Keeping Websites Current by Using a New Evolving JavaScript Framework

Capstone Seminar

for

Master of Science

Information and Communication Technology

Hanna Balaszi

University of Denver University College

March 6, 2022

Faculty: Timothy E. Leddy, MBA

Director: Cathie Wilson, M.S.

Dean: Michael J. McGuire, MLS

Abstract

After being developed, many websites go untouched, leading to outdated content and design. The result of the realization that a business's website is no longer relevant is to perform a complete website redesign and development, which takes a lot of time and money. A solution was theorized called Trend.js by identifying that websites stay current by using yearly trends. Trend.js is an all-inclusive tool for developers to keep websites on-trend and contemporary. With a combination of Angular, APIs, and components, Trend.js creates code modules containing new specific trends that can be implemented on all websites. Trend.js is not only a framework but also a business and community that lends itself to the support and functionality it needs to continue adapting.

Contents

Background

There are over 1.8 billion websites on the internet, and staying modern goes a long way to stand out amongst those 1.8 billion (Bleu 2021). However, staying modern or keeping websites updated is not an easy task. Once a website is created, developers do not usually update the design and functionality unless requested by the client. Clients rely on web developers to make their websites stand out amongst the 1.8 billion websites on the internet for years after it is first built. Unfortunately, developers disappoint their clients because they have no accurate tool to help them accomplish such a monumental task. This disappointment results in more work for developers in the long run or the inevitable departure of the client.

The predicted lifespan of a website that does not make regular updates is two years and seven months (Crestodina, n.d.). Two years and seven months is a fraction of a business's lifespan, indicating that they will have to do a complete redesign and development of their website several times within a business's existence. A total redesign and development can take up valuable resources that a company can spend on other internal necessities, including wasting developers', designs,' and stakeholders' time. Research has shown that keeping up with design trends can extend the lifespan of a website (Crestodina, n.d.). The statistics and research have also demonstrated that developers need to keep adapting to new popular designs to have websites last as long as it was built for. Utilizing trends also sustains the user experience that evolves to a new standard every year. Customers expect to see new functionality on websites when it becomes available, creating a more personal experience than was previously accessible. Trends introduced one year can become a standard for all websites because they showcase new technologies, functions, and designs. Implementing a trend on a website that eventually

becomes a standard provides an advantage to the developer because they do not have to find a way to introduce that trend later, and their website stands out sooner. It becomes an even more significant advantage if the developer finds an effortless way to implement that trend.

One question needs to be answered for developers to keep their clients and maintain their careers. How do developers measure a website's design and functionality and determine if it is outdated? The answer is relatively simple; they do not let websites become obsolete in the first place by adding trends to the websites they manage. Implementing trends should be done yearly as new trends emerge to update websites effectively. Adding to a website, the complexities involved in most trends can take up just as much time and resources as a redesign. While this solution will keep websites modern, it is inefficient, so another solution must be found. One programming language that stands out when looking for that more efficient solution is JavaScript. In 2020 JavaScript was the most utilized tool web developers used to create interactive, advanced, and maintainable websites (Levlin, 2020, i). The maintainability of websites is the most significant power employed to develop a solution. Since it was released, JavaScript has come a long way and provides perfect frameworks to use to build a new framework on top of an existing one.

Research into the JavaScript framework Angular has proven positive compared to the trends seen in the past five years (Freeman 2017). Angular also can render APIs effectively, creating an easy way to add components to any website (Getting Started with Angular 2022). The new framework that will be designed to add trends to websites easily will be called Trend.js. The research will go into making the code snippets for each trend easily incorporated on all types of websites. The University of Southern California Norris Medical Library's website

redesign and the American Scientist magazine website redesign demonstrates the significant differences of updating a website versus relying on a redesign and development (Wu and Brown 2016; American Scientist 2017). Both ended up completing a redesign and could have avoided it if the ability to use Trend.js was an actuality. Trends from 2016, when Norris Medical Library's new website was completed, could have been utilized instead to create a fresh feel (Cardello 2016). Trends from 2017 would have given the American Scientist magazine's old website that same feel (Papp-Dinea 2017). To supply developers with the means to rise to the challenge of keeping websites updated, Trend.js, a JavaScript framework built from Angular, will provide yearly trend components that can be easily added to any website. Furthermore, the monetary and timely cost of using Trend.js will prove to be less than the alternative. The cost to develop Trend.js will play a factor in its overall effectiveness capacity. Every nuance of Trend.js will be examined to guarantee the success of its effectiveness of the solution beyond the development Trend.js.

Approach

I identified that websites are not being updated regularly, ending in businesses needing to do a complete redesign and development inside the predicted existence of the company. The lifespan of a website will be extended when it continues to stay updated. I researched and supplied evidence that websites going out of date, outworn user experience on a website, and the complexity of integrating website trends are the root causes of why websites are not staying current. I did more research and explained through experts that React, Vue, and Angular are the top three JavaScript frameworks and why Angular is the best to implement our solution. I described based on evidence that building an immortal website, updating as clients

request, and doing a complete redesign and development of a website are not efficient and effective solutions. All evidence and research are presented in the literature review. I then used the solution section to indicate that a new JavaScript framework built off Angular will solve the issue of not having an easy and effective way to update websites with trends described in the background section of this paper. Next, I explained why Trend.js solved all the issues revealed in the literature review. I broke Trend.js down into three sections as a solution, the framework, the business side of Trend.js, and the community revolving around Trend.js. I came to this conclusion by looking at the top three JavaScript frameworks and comparing them to the top three trends of the last five years, which is also found in the literature review. As mentioned, Angular excelled beyond the other two top frameworks. The discussion section identifies two case studies that show why creating Trend.js was the only possible solution. I also identified what was needed to complete Trend.js which is found in the discussion section of this paper as well. Lastly, in the recommendations and conclusions section, I outlined the next steps to get Trend.js distributed amongst developers. Identifying the next steps included recognizing ways to build a community around the new framework and a business. The only way to prove that Trend.js is the correct solution is to research other solutions and its functionality.

