# HORISEON WEBPAGE REFACTORING

## ACCEPTANCE CRITERIA

The following items requirements were provided as the scope of work and acceptance criteria.

- **Item 1**
  WHEN I view the source code
  THEN I find semantic HTML elements

- **Item 2**
  WHEN I view the structure of the HTML elements
  THEN I find that the elements follow a logical structure independent of styling and positioning

- **Item 3**
  WHEN I view the image elements
  THEN I find accessible alt attributes

- **Item 4**
  WHEN I view the heading attributes
  THEN they fall in sequential order

- **Item 5**
  WHEN I view the title element
  THEN I find a concise, descriptive title

# CODE REFACTORING

The following narrative details the developer's notes for the refactoring of the horiseon website files to conform with the acceptance criteria. Moreover, the hyperlinks below provide further supporting materials for the refactoring initiative.

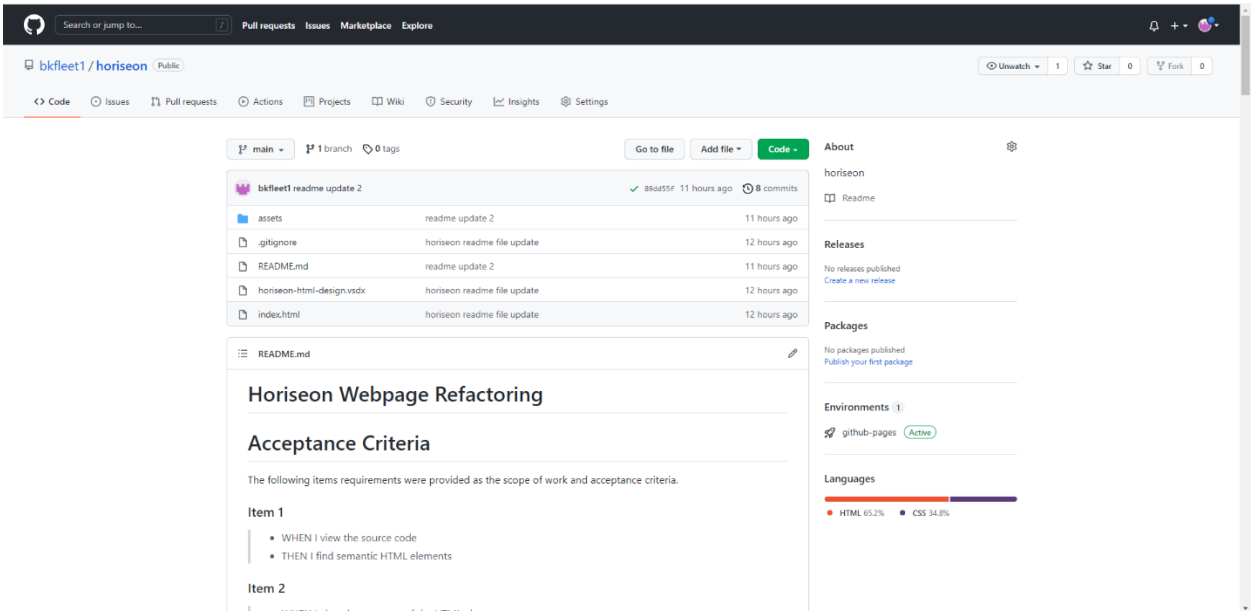
- **Original Code:** https://bkfleet1.github.io/horiseon/assets/documents/original-code.zip
- **Refactored Code:** https://bkfleet1.github.io/horiseon/assets/documents/refactored-code.zip
- **Readme file:** https://github.com/bkfleet1/horiseon/blob/main/README.md?plain=1
- **Project writeup:** https://bkfleet1.github.io/horiseon/assets/documents/horiseon-webpage-refactoring.pdf
- **Live refactored code:** https://bkfleet1.github.io/horiseon/

## REPOSITORY INITIATION

The developer performed the following steps to initiate the code refactoring.

