

Health Advice Group - Development

Table of Contents

Colour and Typography	3
Shared CSS	3
Navigation bar and Footer	4
Shared User Interface	4
Shared Layout	4
Navigation Bar and Footer CSS	5
Home page	5
Home User Interface	5
Home View	6
Home CSS	6
Advice and Information page	7
Advice and Information User Interface	7
Advice and Information View	7
Advice and Information CSS	8
Risk Assessments page	8
Risk Assessments User Interface	8
Risk Assessments View	9
Risk Assessments CSS	9
Weather page	10
Weather Search User Interface	10
Weather City User Interface	10
Search City View	11
City Model	11
Weather Controller	12
Constant Config	12
Forecast Repository	13
City View	13
Weather CSS	14
Support page	14
Support User Interface	14
Support View	15
Support CSS	15
Sign up page	16
Sign up User Interface	16

Sign up View	16
Sign up InputModel	17
Login page	17
Login User Interface	17
Login View	18
Login InputModel	18
Login Validation	19
My Account page	19
My Account User Interface	19
Manage Email User Interface	20
Change Password User Interface	20
Change Password View	21
Change Password InputModel	21
Account Deletion User Interface	22
Account Deletion View	22
Account Deletion Validation	23
Database	23
User Table	23
User Data	23

Colour and Typography

Shared CSS

```
site.css  [X]
1  :root {
2      /* Custom variables*/
3      --blue-color: #006cd1;
4      --brown-color: #994f00;
5      --black-color: #000;
6      --white-color: #fff;
7      --text-font: Tahoma;
8      --heading-font: "Treichet MS";
9  }
10
```

These are custom variables that I have created for the main colours and the font. This makes it easier for future programmers as they will not need to remember the hex code for the colour or the font name. Additionally, they are global and can be used anywhere within the project as they have been declared within :root.

```
51  /*Text CSS*/
52  h1 {
53      font-size: 1.6rem;
54      font-family: var(--heading-font), Arial, sans-serif;
55      color: var(--blue-color);
56      font-weight: bold;
57  }
58
59  h2 {
60      font-size: 1.3rem;
61      font-family: var(--heading-font), Arial, sans-serif;
62      color: var(--brown-color);
63      font-weight: bold;
64      text-align: left;
65  }
66
67  h3 {
68      font-size: 1.3rem;
69      font-family: var(--heading-font), Arial, sans-serif;
70  }
71
72  p {
73      font-size: 1rem;
74      font-family: var(--text-font), Arial, sans-serif;
75      text-align: left;
76  }
77
78  li {
79      font-size: 1rem;
80      font-family: var(--text-font), Arial, sans-serif;
81      text-align: left;
82  }
```

Rem has been used instead of px for the font size to make it more responsive for smaller screen sizes.

A different font style is used for headings and paragraphs so that it is easier to see where a section ends. Additionally fall back fonts have been added just in case the initial font does not display.

Navigation bar and Footer

Shared User Interface

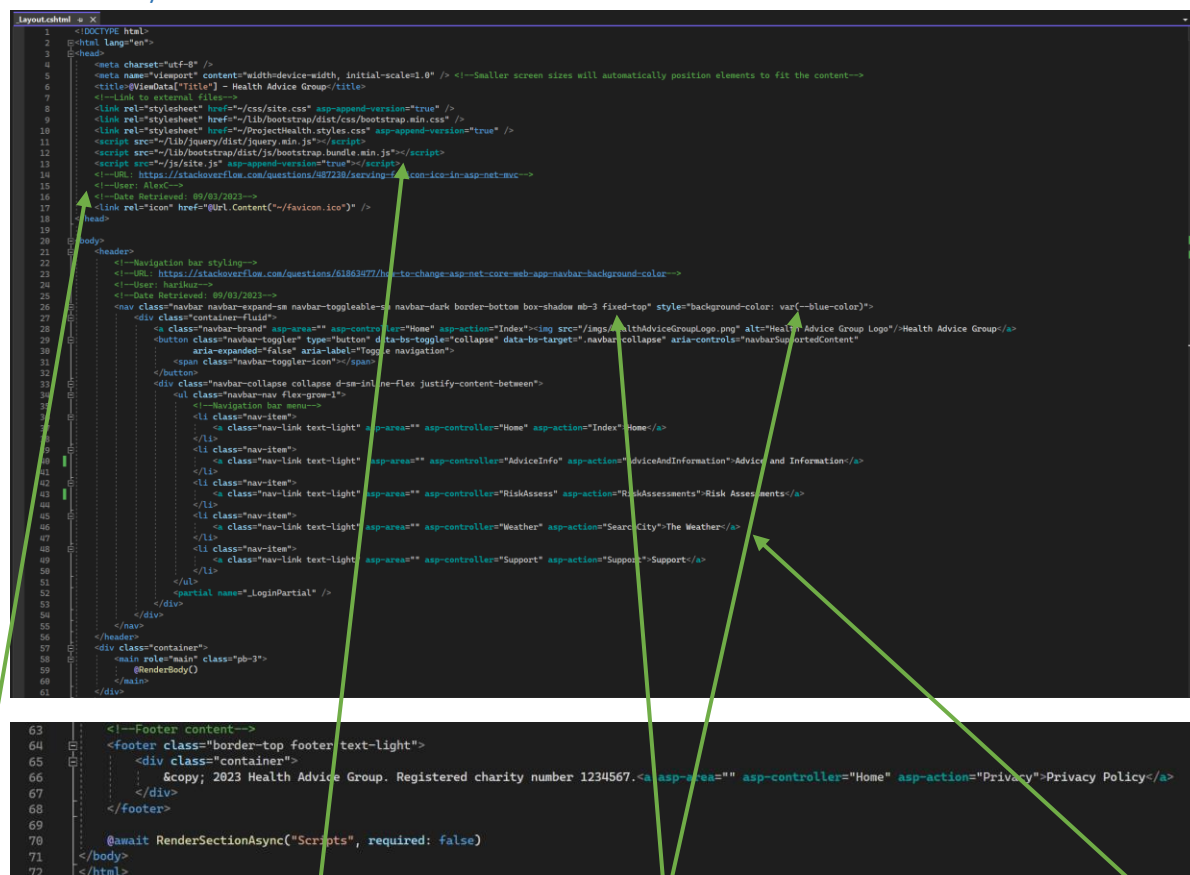


This is the navigation bar for the website. The navigation bar is in blue as it is the primary colour for the website and it gives feelings of trust. A logo has also been added to the left of the navigation bar to add identity to the website so that the user knows this is Health Advice Group's website. There are links to Home, Advice and Information, Risk Assessments, The Weather and Support to the left of the website. These options will not change whether you are logged in or not but Sign up and Login will change to My Account and Logout when logged in, so they are to the right of the navigation bar.



This is the footer for the website. There is a copyright symbol to show that the website was published in 2023. Additionally there will be the registered charity number in the footer to show that the charity is legitimate. The Privacy Policy can also be accessed through the footer on any page but is not on the navigation bar as it is not actively accessed.

Shared Layout



There are comments with references to websites throughout the project which were used to help create the solution.

These links at the top are links to bootstrap as well as the CSS style sheet and the favicon image.

Bootstrap is used for the navigation bar to make it fixed to the top of the page. Also, CSS has been used to make the background of the navigation bar blue.

These are the links on the navigation bar to the controller and action for the page to display.

