJS
List<double?> monthlySalesPersonAsDoubles = monthlyTotalSalesPerson.ConvertAll(s => (double?)s);

#endregion

#region Collect Names for Labels
List<string> salesNames = salesData
    .Select(sale => sale.Salesperson.FullName)
    .Distinct()
    .ToList();
#endregion

#region PieChart creation
//Create the chart object
Chart salesPieChart = new Chart();

//Define the chart type
salesPieChart.Type = Enums.ChartType.Pie;

//Create the chart data object
ChartJSCore.Models.Data pieChartData = new ChartJSCore.Models.Data();

//Assign labels
pieChartData.Labels = salesNames;

//Create dataset object
PieDataset pieDataset = new PieDataset()
{
    Label = $"Total Sales",
    BackgroundColor = new List<ChartColor>()
    {
        ChartColor.FromHexString("#C0C0C0"),
        ChartColor.FromHexString("#2F539B"),
        ChartColor.FromHexString("#0909FF"),
        ChartColor.FromHexString("#006A4E"),
        ChartColor.FromHexString("#00FF00"),
        ChartColor.FromHexString("#FFFF00"),
        ChartColor.FromHexString("#EB5406"),
        ChartColor.FromHexString("#FD1C03"),
        ChartColor.FromHexString("#FF00FF")
    },
    HoverBackgroundColor = new List<ChartColor>()
    {
        ChartColor.FromHexString("#C0C0C0"),
        ChartColor.FromHexString("#2F539B"),
        ChartColor.FromHexString("#0909FF"),
        ChartColor.FromHexString("#006A4E"),
        ChartColor.FromHexString("#00FF00"),
        ChartColor.FromHexString("#FFFF00"),
        ChartColor.FromHexString("#EB5406"),
        ChartColor.FromHexString("#FD1C03"),
        ChartColor.FromHexString("#FF00FF")
    },
    Data = monthlySalesPersonAsDoubles
};
#endregion

#region Packaging and passing to the view
//Create a List<Dataset> to store our Chart data
pieChartData.Datasets = new List<Dataset>();

//Adds the customized PieDataset to the list
pieChartData.Datasets.Add(pieDataset);

//Assigns the List<Dataset> to the data property of our chart object
salesPieChart.Data = pieChartData;

//Stores the completed chart in the ViewData
ViewData["PieChart"] = salesPieChart;

#endregion
#endregion
```

## Bar Chart Code

```
#region Total Sales By Customer
//All time sales by customer
var totalCustSales = totalSalesData
    .GroupBy(sale => new {sale.Customer.CustomerName})
    .Select(group => new {group.Key.CustomerName, TotalSales = group.Sum(sale => sale.SaleTotal) })
    .OrderByDescending(t => t.TotalSales)
    .Take(10)
    .ToList();

//Convert list to doubles for ChartJS
List<double?> totalCustSalesDoubles = totalCustSales.ConvertAll(s => (double?)s.TotalSales);

//Create list of Customer names for column lables
List<string> custNames = totalCustSales
    .Select(sale => sale.CustomerName)
    .Distinct()
    .ToList();

#endregion

#region BarChart creation
//Create the Chart Object
Chart custSalesBar = new Chart();

//Assing the chart type
custSalesBar.Type = Enums.ChartType.Bar;

//Create the chart data object
ChartJSCore.Models.Data custSalesChartData = new ChartJSCore.Models.Data();

//Assing the label
custSalesChartData.Labels = custNames;

//Create the dataset object
BarDataset custSalesDataSet = new BarDataset()
{
    Label = "Total Sales",
    Data = totalCustSalesDoubles,
    BackgroundColor = new List<ChartColor>
    {
        ChartColor.FromHexString("#C04000")
    },
    BorderColor = new List<ChartColor>
    {
        ChartColor.FromHexString("#C04000")
    },
    BorderWidth = new List<int>() { 1 },
    BarPercentage = 1,
    BarThickness = 20,
    MaxBarThickness = 30,
    MinBarLength = 2
};

//Packaging and passing to the view
custSalesChartData.Datasets = new List<Dataset>();
custSalesChartData.Datasets.Add(custSalesDataSet);

custSalesBar.Data = custSalesChartData;

var options2 = new Options
{
    Responsive = true,
    MaintainAspectRatio = false
};

ViewData["CustSalesBar"] = custSalesBar;
#endregion
#endregion
```

## Metadata – Data Annotation

These data annotations help protect proper data interaction with the database during create/edit functions. In some areas, null checks are added to render either N/A or “-” symbol.

```
0 references
public string? UnitType { get; set; }
[Display(Name = "Units in Stock")]
0 references
public short? UnitsInStock { get; set; }
[Display(Name = "Units on Order")]
0 references
public short? UnitsOnOrder { get; set; }
[Display(Name = "Product Status ID")]
0 references
public int? ProductStatusId { get; set; }
[Required(ErrorMessage = "*Product Type ID is required")]
[Display(Name = "Product Type ID")]
0 references
public int ProductTypeId { get; set; }
[Display(Name = "Vendor ID")]
0 references
public int? VendorId { get; set; }
[StringLength(100, ErrorMessage = "*Max 100 Characters")]
[Display(Name = "Product Image")]
[DisplayFormat(NullDisplayText = "-")]
0 references
public string? ProductImage { get; set; }
```

## Partial Class – Custom Properties

These custom properties allow for additional data to be rendered during the CRUD functions.