Literature Review

Outdated Websites Hurt Client Businesses

"When you, your business, or your visitors change, your website ages. It's old as soon as it's out of sync with your business and is not getting you those measurable results (Crestodina, n.d.)." Crestodina (n.d.) outlines website life expectancy factors to help determine if a website has become a problem due to its age. As stated in his quote, the first question Crestodina (n.d.)

asks when looking at a website's lifespan is, has the central message of the business evolved. Other factors include whether the content strategy has changed, if the industry has evolved, if the company is in the design or technology-related sector, if the website is challenging to update, if there has been a decrease in rankings and traffic, and if competitors keep their websites up to date (Crestodina n.d.). Crestodina (n.d) explains that keeping up with design trends and having a mobile responsive website is crucial when expanding its lifespan. His factors indicate how to measure if your website is outdated and how it becomes a problem.

Ninety-four percent of a website's first impression is dependent on the website's design (Bleu 2021). In a list, Hopkins (2017) explains ten signs that your website is grossly outdated; these include archaic design, stale content, low search content, low search rankings, your website not being mobile-friendly, painfully slow load times, if there is too much flash, any inaccuracy on your website, font inconsistencies, broken pages, and poor social media integration.

This issue is further examined in Brinker's (2022) Website Mistakes That Can Hurt Conversation. He states that 75% of customers admit to determining the credibility of a business based on their website design. While Hopkins (2017) lists slow website loading as a sign of an outdated website, Brinker (2022) agrees and continues the statement by pointing out that a two-second load time is an expectation for most customers and that 79% of customers will abandon their carts because of a slow loading website. Website clutter, confusing site navigation, and missing or perplexing call to actions are other mistakes identified on outdated websites (Brinker 2022). Overall, Brinker's (2022) advice for a great website is not to assume

customers know anything, be direct and specific, and gently nudge them where you want them to go.

Outworn User Experience

Who is the intended target of the websites built and put on the internet? The answer is, of course, users, customers, or people in general. When a person goes to a website, they expect a certain level of functionality. An article on Resource Technique's website in 2019 showcases signs that point to poor user experience. First and foremost, if a website has confusing navigation, the user will be lost and become annoyed (Resource Techniques 2019). The article also suggests that forcing requests upon users is a great way to lead a customer to abandon your website; this includes tools like popup windows (Resource Techniques 2019).

Sales can be a huge motivation to have a website, but sales being the only driving factor, can do the opposite and have customers fleeing (Resources Techniques 2019). The Resources Technique's article (2019) talks about how you risk annoying your customers if there is absent or too much information on your website. Outdated content and design, a website not being mobile-friendly, and slow page loading times coincide with signs that your website needs a refresh (Resource Techniques 2019). Not having the newest user experience features on your website is an indication a refresh is necessary as well (Resource Techniques 2019). Not only are indicators of outworn user experiences essential to see what causes the issue of not having an updated website, but looking at how user experience evolves can give insight.

Keeping up with users' needs and expectations has not been easy over the years and can contribute to why websites do not stay consistently current. Kollin (2016) describes some key user experience evolution factors that continue to change in his article User Needs and

Expectations. "Looking at the recent evolution of user experience design patterns might give you a sense of how user expectations have been changing (Kollin 2016)". Kollin (2016) points out that design trends evolve for users; even though designers use them, they cater to users' new needs. The evolution of users' needs and expectations includes changing websites to be continually simplified, creating clean interfaces, and fewer visual hints (Kollin (2016). A prime example of the user experience transformation over time is that Kollin (2016) mentions that a couple of years ago, the biggest challenge for user experience website designers was to make sure the design worked in every browser. Today the most significant challenge is to make sure the design works on every screen and device (Kollin 2016). It has been demonstrated that in order to solve the problem of having an outdated website as well as falling behind on user experience, there is an expectation that designers and developers will keep up with new trends.

The Complexity of Integrating Website Trends

Before even touching the code of a website, the first thing is to figure out how to stay up to date with web design trends. Koening (n.d.) helps explain how to do just that in his article How to Keep up With Web Design Trends**.** One way to achieve this is to subscribe to blogs that report on the latest trends (Koeing n.d.). Social networking in web designer communities and joining reading and contributing to web design internet forums is another way to stay on top of trends. Joining online web design communities to identify new trends and subscribe to web design magazines can also help (Koeing n.d.). Koeing (n.d.) suggests visiting top-ranking websites to track what they are regularly doing. Koeing (n.d.) also indicates that creating alerts on Google for articles with keywords like "web design trends" or "web development" can help

tremendously. Keeping up is only half the problem with updating websites; integrating new trends is a significant factor when trying to implement them.

Ninety percent of the world's data was generated in the past two years alone, with the number growing each year (Fireart 2021). An article on Fireart's (2021) website stipulates that a software developer is required to keep a company's website up to date, indicating that regular front-end web developers will not cut it. The Fireart (2021) article explains that software developers turn code into compelling features. Another example of how implementing new trends is challenging is to examine the implementation of the most significant trend that has become a standard, mobile responsiveness. Today mobile responsiveness is still listed as a trend, but the new trend involves updating the mobile trend to responsive design (Ionos 2022).

Even HTML and CSS standards have been updated to reflect the need for responsive websites (Leon 2016). In Leon's (2016) book Moving to Responsive Web Design, the team working to bring existing static sites into a responsive web design world recounts that they thought the time it would take to make the transitions seemed insurmountable and nearly impossible. The caveat to adding new trends to an existing website is that there is not the luxury of starting from scratch, which would be ideal, especially in the case of adding responsive design (Leon 2016). With all these challenges, very few solutions have been attempted to keep websites updated; instead, a redesign and development are usually administered. However, JavaScript leads us down the path of how to create Trend.js.