Navigation Bar and Footer CSS

```

34  /*Navigation Bar and Footer*/
35  .navbar-brand img {
36      width: 50px;
37      height: auto;
38      padding: 5px 5px;
39  }
40
41  footer {
42      background-color: var(--brown-color);
43      overflow: auto;
44  }
45
46  footer a:link {
47      color: var(--white-color);
48      float: right;
49  }

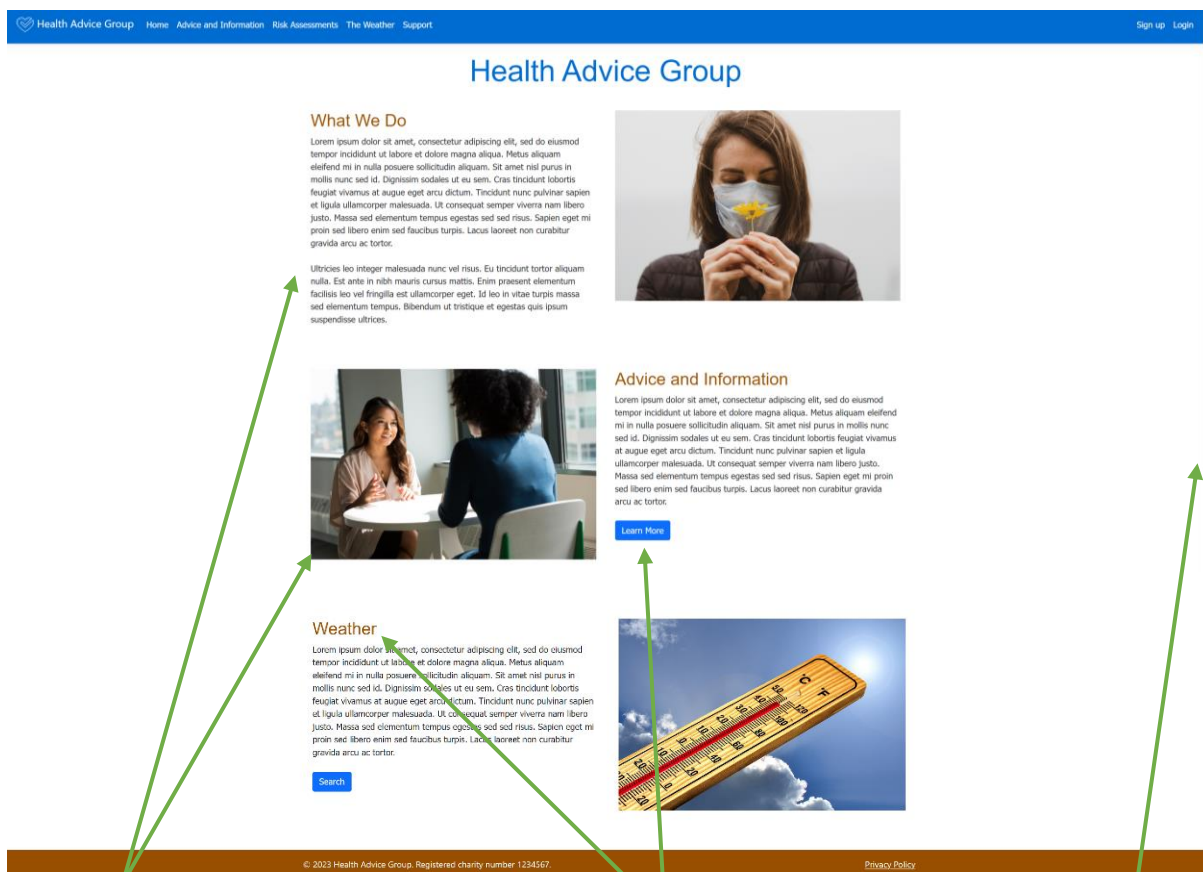
```

To make the image inside the navigation bar fit inside, the width has been set to 50px and padding has been added to add space between the image and the text.

This moves the Privacy Policy link to the right of the footer to separate it from plain text like the registered charity number.

Home page

Home User Interface



There are simple paragraphs explaining what each section does with an image to the left or right to visually represent it.

These sections have been added to the home page with buttons as these pages will be accessed more regularly than the others.

There is a scroll bar to the right so that the user can scroll down as not all of the content is viewable without scrolling.

Home View

```

5 <div class="text-center">
6 <!--Home page title-->
7 <h1 class="display-4">Health Advice Group</h1>
8 <br />
9
10 <div class="homecontainer">
11 <div class="homerow">
12 <!--What We Do subheading-->
13 <div class="homecolumn">
14 <h2>What We Do</h2>
15 <!--What We Do description-->
16 <p class="lefttext">
17 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
18 Metus aliquam eleifend mi in nulla posuere sollicitudin aliquam.
19 Sit amet nisl purus in mollis nunc sed id. Dignissim sodales ut eu sem.
20 Cras tincidunt lobortis feugiat vivamus at augue eget arcu dictum.
21 Tincidunt nunc pulvinar sapien et ligula ullamcorper malesuada.
22 Ut consequat semper viverra nam libero justo. Massa sed elementum tempus egestas sed sed risus.
23 Sapien eget mi proin sed libero enim sed faucibus turpis.
24 Lacus laoreet non curabitur gravida arcu ac tortor.
25 <br />
26 <br />
27 Ultricies leo integer malesuada nunc vel risus.
28 Eu tincidunt tortor aliquam nulla.
29 Est ante in nibh mauris cursus mattis.
30 Enim praesent elementum facilisis leo vel fringilla est ullamcorper eget.
31 Id leo in vitae turpis massa sed elementum tempus.
32 Bibendum ut tristique et egestas quis ipsum suspendisse ultrices.
33 </p>
34 </div>
35 <div class="homecolumn">
36 <!--Image of the charity-->
37 
38 </div>
39 </div>
40 <br />
41 <div class="homerow">
42 <div class="homecolumn">
43 <!--Image for advice and information-->
44 
45 </div>
46 <div class="homecolumn">
47 <!--Advice and Information subheading-->
48 <h2>Advice and Information</h2>
49 <!--Advice and Information description-->
50 <p class="righttext">
51 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
52 Metus aliquam eleifend mi in nulla posuere sollicitudin aliquam.
53 Sit amet nisl purus in mollis nunc sed id. Dignissim sodales ut eu sem.
54 Cras tincidunt lobortis feugiat vivamus at augue eget arcu dictum.
55 Tincidunt nunc pulvinar sapien et ligula ullamcorper malesuada.
56 Ut consequat semper viverra nam libero justo. Massa sed elementum tempus egestas sed sed risus.
57 Sapien eget mi proin sed libero enim sed faucibus turpis.
58 Lacus laoreet non curabitur gravida arcu ac tortor.
59 <br />
60 <a class="btn btn-primary" role="button" asp-area="" asp-controller="AdviceInfo" asp-action="AdviceAndInformation">Learn More</a>
61 </p>
62 </div>
63 </div>
64 </div>

```

Classes have been used to group text and images that will be styled and aligned similarly.

This is a button which has been styled with bootstrap and redirects the user to the Advice and Information page.

