

CST8117

Style Guide Document

Assignment 2 - Simple Client Website with Constraints Group

Group14:

Costa Santiago, Maria Isabel

Lacerda Martins Silva, Deborah

Weng, Joseph

Yick, Jane

Contents

Our page.....	3
API	3
News API	4
Open Weather Map API.....	4
Style.....	5
Inspiration:	5
Guide for the logo	5
Colour schemes.....	5
Palette	5
Colour palette	6
Image & icon usages	6
Typography	6
Design and prototyping.....	7
Web.....	7
Mobile	7
Resolution	8
Semantic Design Guide	8
Main page	12
World Cup page	13
Weather page	14
About US page	17
User Experience	18
Form Validation.....	18
Website Map.....	19
Library and Frameworks	19

Our page

In this project, we explore using an Application Programming Interface - API.

We started the project by researching the public use API; after some discussion, our group selected 2 APIs, one for Weather and the other for News. Thus, the idea of making a page in the format of an electronic newspaper was born. Our goal was to develop an easy-to-navigate newspaper that could inform the public about business news, the city of Ottawa, sports, the world cup, and weather forecasts.

Our page code: <https://github.com/DeborahLacerda/cst8117-assignment2>

API

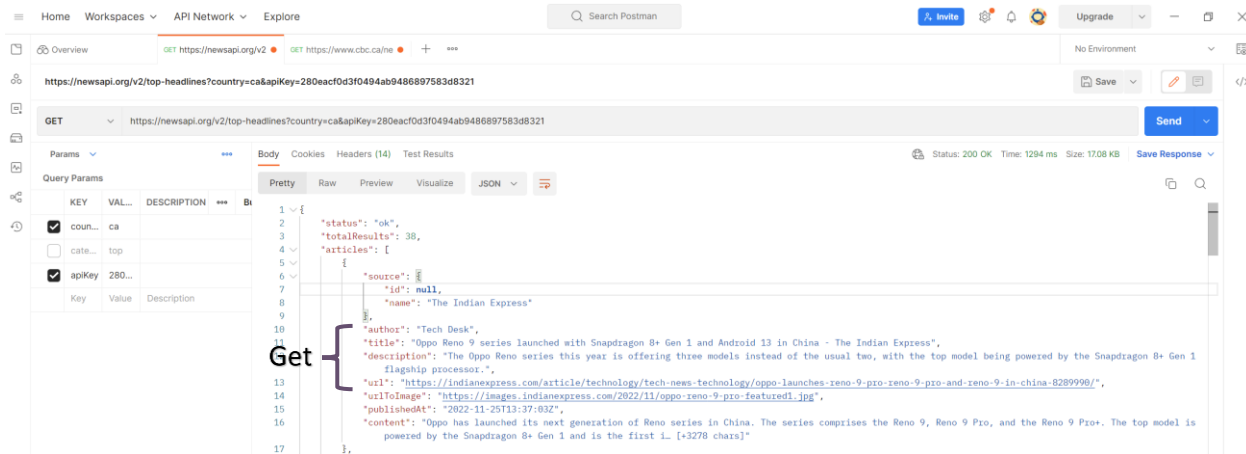
To understand the API's nature, we first used postman to make the code easier to read, and we chose to use Ajax as it is more clearly structured and declarative.

For example:

