“How does the development of a single page application with a client-side framework + server-side API compare to a more traditional web development methodology?”

**Introduction:-**

Single-page application(SPA) page never refresh the browser because web browser will load all the scripting languages just like single page load(Flanagan, 2006). The main aspect of SPA is to feel more like the native app where the goal is that faster transition and user-friendly interactive. Traditional web applications, in which the majority of the application logic is performed on the server, and single-page apps (SPAs), in which the most of the user interface logic is performed in a web browser and communicate with the web server largely via web APIs. Additionally, a hybrid approach is available, with the most straightforward being to host one or more rich SPA-like sub applications inside a larger conventional online application( Choose between traditional web apps and single page apps, 2021).

In Web application there is always consists of two parts client side and server side(Solovei, Olshevska and Bortsova, 2017). The server-side team is responsible for storing all the data and processing it but in client-side team should provide a visual information to browser as requested as user. MPA emerged on the scene at the same time as the Internet. Multiple-page apps are designed to function in a "traditional" manner. Every modification, such as displaying the data or submitting the data back to the server, must be documented. With the introduction of AJAX, MPA learnt to load just a portion of a page, rather of the whole page, which ultimately resulted in the creation of the single page application (SPA). SPA is a development principle in which only one page is transferred to the client part and the content is downloaded only to a specific part of the page, without the need to restart the application. This allows the application to be faster and the user experience of using the application to be simplified to the level of desktop applications. The ONAFT application for the Mechatronics Robot Laboratory was developed in accordance with the SPA in order to automate the management process. The program me is designed to work with a client-server infrastructure. Among the components of the server are a RESTful API, which provides uniform access to the application's functionality, and a database, which stores information. Because the client component is a spa, you may lessen the stress on the connection to the server and enhance the user experience as a result.

**Inspect and Decision to:-**

Client-side functionality is where single page applications excel. It stops user from being able to browse between web pages, or conduct any activities in it. SPAs perform tasks in the background and retrieve data to provide a seamless experience. Loads in SPAs are infrequent, as well. Additionally, SPAs provide client-side functionalities such as drag-and-drop when compared to standard web programmed. Singe page application work even when there is a network difficulty. Changes made by the user on the client will sync with the server when the connection is re-established. SPAs provide more functionality than most HTML scripts. To build single-page apps, you must master JavaScript and TypeScript together with client-side development. developer-friendly frameworks such as AngularJS enable contemporary JavaScript coding the job for us is lifted off our shoulders through APIs. Web APIs are used when a user interacts with your SPA. It is much simplified when a client-side web API is in use. The reason is because APIs take care of the server-side logic(Prabhu, 2021). Implementing typical web functionality, such as providing exact URL representation in the address bar, is essential. A URL should state what activity the SPA facilitates and guide the user to clicking the bookmark to access it again. When the user uses the browser's back and forward buttons, the results should be proper. Read-only and read-mostly are preferred by the majority of users. The simpler web applications can't handle any additional capabilities. What if we used a search engine? To search and display the search results, the search engine shows a textbox on the first page. Using content management systems, CMS systems are mostly content-centric. The traditional web application approach is ideal for building such apps that employ server-side functionality and HTML rendering. Server-side programming and JavaScript skills are required when using a traditional web app development strategy. We may continue using web applications until there is a necessity/demand to provide greater UI/UX.

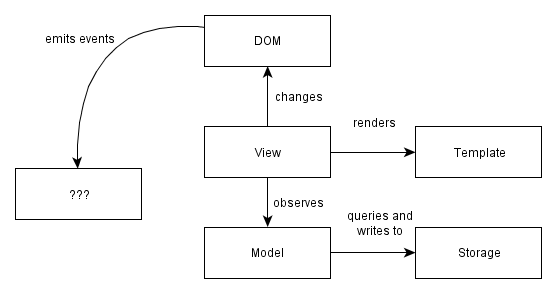


Fig: Overview CURD events occurred.