Home CSS

```

84 /* Home Page */
85
86 #homecontainer{
87   margin: 20px 20px 0 0;
88   border: 5px solid #333;
89 }
90
91 .rightimage {
92   float: right;
93   height: auto;
94   max-width: 100%;
95 }
96
97 .lefttext {
98   float: left;
99   padding-right: 10px;
100   /* URL: https://stackoverflow.com/questions/8369819/how-do-i-make-a-text-go-onto-the-next-line-if-it-overflows */
101   /* User: Serina Patterson & David Farinelli */
102   /* Date Retrieved: 09/03/2023 */
103   word-wrap: break-word;
104 }
105
106 .leftimage {
107   float: left;
108   height: auto;
109   max-width: 100%;
110 }
111
112 .righttext {
113   float: right;
114   padding-left: 10px;
115   word-wrap: break-word;
116 }
117
118 .homerow {
119   display: flex;
120 }
121
122 .homecolumn {
123   flex: 50%;
124   padding: 10px 20px;
125 }

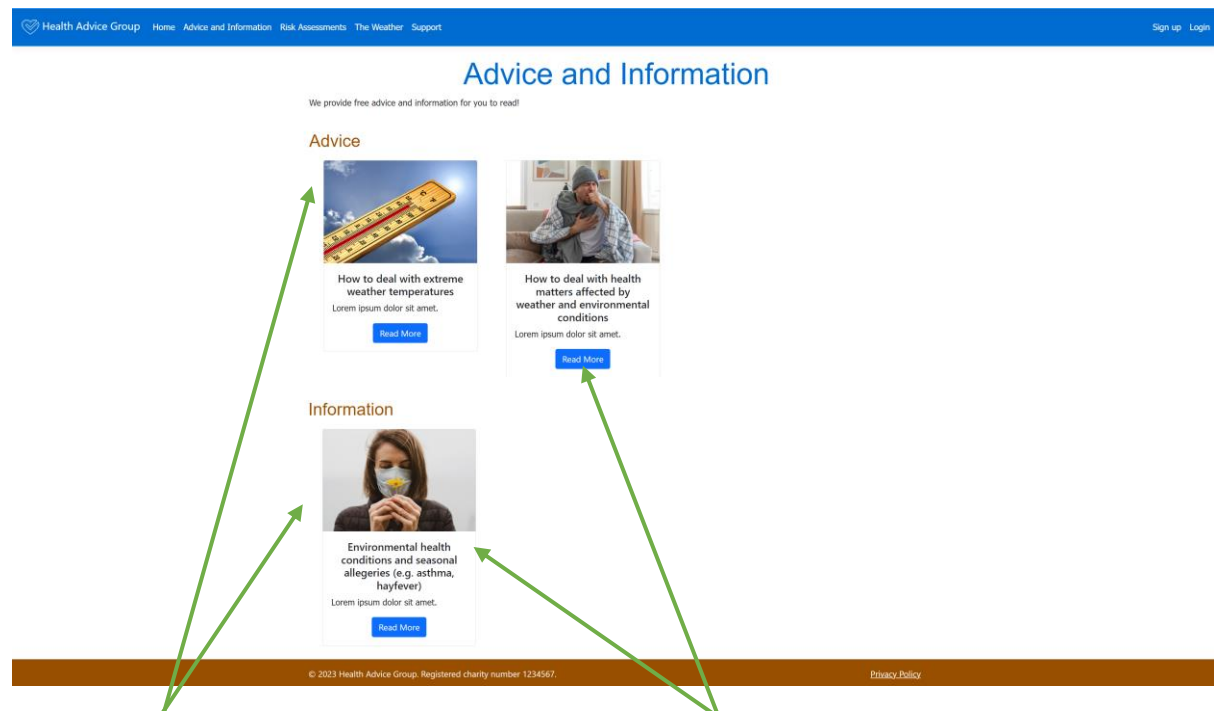
```

The text has been word wrapped to break-word so that the word continues on in the next line and does not overlap with the image.

Columns and rows have been used to align the text and image horizontally with each column taking up 50% of the container.

Advice and Information page

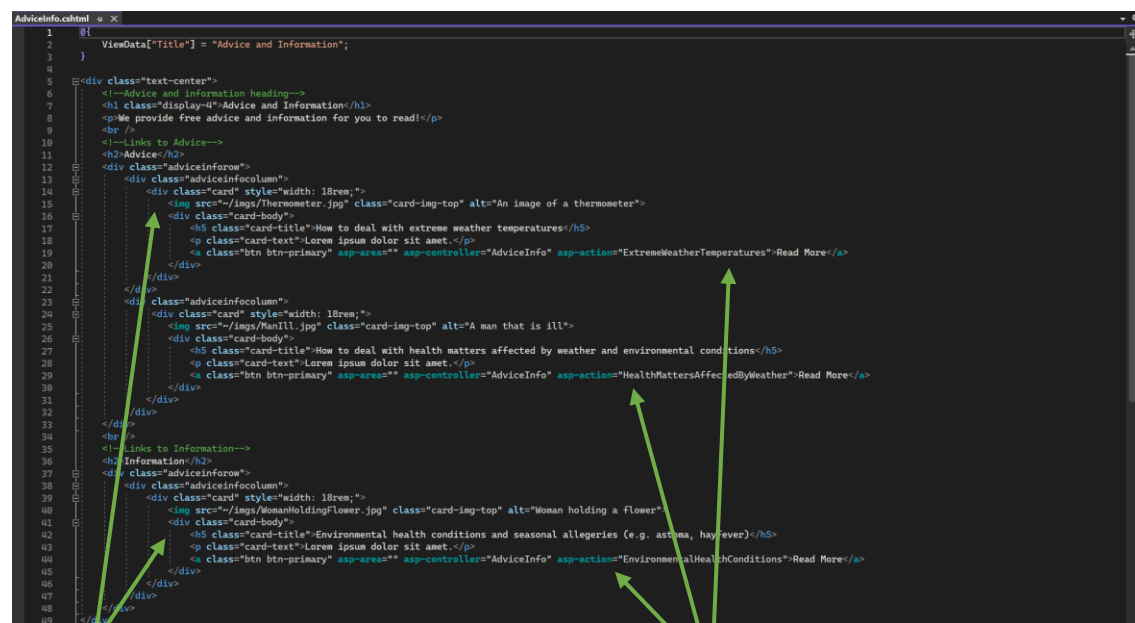
Advice and Information User Interface



Advice and Information have been separated with headings so that the user can differentiate them.

Cards have been created with the help of bootstrap to make the page more pleasing to the eye and organised. These cards have an image, title, a little description and a button which a user can press to view the article.

Advice and Information View



These are the cards that have been styled with bootstrap and organised within rows and columns.

These are buttons that direct the user to the corresponding article page from the card.

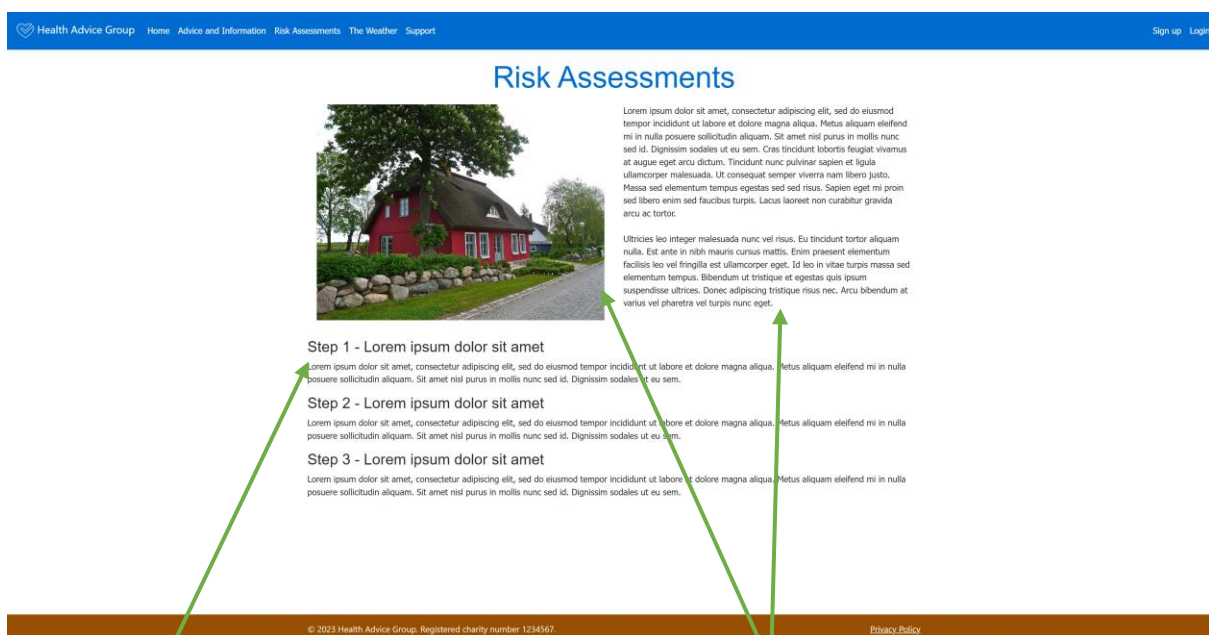
Advice and Information CSS

```
123  /* Advice and Information Page */
124
125  .adviceinfo {
126      display: flex;
127  }
128
129  .adviceinfo {
130      padding: 10px 30px;
131  }
132
```

Flex has been used to align the cards horizontally, next to each other.

Risk Assessments page

Risk Assessments User Interface



A sub heading for steps have been included and a few sentences underneath to explain ways to reduce risks at home.

An image of a house has been added to the left of the page and a couple of paragraphs explaining home risk assessments and how they help.

