



Full Stack Web Development: Vision, Challenges and Future Scope

Gurjeet Singh¹, Madiha Javed², Dr. Balwinder Kaur Dhaliwal³

^{1,2,3}Lovely Professional University

Abstract: This paper provides the summary on varied aspects of full stack web development. Full stack web development could be a pace growing branch of CSE sector because of its successive step towards making the long run in the IT industry. Full stack developers manage the rear end (backend) and front-end development of web site or an online application. They handle the info, clients, system engineering and style. Full stack developer is in demand since of their skills and mastery of web development, due to this vast demand the remuneration of full stack developers is additionally high as compared to alternative jobs. Full stack developer in the main works on web stack, native application stack or mobile stack. Since, they accompany each server-side development and client-side development, full stack developers head the arrange of action and keep a track of the progress of the project. Within the finish of the paper, we will clearly outline Vision, Challenges and Future scope of Full Stack Web Development.

Keywords: Full stack web development, Front-end, Back-end, Programming, Server-side, Client-side, Development, ETC.

I. INTRODUCTION

Full stack web development is the scenario of acting on each, the front-end and back-end of a program. It is a term largely used for those operating in web development. The developers have background on making program and user expertise for front-end, and even have robust information in an exceedingly programming language that's used for handling the logic of the appliance, therefore back-end.

Full stack could be a layer of software system or web development consisting of the front-end and also the back-end parts of associate application. Front-end is what the users can see or act with on your application. Back-end part is what users don't see, like application's logic, database, server, etc. A full-stack web developer is comfortable operating with each, back-end and front-end technologies that build a web site or application perform properly.

Full Stack Developers have the specified technical

information of each and every facet of development – frontend, back end, working of various operating systems in accordance to the required environments of programming as well as tools for programming.

Full stack developers are also termed as “developer generalists”, as they will produce any complicated application from scratch, provided they have understood how each and every technical layer ought to act with the other. As Full stack web development refers to the both of each front-end and back-end of an online application, web application, the development method contains 3 layers, i.e., the logic layer (back-end layer), the presentation layer (front-end layer) and the information layer (database layer).

The major stacks of full stack web development that are commonly used to develop websites and applications are

- Linux Apache MySQL PHP (LAMP)
- Cross-Platform Apache MariaDB PHP (XAMPP)
- MongoDB Express Angular Node.js (MEAN)
- Windows Apache MySQL PHP (WAMP)
- Apache MySQL PHP PERL Softaculous (AMPPS)

LAMP stack stands for **Linux** operating system, **Apache** server, **My SQL** Query Language, and **PHP** server-side scripting language. All of these are open-source tools which are free to use, and has contributed in creating many of the developments. The lamp stack comes as the very common stack in deploying websites and applications, to the internet.

XAMPP is an acronym for **Cross-platform(X)**, **Apache(A)**, **MariaDB**, and **PHP** server-side or **Perl(P)** backend language. XAMPP is the best distribution in the industry that helps developers use local web server for deploying and testing web programs. It is being designed to simply deploy and run the web applications on the web server.

MEAN is commonly used open-source stack that is used to build the production ready web applications. This stack indicates four things which are MongoDB

NoSQL database, Express framework, Angular development platform, and Node.js execution environment. **MongoDB** is the database having NoSQL, and **Express** is most common framework that helps to create the applications like Todo Lists used in everyday life. **Angular** is the JavaScript client-side platform used to make client-side web applications, and **Node.js** is the execution environment, used worldwide to create the on-site and of-site web applications and websites.

Refer to the Fig 1. that explain the Data Flow in Mean Stack.