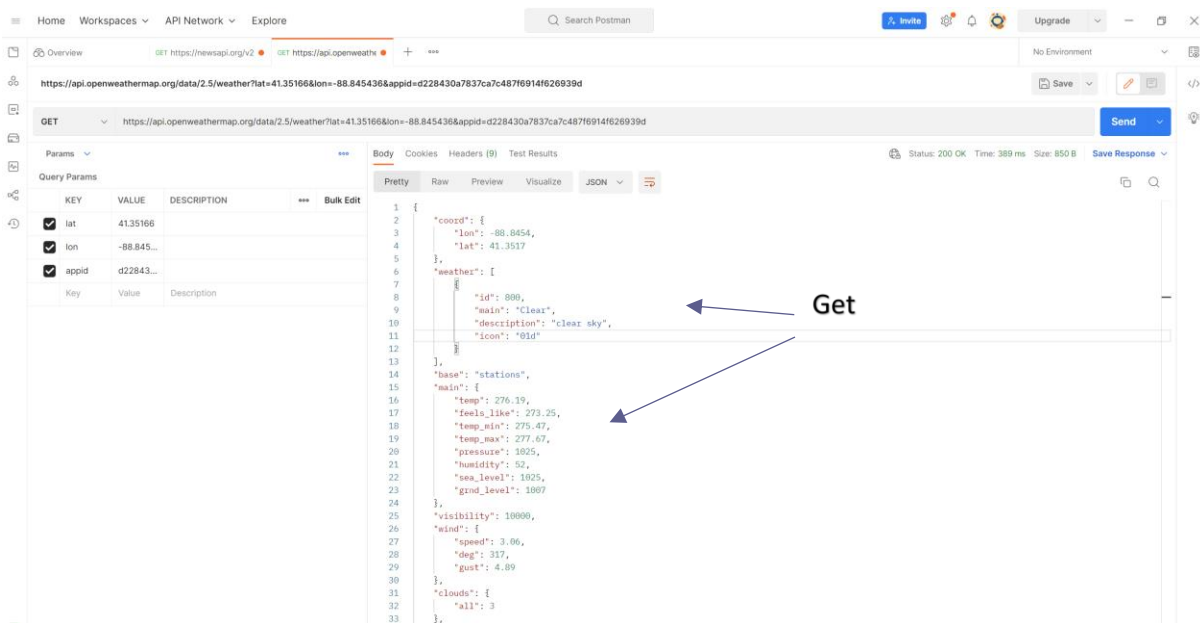
```
let getWeatherByCityName = async (cityString) => {
  let city;
  if (cityString.includes(",")) {
    city =
      cityString.substring(0, cityString.indexOf(",")) +
      cityString.substring(cityString.lastIndexOf(","));
  } else {
    city = cityString;
  }
  let weather = await $.ajax({
    url: `${weatherBaseUrl}/weather`,
    type: "GET",
    data: {
      units: "metric",
      appid: weatherAPIKey,
      q: city,
    },
  },
  error: (error) => {
    showToastRequest("error", error.responseJSON.message);
  },
  });
  return weather;
};
```

News API

A news API refers to an application programming interface that provides access to news sources and databases. And we collect the result using query parameters like country, city, sport, and world cup and more. We collect title, description, and author; in addition to that, we put the link to the original news.



Open Weather Map API was the choice to access free weather data straightforward and intuitive way with good documentation. In this API, we use the city of Ottawa as the default and choose to display the current weather and the five-day forecast. Demonstrated in the image below with the collected parameters.



Style

As our industry will be communication, our main objective in style is to make the layout easily accessible to everyone simultaneously. For this to be possible, the layout of our page does not deviate from what is available in the market and has a certain credibility and attracts the public; we seek inspiration from local and international newspapers.

Inspiration:

We looked for inspiration from newspaper sites so our page would not take me too far away from what the market does.

Follow the links I use to base our page design:

- <https://www.cbc.ca/>
- <https://g1.globo.com/globonews/>
- <https://www.ctvnews.ca/>
- <https://ottawacitizen.com>

Guide for the logo

For the logo we used a website that generates logos (<https://app.logo.com/>), and as we wanted something related to the group members, we used the abbreviations of our names and the colour of our palette.