Risk Assessments View

```

1  <div class="text-center">
2    <!--Risk assessments title-->
3    <h1 class="display-4">Risk Assessments</h1>
4    <div class="riskassessrow">
5      <div class="riskassesscolumn">
6        
7      </div>
8      <div class="riskassesscolumn">
9        <!--Risk assessments text-->
10       <div id="riskassesssctext">
11         <p>
12           Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
13           Metus aliquam eleifend mi in nulla posuere sollicitudin aliquam.
14           Sit amet nisl purus in mollis nunc sed id. Dignissim sodales ut eu sem.
15           Cras tincidunt litora fusce vivamus et augue eget arcu dictum.
16           Tincidunt nunc pulvinar sapien et ligula ullamcorper malesuada.
17           Ut consequat semper viverra nam libero justo. Massa sed elementum tempus egestas sed sed risus.
18           Suspendisse eget mi proin sed libero ante sed fames turpis.
19           Lacus laoreet non curabitur gravida arcu ac tortor.
20         </p>
21         <p>
22           Ultricies leo integer malesuada nunc vel risus.
23           Et tincidunt tortor aliquam nulla.
24           Est ante in nibh mauris cursus mattis.
25           Enim praesent elementum facilisis leo vel fringilla est ullamcorper eget.
26           Id leo in vitae turpis massa sed elementum tempus.
27           Bibendum ut tristique et egestas quis ipsum suspendisse ultrices.
28           Donec adipiscing tristique risus nec. Arcu bibendum at varius vel pharetra vel turpis nunc eget.
29         </p>
30       </div>
31     </div>
32   </div>
33   <!--List of steps-->
34   <div class="riskstep">Step 1 - Lorem ipsum dolor sit amet</div>
35   <div class="riskstep">
36     <p>
37       Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
38       Metus aliquam eleifend mi in nulla posuere sollicitudin aliquam.
39       Sit amet nisl purus in mollis nunc sed id. Dignissim sodales ut eu sem.
40     </p>
41   </div>
42   <div class="riskstep">Step 2 - Lorem ipsum dolor sit amet</div>
43   <div class="riskstep">
44     <p>
45       Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
46       Metus aliquam eleifend mi in nulla posuere sollicitudin aliquam.
47       Sit amet nisl purus in mollis nunc sed id. Dignissim sodales ut eu sem.
48     </p>
49   </div>
50   <div class="riskstep">Step 3 - Lorem ipsum dolor sit amet</div>
51   <div class="riskstep">
52     <p>
53       Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
54       Metus aliquam eleifend mi in nulla posuere sollicitudin aliquam.
55       Sit amet nisl purus in mollis nunc sed id. Dignissim sodales ut eu sem.
56     </p>
57   </div>
58 </div>

```

Columns have been used to align the image and paragraphs next to each other.

All steps have been assigned the same class so that they are all styled the same.

Risk Assessments CSS

```

149 /* Risk Assessments Page */
150 #riskassesssctext {
151   text-align: left;
152   float: left;
153   max-width: 100%;
154   word-wrap: break-word;
155 }
156
157 #riskassessimage {
158   max-width: 100%;
159   height: auto;
160 }
161
162 .riskassessrow {
163   display: flex;
164 }
165
166 .riskassesscolumn {
167   flex: 50%;
168   padding: 10px 20px;
169 }
170
171 .riskstep {
172   text-align: left;
173 }

```

The width of the text and image are 100% so that it takes up all of the space within the 50% column.

Weather page

Weather Search User Interface

Health Advice Group Home Advice and Information Risk Assessments The Weather Support Sign up Login

The Weather

Select a city from the list and press search to view weather data on that city.

Select a City

City Name
London

Search

© 2023 Health Advice Group. Registered charity number 1234567. Privacy Policy

This is a button that can be pressed to search for the weather data of the selected city.

A select menu has been added to limit input errors from users who incorrectly spell a city's name. There are only three cities to select from but more can be added.

Weather City User Interface

Health Advice Group Home Advice and Information Risk Assessments The Weather Support Sign up Login

The Weather

Weather Forecast

City: London

Weather Condition: Clouds
Temperature: 12.97 °C
Wind Speed: 8.75 Km/h
Pressure: 997 Hpa
Humidity: 66 %

Return

© 2023 Health Advice Group. Registered charity number 1234567. Privacy Policy

Real time data is displayed on the weather condition, temperature, wind speed, pressure and humidity of the city that was previously selected.

This a button that can be pressed to return back to the initial weather page to select a different city.

Search City View

```
1 @model ProjectHealth.Models.SearchCity
2
3 @{
4     ViewData["Title"] = "SearchCity";
5 }
6 <h1 class="display-4" style="text-align: center;">The Weather</h1>
7 <div class="container">
8     <p>Select a city from the list and press search to view weather data on that city.</p>
9     <form method="post">
10         <div class="container">
11             <div class="form-group col-md-offset-3 col-md-5">
12                 <h2>Select a City</h2>
13                 <!--URL: https://medium.com/@mignaz/create-your-forecast-application-with-net-core-and-dockerize-it-945329e8fa83-->
14                 <!--User: Carlos Martin-->
15                 <!--Date Retrieved: 17/03/2023-->
16
17                 <label asp-for="CityName" style="font-family: var(--text-font);"></label>
18
19                 <!--Drop down list to select a City-->
20                 <select asp-for="CityName" class="form-select form-select-md" aria-label="form-select-md SelectCity">
21                     <option value="London">London</option>
22                     <option value="Belfast">Belfast</option>
23                     <option value="Edinburgh">Edinburgh</option>
24                 </select>
25                 <span asp-validation-for="CityName" class="text-danger"></span>
26             </div>
27         </div>
28     </form>
29
30     <!--Search button-->
31     <div class="container">
32         <div class="form-group col-md-offset-3 col-md-5">
33             <button asp-controller="Weather" asp-action="SearchCity" class="btn btn-primary">Search</button>
34         </div>
35     </div>
36 </div>
37 </div>
38 </div>
```

This is the select menu and the three cities that have been added: London, Belfast and Edinburgh.

This is the search button that will take the city selected and use it within the controller and actions.

City Model

```
1 using System.Collections.Generic;
2 using System.ComponentModel.DataAnnotations;
3 using System.Linq;
4 using System.Threading.Tasks;
5
6 namespace ProjectHealth.Models
7 {
8     5 references
9     public class City
10     {
11         // URL: https://medium.com/@mignaz/create-your-forecast-application-with-net-core-and-dockerize-it-945329e8fa83
12         // User: Carlos Martin
13         // Date Retrieved: 17/03/2023
14
15         [Display(Name = "City:")]
16         3 references
17         public string? Name { get; set; }
18
19         [Display(Name = "Temperature:")]
20         4 references
21         public float Temp { get; set; }
22
23         [Display(Name = "Humidity:")]
24         3 references
25         public int Humidity { get; set; }
26
27         [Display(Name = "Pressure:")]
28         3 references
29         public int Pressure { get; set; }
30
31         [Display(Name = "Wind Speed:")]
32         3 references
33         public float Wind { get; set; }
34
35         [Display(Name = "Weather Condition:")]
36         3 references
37         public string? WeatherCondition { get; set; }
38     }
39 }
```

A display name is assigned so that it is easier for the user to understand the variable and to include spaces between the words.

Each variable has been declared as public and each have a sensible data type.