1. Logged into developers GitHub portal
2. Created a new repository named "horiseon" and a readme.md file
3. Enabled the repository's web page features, which are found under the repository's settings > pages
4. Created a local directory c:/users/brad/desktop/develop/
5. The root directory for Git Bash on developer's computer is c:/users/brad. Entered command "cd desktop/develop" to change directory
6. Next the developer opened the Git Bash terminal software
7. Entered the following commands in the Git Bash terminal software:
   git init main/horiseon
   git pull origin main/horiseon
8. Downloaded the horiseon .zip file into the local directory
9. Unzipped horiseon website files in the local directory
10. Entered the following commands in the Git Bash terminal software:
    git add - A
    git commit -m "initial horiseon files push"
    git push origin main/horiseon

At this point the development environment on the developer's local machine and GitHub repository were established and ready for code refactoring. Below is a screen capture of the horiseon repository.




The following narrative details the refactoring of the code contained in the .html and .css files to satisfy the acceptance criteria.

# HTML File Modifications

## Head Container <head>

1. Changed <title> content from "website" to "Horiseon | Search Optimization, Online Reputation, and Social Media".

    - Limited character count to 60 in order to optimize browser tab visualization.

    - Incorporated the company name and services to improve search engine ranking and site accessibility. Moreover, the <title> content is an element included in search results and utilized by accessibility resources, such as screen reader extensions.

2. Added "keywords" meta tag with relevant page key words for search engine optimization.
    - Keywords included: "search optimization,reputation management,social media,online marketing,seo,orm,lead generation,brand awareness"

3. Added "description" meta tag to describe the page content for search engine optimization.

    - Description included: "Services to optimize your business digital marketing on search engine and social media platforms."

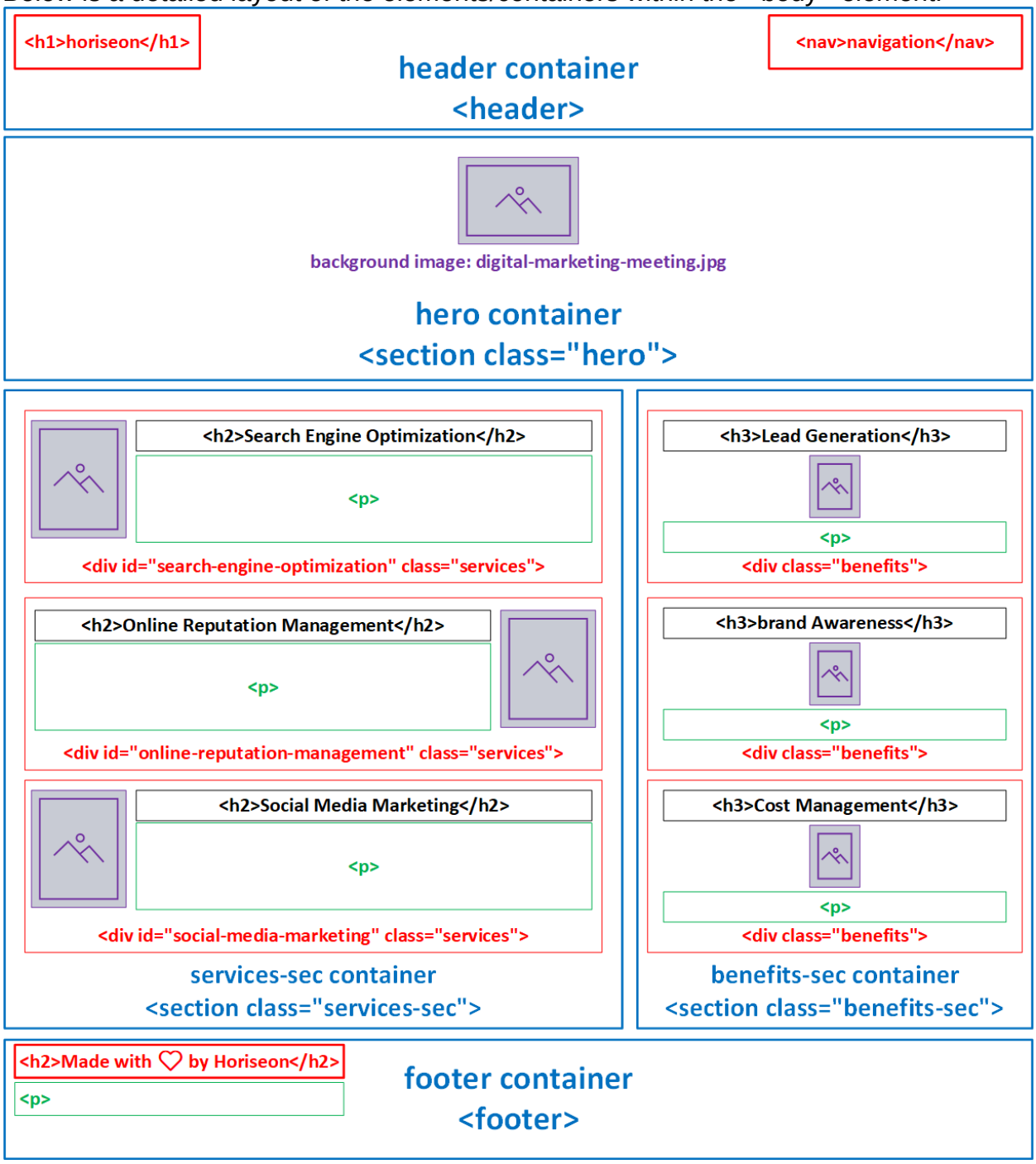
    - The content in the <description> meta tag is an element included in search results of some search engines and utilized by accessibility resources, such as screen reader extensions.