JavaScript and JavaScript Frameworks

In 2020 JavaScript was utilized most by web developers out of all other tools to create interactive, advanced, and maintainable websites (Levlin 2020, i). It was the most used tool

because, according to What is Javascript (2022), it allows the implementation of complex features, including interactive maps, animated 3D, and timely content updates. More specifically, JavaScript can store helpful values inside variables, perform operations on pieces of text, have events trigger code to run on a web page, and introduce the use of Application Programming Interfaces (What is JavaScript 2022). APIs make sets of code easy to implement where it might otherwise be hard or impossible to do so (What is JavaScript 2022). To make writing JavaScript more straightforward JavaScript frameworks were created.

Creating, designing, and maintaining websites has never been easier due to the front-end frameworks developed in the last decade using JavaScript as a foundation (Levlin 2020, i). Levlin (2020) continues to state in his thesis paper that in 2020 the three most popular front-end JavaScript frameworks were React, Vue, and Angular. Even though JavaScript frameworks are viewed as a front-end developer tool, most are used to build web applications, not specifically websites (Freeman 2017). All three frameworks emphasize creating extendable, maintainable, and standardized applications by being built around a pattern called Model-View-Controller (Freeman 2017, 1).

Application programming interfaces, or APIs, are similar to applications found on your phone but contain a software interface (Berlind 2015). When a business wants a function on their website that is not readily available, an API can act as a bridge for that function's code and data to be implemented on their website (Berlind 2015). Berlind's 2015 article, What Are the Benefits of APIs, explains that anyone using the same API will get the same result. APIs are easily moved from one website to another because of the standard interface on APIs (Berlind

2015). Berlind (2015) also points out that so long as the connection through the API is standard, customizations can be made, and the feature should still be accessible.

Angular Vue or React

Table 1 Comparison of React, Vue, and Angular to Trends of the Last Five Years

Trends	React	Vue	Angular
2017 – Innovative Scrolling and Parallax (Papp-Dinea 2017)	No	No	No
2017 – Cinemographs (Papp-Dinea 2017)	No	No	No
2017 – Animations (Papp-Dinea 2017)	No	No	Yes
2018 – Component Based Design System (Cao 2018)	Yes	Yes	Yes
2018 – Polygonal Shapes and Geometric Layers (Cao 2018)	No	No	No
2018 – Complex Desktop & Simple Mobile Design (Cao 2018)	No	No	No
2019 – Animation: Organic Motion (Awwwards 2019)	No	No	Yes
2019 – Micro interactions & Minigames (Awwwards 2019)	No	No	Yes
2019 – Custom Cursors & Mouse Effects (Awwwards 2019)	No	No	No
2020 – Liquid Animation (Cousins 2020)	No	No	Yes
2020 – Audio User Experiences (Cousins 2020)	No	No	No
2020 – Streamlined Navigation (Cousins 2020)	No	No	No
2021 – Neomorphism in User Interfaces (Medium 2021)	No	No	No
2021 – Voice User Interfaces (Medium 2021)	No	No	No
2021 – Accessibility (Medium 2021)	No	No	No
2022 – Scrollytelling (Dvora 2021)	No	No	No
2022 – Typography Animation (Dvora 2021)	No	No	Yes
2022 – Imagery Multilayers (Dvora 2021)	No	No	No

Source: Freeman 2017; Gackenheimer 2015; Nelson 2018

React, Vue, and Angular all have their unique benefits and downsides for using them. React was initially created by engineers at Facebook to handle complex user interfaces and ever-changing datasets Gackenheimer 2015, 1). Although popular with front-end developers, React is technically not a framework exclusive to the web (Getting Started with React 2022). According to Getting Started with React (2022), it renders specific environments by using other libraries; an example of this is that it can be used to build mobile applications. When being utilized as a framework, React works in tandem with ReactDom (Getting Started with React

2022). Getting Started with React (2022) mentions that while it can be used to implement small pieces of an interface, it is not as easy as other frameworks like Vue to add it to an application and would be better to use when building an entire app with React. Table 1 indicates that not many of the trends in the past and upcoming trends can already be accommodated using React (Gackenheimer 2015, 3). All these factors make React not the best framework for a solution.

On the other hand, Vue is a progressive framework because it allows the start of the app development process with minimal effort (Nelson 2018, 1). Nelson (2018, 1) stated that Vue specializes in creating single-page web applications, and most of its library focuses on the view layer. Another great functionality of Vue is that, as required, it can adapt to additional libraries for functionality and scale in complexity (Nelson 2018, 1-2). Vue is also one of the only frameworks that can be used to enhance existing HTML by using a unique HTML template syntax (Getting Started with Vue 2022). Like most other frameworks, Getting Started with Vue (2022) states that it allows the creation of reusable blocks of markup through components. Nelson (2018, 149) points out that one of the reasons Vue is so popular because of how easy it is to get started and produce results without the hassle of other incorporating other popular frameworks. Vue has an enormous amount of flexibility when using it; however, as seen in Table 1, it still does not accommodate the integration of many of the past trends and current trends (Nelson 2018).

Angular, technically a development platform built on TypeScript, includes many benefits that React and Vue cannot produce when integrating trends into a website (Getting Started with Angular 2022). Being sponsored and maintained by one of the world's biggest companies, Google, Angular can be found in some of the largest and most complex web applications

(Freeman 2017, 1). Getting Started with Angular (2022) lists that the Angular platform includes a component-based framework, has a collection of well-integrated libraries, covers an expansive range of features, and has a suite of developer tools to help with current projects. Freeman (2017, 1) indicates that Angular has some of the best aspects of server-side development used to enhance HTML in the browser and create a more straightforward and effortless foundation for building applications. On top of all its functionality, it also allows for APIs that are very effective and efficient (Acosta n.d.). Table 1 shows that Angular can accommodate the most past and current trends compared to Vue and React (Freeman 2017).