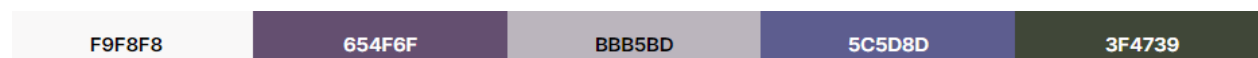


Logo : <https://app.logo.com/>

Colour schemes

We chose the colour palette from <https://coolors.co/>, one of our concerns when choosing the colour was to keep the site accessible for the colorblind, which is why we carefully picked our colour palette. Another factor that made us opt for this colour was the meaning of colours such as shades of blue and green are associated with security and honesty, and purple is normally often used to style mysterious things; we think it will be an excellent colour to sharpen the reader's curiosity.

Palette



Palette : <https://coolors.co/palette/f9f8f8-654f6f-bbb5bd-5c5d8d-3f4739>

Colour palette

Colour Blindness	Palette				
Protanopia	F8F8F8	5B5B67	B8B8BB	5C5C81	42423C
Deutanopia	F8F8F8	5C5E65	B8B9BA	5C5C7E	42413D
Tritanopia	F8F8F8	63615F	BAB9B9	5C7876	3F3F3F
Achromatopsia	F8F8F8	595959	B7B7B7	626262	434343
Protanomaly	F8F8F8	60566B	B9B6BC	5C5C87	40443A
Deuteranomaly	F8F8F8	60546A	B9B6BB	5C5C86	40443A
Tritanomaly	F8F8F8	645769	BAB7BB	5C6984	3F433B
Achromatomaly	F8F8F7	5E5463	B9B6B9	5F5F75	41443E

Image & icon usages

Here we used for most of the images were those provided by the weather and news API. The only one we use static is the banner below found on the official cup website.



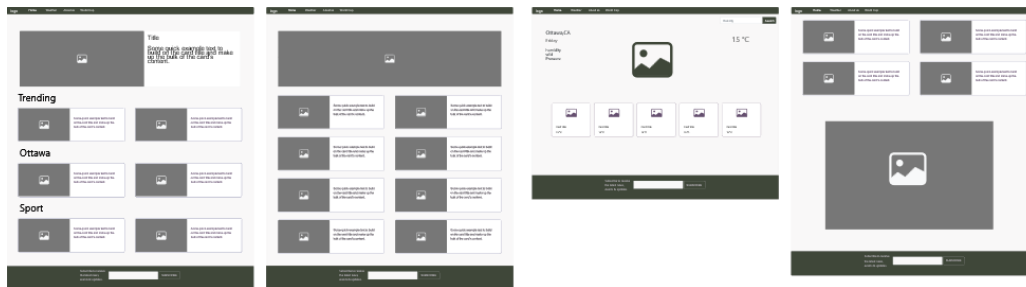
Typography

Use font-family Open Sans, sans- because it has a vertical stress, open shapes and a neutral but friendly look. We chose this font because it has excellent legibility characteristics in its letterforms. Also, it's incredibly readable in small sizes. The best point of all, it's a free font!