```
55 public partial class ProductStatus { }
56
57 [ModelMetadataType(typeof(EmployeeMetadata))]
58 public partial class Employee
59 {
60     4 references
61     public string FullName
62     {
63         get
64         {
65             return string.Format($"{FirstName} {LastName}");
66         }
67     }
68
69     [ModelMetadataType(typeof(ProductMetadata))]
70     public partial class Product
71     {
72         [NotMapped]
73         10 references
74         public IFormFile? Image { get; set; }
75
76         [ModelMetadataType(typeof(ProductTypeMetadata))]
77         22 references
78         public partial class ProductType { }
```

Employee Name
Adam Synger
Addia O'Cuddie
Alexandr Jossum
Alexandre Kerton
Alyson Sutti
Andee Ranvoise
Betsy McConigal
Bette-ann Streight
Bill Kopmann
Binky Partner

## CRUD Views and Functionality

### Customer

Create Customer

Customer Name	<input type="text"/>	
Contact Name	Contact Phone	Contact Email
<input type="text"/>	<input type="text"/>	<input type="text"/>
Billing Address	Billing City	
<input type="text"/>	<input type="text"/>	
Billing State	Billing Zip Code	Country
<input type="text"/>	<input type="text"/>	United States

### Customer

Edit Customer

Customer Name	<input type="text" value="Adams-Greenholt"/>	
Contact Name	Contact Phone	Contact Email
<input type="text" value="Roby Braker"/>	<input type="text" value="5162111675"/>	<input type="text" value="rbrakerv@washingtonpost.com"/>
Billing Address	Billing City	
<input type="text" value="0 Commercial Hill"/>	<input type="text" value="Kolape"/>	
Billing State	Billing Zip Code	Country
<input type="text"/>	<input type="text" value="1106"/>	<input type="text" value="Israel"/>

## Delete

Are you sure you want to delete this?

### Customer

Customer Name	Adams-Greenholt
Contact Name	Roby Braker
Contact Phone	5162111675
Contact Email	<a href="mailto:rbrakerv@washingtonpost.com">rbrakerv@washingtonpost.com</a>
Billing Address	0 Commercial Hill
Billing City	Kolape
Billing State	
Billing Zip Code	1106
Country	Israel

## Search and Pagination

Incorporation of the [Datatables](#) plugin allowed for tables of data to be searched and organized in a page view, including sort functionality.

### Countries

Create New

CopyCSVExcelPDFPrint

Search:

Country Name	Region	Actions
Argentina	South America	Edit   Details   Delete
Austria	Europe	Edit   Details   Delete
Bahrain	Middle East & North Africa	Edit   Details   Delete
Brazil	South America	Edit   Details   Delete
Canada	North America	Edit   Details   Delete
Chile	South America	Edit   Details   Delete
China	East Asia & Pacific	Edit   Details   Delete
Colombia	South America	Edit   Details   Delete
Denmark	Europe	Edit   Details   Delete
Egypt	Middle East & North Africa	Edit   Details   Delete

Showing 1 to 10 of 44 entries

Previous12345Next

### @section Scripts{

```
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script src="https://cdn.datatables.net/1.13.5/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/buttons/2.4.1/js/dataTables.buttons.min.js"></script>
```

<!-- Export Button CSS StyleSheets -->

```
<link href="https://cdn.datatables.net/1.13.5/css/jquery.dataTables.min.css" rel="stylesheet" />
<link href="https://cdn.datatables.net/buttons/2.4.1/css/buttons.dataTables.min.css" rel="stylesheet" />
```

## Clipboard and Export

The [Datatables](#) function also allowed for a copy to clipboard and export functionality of the tables. This was also extended to the Details Views.

Countries

Create New

CopyCSVExcelPDFPrint

Search:

Country Name	Region	Actions
Argentina	South America	Edit   Details   Delete
Austria	Europe	Edit   Details   Delete
Bahrain	Middle East & North Africa	Edit   Details   Delete
Brazil	South America	Edit   Details   Delete
Canada	North America	Edit   Details   Delete
Chile	South America	Edit   Details   Delete
China	East Asia & Pacific	Edit   Details   Delete
Colombia	South America	Edit   Details   Delete
Denmark	Europe	Edit   Details   Delete
Egypt	Middle East & North Africa	Edit   Details   Delete

Showing 1 to 10 of 44 entries

Previous12345Next

## Image Upload

This functionality allows for images to be uploaded to the Products table

Product Image

Choose File

No file chosen

## Wishlist

Below is a brief list of added functionalities that could be incorporated in the future:

- Go through all CRUD views and refine style.
- Style the “email” and “password” boxes.