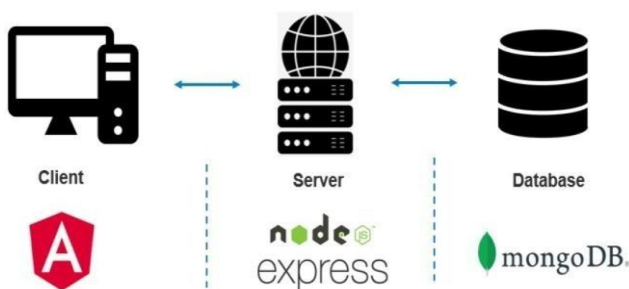


Fig 1. Data Flow Diagram (DFD) -MEAN Stack

WAMP stack stands for **Windows** operating system, **Apache**, **MySQL**, **PHP**. WAMP works same as LAMP; the difference is of operating system. The stack is commonly used for hosting web-applications and webpages from little to medium sized sites. WAMP is quite popular among windows users as it provides flexibility and scalability for the servers.

AMPPS stack stands for **Apache MySQL (MongoDB) PHP (PERL or PYTHON) Softaculous (software stack)**. AMPSS combines three different stacks i.e., LAMP(Linux), WAMP(Windows), MAMP(Macintosh). It has preinstalled open source libraries and dependencies which helps to code the website or application in efficient way and proficiently helps the development process.

II. COMPONENTS OF FSWD

The components of Full Stack Web Development are-

- Front-end
- Back-end
- Database
- Version Control

(a.) **Front-end** is the part that users see when they visit the websites, web-applications.

The two most important types of front-end designs

are-

- User Experience (UX)
- User Interface (UI)

These things are of different kind when we got to know them, otherwise they seem same. The things including visual elements, animations, pictures, videos, etc. which look good in the website, are difficult to create is an example of good **UI** but bad **UX** and vice-versa; intuitive experience that doesn't require the user to think too much is a good way to design the website.

From the programmer's perspective front-end or the part that users see when they visit the website is mainly about the design and to make to look it good somehow. The above elements **UX**, **UI** are taken in consideration in developing web programs or web pages, apps or applications for iOS or Android, Windows or MacOS.

In earlier time when somebody say "front-end" the three languages that came in programmers mind were HTML

(Hyper Text Markup Language), CSS (Cascading Style Sheets), JS(JavaScript). These languages are still used as the most preferred languages in developing webpages, websites because of their major scope in the development industry.

Some drawbacks of the above listed programming languages are there, that is they are very lengthy and time-consuming, apart from that they are very simple to write and code in assembly or high-level language containing words using modern tools like Visual Studio Code (VSC), Atom, Notepad, Notepad++, etc.

To overcome the above drawback of traditional languages, something new was built and they were the tools that provided ready-made components that helped in speeding up the development, these were called Frameworks. The frameworks such as React, Angular, Vue.js, Bootstrap, Node.js, Meteor, Flutter, GitHub, Chrome Dev Tools, Grunt, HTML5 Boilerplate, Code Kit were built from time-to-time and are currently used and upgraded from time-to time to provide developers hand in building the applications they desire to build in lesser time.

Refer to the figure below which demonstrates the various examples of components of FSWD (Full Stack Web Development).

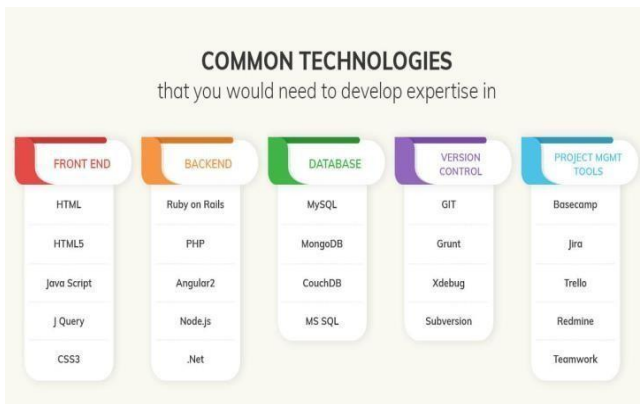


Fig 2. FSWD components

(b.) **Back-end** is the part that is hidden from the users but is a part of the development process.

