ISML Site Documentation

Site Maintainance:

The site is maintained using Github Desktop, via pushes and pulls on the site repository. A tutorial can be found in the link below:

"Learn how to set up, authenticate, and configure GitHub Desktop to allow you to contribute to projects directly from your machine": https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop

Site Hosting:

The site is hosted on github. If the site needs to be hosted on a different github account, such as the unversity account, all the files on the old repositories can be copied to the new one. The host can then follow the tutorial below to host the new repository as a website:

"Steps for Hosting a Website on GitHub": https://gist.github.com/TylerFisher/6127328

HTML Header and CSS:

Each html page has a head/header portion that contains codes for specifying page properties. There is a link to an external css file, a title that specifies web page title when viewed from a browser, and a mobile device scaling instruction.

The pages are built using html and css. Structures and organization are provided by html pages, while the css provides styling. Each html header contain a reference link to an external css file.

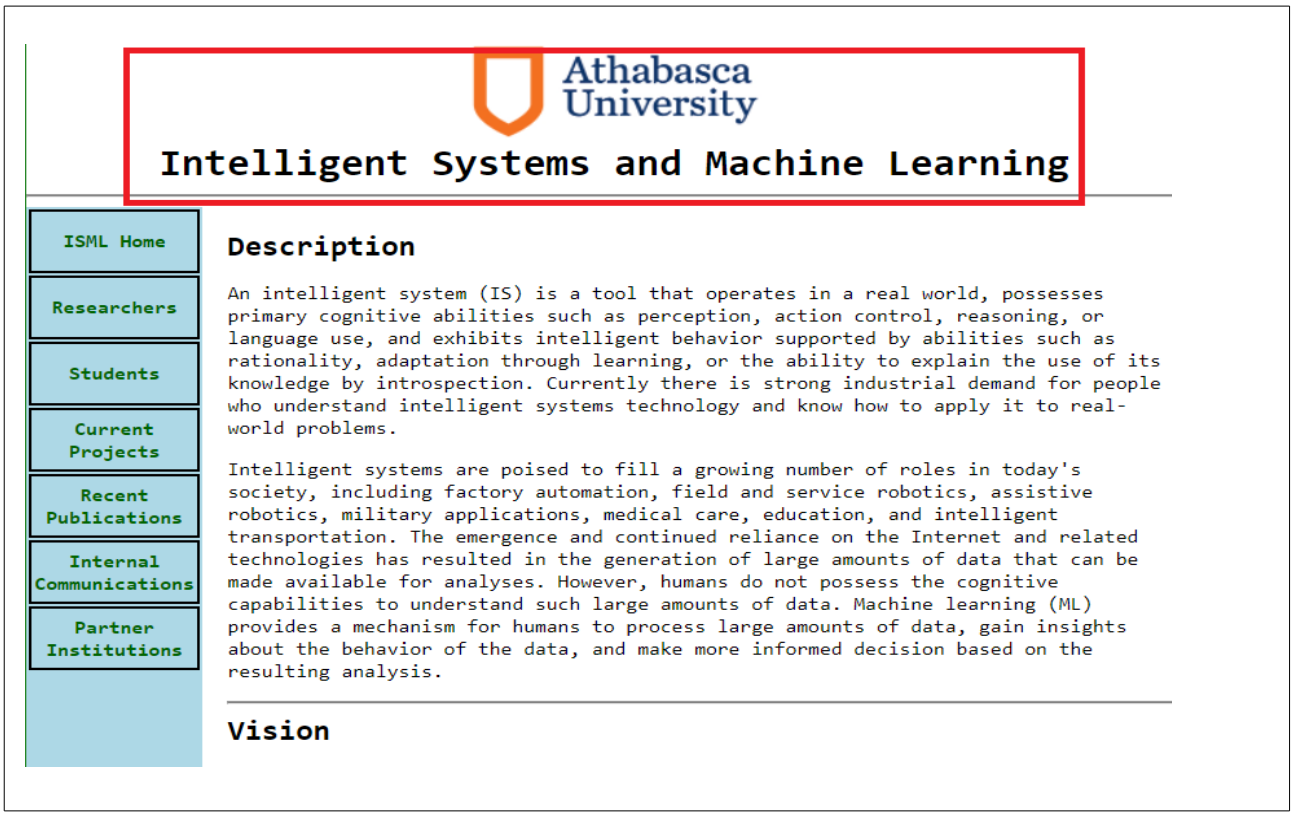
Example of css reference in the header:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!-- Head: -->
  <head>