Weather Controller

```
WeatherController.cs
ProjectHealth
ProjectHealth.Controllers.WeatherController
SearchCity()

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using ProjectHealth.Models;
6 using ProjectHealth.OpenWeatherMapModels;
7 using ProjectHealth.Repositories;
8
9 using Microsoft.AspNetCore.Mvc;
10
11 namespace ProjectHealth.Controllers
12 {
13     public class WeatherController : Controller
14     {
15         // URL: https://medium.com/@signalr/create-your-forecast-application-with-net-core-and-dockerize-it-9d5329e6a81
16         // User: Carlos Martin
17         // Date Retrieved: 17/03/2023
18
19         // GET: SearchCity
20         [HttpGet]
21         public IActionResult SearchCity()
22         {
23             var viewModel = new SearchCity();
24             return View(viewModel);
25         }
26
27         // POST: SearchCity
28         [HttpPost]
29         public IActionResult SearchCity(SearchCity model)
30         {
31             // If the model is valid, consume the Weather API to bring the data of the city
32             if (ModelState.IsValid)
33             {
34                 return RedirectToAction("City", "Weather", new { city = model.CityName });
35             }
36             return View(model);
37         }
38
39         // GET: City
40         [HttpGet]
41         public IActionResult City(string city)
42         {
43             // Consume the OpenWeatherAPI in order to bring forecast data to the page.
44             WeatherResponse weatherResponse = _forecastRepository.GetForecast(city);
45             City viewModel = new City();
46
47             if (weatherResponse != null)
48             {
49                 viewModel.Name = weatherResponse.Name;
50                 viewModel.Humidity = weatherResponse.Main.Humidity;
51                 viewModel.Pressure = weatherResponse.Main.Pressure;
52                 viewModel.Temp = weatherResponse.Main.Temp;
53                 viewModel.WeatherCondition = weatherResponse.Weather[0].Main;
54                 viewModel.Wind = weatherResponse.Wind.Speed;
55             }
56             return View(viewModel);
57         }
58
59         private readonly IForecastRepository _forecastRepository;
60
61         // Dependency Injection
62         [Inject]
63         public WeatherController(IForecastRepository forecastAppRepo)
64         {
65             _forecastRepository = forecastAppRepo;
66         }
67     }
68 }
```

When there is valid data it will be assigned to the viewModel variables which can then be displayed on the page.

When a valid city is selected then it will be used to get its data from the API.

Constant Config

```
Constants.cs
ProjectHealth
ProjectHealth.Config.Constants
API_Key

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5
6 namespace ProjectHealth.Config
7 {
8     public static class Constants
9     {
10         // URL: https://openweathermap.org/
11         // User: OpenWeather
12         // Date Retrieved: 17/03/2023
13         public const string API_Key = "5839b5725d46adf0b15dab7f5661d9d08";
14     }
15 }
```

This is a constant variable for the API Key.

Forecast Repository

```

ForecastRepository.cs
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using ProjectHealth.Config;
6 using ProjectHealth.OpenWeatherMapModels;
7 using Newtonsoft.Json;
8 using RestSharp;
9 using Newtonsoft.Json.Linq;
10
11 namespace ProjectHealth.Repositories
12 {
13     1 reference
14     public class ForecastRepository : IForecastRepository
15     {
16         2 references
17         WeatherResponse IForecastRepository.GetForecast(string city)
18         {
19             // URL: https://medium.com/@kmgiaz/create-your-forecast-application-with-net-core-and-dockerize-it-945329e0fa83
20             // User: Carlos Martin
21             // Date Retrieved: 17/03/2023
22
23             string IDOWeather = Constants.API_Key;
24             // Connection String
25             var client = new RestClient($"http://api.openweathermap.org/data/2.5/weather?q={city}&units=metric&appid={IDOWeather}");
26             var request = new RestRequest();
27             var response = client.Execute(request);
28
29             if (response.IsSuccessfull)
30             {
31                 // Deserialize the string content into JObject object
32                 var content = JsonConvert.DeserializeObject<JToken>(response.Content);
33
34                 // Deserialize the JObject object into our WeatherResponse Class
35                 return content.ToObject<WeatherResponse>();
36             }
37
38             return null;
39         }
40     }

```

This deserializes the weather data gathered from the API from string into the class data types.

This uses the selected city and the API Key to get the weather data by using RestSharp.

City View

```

City.cshtml
1 @model ProjectHealth.Models.City
2 @{
3     ViewData["Title"] = "The Weather";
4 }
5 <h1 class="display-4" style="text-align: center;">The Weather</h1>
6 <div class="container text-md-center">
7     <div class="form-group col-md-offset-3 text-black text-md-center">
8         <!--URL: https://medium.com/@kmgiaz/create-your-forecast-application-with-net-core-and-dockerize-it-945329e0fa83-->
9         <!--User: Carlos Martin-->
10         <!--Date Retrieved: 17/03/2023-->
11
12         <!--Weather forecast output-->
13         <h2 style="text-align: center;">Weather Forecast</h2>
14
15         <label asp-for="Name"></label>
16         <span class="badge">@Model.Name</span>
17         <br />
18
19         <label asp-for="WeatherCondition"></label>
20         <span class="badge">@Model.WeatherCondition</span>
21         <br />
22
23         <label asp-for="Temp"></label>
24         <span class="badge">@Model.Temp °C</span>
25         <br />
26
27         <label asp-for="Wind"></label>
28         <span class="badge">@Model.Wind Km/h</span>
29         <br />
30
31         <label asp-for="Pressure"></label>
32         <span class="badge">@Model.Pressure Hpa</span>
33         <br />
34
35         <label asp-for="Humidity"></label>
36         <span class="badge">@Model.Humidity %</span>
37         <br />
38     </div>
39 </div>
40 <div>
41     <!--Return to select a city-->
42     <div class="container text-md-center">
43         <div class="form-group">
44             <button style="background-color: var(--brown-color)" asp-controller="Weather" asp-action="SearchCity" class="btn btn-success">Return</button>
45         </div>
46     </div>
47 </div>

```

These get the data and display it for the user to view on the page.

This is the return button to go back to the search city page.

Weather CSS

```
176  /* Weather Page */
177
178  .badge {
179      font-size: 1rem !important;
180      color: var(--black-color) !important;
181  }
```

!important has been used as it was being overridden by another piece of code.

Support page

Support User Interface

The screenshot shows a web interface for the 'Health Advice Group'. The top navigation bar is blue and contains links: Home, Advice and Information, Risk Assessments, The Weather, and Support. On the right of the bar are 'Sign up' and 'Login' links. The main content area has a light blue header with the word 'Support'. Below this, there are two columns. The left column is titled 'FAQ' and contains three question-and-answer pairs, each with a 'Question:' and 'Answer:' section. The right column is titled 'Contact Information' and contains contact details: Email (support@healthadvicegroup.org.uk), Telephone (01234567890), and Address (1 Health Advice Way, London, United Kingdom). At the bottom, there is a dark blue footer with copyright information: '© 2023 Health Advice Group. Registered charity number 1234567.' and a 'Privacy Policy' link.

This is the FAQ with questions and answers underneath that the user can read.

This is the contact information that the user can read and use to contact the charity.

Support View

```

1  ViewData["Title"] = "Support";
2  }
3
4
5  <div class="text-center">
6      <!--Support page title-->
7      <h1 class="display-4">Support</h1>
8      <div class="supportrow">
9          <div id="faqcolumn">
10             <!--Support page FAQ-->
11             <h2>FAQ</h2>
12             <h3 class="question"><b>Question:</b> Lorem ipsum dolor sit amet, consectetur adipiscing elit?</h3>
13             <p>
14                 <b>Answer:</b> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. A erat nam at lectus.
15                 Aliquam eleifend mi in nulla posuere sollicitudin. Sed felis eget velit aliquet sagittis id.
16                 Et malesuada fames ac turpis egestas sed. Varius morbi enim nunc faucibus a.
17                 Mauris rhoncus aenean vel elit. Pellentesque eu tincidunt tortor aliquam.
18                 In hendrerit gravida rutrum quisque non tellus orci ac auctor. Sed egestas egestas fringilla phasellus faucibus scelerisque eleifend.
19             </p>
20             <br />
21             <h3 class="question"><b>Question:</b> A erat nam at lectus?</h3>
22             <p>
23                 <b>Answer:</b> Nisl suscipit adipiscing bibendum est ultricies integer quis. Sed turpis tincidunt id aliquet risus feugiat.
24                 Tempus imperdiet nulla malesuada pellentesque elit eget gravida cum sociis. Et tortor at risus viverra adipiscing at. Vulputate mi sit amet mauris.
25                 Pretium nibh ipsum consequat nisl vel. Tristique senectus et netus et malesuada fames ac turpis egestas. Integer enim neque volutpat ac tincidunt vitae semper.
26                 Integer vitae justo eget magna. Pulvinar sapien et ligula ullamcorper malesuada proin. Varius quam quisque id diam vel quam.
27                 Habitant morbi tristique senectus et. Porta lorem mollis aliquam ut porttitor leo. Condimentum id venenatis a condimentum vitae sapien pellentesque habitant.
28                 Pellentesque habitant morbi tristique senectus.
29             </p>
30             <br />
31             <h3 class="question"><b>Question:</b> Aliquam eleifend mi in nulla posuere sollicitudin?</h3>
32             <p>
33                 <b>Answer:</b> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
34                 Tristique magna sit amet purus gravida.
35                 Lectus mauris ultrices eros in cursus turpis massa tincidunt.
36             </p>
37         </div>
38         <!--Support page contact information-->
39         <div id="contactcolumn">
40             <h2 id="contact" style="text-align: center">Contact Information</h2>
41             <p><b>Email:</b> support@healthadvicegroup.org.uk</p>
42             <p><b>Telephone:</b> 01234567890</p>
43             <p><b>Address:</b> 1 Heathcote Drive May, <br /> London, United Kingdom</p>
44         </div>
45     </div>
46

```

All of the question headings were assigned to the same class so that they are styled the same.