Other Solutions

Currently, there are three solutions to the problem of how to keep websites updated, build an immortal website, update as we go, and complete a redesign and development. Many sources have demonstrated that the root causes of not keeping websites up to date is that they become outdated with old content and design, the user experience of a website becomes obsolete, and that implementing website trends into websites are complex and demanding, so a solution that addresses those issues from the beginning could be a good solution. How is creating an immortal website possible? According to an article by Kingsley-Smith (2018) called Web Design That Stands the Test of Time, four principles will help your website stand the test of time, appearance, content, functionality, and usability.

Kinsley-Smith (2018) states that an attractive website is the key to a positive first impression and is more likely to get your customers to stay. He suggests using a color palette that compliments the brand, ensuring the website is easy to read, the photography is good quality, and the website is simply designed (Kinsley-Smith 2018). Keeping content organized

and up to date is a must, and verifying that it is speaking to the visitor of the website will also extend the website's lifespan (Kinsley-Smith 2018). According to Kinsley-Smith (2018), making sure everything on the website functions correctly and is nice to use is critical when wanting it to stand the test of time. While these are all good practices in general when building a website, Kollin (2016) and Koenig (n.d.) demonstrate that trends and user experience evolve, creating signs of an outdated website laid out by Crestodina (n.d.), making this not a very practical solution.

Another solution in practice is to update a website periodically as needed. Leto (2022) put together a list of reasons why a website should be updated. Among the top listed reasons are that the website does not stand out, it does not reflect the business's brand, the right clients are not being attracted to it, the website uses outdated themes and technologies, and it is not mobile-friendly (Leto 2022). Other reasons include the website loading slowly, the bounce rate is high, the security of your website being compromised, and if the website needs more call to actions (Leto 2022). Finally, Leto (2022) indicates that new apps and integrations is another reason to update a website. Except for an app or integration that allows for easy updates, the article on Fireart's (2021) website states evidence that these updates are not easy and would require a software developer to keep any website up to date.

Finally, the last solution, and most popular, is a complete website redesign and development. Sankarnarayan (2018) discusses in his article Why Should You Redesign Your Website, why it is necessary to do a complete redesign. He states that a redesign is to show customers the business still cares about their website, match the company's rebranding, include better content strategies, and better compete with competitors (Sankarnarayan 2018).

Like Crestidona (n.d.) offers evidence that a website's lifespan is two years and seven months, so does Sankarnarayan (2018) when no one is updating it. He uses this information and signs of an outdated website like Hopkins (2017) to indicate a complete redesign is the solution (Sankarnarayan 2018). Next, there is an overview of how much a website redesign will cost and how long it takes to complete. Sankarnarayan (2018) indicates the cost could be between $650 to $20,000 depending on the complexity of the redesign and could take from 5 working days to 30 working days. While this solution ticks all the boxes as far as what is needed for a solution, it would be much more cost-effective and time-sensitive to use all the benefits of a framework like Angular described by Freeman (2017), Acosta (n.d.), and Getting Started with Angular (2022) for a solution.

Solution

The solution to the issue of developers keeping websites current is relatively simple; a tool is needed to help them stay on top of updates. Tools that are presently used do not sustain the turnaround time required to keep a website fresh, let alone multiple websites. The Trend.js framework will be the new tool that maintains a contemporary website using trends. Trend.js will not just be a tool, but a community gathered around the need to create a solution to this issue. It will also be a company that will provide the service needed to sustain an evolving framework. Trend.js being a tool, community, and company will continue to solve the outdated website problem year after year.

Trend.js as a Tool

Trend.js needs to produce an effortless process to add new trends to all websites. This goal will be completed by constructing Trend.js from Angular, a popular JavaScript framework,

that showed evidence of accommodating five more trends than React and Vue from the last five years (Freeman 2017). Continuing to utilize Angular's assets, Trend.js will retain its component fragment, making code for each new trend contained in its own unit. Angular's impeccable API rendering rate will be employed for Trend.js as well, creating a simple singular connection point for all new trend components (Getting Started with Angular 2022). Trend.js features a database of code shortcuts that are used to activate and implement a trend once its component is plugged into the website.

Trends are the driving force behind Trend.js and will be what is used to sustain fresh websites. Trends will be researched for the coming year, and the most important ones will be selected to turn into the components part of Trend.js. Then those new trend components will be provided for utilization. It does not matter if the latest trends are based on design or functionality, Trend.js provides code components for each type. The trends aspect of Trend.js is where the framework, community, and business characteristics have crossover.

Trend.js as a Community

The community feature of Trend.js, hosted on its website, creates an opportunity for all Trend.js developers to network. It will promote ways to find motivation and new knowledge when using components on websites. Community members will also have a chance to expand their areas of expertise by immersing themselves in the diverse pool of Trend.js developers. It will host a place for issues and questions to be discussed in a timelier manner and by multiple fellow Trend.js developer perspectives. Examples of other developers' work will be displayed to provide insight into different ways to use trend components on websites. The community hub can help developers realize the potential that can be discovered when participating in Trend.js

projects and challenges. More specifically, the community section of the website will include a place to vote for the trends that developers want to turn into components and a form for developers to submit other trend ideas. However, none of this is possible without the third part of the Trend.js solution, Trend.js as a business.

Trend.js as a Business

The business characteristic of Trend.js will be a highly functional website maintained by developers; the staff needed to do trend research and manage the developers, and determine and implement outside elements required to keep the business going. The website maintained by developers will offer a place to find all trend component code, API instructions, new trends that Trend.js is looking into for the next year, the database of code shortcuts used by Trend.js, and any other updates to the Trend.js framework that developers need to know. The team of developers at Trend.js will also provide year-round customer service needs by fixing any issues found with components and answering questions presented in the website's chat application. The additional staff will tend to all the trend duties, including researching which trends are coming out the following year and deciding which ones to make components. They will also support developers who have problems with their Trend.js website account. Funding will need to be provided to keep the staff working all year long and for many years to come.

