IMPLEMENTATION OF RESPONSIVE WEBSITE WITH CSS FLUID GRIDS

Janne Umapathi[1], Dr. Vishal Khatri[2]

[1]Research Scholar, Department of Computing and Information Technology, Himalayan University, Itanagar, Arunachal Pradesh, INDIA.

[2]Research Guide, Department of Computing and Information Technology, Himalayan University, Itanagar, Arunachal Pradesh, INDIA.

Abstract

Web pages that are built with responsive design adapt to the viewing device's screen size without compromising readability or functionality. Utilizing modern web development tools like CSS3 and HTML5 will help to accomplish this interactive dynamic pages. This main aim of this research paper is to discuss insights about the CSS Fluid grids for responsive website development. The objective of the research is to attempt development ofnew CSS fluid grids for fundamental layout in responsive website and analyse the insights of CSS fluid grid for dynamic website creation. Primarily the contextual analysis was discussed and further, the outcome of the implemented fluid grid method and the key features were discussed extensively.

Keyword: *Dynamic websites, CSS Fluid grids, Responsive web design, Web development.*

Introduction

An organisation can boost its exposure to prospective clients by developing a user-friendly website in this day and age. In today's modern era, having a website is a typical way for businesses, institutions, organisations, and even individuals to communicate with their target market(Bryant & Jones, 2012). However, merely existing online and being discoverable via search engines is no longer adequate. Nowadays,the websites need to be optimised for all these devices in order to deliver the greatest user experience possible, as people spend the majority of their time online and access the Internet via mobile devices. Input methods vary from device to device, including the use of a mouse, finger touches, and gestures, in addition to the aforementioned variations in screen resolutions, screen sizes, web browsers, and platforms(Ben Frain, 2020). While responsive web design has been in existence for considerable time, many sites still aren't taking advantage of the tools available to make their content mobile friendly. It takes a lot of time, money, and effort to convert an existing website. The demand for HTML5 and Cascading Style Sheets model - CSS3 for responsive designing of websites is growing substantially, whereas other web technologies are outdated (for example, Flash) due to their incompatibility with different versions of the browser.

The fundamental complexity appears when a web app's requirements collide with those of the web platform itself. Raw materials for developing web apps have not even come close to keeping pace. The HTML elements "<div>," "<p>," "<h1>," and "ul>" are fantastic for document markup, but they are insufficient for developing the sophisticated app-like interfaces and websites of today(Savage, 2015).Challenges can be fixed with the use of grid layout's features.The new CSS grids are very much in the tradition of table-based layouts. The main aim of this research work is to discuss the implementation of responsive websites with CSS fluid grids. The objective of the research work focuses to propose and test asimple CSS fluid grid program working and discuss the implementation, features of responsive websites.

Literature Review

Need for responsive web design

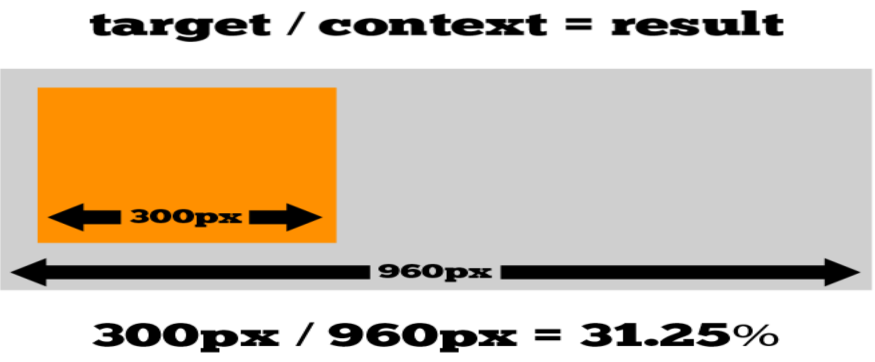
The advent of portable electronic devices like smartphones and tablets has transformed the way that product planning and development are conducted. Before the proliferation of smartphones and tablets with powerful web browsers, the biggest challenge for web designers was working to ensure their sites looked the same across different browsers and versions(Lundmark, 2022). Most frameworks are so all-encompassing that they mandate that every single bit of code and markup be created with the framework in mind. All the way from a user interface library to database drivers, Meteor has the user covered. These are tremendously feature-rich application platforms on their own, but they cause lock-in immediately and make it very difficult to switch to another system. This allows designers to create articulate content and divide the available layout space into columns and rows with a consistent set of size actions. This allows web developers to create applications with consistent layouts by placing elements on a grid with fixed column and row widths(Brett S. Gardner, 2011). Instead of using tables, as is conventional way to distribute content on the web, the designer can now make use of the new grid fluid layout.