    <!-- Link to CSS -->
    <link rel="stylesheet" href="ExternalStyle.css">

    <!-- Page bar title: -->
    <title>
      AU ISML Cluster
    </title>

    <!-- Mobile device scaling: -->
    <meta name="viewport" content="width = device-width, initial-scale = 1">
  </head>

  <!-- Body: -->
  <body>
    <!-- Header: -->
    <div class="header">
      <h1>
        <img src= "Img/AU_Logo_Horiz.jpg" class="img_Standalone_Small">
        Intelligent Systems and Machine Learning
      </h1>
    </div>
```

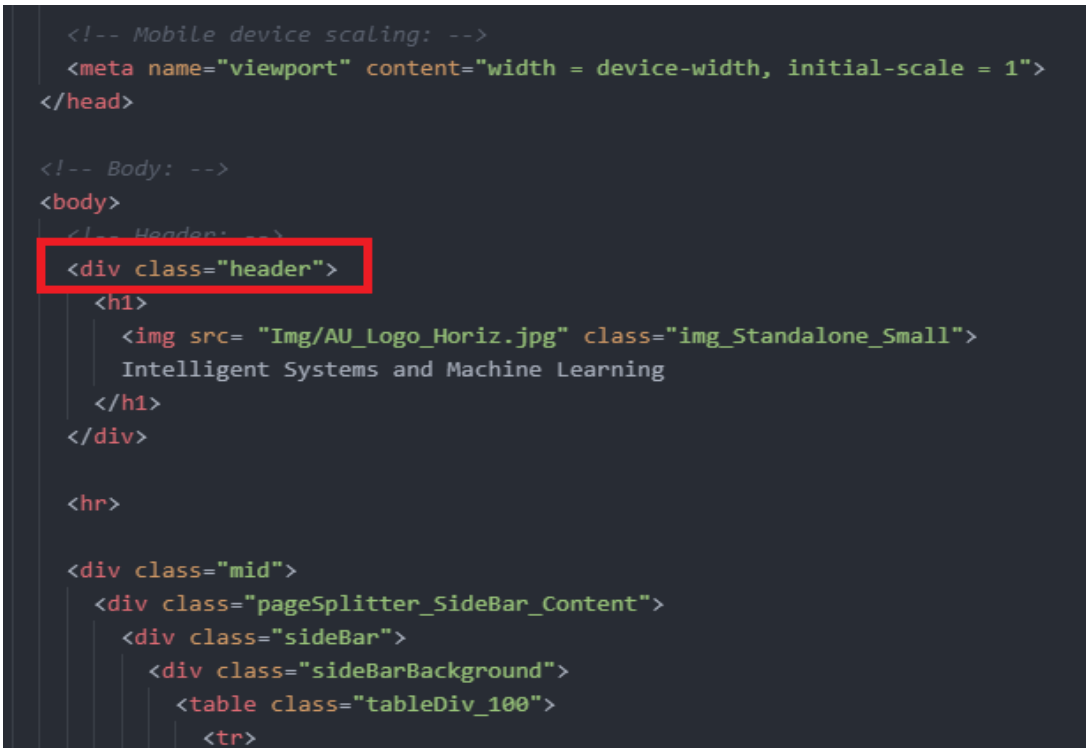
HTML Body Structure:

The body contains actual web page contents. Website elements are organized under divisions, implemented as , as shown in the example below.

The header element is highlighted in red:



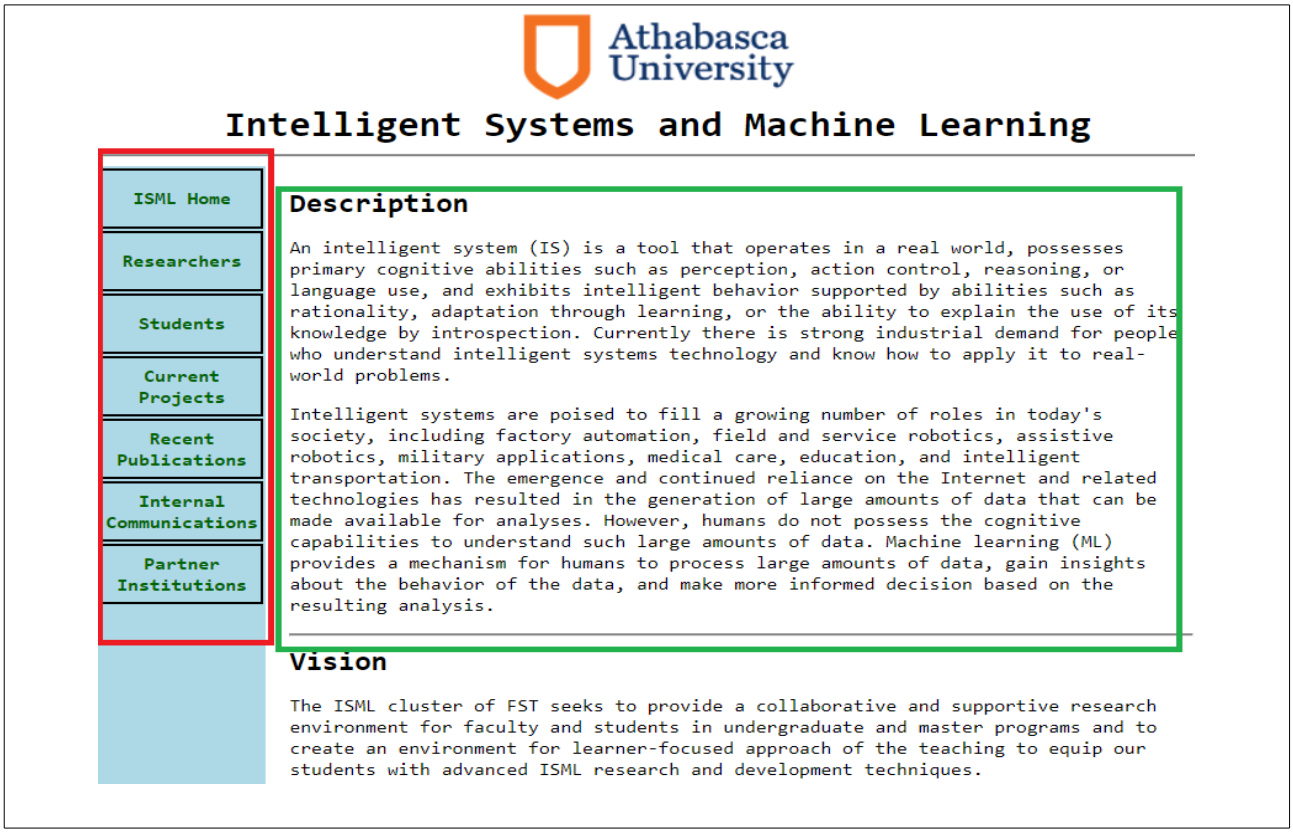
Header code:



