~~~~~~

Changeset 2020 - 6/14/2018 4:16:54 PM

(1) I added the actionresult method to return the "buyer" in the BuyerDetails view.

(2) I also have been updating the CSS on the reviews page, added a couple hr's with classes to seperate the reviews, and hr's within the review details dropwdown that were ehavier to strongly imply their seperation from the next view. I added a gradiant shading to the review backgrounds to make them stand out more, with an easy view on the eyes.

*****(1)*****

To pass the view I had to convert the datatype from the campaign model to the AnalyticsViewModel type, then pass that view.

-----

```
public ActionResult BuyerDetails(int? id)
{
        if (id == null)
        {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Campaign buyers = db.Campaigns.Find(id);
        AnalyticsViewModel buyer = new AnalyticsViewModel(buyers);
        {

        }
        if (buyer == null)
        {
        return HttpNotFound();
        }
        return View(buyer);
}
```

-----

*****(2)*****

I added <hr>'s and gave them a class to target in CSS

-----

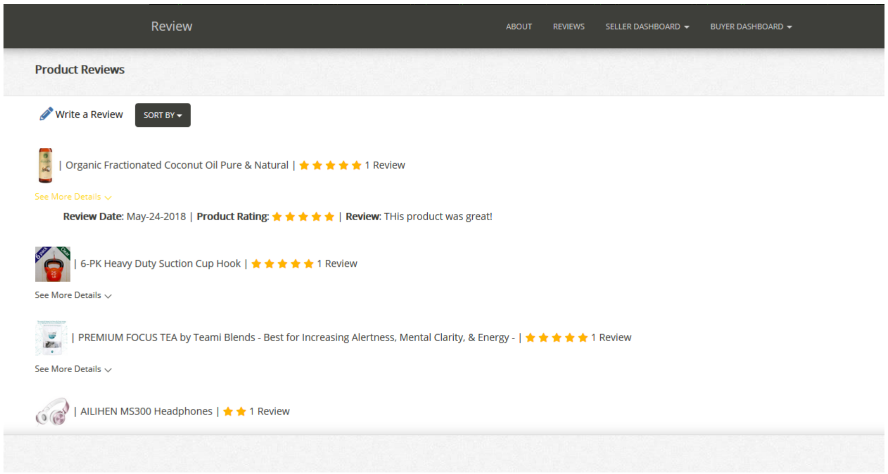
```
/*For the "See more details" dropdown*/
hr.style-seven {
   overflow: visible; /* For IE */
   height: 30px;
   border-style: solid;
   border-color: black;
   border-width: 1px 0 0 0;
   border-radius: 20px;
}

   hr.style-seven:before {
      display: block;
      content: "";
      height: 30px;
      margin-top: -31px;
      border-style: solid;
      border-color: black;
      border-width: 0 0 1px 0;
```
~~~~~~

```
        border-radius: 20px;
    }
*
/*For the Review borders*/
hr.style-four {
    border: 0;
    height: 1px;
    background-image: linear-gradient(to right, rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.75), rgba(0, 0, 0, 0));
}

.accordion-item {
    background: -webkit-gradient(radial, 50% 50%, 0, 50% 50%, 350, from(#e9e9e9), to(#fff));
}
-----
```

[Example of before / after CSS manipulation]



Product Reviews

| Organic Fractionated Coconut Oil Pure & Natural | ★★★★★ 1 Review

See More Details

Review Date: May-24-2018 | Product Rating: ★★★★★ | Review: THis product was great!

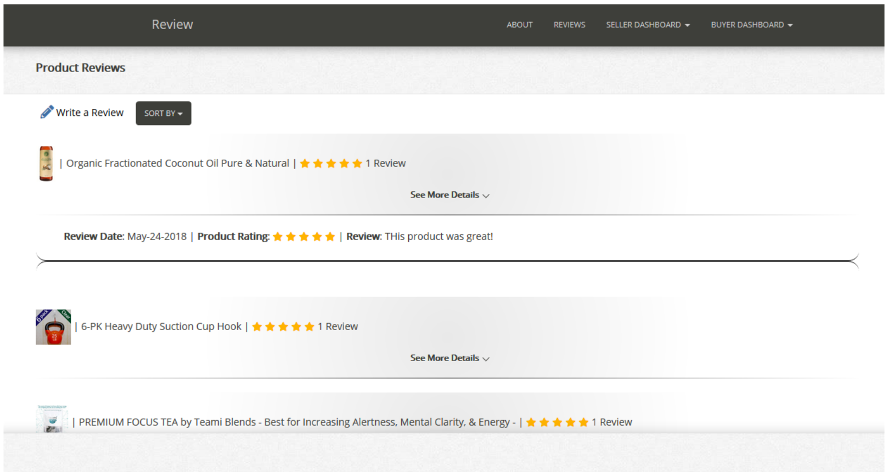
| 6-PK Heavy Duty Suction Cup Hook | ★★★★★ 1 Review