Categorization of responsive web design

There are three technological aspects of responsible web design such as fluid grids, flexible images, and media queries(Gardener, 2022). It is the responsibility of fluid grids to size flexible pictures and page elements in relative units. Common practise has been followed to use pixel dimensions when discussing website design. This is the same as a magazine or newspaper that has a set page count and width. Websites can also be viewed on very big screens, like those of televisions, or on very small screens, like those of smartphones (or even a smart watch). Therefore, instead of using absolute units like pixels, responsive websites use relative quantities like percentages in their development and design. A following simple mathematical formula can ease the transition from working with pixels to percentages for creation(Georg Niess et al., 2019).

$$Result = Target/Context \qquad (1)$$

Figure 1 represents the expression 1 and it can be described as followingly. Based on the proportional element calculation, the target element is divided by its context. Creating a high-fidelity mock-up in image development applications (e.g., photoshop)was the best method to achieve this idea. With a high-fidelity mock-up, it is feasible to divide a page element by the page's width.



Media queries allow for the use of various CSS depending on the capabilities of the device. In order to create a responsive design, one must adjust the layout of the page to fit the device's screen resolution, resize images automatically so that they fit on the screen, hide unnecessary elements in smaller devices and modify the size of buttons and links to touch interfaces where the user's finger replaces the mouse pointer; and make intelligent use of features in mobile devices, such as geo localization and rotating the content view. Using logical operators like "not," "and," and "only," designers can construct even more intricate media queries(Elena Parvanova, 2018; Zea, 2015). In order to conduct a media query, the AND operator must be used to combine multiple media features into a single media query, with ALL of the combined features needing to return TRUE. Using not operator, facilitates to cancel out an entire media query. Somestyles from being applied in earlier browsers by using the only operator can be prevented, which applies the style only if the full query matches.

Methodology

Methodology proposed in this research work attempts to structure the subsections with the grid line program. The grid line code is implemented in order to understand the working of gridlines for versatile screen usage(Keda I.O, 2021). Considering a grid container with four sub sections/separate boxes. The default auto-placement rules will assign one item to each of the first four cells in case of non-assistance.The following code is implemented to create four grids with columns and rows divided. The program was executed in CSS 3. Using CSS fluid grids, the design's pixel-based widths are transformed into percentages while maintaining the proportions of our grid across a range of viewport sizes. The 960px approach required us to create under strict grids. Designers make sure the project is easily maintained and efficient by designing each piece using proportional widths and heights rather than pixel-based dimensions. Therefore, elements' widths and heights will automatically readjust to their parent container's dimensions if the device or screen size is altered.

```
.wrapper {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: repeat(3, 100px);
}
Copy to Clipboard
<div class="wrapper">
<div class="Menu">One</div>
<div class="Login">Two</div>
<div class="Description">Three</div>
<div class="Subscribe">Four</div>
</div>
```

Outcome and Discussions

Implementation

It's necessary to generalise writing a single code for all devices, since responsive design is a complex task. There are two possible methods to go about creating a mobile-friendly website, either to begin with the final objective in mindand work backwards to accommodate mobile devices, or vice versa. In order to implement Responsive Design, in which a single website can be viewed across a wide range of devices without needing to be rewritten for each one, it must be determined what can be generalised so that writing duplicate code can be avoided for different devices. A single codebase can serve visitors with varying-sized viewports, eliminating the need for a separate site and codebase for wide-screen displays, desktops, laptops, tablets, and phones of various screen resolutions. In this method, content on the page is rearranged dynamically based on the size of the viewer's window. A three-column desktop layout may be rearranged into two columns for a tablet and a single column for a Smartphone. The proposed program was implemented as each grid division were labelled, as Menu, login, Description, Subscribe. The following structure in figure 1 represents the outline of the gridline code implemented page,

The structure depicts the cells with 3x3 row and column arranged with given labels. By rearranging text and design elements in a cell based on relative sizes, responsive layouts adapt to the viewing environment. Just making sure everything fits on a page isn't enough. Usability across all device sizes and resolutions is crucial to the success of a responsive design.