The back-end part can be divided into two types –

- The server
- The application

The **server** handles all the requests that came from application’s code. There are many relations between application source code and server requests, so therefore server keeps synchronizing from time-to-time with application’s code. The block parts of the code such as `async- wait`, `try-catch`, `sync blocks` are good examples of such codes. The server handles them according to their request type and response to client-side(front-end) or server-side(back-end) depending on the types of callbacks present in the code. The architecture of the server contains requests-responses tier that communicates to make synchronization successful and helps the application run smoothly.

The **application** acts as an intermediate between the server and the database communicating all the data and requests that server or database needed in order to get the required output. The application basically contains all the code that is logical and functional which run on the server/host of website/application based on the specific requirements of the product/app/website.

There are many technologies available on the Internet that helps in creating the backend of the website/application. Some examples including PHP, Node.js, Angular2, Ruby on Rails, Java (for coding of mobile applications), Laravel (which is known as framework of PHP) are most trending and helpful in creating the back-end part of the web applications, websites, mobile-apps, etc.

The various tools used to program in these languages are: Visual Studio Code (VSC), Atom, Notepad, Notepad++, etc.

Refer to the figure 3 for the workflow process of FSWD.

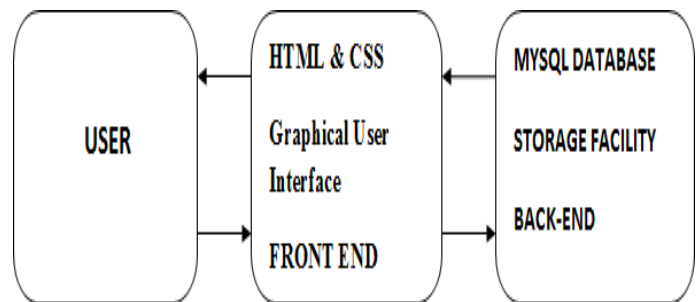


Fig 3. FSWD -Workflow Process

(c.) **Database** is the storage component in the full-stack web-development that contains all the data present in form of tables containing rows(tuples) and columns(attributes) which on requests sends data through secure transmission channel to help the application perform dynamically with a huge size of data able to send and receive simultaneously. Some examples of databases are MY-SQL, MongoDB, CouchDB, MS-SQL, etc. which comes handy while creating strong applications needing a good backend-support.

(d.) **Version Control** being called as source control, is the type of component in the full stack web development that is used to track and manage the changes to software code from time to time. There are mainly three types of version control system:

- Local Version Control System
- Centralized Version Control System
- Distributed Version Control System

The **local version control system** is available locally on your system and users are able to store every file as a repository in this system. To check the last version of the file it is important to add up all the file versions to the repository in order.

The main problem with this system is that everything is stored locally, if anything happened to the local database, all of the patches will be lost. If something happened to the single version, all the changes made would also be lost.

Also, collaborating with other developers for the same project or team becomes hard.

The **centralized version control** is the system in which there is one single center server which has all the file versions. This enables the multiple clients to work simultaneously and helps them to access files from server, pull them into own computer or push them all again onto the server from their computer. This helps in easy collaboration with other developers or a team for any project they are on to.

The biggest issue with this system is that everything is stored on the centralized server. If something went wrong to that server, everybody will lose their important data and would never able to collaborate at all.

The **distributed version control system** is a system in which clients don't just check out the files from the server, they fully clone the repository, including its full history.

Therefore, everyone collaborating on a project owns a local copy of the whole project. The main advantage of this model is that if the server becomes unavailable or destroyed, any of the client can send a copy of the project version to any other client or project member back to the server, when server is available and working fine.

Example for the distributed version control systems is Git/GitHub.