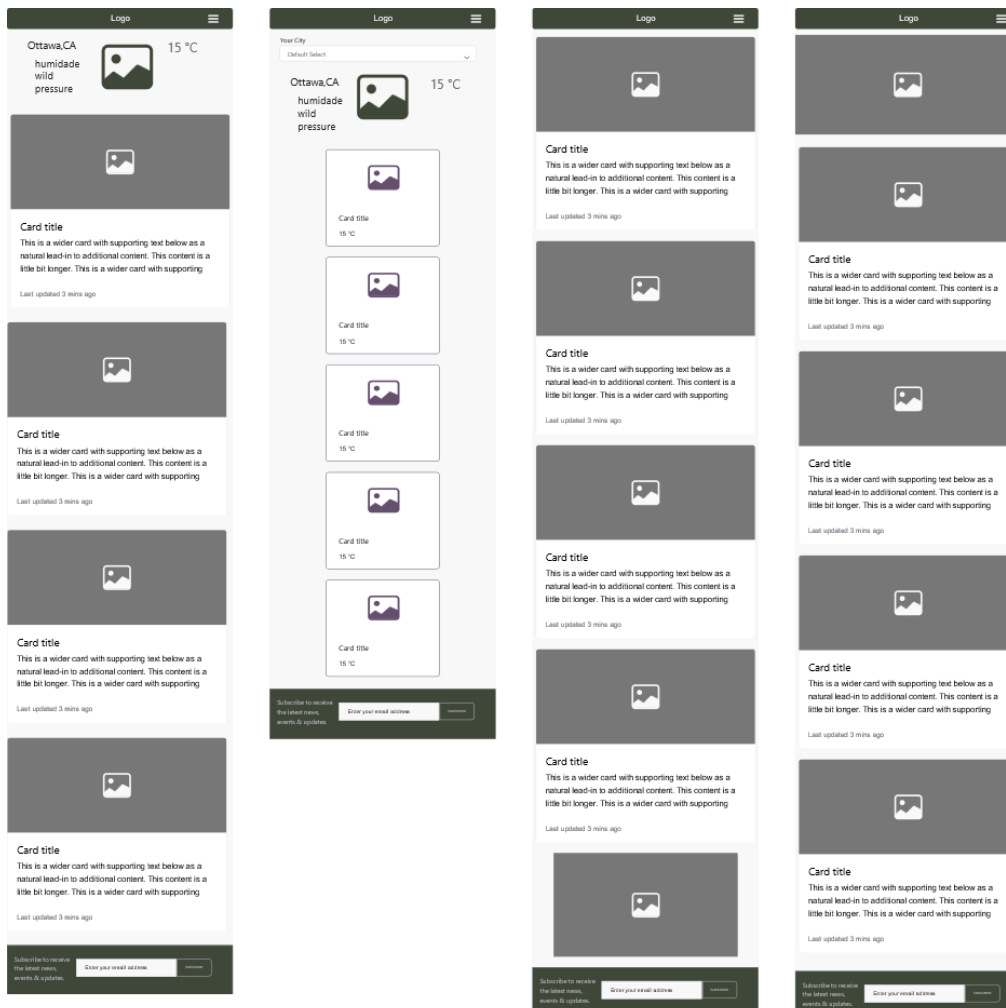
Design and prototyping

We used Xd design for the prototyping of our page to ensure identity in the layout and facilitate the construction of the CSS, and we chose to use the bootstrap library with the main objective of applying the size and layout options, reducing the amount of code.

Web



Mobile



Resolution

The website will target traditional laptop monitor resolutions of 1366x768 and cell (iPhone 14 Pro Max) 430x932

Semantic Design Guide

I used the landing page as a Base for all the others, in addition to maintaining the identity. It ended up being simpler when customizing the other pages.

For consistency and standards, we chose to use the bootstrap library version 5.2.2.

Navbars

As source at navbar we use a wrapping `.navbar` with `.navbar-expand-md` for responsive collapsing and we change background.

```
<nav class="navbar navbar-expand-md navbar-dark fixed-top">
```

Use `.container-fluid` for a full width container, spanning the entire width of the viewport.

```
div class="container-fluid">
```

Hamburger menu

`.collapse .navbar-collapse` for grouping and hiding navbar contents by a parent breakpoint

```
<button
  class="navbar-toggler"
  type="button"
  data-bs-toggle="collapse"
  data-bs-target="#navbarCollapse"
  aria-controls="navbarCollapse"
  aria-expanded="false"
  aria-label="Toggle navigation"
>
  <span class="navbar-toggler-icon"></span>
</button>
```

This example uses spacing (`me-auto`, `mb-2`) utility classes and (`mb-lg-0`, `me-2`) modal component meaning its position and display have been overridden

```
<ul class="navbar-nav me-auto mb-2 mb-md-0">
```


Use .flex-shrink-0

```
<main class="flex-shrink-0">
  <div class="container">
    <section class="banner">
      <div class="weather"></div>
      <div id="banner"></div>
    </section>
```

Here's .row-cols-1 laying out the cards on one column, and .row-cols-md-2 splitting four cards to equal width across multiple rows, from the medium breakpoint up.

```
<div id="city-news" class="row row-cols-1 row-cols-md-2 g-4"></div>
```