Trend.js is an all-encompassing multifaceted solution for outdated websites. Trend.js as a framework provides the physical solution, as a community provides a supporting solution, and as a business provides a maintenance solution. Sustaining a website's lifespan by using trends is a substantial task, so what makes this solution the correct solution for that task? Are all the pieces of such a big solution covered, and are there any limitations to the proposed solution?

Discussion

The effectiveness of Trend.js, until put into use, can only be measured by the functionality it is built to implement. The best way to demonstrate Trend.js's function is to show examples of times when using it would have either expanded the predicted lifespan of two years and seven months of a website or prevented a complete website redesign and development (Crestodina, n.d.). Two companies, the University of Southern California Norris Medical Library (NML) and the American Scientist magazine, will showcase that if Trend.js was used, they would not have had to complete a redesign and development of their websites. If trends like user experience and responsive web design had an easy way to be updated and executed on their websites, the lifespan of those websites would have extended exponentially. Lastly, if they had known the complexities of a redesign and development along with the time and money it would have wasted, they would have used Trend.js from the beginning of its inception.

User Experience

User experience was constantly evolving in 2016 and 2017 when NML and the American Scientist magazine did their website refreshes. Kollin (2016), in his article User Needs and Expectations, gave examples of the ever-evolving expectations of users, including websites becoming more and more simplified, with clean interfaces and fewer visual hints. These new expectations were one of the reasons NML decided it was time to do a website redesign after ten years of outdated user experience (Wu and Brown 2016, 158). The featured article about their new website described a couple of significant issues with their old website's user experience. First, the parts of their website that were accessed most often were not visible

from the homepage (Wu and Brown 2016, 158). Necessary resources were hard to find by their website's users (Wu and Brown 2016, 158). The physical library's usability had changed and evolved, and they needed their website to reflect its current state by updating its usability (Wu and Brown 2016, 158).

If Trend.js was used from the beginning of the NML website's creation, it would have evolved throughout time, making its lifespan indefinite. However, if Trend.js was created in 2015 and NML wanted to utilize it to update their website instead of redesigning and developing, there would have been a few trends they could have taken advantage of, provided they were built into components. NML was concerned about hidden content, but one of the new trends in 2016 was a scrolling trend where if administered, they would have had much more room on the homepage of their website to add additional links to bring the user to the hidden content (Cardello, 2016). Direct chat applications were another trend that appeared in 2016 (Cardello, 2016). It could have helped users have the ability to reach out any time they needed help without having to move off whichever page they were on. Using the trend components from Trend.js to change NML's old website's functionality to match how the library had changed would have saved them time and money. An example of updating their website as the library changed could have been to use the 2016 trend of relying on images over text (Cardello, 2016). Showcasing new photos of the adapted library and minimizing the text on their website would have given it a fresh feel.

The American Scientist magazine had more specific outdated user experience issues they wanted to correct with a new website. They partnered with Sigma Xi staff, Systems Solutions Incorporated, and Advontemedia to create a user-friendly website (American Scientist

2017, 256). A feature they added with the new website was integrating images and videos (American Scientist 2017, 256). The new design aligned perfectly with the goals for its audiences and partners (American Scientist 2017, 256). One of the most significant advancements about the revamp was offering users relative content based on visitor preferences, such as engineering, cartoons, and online exclusives (American Scientist 2017, 256). This achievement alone offered users better functionality and catapulted their website so far forward from where it was. Being able to offer content based on preferences is something a user would expect to have the capability to do in 2017. Finally, the American Scientist magazine wanted to create alignment between technology and visually through customization (American Scientist 2017, 256). This alignment would heighten the reader's experience by building content discovery and presentation (American Scientist 2017, 256).

The best way to use trends that Trend.js would have created into components in 2017 would have been to use the trends that push the customizations even further. 3D elements, bold colors, and animations, all trends in 2017, could have been used to make customization for each user stand out even more (Papp-Dinea 2017). Custom graphics and illustrations would have been another layer of customization for users to experience on the American Scientist magazine's website (Papp-Dinea 2017). Cinemagraphs, a trend that appeared in 2017 that showcases a still image with small parts of that image moving, would have solved the goal to align technology with visualization (Papp-Dinea 2017). Trends from the year each new website was created would have helped in the complexity of making a website have better user experience and take on responsive web design.

Responsive Web Design

In 2010 mobile web design became a new trend (Leon 2016). Who knew that a form of mobile design, responsive web design, would still be a trend even today? For a trend to be a trend for over ten years, demonstrates how it become a standard in the web design and development industry. The fact that a trend became a standard indicates how important it is for developers to follow trends and add them to websites whenever possible. It is also evidence that Trend.js will aid in the effectiveness and efficiency of updates on a website.

When the NML staff thought about what needed to be updated the most when going from their old website to their new website, the unanimous answer was that it required a mobile-friendly version (Wu and Brown 2016, 158). They could have added a mobile design to their new website but instead wanted to focus on responsive design so their website would look good on a range of devices (Wu and Brown 2016, 158). Many people, including students, use the library, so it was essential to create accessibility on any device anyone could use to visit their website. Sticking to the idea that NML decided to use Trend.js in 2016 instead of doing a complete redesign, they could have chosen to apply a mobile responsive trend component to their website, solving the issue before it became one.

When the American Scientist magazine researched ways to make their new website better, they realized that a third of their users were coming to it using a mobile device (American Scientist 2017, 256). With that information in hand, the main goal of their new website was to focus on incorporating a mobile-optimized version (American Scientist 2017, 256). Creating an optimized version of their website meant converting the thousands of articles hosted on their old website to a mobile-friendly format, which could be a daunting task

(American Scientist 2017, 256). The last reason the American Scientist magazine wanted to create a mobile-friendly version of their website was to aid in the capitalization of organic content discovery through social media and shared links (American Scientist 2017, 256). If by 2017 the American Scientist magazine had already been adding the trends Trend.js had turned into components over the years, they would not have had to worry about transforming all their articles to mobile versions at one point. They could have slowly and easily incorporated their articles as they integrated the new trend into their website.