Gmail, Google, Google Maps, Twitter, and Google Drive are just a few of the many programmers that bring advantages to everyone in their everyday lives, even children. All of these are examples of Single Page Applications. A Single Page Program (SPA) is a kind of program me that operates entirely inside a single browser page while the application is being used. The Single Page Application has its own set of characteristics, each with its own set of pros and downsides. The most important characteristic of a Single Page Application is most likely the user experience, which allows the end-user to work in a welcoming environment without having to wait for any pages to refresh since the user remains on the same page. Because the application does not have to wait for the answer from the client-server all of the time while executing the procedures, the SPA is more responsive to the user's inquiry. In addition, the Single Page Application is much quicker than the Multiple Page Application (MPA). The resources like as HTML, CSS, and JavaScript are loaded during the installation of the program me and do not need to be refreshed in accordance with time norms. In order to function, the Single Page Application must follow a certain protocol or architecture. The client-side technologies such as React.js, VUE.js, and Angular are used in the SPA, whilst the server-side technologies are used in the application. Node.js is one example of this in accordance with the guidelines, the SPA has provided three choices to pick from, which are as follows: (Monteiro, 2014).

* Interpretation on the client's behalf
* Interpretation on the server-side
* Interpretation on the static side

**Pros of MPA:-**

* The Multiple-Page Program (MPA) offers consumers with a more accurate representation of the application. The essential component of a multipage application is the multilevel navigation system.
* Since MPA has numerous pages, it is simpler to advertise them with keywords because they can be specified independently for each page. which, at the time, is the most appropriate option.
* SEO (Search Engine Optimization) it is now the most widely used option for designing web-based applications today. This implies that you have a large number of options.

**Cons of MPA:-**

* The business logic of multi-page applications is intimately tied to the client side and the server side, which is not the case with single-page apps.
* Allow for changes to the server portion of the system without impacting the client portion.
* As the development progresses, it gets more sophisticated. Frameworks are required on both the client and server sides, and the developer must utilize them.
* As a consequence, the time required for application development is increased.
* It will be necessary to restart the page each time a user clicks on a link on the web site if you do not utilize Ajax.
* The development of mobile apps takes much longer. In the majority of situations, you will have to create the back-end from scratch.

**Pros of SPA:-**

* Because SPAs are compatible with a huge number of devices, by developing a single application, you may reach a far bigger audience of people than you would by utilizing a typical method.
* Application development is much simpler since there is no need to create code to render on the server.
* Single-page applications (SPAs) may be readily debugged in a web browser without the need for extra plugins. You may have a peek
* During network operations, it is possible to monitor code modifications on the fly.
* SPA is capable of caching any local storage efficiently. A programmer makes just one request, which stores all of the data, and then uses this information.
* data and functionality even while not connected to the internet.
* Because the web page is a single entity, it is much simpler to create a sophisticated, rich user interface.
* It is possible to construct large sites for the purpose of performing non-trivial activities with the aid of the SPA.
* User experience that is improved.
* Low overall support costs over the long run.

**Cons of SPA:-**

* Client frameworks that are large and need a lot of resources to be loaded on the client.
* Since not all search engines enable page rendering on the client side, it is not possible to use efficient search engine optimization strategies.
* SEO management Duplication of routers is an issue (in comparison with the classical approach).
* Because the programmed only has one entry point, there is a danger that a single mistake may result in the programmed being inoperable as a method of implementing.

**Use cases for general web applications:-**

* Traditional web apps may be started with relative ease in browsers that do not support JavaScript.
* The application is really simple and is mostly intended to provide read-only information.
* Our resources (developers) are not experienced with JavaScript, TypeScript, or front-end development, which is a problem.

**Experience through this Module:-**

First the tools or technologies I used for this assessment and the reason why used it.

Angular:-

* I am able to provide the user with a much-enhanced experience.
* Because less bandwidth is being consumed and no complete page refreshes are happening while the user navigates around the program, the program will seem to be quicker(Angular SPA: Why Single Page Applications, 2021).
* This application will be much simpler to deploy in production, at least in terms of the client portion: all we need is a static server to serve.
* I also use code splitting to divide the bundles into various portions if the situation requires it.
* Using versioning in production, the frontend portion of the program is quite basic, allowing for simpler deployment and rollbacks to prior versions of the frontend if necessary.