## Body Container - <body>

The <body> element includes the following major elements / containers. Moreover, both the .css and .html files are organized and noted by the following major elements / containers.

| | |
|---|---|
| **header container** | **<header>** |
| **hero container** | **<section class="hero">** |
| **services-sec container** | **<section class="services-sec">** |
| **benefits-sec container** | **<aside class="benefits-sec">** |
| **footer container** | **<footer>** |

Below is a detailed layout of the elements/containers within the <body> element.




## HEADER CONTAINER - <HEADER>

1. Changed the header container from a  to a <header> tag. Designating the container with the <header> element formalizes it as the top-most viewable section of the web page and includes the company's badge and site navigation links.

**Original code**

**New Code**
<header>

2. Changed from a  to a <nav> tag. Although both are block elements the <nav> tag appropriately categorizes the <ul>,<li>,and <a href> content as page navigation code.

## HERO CONTAINER - <SECTION CLASS="HERO">

1. Changed the header container from a  to a <section> tag. Although both are block elements the <section> tag formalizes the hero container as an important section of the page.

**Original code**

**New Code**
<section class="hero"></section>

## SERVICES-SEC CONTAINER - <SECTION CLASS="SERVICES-SEC">

1. Changed services-sec container from a  to a <section> tag. Although both are block elements the <section> tag formalizes the services-sec container as an important section of the page; and

2. Changed the name of the class name for the container from "content" to "services-sec". The class name is more specific to the content nested within its section of the page.

**Original code**

**New Code**
<section class="services-sec">

3. Added id="search-engine-optimization" to the Search Engine Optimization . This enabled the page's navigation link for "Search Engine Optimization" located in the <header>; and
4. Removed the three separate css classes and replaced them with a single class named "services". All three classes had the same styling parameters in the css file.

**Original code**

```
<div class="search-engine-optimization">
<div id="online-reputation-management" class="online-reputation-management">
<div id="social-media-marketing" class="social-media-marketing">
```

**New Code**

```
<div id="search-engine-optimization" class="services">
<div id="online-reputation-management" class="services">
<div id="social-media-marketing" class="services">
```

5. Added alt tags to images, which were missing.

**Original code**

```
<img src="./assets/images/search-engine-optimization.jpg" class="float-left" />
<img src="./assets/images/online-reputation-management.jpg" class="float-right" />
<img src="./assets/images/social-media-marketing.jpg" class="float-left" />
```

**New Code**

```
<img src="./assets/images/search-engine-optimization.jpg" alt="Enhancing search engine placement" class="float-left" />
<img src="./assets/images/online-reputation-management.jpg" alt="Managing online reputation" class="float-right" />
<img src="./assets/images/social-media-marketing.jpg" alt="Effective social media for marketing" class="float-left" />
```

## BENEFITS-SEC CONTAINER <ASIDE CLASS="BENEFITS-SEC">

1. Changed benefits-sec container from a  to the <aside> tag. Although both are block elements the <aside> tag is more appropriate since it's defining the benefits of the services in the Services-Sec Container; and

2. Changed the name of the class name for the container from "benefits" to "benefits-sec". The class name is more specific to the content nested within its section of the page.