General changes:

Create a variable with colors to avoid errors, maintain the visual identity of the site on all pages. alem days we collect the source and apply it to the body.

```
:root {
  --english-violet: #654f6f;
  --bg: #f9f8f8;
  --lavender-gray: #bbb5bd;
  --dark-blue-gray: #5c5d8d;
  --rifle-green: #3f4739;
  --dark-back: #5c5e65;
  --dark-card: #5b5b67;
  --wcard-light: #3f473977;
  --wcard-dark: #60675a;
}
```

```
html {
  position: relative;
  min-height: 100%;
}
```

```
body {
  margin-bottom: 60px;
  font-family: Open Sans, sans-serif !important;
}
```



1. Nav Bar

We change background-color, logo size

```
nav {  
  background-color: var(--rifle-green);  
}  
  
#logo {  
  width: 125px;  
  padding: 5px;  
}
```

2. Footer

Changes in position, background color, and font color.

```
.footer {  
  position: absolute;  
  display: flex;  
  text-align: center;  
  bottom: 0;  
  width: 100%;  
  background-color: var(--rifle-green);  
  padding: 20px;  
  color: var(--lavender-gray);  
}  
  
.form {  
  text-align: center;  
  color: var(--lavender-gray);  
}
```

In the buttons of the pages we change background color, and font color.

```
.button {  
  background-color: var(--dark-blue-gray);  
  color: var(--lavender-gray);  
}  
  
.button:hover {  
  background-color: var(--english-violet);  
}
```

In the button to change to dark or light mode we use the html below.

```
<input type="checkbox" class="checkbox" id="checkbox" />
  <label for="checkbox" class="label">
    <div class="ball"></div>
  </label>
```

We used a series of modifications as shown below.

```
.checkbox {
  opacity: 0;
  position: absolute;
}

.label {
  width: 40px;
  height: 18px;
  background-color: var(--dark-blue-gray);
  display: flex;
  border-radius: 50px;
  align-items: center;
  justify-content: space-between;
  position: relative;
  transform: scale(1.2);
}

.ball {
  width: 15px;
  height: 15px;
  background-color: var(--bg);
  position: absolute;
  top: 2px;
  left: 2px;
  border-radius: 50%;
  transition: transform 0.2s linear;
}
```

Main page

On that page we made some changes to the card for auto at the time because the content changes. And we also change the width and text of the first letter to uppercase using capitalize.



Subscribe to receive the latest updates, news & events.

☐ Change preferences

Favorite options

<input type="checkbox"/> Business	<input type="checkbox"/> Ottawa	<input type="checkbox"/> Sports
<input type="checkbox"/> Basketball	<input type="checkbox"/> Hockey	<input type="checkbox"/> Food
<input type="checkbox"/> Car	<input type="checkbox"/> Entertainment	<input type="checkbox"/> Movies
<input type="checkbox"/> Music	<input type="checkbox"/> World	

```
.modal-body {  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
}  
  
.checkbox-preferences {  
  width: 33%;  
  text-transform: capitalize;  
}
```

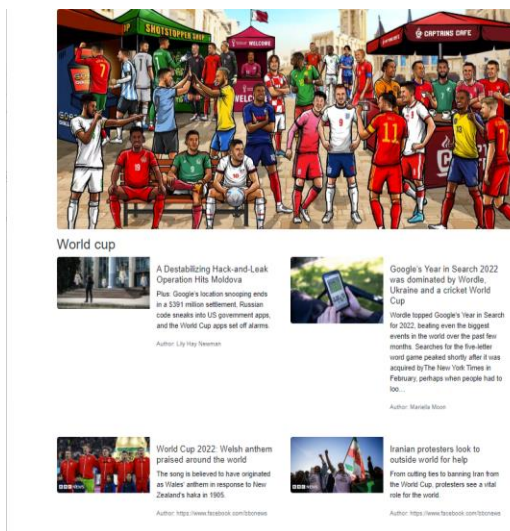
```

.card-body {
    height: auto;
    position: relative;
}
.dark-mode .card {
    background-color: var(--wcard-dark);
}
.dark-mode .text-muted {
    color: var(--lavender-gray) !important;
}
.dark-mode .backcard {
    background-color: var(--dark-back);
}