Node JS:-

* + - Because of its single-threaded nature, Node.js is generally used for non-blocking, event-driven services such as web servers. However, it was created with real-time, push-based architectures in mind and is thus utilized for regular web sites and back-end API services as well (Capan, 2021).
    - It is an environment in which back-end frameworks may be found and used. HTTP servers, such as Express.js (or just Express), and WebSocket servers, such as Socket.IO, are among the most widely used.
    - I used Node.js it is good not just because is of popular at what it does, but also because it has a large, active, open-source, JavaScript-based ecosystem. Furthermore, it does not have a significant impact on the compatibility across different versions.
    - Node.js is a JavaScript framework that runs on the server and executes JavaScript code in its environment, while Angular is a JavaScript framework that runs on the client (i.e., within a web browser.)
    - Because Node.js is a single-threaded programming language, it may be a poor fit for web servers that also function as computational servers, since excessive processing would slow the server's response. However, Node.js is not a poor technology in and of itself: the technology is fairly mature and is extensively used for a broad variety of various sorts of servers.

I done in node.js but because of this module I gained lot of knowledge of PHP(not familiar with language). At starting I decided do **Farmer’s welfare system** for this assessment because I alredy under post-graduation I done front-end coding including integrated PHP. But that site was left in half way. Due to this module, I have clear idea how to code for PHP and also, I restarted that project for my portfolio. I never done testing for any code but now I am having clear idea about testing the code.

**Conclusion:-**

As more and more businesses make their programs accessible on the Internet, some of them provide extensive functionality that is equivalent to that of desktop apps. This offers a competitively similar product on the market, relieves the user of the need to install software on the desktop, and ensures that the program will always be accessible on the Internet, regardless of the time of day. In order to pick the development approach, you must consider the aims of the final product. For example, if the application is tiny, has a few dynamic pages, and the primary content is static, MPA is a good choice. However, if the application has a complicated structure, contains large amounts of data, and requires a decent user experience, it is recommended that you utilize a single page application (SPA). Because SPA is a more recent method of developing web apps.

**References:-**

1. Solovei, V., Olshevska, O. and Bortsova, Y., 2017. Automation of technological and business-processes. THE DIFFERENCE BETWEEN DEVELOPING SINGLE PAGE APPLICATION AND TRADITIONAL WEB APPLICATION BASED ON MECHATRONICS ROBOT LABORATORY ONAFT APPLICATION, [online] (26324:2012). Available at: <http://www.atbp.onaft.edu.ua/> [Accessed 1 June 2021].
2. Capan, T., 2021. Why The Hell Would I Use Node.js? A Case-by-Case Tutorial. [online] Toptal Engineering Blog. Available at: <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js> [Accessed 8 June 2021].
3. Fadhilah Iskandar, T., Lubis, M., Fabrianti Kusumasari, T. and Ridho Lubis, A., 2020. Comparison between client-side and server-side rendering in the web development. IOP Conference Series: Materials Science and Engineering, 801, p.012136.
4. Flanagan, David, "JavaScript - The Definitive Guide", 5th ed., O'Reilly, Sebastopol, CA, 2006, p.497
5. Monteiro, F., 2014. Learning single-page web application development. Packt Publishing Ltd.
6. Docs.microsoft.com. 2021. Choose between traditional web apps and single page apps. [online] Available at: <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/choose-between-traditional-web-and-single-page-apps> [Accessed 11 May 2021].
7. Prabhu, J., 2021. When to Choose Single Page Apps (SPAs) and Traditional Web Apps. [online] Tech Blogs by TechAffinity. Available at: <https://techaffinity.com/blog/when-to-choose-single-page-apps-spas-and-traditional-web-apps/> [Accessed 7 June 2021].
8. Angular University. 2021. Angular SPA: Why Single Page Applications?. [online] Available at: <https://blog.angular-university.io/why-a-single-page-application-what-are-the-benefits-what-is-a-spa/> [Accessed 11 June 2021].