Bold tags were used to bring emphasis to the contact information for users to notice them quicker.

Support CSS

```

183  /* Support Page */
184  .question {
185      text-align: left;
186      font-size: 1.3rem;
187  }
188
189  .supportrow {
190      display: flex;
191  }
192
193  #faqcolumn {
194      flex: 70%;
195      padding: 10px 20px;
196  }
197
198  #contactcolumn {
199      flex: 30%;
200      padding: 10px 20px;
201  }

```

The FAQ column was given a larger flex area as the questions and answers are longer and contain more text than the contact information.

Sign up page

Sign up User Interface

The screenshot shows the 'Sign up' page of the Health Advice Group website. The page has a blue header with navigation links: Home, Advice and Information, Risk Assessments, The Weather, Support, Sign up, and Login. The main content area is titled 'Sign up' and 'Create an account'. It contains a form with three input fields: 'Email', 'Password', and 'Confirm Password'. Below the fields is a blue 'Sign up' button. A green arrow points from the 'Sign up' button to the 'Sign up' text in the header. Another green arrow points from the 'Sign up' button to the 'Sign up' text in the main content area. A third green arrow points from the 'Sign up' button to the 'Sign up' text in the footer.

This is where a user can input their email, a suitable password and the confirmation password.

This is the sign up button that the user can press after inputting their email and password.

Sign up View

```
1 @page
2 @model RegisterModel
3 @*
4   ViewData["Title"] = "Sign up";
5   *@
6
7 <h1 class="display-4">@ViewData["Title"]</h1>
8
9 <div class="row">
10   <div class="col-md-4">
11     <form id="registerForm" asp-route-returnUrl="@Model.ReturnUrl" method="post">
12       <!-- Sign up heading -->
13       <h2>Create an account</h2>
14       <hr />
15       <div asp-validation-summary="ModelOnly" class="text-danger" role="alert"></div>
16       <!-- Email input -->
17       <div class="form-floating mb-3">
18         <input asp-for="Input.Email" class="form-control" autocomplete="username" aria-required="true" placeholder="name@example.com" />
19         <label asp-for="Input.Email">Email</label>
20       </div>
21       <span asp-validation-for="Input.Email" class="text-danger"></span>
22     </div>
23     <!-- Password input -->
24     <div class="form-floating mb-3">
25       <input asp-for="Input.Password" class="form-control" autocomplete="new-password" aria-required="true" placeholder="password" />
26       <label asp-for="Input.Password">Password</label>
27     </div>
28     <span asp-validation-for="Input.Password" class="text-danger"></span>
29     <!-- Confirm password input -->
30     <div class="form-floating mb-3">
31       <input asp-for="Input.ConfirmPassword" class="form-control" autocomplete="new-password" aria-required="true" placeholder="password" />
32       <label asp-for="Input.ConfirmPassword">Confirm Password</label>
33     </div>
34     <span asp-validation-for="Input.ConfirmPassword" class="text-danger"></span>
35   </div>
36   <div class="col-md-4">
37     <!-- Sign up button -->
38     <button id="registerSubmit" type="submit" class="w-100 btn btn-lg btn-primary">Sign up</button>
39   </div>
40 </div>
41
42 <section Scripts>
43   <partial name="_ValidationScriptsPartial" />
44 </section>
```

A form has been used to allow users to input their email, password and confirmation password.

Aria-required is used to make sure that the user has inputted an email and password before the account is created.

Sign up InputModel

```

58
59 // Sign up InputModel
60 public class InputModel
61 {
62     [Required]
63     [EmailAddress]
64     [Display(Name = "Email")]
65     public string Email { get; set; }
66
67     [Required]
68     [StringLength(20, ErrorMessage = "The {0} must be at least {2} and at max {1} characters long.", MinimumLength = 8)]
69     [DataType(DataType.Password)]
70     [Display(Name = "Password")]
71     public string Password { get; set; }
72
73     [DataType(DataType.Password)]
74     [Display(Name = "Confirm password")]
75     [Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]
76     public string ConfirmPassword { get; set; }
77
78 }
79

```

Compare has been used to make sure that the confirm password is the same as the inputted password.

Error messages have been added to inform the user that what they inputted is invalid.

Login page

Login User Interface

This is where a user can input their email and password to login to their account.

If a user has forgotten their password they can press this link which will send them to the forgot password page

The sign up page can either be accessed by clicking the hyperlink or through the navigation bar.

Login View

```

1 @page
2 @model LoginModel
3
4 @{
5     ViewData["Title"] = "Log in";
6 }
7
8 <h1 class="display-4">@ViewData["Title"]</h1>
9 <div class="row">
10     <div class="col-md-4">
11         <section>
12             <form id="account" method="post">
13                 <!-- Login heading -->
14                 <h2>Log in to your account</h2>
15                 <hr />
16                 <div asp-validation-summary="ModelOnly" class="text-danger" role="alert"></div>
17                 <!-- Email input -->
18                 <div class="form-floating mb-3">
19                     <input asp-for="Input.Email" class="form-control" autocomplete="email" aria-required="true" placeholder="email@example.co.uk" />
20                     <label asp-for="Input.Email" class="form-label">Email</label>
21                     <span asp-validation-for="Input.Email" class="text-danger"></span>
22                 </div>
23                 <!-- Password input -->
24                 <div class="form-floating mb-3">
25                     <input asp-for="Input.Password" class="form-control" autocomplete="current-password" aria-required="true" placeholder="password" />
26                     <label asp-for="Input.Password" class="form-label">Password</label>
27                     <span asp-validation-for="Input.Password" class="text-danger"></span>
28                 </div>
29                 <!-- Remember me checkbox -->
30                 <div class="checkbox mb-3">
31                     <label asp-for="Input.RememberMe" class="form-label">
32                         <input class="form-check-input" asp-for="Input.RememberMe" />
33                         @Html.DisplayNameFor(m => m.Input.RememberMe)
34                     </label>
35                 </div>
36                 <!-- Login button -->
37                 <div>
38                     <button id="login-submit" type="submit" class="w-100 btn btn-lg btn-primary">Log in</button>
39                 </div>
40                 <br />
41                 <div>
42                     <!-- Forgot password link -->
43                     <p>
44                         <a id="forgot-password" asp-page="/ForgotPassword">Forgot your password?</a>
45                     </p>
46                     <!-- Register link -->
47                     <p>
48                         <a asp-page="/Register" asp-route-returnUrl="@Model.ReturnUrl">Sign up and create an account</a>
49                     </p>
50                     <!-- Resend email confirmation link -->
51                     <p>
52                         <a id="resend-confirmation" asp-page="/ResendEmailConfirmation">Resend email confirmation</a>
53                     </p>
54                 </div>
55             </form>
56         </section>
57     </div>
58 </div>
59
60 @section Scripts {
61     @await RenderPageAsync("~/ValidationScriptsPartial")
62 }

```

A form has again been used to allow users to input their email and password.

These are the hyperlinks which will send the user to the forgot password page, sign up page or the resend email confirmation page.