Drawbacks of a Website Doing a Full Redesign and Development

Currently, it is the standard to create a redesign and then develop a website redesign when significant updates need to be made; however, this has proven to be very expensive and time-consuming. For example, when the NML staff set out to redesign and develop their old website, they realized that the process was a lot more complex and challenging than they initially anticipated (Wu and Brown 2016, 159). The time it took to complete their new website was beyond expectations and demonstrated that big projects like this are complex and challenging to manage effectively (Wu and Brown 2016, 159).

The American Scientist magazine had a huge hurdle when completing a full redesign and development of their old website because they wanted to create mobile versions of all their articles (American Scientist 2017, 256). This task was enormous because not only did it include the written words of thousands of articles, but they also had multiple images, graphics, captions, pull quotes, and links associated with each of them (American Scientist 2017, 256). With this massive undertaking in mind, the decision to only convert the past five years in time for the new website launch was made (American Scientist 2017, 256). American Scientist

magazine partnered with numerous companies to create their redesigned website, which cost more money than the monthly subscription fee Trend.js would charge (American Scientist 2017, 256).

Crestodina's (n.d.) research has shown that keeping up with design trends will extend the lifespan of a website, and with these examples, it is easy to see why. If the American Scientist magazine had started slowly transitioning articles and everything that came along with each article when responsive web design emerged as a trend, they would have completed the transition long before the date they launched their new website. One of the best concepts of using trends to keep a website current and not creating a completely new one is that the updates could have been done in small chunks. Perhaps the American Scientist magazine could have taken six months to focus on the responsive web design trend instead of splitting the focus between everything that goes into a new redesign and development. The NML could have done the same thing, focusing on one update at a time, making the transition less complex and overwhelming. It is vital to see real cases of how Trend.js can keep websites current, but it is also imperative to talk about Trend.js and why it specifically will solve the problem.

Why Trend.js?

Trend.js will be built upon a commonly used JavaScript framework, Angular, which has excellent features to implement into Trend.js as a foundation. Angular utilizes components, which are small code snippets that solve a specific purpose (Freeman 2017). In this case, each component will have a particular trend coded and encapsulated into them. These trend components make it easier to pick and play with different trends without adding the code for all the trends Trend.js hosts in its framework. That much code added to apply one trend would

cause the website to run slowly, which would negate some of the benefits of Trend.js. APIs are incredible complex code connectors that will enable any complex trend code to be added to any website. Not only will an API be utilized, but Angular has a reputation for being able to be added to any website with minimal effort (Acosta n.d.). Finally, Trend.js is the perfect solution because no other framework, program, or application is employed in the web design and development field that supplies the ability to easily and effectively add new trends to websites, giving them a continuously modern feel.

Why does Trend.js utilize trends to keep websites up to date? Trends are what are currently prevalent in the industry, already working for other websites, presently creating new interest, and usually includes the latest technology when building sites. If we already have access to lists of trends for the upcoming year, why not utilize them. The website and development field evolves rapidly, and if Trend.js can put that movement to use, it has officially found a productive way to take advantage of it. No one else is currently taking the same benefit and turning it into a tool the entire field could use.

Strengths and Limitations of Trend.js

Trend.js has many strengths, but one of the biggest is that developers would no longer disappoint their clients because their websites would stay contemporary. Disappointed clients can hurt a website management business's reputation and having enough of them can cause those clients to leave or even put developers out of a job. Another great strength of using Trend.js is that marketing companies have an additional monetizing feature. There will still be some indication that clients will need to supply and let developers know when and how to change or update their website content. Presenting Trend.js as a feature where the client does

not need to ever worry about their website looking outdated is a great benefit. Furthermore, it prevents a complete redesign and development. It has been made apparent of the complexity seen by clients when going through the redesign process; however, it is complex for the developer as well. It would take much less time to create updates by using Trend.js.

Trend.js will never become stagnant and outdated since it is an ever-evolving framework. It must change every year to keep up with new trends. A significant statistic mentioned at the beginning of this paper was that a typical, not updated website's lifespan is only two years and seven months (Crestodina, n.d.). Trend.js would extend the lifespan of websites exponentially. Trend.js as a business with year-round staff is a massive strength because customer support will always be present when a developer needs help. A community built around Trend.js will also help to expand the reach of what it can do and help developers learn every facet of Trend.js. While there are many reasons Trend.js is an optimal solution, it also has its limitations.

Evidence was provided by Kinsley-Smith (2018) that demonstrates a solution to keeping websites up to date currently in use is to build a website that has timeless features when first creating it. The problem with that method is that websites can not look generic; they need to stand out and be eye-catching. Another solution discussed was regularly making updates to a website, which unfortunately is very time-consuming and cumbersome. It is challenging to retrofit a new feature to a website, and developers should not have to when using Trend.js. Lastly, the most commonly used practice is redesigning the outdated website and developing it, which will not work for developers because it costs more money and time than using Trend.js. If an alternative solution were presented that merits a comparison, it would be something to look

into; however, challenging Trend.js as the ultimate solution will only make it better. A guideline on moving forward with this solution will be recommended next.