III. ADVANTAGES OF FSWD

There are various advantages of FSWD (Full Stack Web Development), some of them are listed below:

a) Accomplishment of Website/App Development

Full-stack developers have the keen knowledge and experience of the web programming languages whose work is to add groundbreaking features to the project and to make it reach to the market in a trend. The pro full stack developers can manage the whole stack from the scratch.

b) Versatility

Full Stack Developers are versatile in nature because they have skill of developing both front-end and back-end. Back-end development need application code that helps to connect to websites with other management systems. Front-end development need application code that handles the design and interaction of a site within a browser. The both types of development are taken care of by full stack developers.

c) Experience

The biggest benefit of full stack web developers is the experience of the web and app development. As full stack web developers already had experience in lots of part of web development field, so therefore they have keen knowledge of the web and app development. Thus, they are highly beneficial to the performance of the products/websites/web-apps/interfaces in the market and helps them keep updated with the ongoing market trend by their vast experience and knowledge of the components used in new technology in order to enhance the performance.

d) Jack of all trades

Full Stack Developers are commonly referred to as "Jack of All Trades". This term is most suited to them because they can work on any level of the project and provide the whole design and structure required in development.

They are also very helpful in maintaining and optimizing the existing systems as per need of the development.

Refer to the figure 4. for the advantages of FSWD.



Fig 4. Advantages of FSWD

e) Smart upgraders

From the recent study, 90% people found that website appearance is the main reason for accepting or rejecting the website. Therefore, stack developers are highly beneficial as they make websites highly user-friendly and are updated with all the latest technologies, tools which help in implementation of smart features to create the best responsive and interactive webpages, websites, apps for your clients.

f) Troubleshooting problems

Full stack developers are very experienced in troubleshooting the problems of web development field. They are acquired with necessary skills to detect the issues that come while creating the project and have necessary tools required for implementing solutions for making the website or application optimal. Therefore, they are very helpful in troubleshooting issues in the development.

g) Convenient web developers

It is appropriate thing to trust the full-stack developer for your project needs. It is because they are accountable for all the software development as they can handle web development from the initial to final phase by themselves and are very fitted in any role they put to work and help to give us the forward way. They also help with making the project quality approved and assurance of the design of the project.

h) Handling both server-side and client-side

Full stack developer helps the website, application give the required edge which it needs to shine in technology market. The cutting edge-technology of modern era requires sharp and well-qualified persons for the projects. So, full stack developers are perfect match for that as they can handle the development from initial to final phase and the server-side and client-side of the application in the development industry and give fast-track delivery.

IV. CHALLENGES AND OBSTACLES TO FSWD

a) Bugs & Broken Codes

Codes usually need scopes which is most common problem of broken codes during web development.

If the project has not been given scope properly within required blocks, under each specific labels that define the part of the code, then the developers have to work in a hurry to complete work on the scheduled time.

This cause bugs and code starts to get broken when semicolon is missing from the code leading to application crashes and bugs.

b) Premature Optimization

Speed is commonly treated as the highest good when it comes to code, at the expense of the code which is clean or legible. It is a pitfall for developers to focus

on writing optimized code that runs faster than code which can be read, understood, and maintained over the long term.

Hard to read code can slow your team down, especially when engineers come and go. If an engineer comes in and can't decipher code, it will be a big problem for team.

Premature Optimization happens when there isn't a strong code review process. Without this essential step, things get messy and cause problems in your way.

c) Stuck with Complicated New Technologies

Full Stack Developers are very much familiar with new technologies that with no surprise gives increased efficiency and flexibility in production.

They are prominent to use when developers become expert with them, but also it takes so much time in learning to install and test these new technologies. Also, it is like learning a new technology in which starting from the basics, some experience is required, which is time-taking and very challengeable in navigation. The area which comes with this issue is database storage. The problem behind all this is a lack of required leadership in handling software teams.

d) Too much or Less Abstraction

Running your code of project out of so many lines of code, could cause a massive crash and slow down your development process. Also, less abstraction makes the code looks lengthier and we are unable to understand the parts necessary for implementation of the modules we require because it is only one file.

By this way our code is more complex and cause compile-time, run-time errors which are burden to encounter.

This makes the development a serious problem for the developers and is a big obstacle for achieving the goal they need to achieve.

e) Overlooking the little things

The more you look, more you see. This phrase matches perfectly with the problem we are going to discuss here. Sometimes, when we are coding, we start to code in a such a way that creates alternative methods for achieving the same goal. Due to this, we are overlooking the little things which may or may not matter in the code.