```

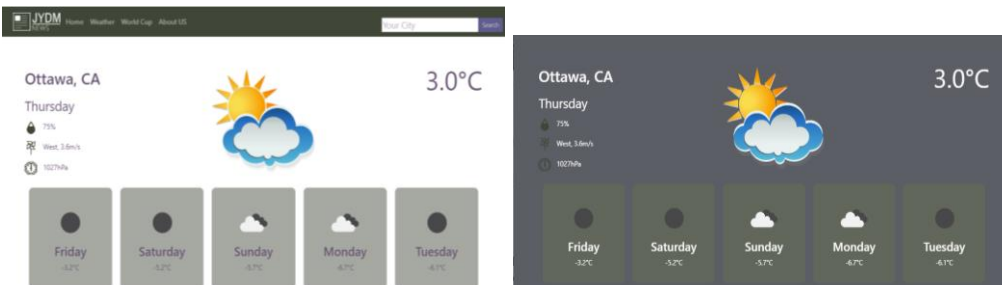
World Cup page

This page is very similar to the news page, but focused on the World Cup, it has updated news about the World Cup. Here we use the same css as the main page. That is, the cards have the same nature. The main difference is that we have a banner at the top of the page.



Weather page

This page was the one we used the most to style change independent of the bootstrap library due to a large number of images in the nature of the API. In addition, we manipulate the colours of some icons to match the colour of our palette, thus maintaining the identity of the site. For greater convenience and user interaction on the site, we leave the search option by city with autocomplete.



```
.citysearch label {
  display: block;
}

.weather {
  width: 90%;
  margin: 0 auto;
  padding: 120px 0;
  box-sizing: border-box;
  white-space: nowrap;
}

.citysearch {
  margin-top: 50px;
}

.weather_search {
  width: 100%;
  border: none;
  font-size: 20px;
  color: lava;
}

.weather_search::placeholder {
  color: #bbb5bd;
  margin-top: 5px;
}
```

```
.weather_today {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  padding-bottom: 10pt;
}

.weather_image {
  width: 30vmin;
  align-self: center;
}

.weather_city {
  font-size: 36px;
  font-weight: 600;
  margin: 20px 0;
}

.weather_day {
  font-size: 30px;
  margin: 10px 0;
  font-weight: 400;
}

.weather_indicator::before {
  content: "";
  display: inline-block;
  vertical-align: middle;
  width: 30px;
  height: 30px;
  background-size: cover;
  margin-right: 10px;
}

.weather_indicator-humidity::before {
  background-image: url(../images/humidity.png);
}

.weather_indicator-wind::before {
  background-image: url(../images/wind.png);
}

.weather_indicator-pressure::before {
  background-image: url(../images/pressure.png);
}
```

```
}

.weather_temperature {
  font-size: 60px;
}

.weather_forecast {
  width: 100%;
  text-align: center;
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
}

.weather_forecast_icon {
  max-width: 50%;
  color: var(--english-violet);
}

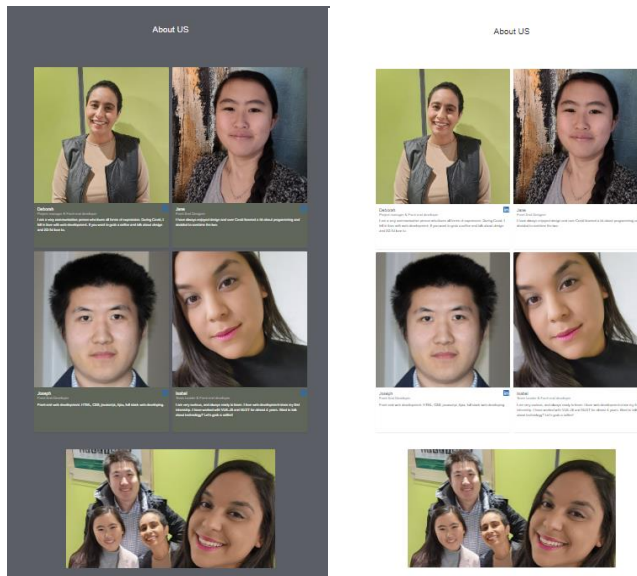
.weather_forecast_item {
  color: var(--english-violet);
  background-color: var(--wcard-light);
  border-radius: 10px;
  margin: 0 10px 10px;
  padding: 30px 5px 20px;
  box-sizing: border-box;
  min-width: 150px;
  width: calc(20% - 20px);
  flex-grow: 1;
}

dark theme */
.dark-mode {
  background-color: var(--dark-back);
  color: var(--bg);
}

.dark-mode .weather_forecast_item {
  background-color: var(--wcard-dark);
  color: var(--bg);
}
```