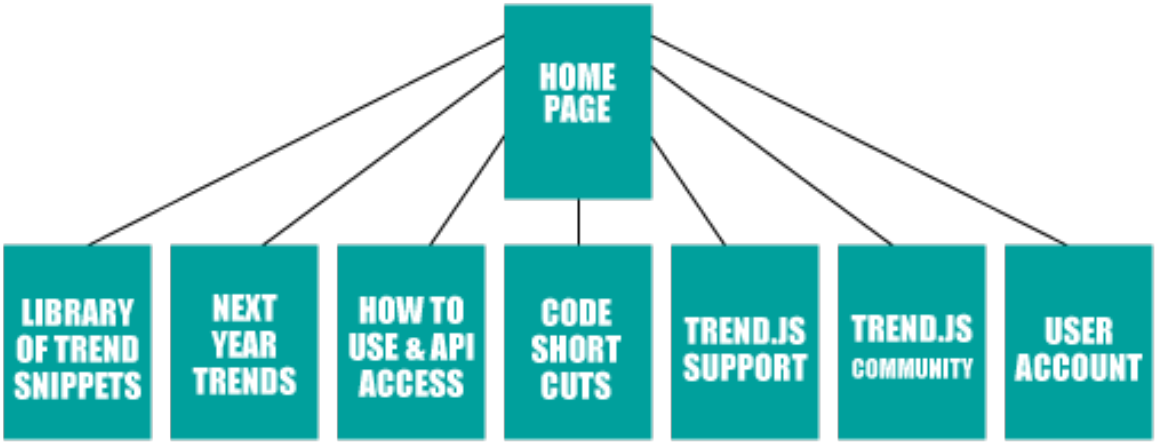
Recommendations

Now that a solution has been defined, how would it be implemented? Moving forward, Trend.js will need to be looked at in three different ways besides its framework. First, it will need to be seen as a project. Projects have an organizational factor that will keep progress happening. Next, to truly get Trend.js out to developers, it needs to be seen as a product. The business side of Trend.js will back up both the project and product aspects. Finally, it needs to be seen as a community to grow forever and flourish. What other elements do looking at Trend.js as a project bring?

First, a project proposal must be created and presented to a sponsor to get funding. The project proposal will include the scope of work, what personnel is needed to develop Trend.js, what sort of budget is required, goals of setting up this project, and what makes it a successful project. A project manager should be hired to get Trend.js off the ground and moving. They would also act as ambassadors for the project between the stakeholders and developers. All project matters will be worked out in this phase, including what the monthly subscription rate will be. It takes Trend.js out of the idea person's hands and into hands that know how to move it along. Treating Trend.js as a project is a great way to ensure that the necessary steps are taken to develop it in an organized fashion.

In tandem with Trend.js being developed is a website for it. The website will include everything that is needed to know about Trend.js. Figure 1 demonstrates the website will present as.

Figure 1 Representation of Trend.js's Website



Obviously, like most websites, the user will land on the home page. From there, access to the library of trend components, next year's trends, how to use Trend.js and API access, code short cuts, Trend.js support, the Trend.js community, and the users' account is available. The library of trend components will host an index of all the trend components ever created by Trend.js. From the index list, a link will take you to a page with the trend's component and the details of the trend. The following years' trends page will contain an active list of new up and coming trends that could be considered to add to Trend.js and two forms. The first form will give developers the chance to vote on which of the up-and-coming trends they want to see developed on Trend.js, and the second is for trends that are not on the list that developers think Trend.js should consider. The how-to-use Trend.js and API access page will include step-by-step instructions on setting up and using Trend.js. This page will consist of directions on accessing the API and having it on websites. The code shortcuts page is a list of all the small

ways to write code to work with Trend.js to make any coding to add trends easier. Trend.js support is where developers can find help when trend components are not working correctly. They will have the ability to talk to a Trend.js developer or be sent in the direction of the Trend.js community. The Trend.js community is for all developers to showcase the work they put together using Trend.js, ask for help from other Trend.js users, and discuss trends they have seen in the industry lately. Lastly, developers will access their account to manage it in the way they see fit and pay their monthly subscription fee.

Trend.js, although a JavaScript framework, will also be a product to sell. With this realization, a marketing team will need to be assembled. A marketing strategy will need to be created by that marketing team. Research needs to be done by the marketing team to narrow down platforms that advertisements would be seen by the correct demographic. Ad campaigns will need to be created by graphic designers, and Trend.js will need to continue to be marketed through the product's life. Funding will need to be allocated for the marketing piece as well.

Overall, Trend.js will define a community of developers that want to create a more meaningful way to update websites. The community is for developers who want to help each other get the most out of it and have the ability to ask their peers when needed. Passionate people who will share their experiences when using Trend.js. The community will also create a path for Trend.js to grow and prosper. Following these recommendations will ensure that the idea of Trend.js moves forward into a reality.

Conclusion

Websites can sit with no updates for a very long time. Website developers usually wait until clients indicate they want updates before they touch the websites they manage after the

first development process. Updating a website is difficult; not only does it take time and development to integrate updates, but it is also time consuming to figure out which updates to make without any indication from the client. The best way to implement any changes is to use tools to help speed up the process. Research has also shown that keeping up with design trends can help extend the life of websites (Crestodina n.d.) Unfortunately, no tools are created right now that cater to this need.

Research would suggest that by using Angular as a foundation, a new JavaScript framework can be created to become the tool developers need. Out of the top three JavaScript frameworks, React, Vue, and Angular, Angular already can accommodate the most trends from the past five years (Freeman 2017). Trend.js, the new foundation that will be created to solve this issue, uses recent up-and-coming trends to keep websites up to date. Keeping up on trends is no easy task. That is why Trend.js does it for developers, so they can add it to their websites and be done. Trend.js is the only solution that helps developers be highly effective and efficient, and also keeps their clients happy because their website is constantly updated. There are over 1.8 billion websites on the internet, and Trend.js will help websites stay current and stand out (Bleu 2021).

References

Acosta, David. n.d. "What is Angular". Published by Packt Publishing. Course video, https://learning.oreilly.com/videos/modern-web-design/9781789612813/9781789612813-video1_2/

American Scientist. 2017. "American Scientist Launches Website Redesign." American Scientist, July 1, 2017. https://web-p-ebscohost-com.du.idm.oclc.org/ehost/pdfviewer/pdfviewer?vid=2&sid=64c2ff91-0426-422b-bb72-52ff08c6b384%40redis

Awwwards. April 11, 2019. "Web Design Trends 2019: Voice Interfaces, Image Search, Alexa and Other Crazy Things That Are Rocking Our World". Accessed January 14, 2022. https://www.awwwards.com/web-design-trends-2019.html.

