**Responsive web:**

Current trends and improvements offered in terms of both software and hardware led us to develop websites in large number and to cater to vast screen sizes, which includes smart phones, tablets, dual screened mobile phones and large displays.

These comes with diverse characteristics with respect to their screen resolutions. When building this news website for assignment 2, I explored the possibility of building one webpage that suits all screen sizes, i.e., ‘**Responsive web design’.**

This kind of webpage helped solve a lot of problems for the website owner as well as proved to be sustainable for the environment. Building responsive website made it mobile friendly as well as rendered as expected on larger screens. Website automatically scales its contents to match the screen size it is being viewed on. This prevented unnecessary zooming, panning or scrolling to view the contents.

The effectiveness of such a website correlated to the following benefits:

* One page fits all screens, therefore being flexible
* Developer need not worry about the screen the page is being rendered upon.
* Single deployment, improving the effective time to ship the product to market.
* Ease of management as the codebase was relatively smaller than 2 separate adaptive websites built to cater to different screen sizes.
* Improved User experience, as the transition from desktop to mobile and vice versa was seamless.

Google’s page indexing and search engine result, ranking mobile friendly websites higher in the order and promotes responsive design as a standard when it comes to SEO(Search Engine Optimization) [1].

Challenges faced while building responsive design:

There were intricacies when building a website that renders on all devices and validated using an emulator. The following difficulties was faced when designing the same:

* **Testing**: Each element added to the website needed to be tested on all possible screen resolutions to check for backlogs and breakages. Focus was to not lose the element when being rendered on different screens. Care needed to be taken after every addition, to see if there were no effect on other components of the page.
* **Scalability**: Each element in the webpage has its own styles of implementation with regards to scalability. Font, images, and videos (both from local and referenced websites) needed to scale as per the screen. This was rectified by using relative values such as **EM** (relative to parent), **%** (relative to parent), **VW** and **VH** (relative to viewports width and height), rather than using absolute value such as **PX** (pixels) in most cases.

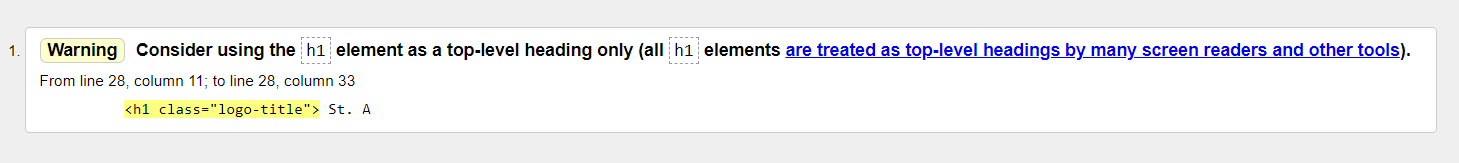
**Bootstrap Framework**

The responsive website's components were built using the **Bootstrap** framework since it has a fluid grid structure that expands suitably up to 12 columns as the device or viewport size rises. It has strong mixins for creating more semantic layouts as well as preset classes for quick layout options. Grid layout offered by bootstrap was used in home screen to render cards within a row and 4 columns.

**Validation**

On validating the page after development, the following errors were found

1. **<h1>** tag was used inside lower ranked elements. h1 is a blocking element so it should not be inside a non-blocking element . This violates the html standards [2]. This was later replaced with **<span>** tag.

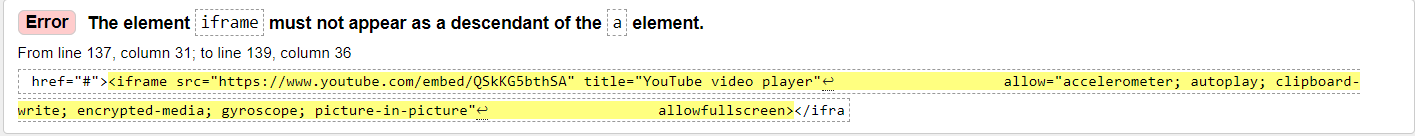


1. The following error was thrown like the one stated above. Rectification used for this error was to replace **p1** with **p** tag.

Text

Description automatically generated with medium confidence

1. As per the Html specification stated in **WHATWG** for **‘a’ tag** [3], there must be no interactive content as descendant, and **iframe** is an interactive element. The **source** used in the iframe element allows for navigation and interaction, thereby removing the dependency of wrapping around **<a>**



4. **<hr>** tag was used immediately after ordered list **<ol>** tag was used, therefor it had to be removed and added after the list item is added. Rectification used to resolve this was: <ol><hr><li>item</li></ol> is not valid, but <ol><li>item<hr ></li></ol> is valid as the direct child of <ol> is <li>. [4]

Graphical user interface, text, application

Description automatically generated

The Screenshot of the page rendered on a 1739 X 979-pixel dimension display is added as part of the source code, under the directory name ‘**screenshot’.**

**2. Adaptive mobile**

The mobile version was built and tested on 412px \* 900px display with ‘fit to window’ zoom option selected. This website is made non-responsive and will only be rendered correctly on the above-mentioned resolutions. Any drastic change from the aforementioned resolution will lead to breakage and run overs in terms of the responsiveness.

****

Figure illustrates the screen resolution considered to build adaptive mobile version of the website. The dimension is provided by google chrome web browser under its list of devices.