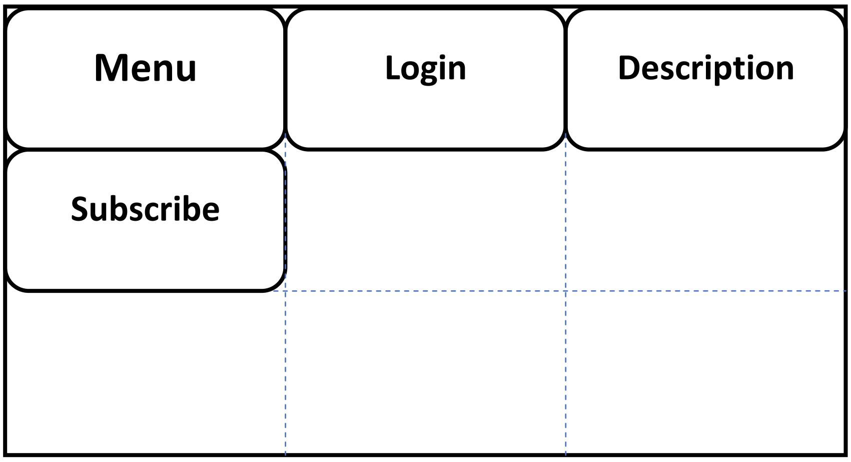


Figure 1: Representation of gridline trial design

Advantages and disadvantages

There are several advantages to using responsive design since it eliminates the need to create multiple versions of a website for various screen sizes. With a responsive design, creators need to design social shares for a single URL, and when the site is shared, the recipient can view it on any device (phone, tablet, or computer) and find the content easily. Simplified content maintenance attributable to a single point of update. Since there will only be one website, the value of material will increase while the likelihood of content duplication decreases. Website performance, search engine optimization, and Google search engine rankings can all benefit from this kind of excellent user experience. The costs associated with updating several website iterations (mobile and desktop) are reduced with responsive design which turns out to be a long term contended maintenance.

In the parallel view, this technique also has few limitations, which has to be considered in the following critical cases. Some devices may not support a "one size fits all" approach to Responsive Web Design, where user's browser in either device commands to download extensions based on CSS(Andy Clarke, 2019). Both the flexible web design and the small screen size of mobile devices make it tricky to lay out the navigation in an intuitive and complete fashion. Most crucial limitation to have an alternative solution is to cut down the loading duration, which in case was observed higher in mobile than desktop.

Conclusion

In respect to the objective of this work, a CSS grid program was executed, and the outline of the result was represented. The excellent features, categorization and limitationsof the fluid grids was discussed extensively. Each and every detail of the design and CSS 3 code have been meticulously implemented. Based on the implementation attempted and discussion, this work concludes that using the CSS fluid grid, programmers have more ability to structure their data with intact ability and automatic control over any devices.

References

Andy Clarke. (2019). *Art Direction for the Web*. Smashing Magazine GmbH. https://stuffandnonsense.co.uk/books

Ben Frain. (2020). *Responsive Web Design with HTML5 and CSS: Develop Future-proof Responsive Websites Using the Latest HTML5 and CSS Techniques* (3rd ed.). https://github.com/PacktPublishing/Responsive-Web-Design-with-HTML5-and-CSS-Third-Edition

Brett S. Gardner. (2011). Responsive Web Design: Enriching the User Experience. *Sigma, Noblis, 11*. https://www.webdesignblog.gr/wp-content/uploads/2012/03/5.pdf#page=15

Bryant, J., & Jones, M. (2012). Responsive Web Design. *Pro HTML5 Performance*, 37–49.

https://doi.org/10.1007/978-1-4302-4525-4_4

Elena Parvanova. (2018). Explore Modern Responsive Web design techniques. *Proceedings of the International Conference on Information Technologies (InfoTech-2018)*, 1–6. http://infotech-bg.com/sites/default/files/2018/A04.pdf

Gardener, S. D. (2022). *The Algorithmic Composition Explorer Using a learning environment to enable users to experience intrinsic feedback on musical algorithms*. https://www.duo.uio.no/handle/10852/96082

Georg Niess, Arwin Roubal, Stefan Thurner, & Enrique Barba Roque. (2019). *CSS Grid Layouts*. Graz University of Technology. https://courses.isds.tugraz.at/iaweb/surveys/ws2019/iaweb-ws2019-g1-survey-css-grid.pdf

Lundmark, O. (2022). *Responsive Web Design for Modern Devices*. http://urn.kb.se/resolve?urn=urn:nbn:se:bth-23256

Savage, T. (2015). Componetizing the web. *Queue*, *13*(8), 1–20.

Zea, R. (2015). *Mastering Responsive Web Design with HTML5 and CSS3: Become a Master in RWD with this Practical and Engaging Guide* (1st ed.). Packt Publishing. https://www.oreilly.com/library/view/mastering-responsive-web/9781783550234/

Keda I.O. (2021). Artificial Intelligence as a new artist. *Ministry of Education and Science of Ukraine Tavria State Agrotechnological University The Names of Dmytra Motorny Council of Young Scientists and Higher Education Acquires*, 15–18. http://www.tsatu.edu.ua/im/wp-content/uploads/sites/24/materialy-ix-vseukr.-nauk.-tehn.-konf.2021.pdf#page=16