```
    <!-- Mobile device scaling: -->
    <meta name="viewport" content="width = device-width, initial-scale = 1">
</head>

<!-- Body: -->
<body>
    <!-- Header: -->
    <div class="header">
        <h1>
            <img src= "Img/AU_Logo_Horiz.jpg" class="img_Standalone_Small">
            Intelligent Systems and Machine Learning
        </h1>
    </div>

    <hr>

    <div class="mid">
        <div class="pageSplitter_SideBar_Content">
            <div class="sideBar">
                <div class="sideBarBackground">
                    <table class="tableDiv_100">
                        <tr>
```

Elements which are can be nested under other , based on their visual structure. In the example below, the side bar and page content divisions are both nested under body. This provides a way to specify classes, for further css styling.

The side bar element is highlighted in red, while the page content is in green:



Side bar and page content codes:

```
<!-- Header: -->
<div class="header">
  <h1>
    <img src= "Img/AU_Logo_Horiz.jpg" class="img_Standalone_Small">
    Intelligent Systems and Machine Learning
  </h1>
</div>

<hr>

<div class="mid">
  <div class="pageSplitter_SideBar_Content">
    <div class="sideBar">
      <div class="sideBarBackground">
      </div>
    </div> <!-- End side bar -->

    <div class="pageContent">
    </div>
  </div>
</div>
<hr>

<!-- Footer: -->
<div class="footer">
  <div class="tableDiv_10">
```

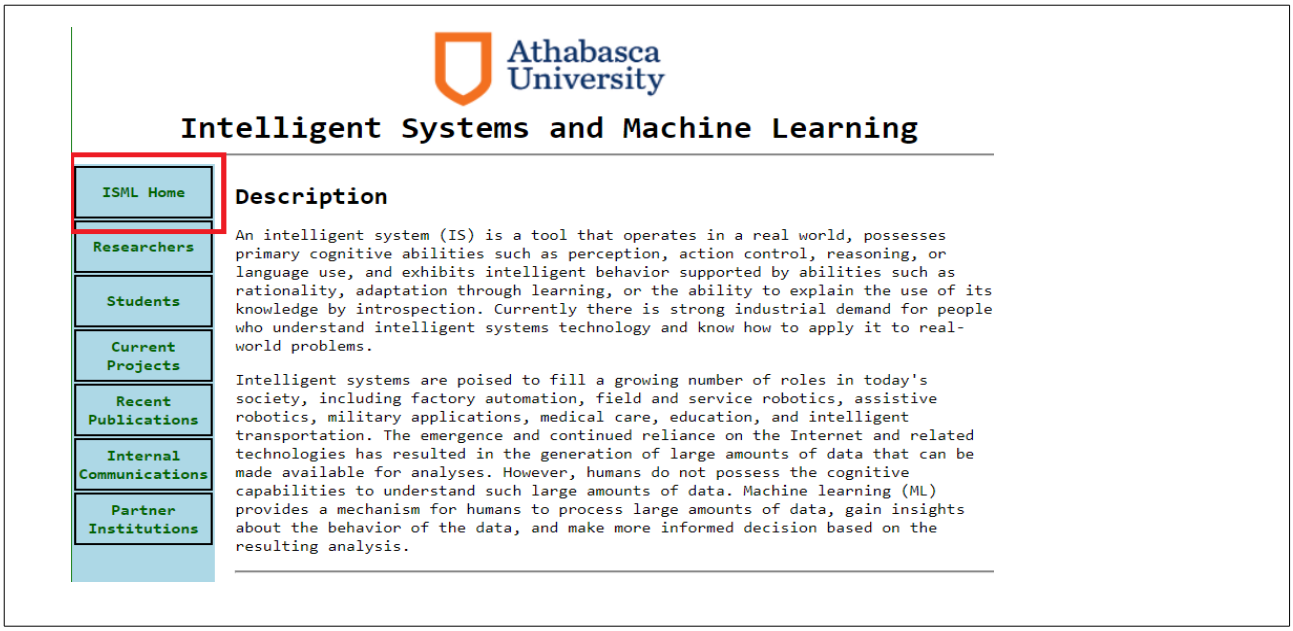
CSS Sheet:

ExternalStyle.css contains styling properties for all html classes. It can also be used as a reference for all html classes; comments in the css should describe which elements are being styled. Ideally each element, such as a or <table>, in the pages would have an associated class, and the class style would be described in the css sheet. Further works on the pages should continue to do this, so css styling can be applied uniformly.

Developers should be aware of the default classes. These are highlighted in red. They can conflict with more specific classes. Ideally they are used to describe very general elements, such as all the links on a page. If there is a need for a specially styled link, then the general class "a {}" in css should be deleted; afterward, specific <a class=customLink> in html and ".customLink {}" in css should be implemented.

An example of visual appearence, html structure, and css styling is described below.

Link as shown:



HTML implementation:

```
<div class="mid">
  <div class="pageSplitter_SideBar_Content">
    <div class="sideBar">
      <div class="sideBarBackground">
        <table class="tableDiv_100">
          <tr>
            <th>
              <a href="IMSL_Home.html"> <div class="a_Backbround"> ISML Home </div> </a>
            </th>
          </tr>
          <tr>
            <th>
              <a href="IMSL_Researchers.html"> <div class="a_Backbround"> Researchers </div>  </a>
            </th>
          </tr>
          <tr>
            <th>
              <a href="IMSL_Students.html"> <div class="a_Backbround"> Students </div> </a>
            </th>
          </tr>
```

CSS styling of class a{}, which represents the link text, and class .a_Background, which represents the blue area behind the link text:

```css
/* Default class for all links.  */
a {
  font-family: "Serif", University, monospace;
  font-size: 100%;
  font-weight: bold;
  text-decoration: none; /* No underline.  */
  color: darkgreen;
  text-align: center;
  height: 100%;
}

/* Style for the background of all links. */
.a_Background{
  height: 100%;
  background-color: lightblue;
  display: flex;
  align-items: center;
  justify-content: center;
}
```