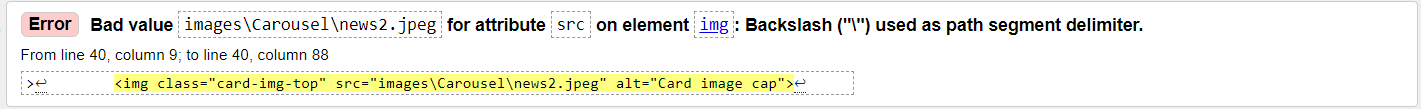
This approach towards building a native website came with a set of challenges:

* **Testing**: The websites are built and viewed on browsers on a larger screen and is heavily dependent on tools offered by the browsers such as ‘Toggle device toolbar’ menu in Google chrome’s DevTools, ‘Toggle device emulation’ on Microsoft edge, etc. These offer a range a device list from which preferred screen dimension was chosen. The rendered page was not consistent on switching between browsers, leading to inconvenience in adding adequate CSS styles for respective elements. Google chrome was chosen as a default browser when testing for adaptive screens.
* **Static elements and layout:** Mobile first approach when building a responsive website, gave good insight on understanding the scalability. But in case of adaptive web design, constraining to a specific fixed pixels and making a static layout at the cost of sacrificing user experience when scaled, seemed difficult to comprehend.

**Validation:**

On validating the adaptive mobile after development, the following errors were found:

1. The relative path added for the images was using



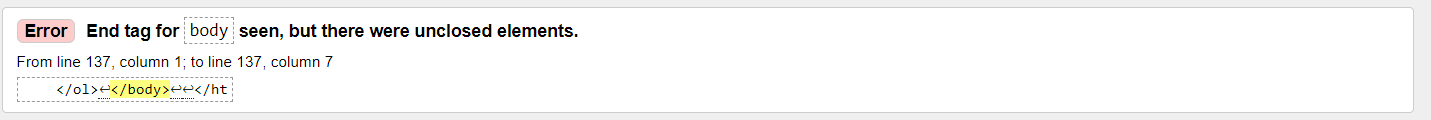
Rectification:

Bad input : src="images\Carousel\category.png"

Correct form of relative path was to use ‘/’ : src="images/Carousel/category.png"

1. Unclose elements were caught by the validator and was added post validation.

Missed a div element closure before the body completion.



1. **<img>** or **<video>** elements accept a width attribute to specify the size in pixels. This number should not contain units or % and can only be an integer. If you need to specify a percentage width, you need to do that with CSS. The browser truncates the value "420px" for key "width" to its numeric prefix and renders it without any difficulties.



**CSS Error**

A picture containing graphical user interface

Description automatically generated

The CSS validator [5] threw the above shown error, because of incorrect spelling used on the CSS property value.

The Screenshot of the page rendered on a 1739 X 979-pixel dimension display is added as part of the source code, under the directory name ‘**screenshot’.** The screenshot shows how an adaptive website built for mobile screen renders on a larger display.

**3. Adaptive Desktop**

The desktop version was built and tested on 1600px \* 900px display with ‘fit to window’ zoom option selected. This website is non-responsive and will only be rendered correctly on the above-mentioned resolutions. Any drastic change from the aforementioned resolution will lead to breakage and run overs in terms of the responsiveness.

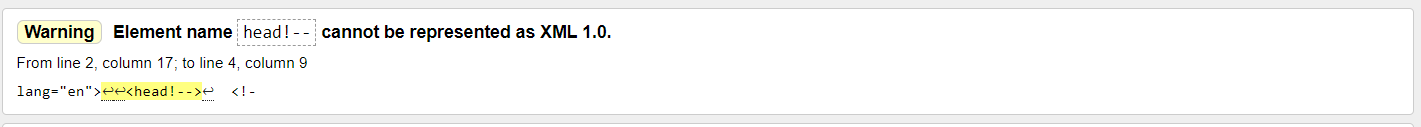
The challenges faced were similar to the ones faced when building an adaptive mobile website.

The following errors were identified on validating HTML and CSS.

1. This error was thrown when the head tag was written in the form <head!-->

**A picture containing graphical user interface

Description automatically generated**

This led to the following warning 

Document type of HTML was rendered correctly on the browsers even when these validation errors existed indicating that the modern-day browser ignores the above-mentioned errors.

The Screenshot of the page rendered on a 1600 X 900-pixel dimension display is added as part of the source code, under the directory name ‘**screenshot’.**

**Bootstrap Framework via CDN**

The following bootstrap compiled CSS was included in responsive websites to use components provided by Bootstrap.

Version : Bootstrap 4.5.3

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"

integrity="sha384-X8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

Content Delivery Network (CDN) provide a great solution to serving assets like image files from a network of data centers worldwide. This means that the largest files will be loaded from the CDN location in the user’s own region, reducing the distance that data is moving each time a page is loaded. This improves energy efficiency and improves page load times.

**REFERENCES**

[1] Google Search essentials , <https://developers.google.com/search/docs/essentials>, accessed on 10-11-2022

[2] HTML5, <https://www.w3.org/TR/2010/WD-html5-20101019/sections.html#headings-and-sections/>, accessed 17-11-2022

[3] HTML living standards, <https://html.spec.whatwg.org/multipage/text-level-semantics.html#the-a-element>, updated on 19-11-2022

[4] HTML Checking for Large Sites, <https://rocketvalidator.com/html-validation/element-div-not-allowed-as-child-of-element-ul-in-this-context-suppressing-further-errors-from-this-subtree>, accessed 17-112022.

[5] CSS validator, <https://jigsaw.w3.org/css-validator/>, accessed on 17-11-2002