Berlind, David. December 3, 2015. "What is an API, Exactly?". Accessed March 1, 2022. https://www.programmableweb.com/news/what-api-exactly/analysis/2015/12/03.

Berlind, David. December 3, 2015. "What are the Benefits of APIs?". Accessed March 1, 2022. https://www.programmableweb.com/news/what-are-benefits-apis/analysis/2015/12/03

Bleu, Nicola. December 13, 2021. "37 Latest Web Design Statistics And Trends For 2022." Accessed January 9, 2022. https://bloggingwizard.com/web-design-statistics/.

Brinker, Mark. January 26, 2022. "Research Shows Having A Bad Website Can Hurt Your Business". Accessed February 6, 2022. https://www.markbrinker.com/a-bad-website-can-hurt-your-business\

Cardello, Jeff. July 26, 2016. "17 Web Design Trends for 2016". Accessed February 19, 2022. https://webflow.com/blog/17-web-design-trends-for-2016

Cao, Jerry. April 16, 2018. "Web Design Trends 2018: The Complete Guide for Designers". Accessed January 13, 2022. https://www.uxpin.com/studio/web-design/web-design-trends-2018-the-complete-guide-for-designers/.

Cousins, Carrie. January 14, 2020. "Top 11 Web Design and UI Trends for 2020". Accessed January 14, 2022. https://designmodo.com/web-design-trends-2020/.

Crestodina, Andy. Orbitmedia. n.d. "What is the average website lifespan? 10 Factors in Website Life Expectancy." Accessed January 9, 2022. https://www.orbitmedia.com/blog/website-lifespan-and-you/

Dvora, Shany. December 21, 2021. "10 Web Design Trends to Expect in 2022". Accessed January 13, 2022. https://elementor.com/blog/web-design-trends/.

Fireart. December 15, 2021. "Top Programming Languages That Will Rule in 2022". Accessed February 8, 2022. https://fireart.studio/blog/top-programming-languages-that-will-rule-in-2021/

Freeman, Adam. 2017. "Pro Angular: Second Edition". Springer Science+Business Media New York.

Gackenheimer, Cory. 2015. "Introduction to React". Springer Science+Business Media New York.

Getting Started with Angular. 2022. "Getting Started with Angular". MDN Web Docs. Accessed February 13, 2022. https://developer.mozilla.org/en-

US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Angular_getting_started.

Getting Started with React. 2022. "Getting Started with React". MDN Web Docs. Accessed February 13, 2022. https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started.

Getting Started with Vue. 2022. "Getting Started with Vue". MDN Web Docs. Accessed February 13, 2022. https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Vue_getting_started.

Hopkins, Matt. 2017. "10 Warning Signs Your Website is Grossly Outdated." Croplife 180, no. 2: 42, https://du.idm.oclc.org/login?url=https://www.proquest.com/trade-journals/10-warning-signs-your-website-is-grossly-outdated/docview/1888648938/se-2?accountid=14608.

Ionos. January 12, 2022. "The Most Popular Web Design Trends for 2022". Accessed February 8, 2022. https://www.ionos.com/digitalguide/websites/web-design/webdesign-trends/.

Kingsley-Smith Guy. November 23, 2018. "Web Design That Stands The Test Of Time". Accessed February 8, 2022. https://www.linkedin.com/pulse/web-design-stands-test-time-guy-kingsley-smith.

Koenig, David. n.d. "How to Keep Up With Web Design Trends". Accessed February 6, 2022. https://smallbusiness.chron.com/add-graphics-blogger-26478.html.

Kollin, Zoltan. May 23, 2016. "The Evolution of UX Challenges". Accessed February 6, 2022. https://uxdesign.cc/the-evolution-of-ux-challenges-5e1748b82ede

Leto, Nata. 2022 "11 Reasons Why You Should Update Your Site this Year". Accessed February 13, 2022. https://flothemes.com/why-update-your-site/

Leon de, Inayaili. 2016 "Moving to Responsive Web Design: Bring Existing Static Sites into Today's Multi-Device World with Responsive Web Design." Springer Science+Business Media New York.

Levlin, Mattias. 2020. "DOM benchmark DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue, and Svelte." Masters thesis, Abo Akademi University.

Medium. January 4, 2021. "Coolest 25+ Web Design Trends In 2021, That'll Rock The World". Accessed January 15, 2022. https://medium.com/quick-code/coolest-25-web-design-trends-in-2021-thatll-rock-the-world-388eb1144b7e.

Nelson, Brett. 2018. "Getting to Know Angular: Learn to Build Single page Applications in Vue from Scratch". Springer Science+Business Media New York.

Papp-Dinea, Loredana, and Mihai Baldean. January 30, 2017. "2017 Design Trends Guide". Accessed January 13, 2022. https://www.behance.net/gallery/47810259/2017-Design-Trends-Guide.

Sankarnarayan. August 31, 2018. "Why Should You Redesign Your Website?". Accessed February 13, 2022. https://colorwhistle.com/why-should-you-redesign-your-website/.

Resource Techniques. April 24, 2019. "Signs That Point To Poor User Experience". Accessed February 6, 2022. https://www.resourcetechniques.co.uk/news/web-design/signs-that-point-to-poor-user-experience-101495.

What is JavaScript?. 2022. "What is JavaScript?". MDN Web Docs. Accessed February 13, 2022. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript.

Wu, Jin, and Janis F. Brown. April 7, 2016. "Website Redesign: A Case Study". Accessed January 18, 2022. https://doi.org/10.1080/02763869.2016.1152142.