One of overlooking of little thing is the failure to add proper sections then looking in every part of the code for required feature. Another is not giving proper logic to the code.

These little things in your code are dangerous as it would create massive problems in your project in future.

V. CONCLUSION AND FUTURE SCOPE

These days, full stack developers are in huge demand due to trend of websites and e-commerce in the modern world. Companies like to hire them due to number of reasons. FSWDs are able to work in multiple environments such as Front-end and back-end simultaneously. It gives companies the huge profit as they could get lot of work done by few professionals rather than hiring a bunch of developers to work on all parts of websites/applications. Having knowledge of different platforms stack such as MEAN stack, MERN stack, LAMP stack, WAMP stack, etc. FSWDs are able to put their effort to the project easily and get the required outputs as per benefits of company.

The salary of FSWDs is at the increasing rate worldwide as the demand is increasing for the same. Companies are willing to pay more on the basis of more unique skills and abilities of stack development to the talented full stack web developers out there.

The future of FSWDs is very bright as the developers are in huge demand and it will keep on increasing with a big rate. Listed below are some of the reasons for the same:

Smaller Teams- With a full-stack web developer in the team, you can keep the team short and smaller teams have many advantages over large ones. Even Jeff Bezos believes that a small team is better than a large team. A large team requires more communication and resources which is less required in smaller team.

Adaptability- Web development is quite a challenging field. There are a lot of factors to consider when developing a new product such as product requirements, languages, tools, plus a large developing team. In case, any of the factors got any problem such as product requirements got changed at any time and also a member leaves the team in such scenarios full stack developers can adapt to these changes quickly and more effectively than other developers.

New Technologies- Web Development tools and software keep getting new updates for better performance. Even the programming languages keep getting newer versions regularly to ensure they function better and cause less hassle. An average developer might have to put a lot of effort to keep up with these new developments. On the other hand, a full stack web developer already knows the relations between different technologies. So, keeping up with new updates and integrations is an easy task for him/her. Last but not least, full stack developers already keep learning new things about the web development, so they will remain in the huge demand for dealing with new technologies for now and in future as well.

REFERENCES

- [1] Valerio De Sanctis, Full Stack Web Development-Third Edition,2020 book
- [2] Chris Northwood, Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Developer
- [3] Frank Zammeti, Modern Full Stack Development: Using Typescript, React, Node.js, Webpack, and Docker Full
- [4] Juha Hinkula, Hands-on- Full Stack Development with Spring Boot 2 and React, Second Edition
- [5] Hassan Djirdeh, Nate Murray, Ari Lerner, Full Stack Vue: The Complete Guide to Vue.js
- [6] Nader Dabit, Full Stack Serverless, Modern Application Development with React, AWS, and GraphQL
- [7] Dr. Balwinder Kaur Dhaliwal, Gurpreet Singh, "Use of Clustering with CNN to Categorize image inside different directories on OS" - Electronic ISBN: 978-1- 6654-1703-7 Published in "2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)" (ICRITO) 3-4 Sept, 2021
- [8] G.S Shahi, R.S Batth, S. Egerton, 2020 "MRGM: An Adaptive Mechanism for Congestion Control in Smart Vehicular Network", published in "International Journal of Communication Networks and Information Security 12".
- [9] Balwinder Kaur Dhaliwal, Rattan K. Datta, "Secure and Energy Efficient Trust Aware Routing Protocol in IoT using the Optimized

Artificial Neural Network: SEETA-IoT” published in ‘International Journal of Engineering and Advanced Technology (IJEAT)’, ISSN: 2249-8958 (Online), Volume-8 Issue-6, August 2019, Page No.:4341-4353.

- [10] Balwinder Kaur Dhaliwal, Rattan K Datta,” A Trust Aware Based Predictive Model using Hybrid Convolutional Neural Network for Mobile AD HOC Network in Internet of Things” published in ‘International Journal of Engineering and Advanced Technology (IJEAT)’, ISSN: 2249-8958 (Online), Volume-9 Issue-1, October 2019, Page No.1276- 1285.