**Original Code**

```
<div class="benefits">
```

**New Code**

```
<aside class="benefits-sec">
```

3. Corrected <img> tag closing; and
4. Added alt tags to images, which were missing

**Old Code**

```
<img src="./assets/images/lead-generation.png" />
<img src="./assets/images/brand-awareness.png" />
<img src="./assets/images/cost-management.png"></img>
```

**New Code**

```
<img src="./assets/images/lead-generation.png" alt="Generating business using digital platforms" />
<img src="./assets/images/brand-awareness.png" alt="Increasing visibility of business" />
<img src="./assets/images/cost-management.png" alt="Control digital advertising cost" />
```

5. Removed the three separate css classes and replaced them with a single class named "services". All three classes had the same styling parameters in the css file.

**Original Code**

```
<div class="benefit-lead">
<div class="benefit-brand">
<div class="benefit-cost">
```

**New Code**

```
<div class="benefits">
```

## FOOTER CONTAINER

1. Changed footer from a  to the <footer> tag. Although both are block elements the <footer> tag is more appropriate since it is at the bottom of the page and contain copyright information. By using the <footer> element a separate class name is not necessary.

**Original Code**

```
<div class="footer">
```

**New Code**

```
<footer>
```

# #CSS FILE MODIFICATIONS

1. As previously noted, both the .css and .html files are organized and noted by the following major elements / containers.

| | |
|---|---|
| **header container** | **<header>** |
| **hero container** | **<section class="hero">** |
| **services-sec container** | **<section class="services-sec">** |

| | |
|---|---|
| **benefits-sec container** | **<aside class="benefits-sec">** |
| **footer container** | **<footer>** |

2. Added :root class to CSS to provide definitions for frequently used color and font-family styles

**New Code**

```
:root {
    color-one: #d9dcd6;
    color-two: #ffffff;
    color-three: #0072bb;
    font-family-one: "Trebuchet MS", "Lucida Sans Unicode", "Lucida Grande",
    "Lucida Sans", Arial, Sans-Serif;
    font-family-two: "Gill Sans", "Gill Sans MT", "Calibri", "Trebuchet MS",
    Sans-Serif;
}
```

3. Used var() to replace individual definitions for colors and font-family throughout the style sheet. The var() deployed referenced the variables defined in the newly added :root style.

**Original Code**

```
background-color: #d9dcd6;
background-color: #0072bb;
color: #ffffff;
font-family: "Trebuchet MS", "Lucida Sans Unicode", "Lucida Grande",
"Lucida Sans", Arial, Sans-Serif;
font-family: "Gill Sans", "Gill Sans MT", "Calibri", "Trebuchet MS", Sans-Serif;
```

**New Code**

```
background-color: var(--color-one);
background-color: var(--color-three);
color: var(--color-two);
font-family: var(--font-family-one);
font-family: var(--font-family-two);
```

4. Replaced three classes ("search-engine-optimization","online-reputation-management","online-reputation-management") with a single class named "services". The three classes that were replaced contained the same styling parameters, which were retained in the new class.

5. Replaced three classes ("benefit-lead","benefit-brand","benefit-cost") with a single class named "benefits". The three classes that were replaced contained the same styling parameters, which were retained in the new class.

6. Renamed class from ".header " to "header "
7. Renamed class from ".header h1" to "header h1"
8. Renamed class from ".header div" to "header nav"
9. Renamed class from ".header div ul" to "header nav ul"
10. Renamed class from ".header div ul li" to "header nav ul li"
11. Renamed class from ".footer" to "footer"
12. Renamed class from ".footer h2" to "footer h2"