Login InputModel

```

1 reference
2 public class InputModel
3 {
4     [Required]
5     [EmailAddress]
6     public string Email { get; set; }
7
8     [Required]
9     [DataType(DataType.Password)]
10    public string Password { get; set; }
11
12    [Display(Name = "Remember me?")]
13    public bool RememberMe { get; set; }
14 }

```

RememberMe has the bool datatype as a checkbox is used for input.

Login Validation

```
73 0 references
74 public async Task<IActionResult> OnPostAsync(string returnUrl = null)
75 {
76     returnUrl ??= Url.Content("~/");
77     ExternalLogins = (await _signInManager.GetExternalAuthenticationSchemesAsync()).ToList();
78
79     if (ModelState.IsValid)
80     {
81         // This doesn't count login failures towards account lockout
82         // To enable password failures to trigger account lockout, set lockoutOnFailure: true
83         var result = await _signInManager.PasswordSignInAsync(Input.Email, Input.Password, Input.RememberMe, lockoutOnFailure: false);
84         if (result.Succeeded)
85         {
86             _logger.LogInformation("You are logged in.");
87             return LocalRedirect(returnUrl);
88         }
89         if (result.RequiresTwoFactor)
90         {
91             return RedirectToPage("./LoginWith2fa", new { ReturnUrl = returnUrl, RememberMe = Input.RememberMe });
92         }
93         if (result.IsLockedOut)
94         {
95             _logger.LogWarning("Account locked out.");
96             return RedirectToPage("./Lockout");
97         }
98         else
99         {
100             ModelState.AddModelError(string.Empty, "Invalid login attempt.");
101             return Page();
102         }
103     }
104
105     // If something failed form will redisplay
106     return Page();
107 }
108
109 }
```

Messages are outputted to the user depending on the situation after the login button is pressed.

If none of the options from the if statement passes then the page will redisplay.

My Account page

My Account User Interface

Health Advice Group Home Advice and Information Risk Assessments The Weather Support My Account Logout

Manage your account

Change your account settings

Profile

Email
Password
Personal data

Profile

Email
test@healthadvicegroup.co.uk

Phone number

Save

© 2023 Health Advice Group. Registered charity number 1234567. Privacy Policy

A sidebar menu to the left was added to allow the user to select more account options.

This is where a user can input their phone number and press the save button to save it to their account.

Manage Email User Interface

The screenshot shows the 'Manage your account' page with the 'Change your account settings' link. Under the 'Manage Email' section, there is a 'Profile' sidebar with 'Email' selected. The main area shows the current email 'test@healthadvicegroup.co.uk' with a checkmark, and a 'New email' field containing 'test@healthadvicegroup.co.uk'. A blue 'Change email' button is below the new email field. Two green arrows point from the bottom text blocks to this button.

Health Advice Group Home Advice and Information Risk Assessments The Weather Support My Account Logout

Manage your account
Change your account settings

Profile
Email
Password
Personal data

Manage Email

Email
test@healthadvicegroup.co.uk ✓

New email
test@healthadvicegroup.co.uk

Change email

© 2023 Health Advice Group. Registered charity number 1234567. Privacy Policy

This is where a user can change their existing email to a new one by inputting their new email.

When this button is pressed the email will be changed to the new email provided.

Change Password User Interface

The screenshot shows the 'Manage your account' page with the 'Change your account settings' link. Under the 'Change password' section, there is a 'Profile' sidebar with 'Password' selected. The main area has three input fields: 'Current password', 'New password', and 'Confirm new password'. A blue 'Update password' button is at the bottom. Three green arrows point from the bottom text blocks to these three input fields, and one green arrow points from the bottom text block to the 'Update password' button.

Health Advice Group Home Advice and Information Risk Assessments The Weather Support My Account Logout

Manage your account
Change your account settings

Profile
Email
Password
Personal data

Change password

Current password

New password

Confirm new password

Update password

© 2023 Health Advice Group. Registered charity number 1234567. Privacy Policy

This is where a user can change their password by inputting their current password and then the new password. The current password is required to prevent accidental password changes.

This is the update password button which will change the current password to the new password when pressed.

Change Password View

```
1 @page
2 @model ChangePasswordModel
3 {
4     ViewData["Title"] = "Change password";
5     ViewData["ActivePage"] = ManageNavPages.ChangePassword;
6 }
7 <!--Change Password heading-->
8 <h3>@ViewData["Title"]</h3>
9 <partial name="_StatusMessage" for="StatusMessage" />
10 <div class="row">
11     <div class="col-md-6">
12         <form id="change-password-form" method="post">
13             <div asp-validation-summary="ModelOnly" class="text-danger" role="alert"></div>
14             <!--Input old password-->
15             <div class="form-floating mb-3">
16                 <input asp-for="Input.OldPassword" class="form-control" autocomplete="current-password" aria-required="true" placeholder="Please enter your old password." />
17                 <label asp-for="Input.OldPassword" class="form-label"></label>
18                 <span asp-validation-for="Input.OldPassword" class="text-danger"></span>
19             </div>
20             <!--Input new password-->
21             <div class="form-floating mb-3">
22                 <input asp-for="Input.NewPassword" class="form-control" autocomplete="new-password" aria-required="true" placeholder="Please enter your new password." />
23                 <label asp-for="Input.NewPassword" class="form-label"></label>
24                 <span asp-validation-for="Input.NewPassword" class="text-danger"></span>
25             </div>
26             <!--Input confirm password-->
27             <div class="form-floating mb-3">
28                 <input asp-for="Input.ConfirmPassword" class="form-control" autocomplete="new-password" aria-required="true" placeholder="Please confirm your new password." />
29                 <label asp-for="Input.ConfirmPassword" class="form-label"></label>
30                 <span asp-validation-for="Input.ConfirmPassword" class="text-danger"></span>
31             </div>
32             <!--Update password button-->
33             <button type="submit" class="w-100 btn btn-lg btn-primary">Update password</button>
34         </form>
35     </div>
36 </div>
37
38 @section Scripts {
39     <partial name="_ValidationScriptsPartial" />
40 }
```

This makes sure that the current password inputted is valid.

This is the update password button that needs to be pressed to make the changes.

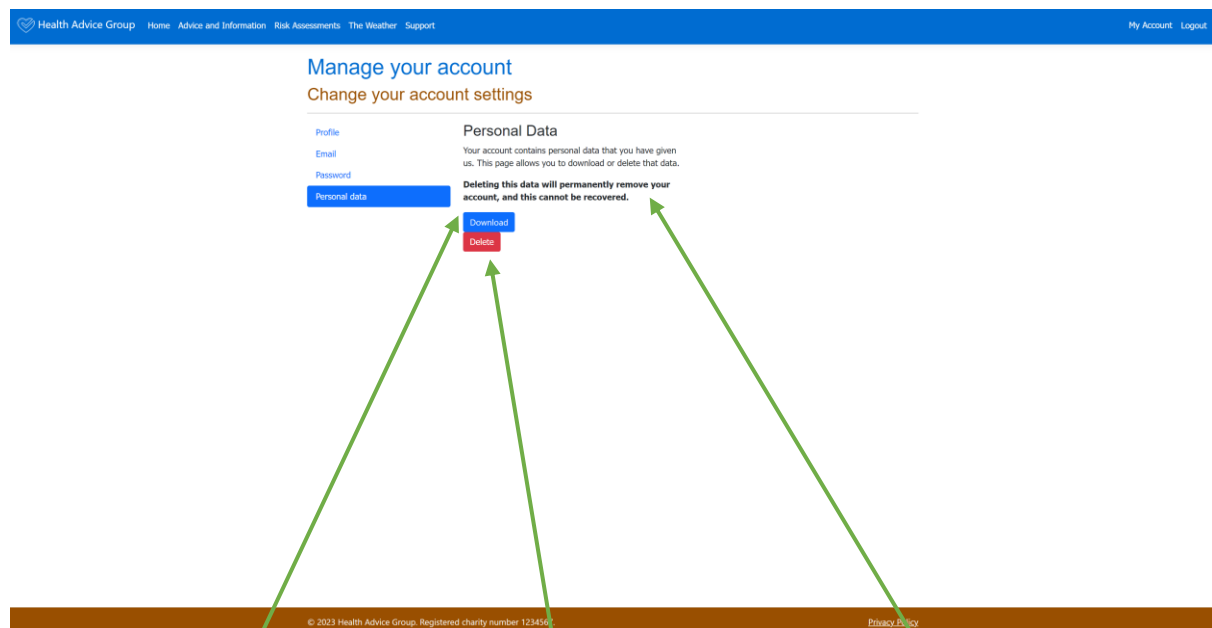
Change Password InputModel

```
39
40
41 public class InputModel
42 {
43     [Required]
44     [DataType(DataType.Password)]
45     [Display(Name = "Current password")]
46     public string OldPassword { get; set; }
47
48     [Required]
49     [StringLength(100, ErrorMessage = "The {0} must be at least {2} and at max {1} characters long.", MinimumLength = 6)]
50     [DataType(DataType.Password)]
51     [Display(Name = "New password")]
52     public string NewPassword { get; set; }
53
54     [DataType(DataType.Password)]
55     [Display(Name = "Confirm new password")]
56     [Compare("NewPassword", ErrorMessage = "The new password and confirmation password do not match.")]
57     public string ConfirmPassword { get; set; }
58 }
```

Error messages were added to inform the user that the passwords inputted aren't valid.