See More Details

| PREMIUM FOCUS TEA by Teami Blends - Best for Increasing Alertness, Mental Clarity, & Energy - | ★★★★★ 1 Review

See More Details

| AILIHEN MS300 Headphones | ★★ 1 Review



Product Reviews

| Organic Fractionated Coconut Oil Pure & Natural | ★★★★★ 1 Review

See More Details

Review Date: May-24-2018 | Product Rating: ★★★★★ | Review: THis product was great!

| 6-PK Heavy Duty Suction Cup Hook | ★★★★★ 1 Review

See More Details

| PREMIUM FOCUS TEA by Teami Blends - Best for Increasing Alertness, Mental Clarity, & Energy - | ★★★★★ 1 Review

~~~~~~

Changeset 1987 - 6/12/2018 7:48:34 PM

(1) created a new view in the Campaigns folder called BuyerDetails
~~~~~~

(2) Added the name and Email of the person who referred the converted referral.

*****(1)*****

Right click on view folder, add view, enter in the scaffolding, press ok.

created a new view in the Campaigns folder called BuyerDetails and scaffolded it off the AnalyticsViewModel with a Details template and the ApplicationDbContext as the Data Context. The trick with this is that there is no actual model being used, we were using a viewmodel which means that the data context is not needed as the viewmodel is not actually in the database. To fix this I simply removed the data context field, asnd the view was created.

*****(2)*****

This was the original code,

```
-----
<tr>
        <th>
            @Html.DisplayNameFor(model => model.ReferralName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.ReferralEmail)
        </th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.ReferralName)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.ReferralEmail)
        </td>

    </tr>
-----
```

And I needed to add the the name and Email of the person who referred the converted referral,

```
-----
 <tr>
        <th>
            @Html.DisplayNameFor(model => model.ReferralName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.ReferralEmail)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.ApplicationUser.FullName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.ApplicationUser.Email)
        </th>
    </tr>
```

```
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.ReferralName)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.ReferralEmail)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.ApplicationUser.FullName)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.ApplicationUser.Email)
        </td>
    </tr>
-----
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Changeset 1976 - 6/11/2018 6:20:54 PM
Added a paper-plane icon as a mailto: link for the referral emails.

This was the original line,

```
-----
<td>
            @Html.DisplayFor(modelItem => item.ReferralEmail)
</td>
-----
```

And I added an icon as a Mailto: link,

```
-----
 <td>
@Html.DisplayFor(modelItem => item.ReferralEmail)
@{
<a href="mailto:@(item.ReferralEmail)"><i class="fas fa-paper-plane"></i></a>
}
</td>
-----
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Changeset 1974 - 6/11/2018 5:30:48 PM
Entered in a review count after the star rating (But before Info icon) for the reviews page.

```
-----
 @{ int ReviewCount = item.Campaign.Reviews.Count(); }
     @Html.Raw(ReviewCount)
     @if (ReviewCount != 1)
     {
         @Html.Raw(" Reviews")
     }
     else
     {
         @Html.Raw(" Review")
```

```
     }
-----
```

~~~~~~

Changeset 1969 - 6/11/2018 1:16:59 PM
Changed banner title to "Blue Ribbons Review"

Changed the following line,

```
-----
@Html.ActionLink("Blue Ribbons Reviews", "Index", "Campaigns", new { area = "" }, new { @class
= "navbar-brand" })
-----
```

to,

```
-----
@Html.ActionLink("Blue Ribbons Review", "Index", "Campaigns", new { area = "" }, new { @class =
"navbar-brand" })
-----
```

~~~~~~

Changeset 1947 - 6/8/2018 1:14:27 PM
Added name of referrer and email of referrer to referral details page.

```
-----
 <dt>
            @Html.DisplayName("Referrer Name")
</dt>

<dd>
            @Html.DisplayFor(model => model.ApplicationUser.FirstName)  @Html.DisplayFor(model
=> model.ApplicationUser.LastName)
</dd>

<dt>
            @Html.DisplayName("Referrer Email")
</dt>

<dd>
            @Html.DisplayFor(model => model.ApplicationUser.Email)
</dd>
-----
```

~~~~~~

Changeset 1945 - 6/8/2018 11:34:49 AM

(1) Created a drop-down list for referrals on the admin details page.
(2) I created a new class for the button as it disoriented the text layout with it's size. This smaller version fits the layout better and still functions fine as a drop-down list.

*****(1)*****

First I changed the following line,
~~~~~~

```
 <dd>
    @Html.DisplayFor(model => model.ReferralCount)
</dd>
```

Into,

```
 <dd>

<div class="btn-group">
            <button type="button" class="btn btn-darksmall">@Html.DisplayFor(model =>
model.ReferralCount)</button><button type="button"  class="btn btn-darksmall dropdown-toggle"
data-toggle="dropdown">
            <span class="caret"></span>
            <span class="sr-only">Toggle Dropdown</span>
</button>
<ul class="dropdown-menu">

@*foreach user create link to all referals they have made in the dropdown.*@
@{
foreach (var referrals in Model.Referrals)
{
            <li>@Html.DisplayFor(model => referrals.ReferralName)</li>
}
}

</ul>
</div>
</dd>
```