About US page

Here we remove the border and change the group image.



```
.card-body-staff {  
  height: 160pt;  
  position: relative;  
}  
.group {  
  margin-top: 5% !important;  
  margin-bottom: 15% !important;  
}  
.dark-mode .card {  
  background-color: var(--wcard-dark);  
}  
.dark-mode .text-muted {  
  color: var(--lavender-gray) !important;  
}  
.dark-mode .backcard {  
  background-color: var(--dark-back);  
}
```

User Experience

Our user Experience it focuses on having a deep understanding our users, what they need, their abilities, and also their limitations.

A good user experience would mean that:

1. User visited your website
2. Scrolled through our pages
3. Visualized the News and Weather
4. Chose news and theme

The factors that influence our choices.

- Usefulness
- Findable
- Accessible

To improve the use of our website, we let the user choose which type of news he would like to see through the **showFavoriteList** function. We use the library in bootstrap to do this and save the preferences in **localStorage** using the **savePreferences** function.

With the list below:

- Business
- Ottawa
- Sports
- Basketball
- Hockey
- Food
- Far
- Entertainment
- Movies
- Music
- World

We use css to change the list.

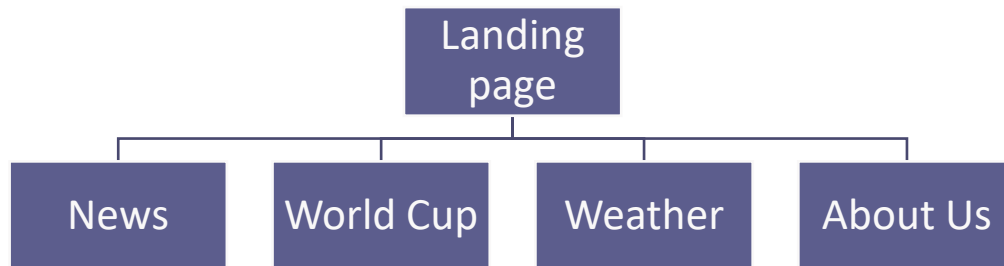
```
.checkbox {  
  width: 33%;  
  text-transform: capitalize;  
}
```

Form Validation

We used the Toastify JS library and created the **showToastRequest** function to push notifications to your visitors with a toast. On all pages we apply the validation of the subscribe field and on the weather page we also use the validation in the search field. For the subscribe field any time visitor send the e-mail will be show ne toast green said "e-mail send!"

Website Map

All pages are connected to each other because they have the same header.



Library and Frameworks

Toastify JS - <https://apvarun.github.io/toastify-js/>

Bootstrap (5.2.2) - <https://blog.getbootstrap.com/>

jQuery (3.6.1) - <https://jquery.com/>

Ajax (3.6.1) - <https://api.jquery.com/jquery.ajax/>