This is a length check to make sure that the new password is at least 6 characters long.

Account Deletion User Interface

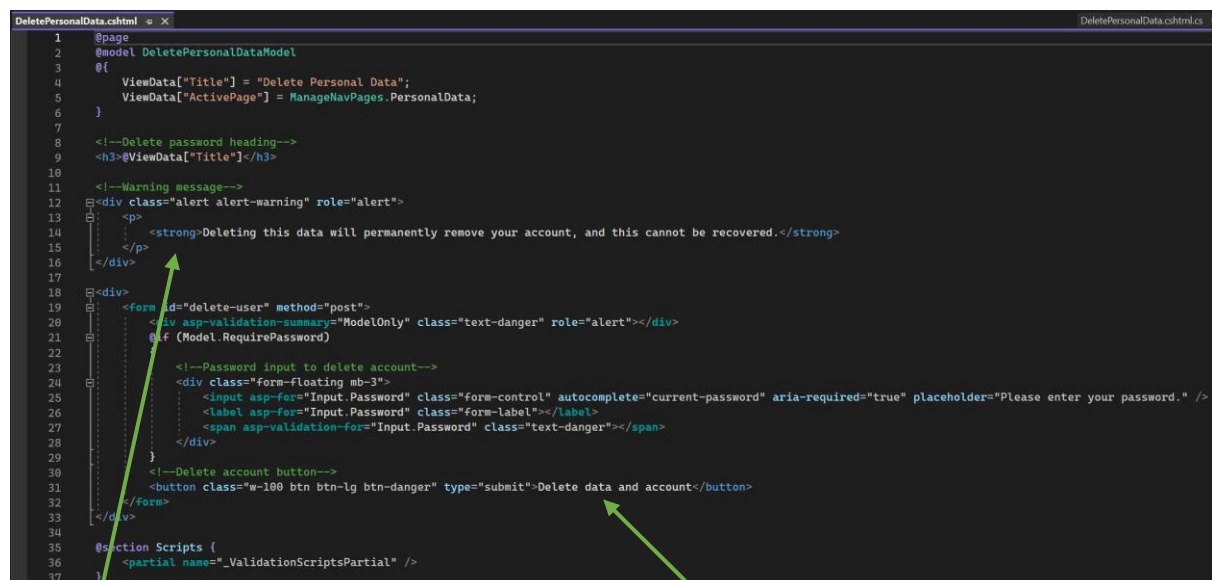


A download button was added so that the user can see what data was collected.

The delete button is in red to signify danger.

This is a warning to make sure that the user is aware that when their account data is deleted it cannot be retrieved again.

Account Deletion View



This warning is in a strong tag to give emphasis as the data will be permanently deleted.

This is the delete button which when pressed will delete the account and all data stored.

Account Deletion Validation

```

57 public async Task<IActionResult> OnPostAsync()
58 {
59     var user = await _userManager.GetUserAsync(User);
60     if (user == null)
61     {
62         return NotFound($"Unable to load user with ID '{_userManager.GetUserId(User)}'.");
63     }
64
65     RequirePassword = await _userManager.HasPasswordAsync(user);
66     if (RequirePassword)
67     {
68         if (!await _userManager.CheckPasswordAsync(user, Input.Password))
69         {
70             ModelState.AddModelError(string.Empty, "Incorrect password.");
71             return Page();
72         }
73     }
74
75     var result = await _userManager.DeleteAsync(user);
76     var userId = await _userManager.GetUserIdAsync(user);
77     if (!result.Succeeded)
78     {
79         throw new InvalidOperationException($"Unexpected error occurred deleting user.");
80     }
81
82     await _signInManager.SignOutAsync();
83
84     _logger.LogInformation("User with ID '{UserId}' deleted themselves.", userId);
85
86     return Redirect("~/");
87 }
88
89

```

This will delete the user's account from the database.

This will sign out the user once the account has been deleted.

If the incorrect password is inputted then the user will be informed and the user is redirected to the page again.

Database

User Table

Name	Data Type	Allow Nulls	Default
Id	nvarchar(450)	<input checked="" type="checkbox"/>	
UserName	nvarchar(256)	<input checked="" type="checkbox"/>	
NormalizedUserName	nvarchar(256)	<input checked="" type="checkbox"/>	
Email	nvarchar(256)	<input checked="" type="checkbox"/>	
NormalizedEmail	nvarchar(256)	<input checked="" type="checkbox"/>	
EmailConfirmed	bit	<input checked="" type="checkbox"/>	
PasswordHash	nvarchar(MAX)	<input checked="" type="checkbox"/>	
SecurityStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>	
ConcurrencyStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>	
PhoneNumber	nvarchar(MAX)	<input checked="" type="checkbox"/>	
PhoneNumberConfirmed	bit	<input checked="" type="checkbox"/>	
TwoFactorEnabled	bit	<input checked="" type="checkbox"/>	
LockoutEnd	datetimeoffset(7)	<input checked="" type="checkbox"/>	
LockoutEnabled	bit	<input checked="" type="checkbox"/>	
AccessFailedCount	int	<input checked="" type="checkbox"/>	


```

1 CREATE TABLE [dbo].[AspNetUsers] (
2     [Id] NVARCHAR (450) NOT NULL,
3     [UserName] NVARCHAR (256) NULL,
4     [NormalizedUserName] NVARCHAR (256) NULL,
5     [Email] NVARCHAR (256) NULL,
6     [NormalizedEmail] NVARCHAR (256) NULL,
7     [EmailConfirmed] BIT NOT NULL,
8     [PasswordHash] NVARCHAR (MAX) NULL,
9     [SecurityStamp] NVARCHAR (MAX) NULL,
10    [ConcurrencyStamp] NVARCHAR (MAX) NULL,
11    [PhoneNumber] NVARCHAR (MAX) NULL,
12    [PhoneNumberConfirmed] BIT NOT NULL,
13    [TwoFactorEnabled] BIT NOT NULL,
14    [LockoutEnd] DATETIMEOFFSET (7) NULL,
15    [LockoutEnabled] BIT NOT NULL,
16    [AccessFailedCount] INT NOT NULL,
17    CONSTRAINT [PK_AspNetUsers] PRIMARY KEY CLUSTERED ([Id] ASC)
18 );

```

This is the User Table with the attribute's Name, Data Type and whether a NULL value is allowed.

This is the same User Table but in SQL.

This symbol shows that Id is the primary key.

User Data

Id	UserName	NormalizedUser...	Email	NormalizedEm...	EmailConfirmed	PasswordHash	SecurityStamp	ConcurrencySt...	PhoneNumber	PhoneNumbe...	TwoFactorEnab...	LockoutEnd	LockoutEnabled	AccessFailedCo...
2563742-8950...	test@healthad...	test@healthad...	test@healthad...	test@healthad...	true	AQAAAAAAYw...	74QXIH6AS5I6...	8ac804be-2f8...	NULL	NULL	NULL	NULL	NULL	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

User accounts are saved as rows in a table.

Passwords are hashed for security.