*****(2)*****

Then I added and called the class to create a smaller cleaner look for the buttons,

-----

```
 .btn-darksmall {
    background: #3e3f3a;
    border-color: #204d74;
    color: #fff;
    padding: 1px 5px;
    }
    .btn-darksmall:hover {
    background: #3e3f3a;
    border-color: #204d74;
    color: #fff;
    padding: 1px 5px;
    }
```

~~~~~~

Changeset 1943 - 6/7/2018 9:22:32 PM
~~~~~~

Added name of user to the top of the page on the "Referral details page

Changed,

```
-----
@{
    ViewBag.Title = "Referral Details";
    ViewBag.Tagline = "";
}
-----
```

To,

```
-----
@{
    ViewBag.Title = "Referral Details for " + (Html.DisplayFor(model => model.ReferralName));
    ViewBag.Tagline = "";
}
-----
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

1942   carter spade   6/7/2018 9:15:08 PM   Added name of user to the top of the page after "Details" per user story 1707

Changed line,

```
-----
@{
    ViewBag.Title = "Details";
}
-----
```

To,

```
-----
@{
    ViewBag.Title = "Details for " + (Html.DisplayFor(model => model.FirstName))+ " " +
(Html.DisplayFor(model => model.LastName));
}
-----

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Changeset 1941 - 6/7/2018 8:59:42 PM
Added mailto: function to emails on Campaign AdminView. Added "N/A" for applicationUsers with a null value as the mailto function was causing a NullReferenceException when the fullname value was null.

```
-----
<td>

 @{
    if (item.ApplicationUser == null)
    {
        @Html.Raw("(N/A)");
    }
    else
    {
```

```
        <a href="mailto:@(item.ApplicationUser.Email)">@Html.DisplayFor(modelItem
=>item.ApplicationUser.Email)</a>
    }
 }
</td>
-----

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Changeset 1929 - 6/7/2018 12:21:58 AM

(1) in the user table on the Index page of the Admin controller, I converted the user's email address that appears in the email address column into a link that sends an email to that user.

(2) Changed the color of the buttons on the index page to match the color of the navbar.

*****(1)*****

To change the email into a Mailto: link I changed the following line,

```
-----
<td>
    @Html.DisplayFor(modelItem => item.Email)
</td>
```

into,

```
-----
 @{
    <a href="mailto:@(item.Email)">@Html.DisplayFor(modelItem => item.Email)</a>
}
-----
```

*****(2)*****

Changed the name of the specific buttons class,

```
-----
 <button type="button" class="btn btn-dark">@Html.DisplayFor(modelItem =>
item.CampaignCount)</button>
 <button type="button" class="btn btn-dark dropdown-toggle" data-toggle="dropdown">
-----
.btn-dark {
    background: #3e3f3a;
    border-color: #204d74;
    color: #fff;
}
-----
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Changeset 1920 - 6/5/2018 11:09:59 PM
(1)moved the log off option to be a list item under the "Hello Username" in the navbar, and
(2)displayed the referrals associated with the users that have referred others in a dropdown on the Index page of the Admin

*****(1)*****

I converted this list item,

```
<li><a href="javascript:document.getElementById('logoutForm').submit()">Log off</a></li>
```

-----

to part of a drop-down menu,

```
<ul class="dropdown-menu">
    <li>@Html.ActionLink("Change Password", "ChangePassword", "Manage")</li>
    <li><a href="javascript:document.getElementById('logoutForm').submit()">Log off</a></li>
</ul>
```

*****(2)*****

First step, I had to create a list of Referrals.

```
 /*Created list so as to enable drop-downs on the index page of the admin - Carter*/
        public List<Referral> Referrals
        {
            get
            {
                ApplicationDbContext db = new ApplicationDbContext();
                var referrals = from c in db.Referrals
                                where c.ApplicationUser.Id == UserID
                                select c;
                return referrals.ToList<Referral>();
            }
        }
```

Then adding a drop-down button from a list of information

```
            <div class="btn-group">
                <button type="button" class="btn btn-dark">@Html.DisplayFor(model =>
model.ReferralCount)</button>
                <button type="button" class="btn btn-dark dropdown-toggle" data-toggle="dropdown">
                    <span class="caret"></span>
                    <span class="sr-only">Toggle Dropdown</span>
                </button>
                <ul class="dropdown-menu">

                    @*foreach user create link to all referals they have made in the dropdown.*@
                    @{
                        foreach (var referrals in Model.Referrals)
                        {
                            <li>@Html.DisplayFor(model => referrals.ReferralName)</li>

                        }
                    }

                </ul>
            </div>
```

~~~~~~
~~~~~~