

# Full Stack Web Development: Vision, Challenges and Future Scope

Kashif Iqbal<sup>1</sup>, Vaishali Hatkar<sup>2</sup>

1-2Dept of MCA, Trinity Academy of Engineering, Pune, India

1,2

\*\*\*

**Abstract:** This paper presents a comprehensive overview of various facets of full stack web development. Rapidly emerging as a pivotal branch within the Computer Science and Engineering sector, full stack web development represents a significant stride towards shaping the future of the IT industry. Full stack developers play a central role in managing both the back-end and front-end development of websites and web applications. Their responsibilities span across data management, client interactions, system engineering, and design. The demand for full stack developers remains robust owing to their versatile skill set and adeptness in web development, consequently commanding high remuneration compared to other professions.

Full stack developers primarily operate within the web stack, native application stack, or mobile stack domains. Given their involvement in both server-side and client-side development, full stack developers assume leadership in project management, ensuring cohesive coordination and monitoring project progression.

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

## I. INTRODUCTION

Full stack web development encapsulates the holistic approach of managing both the front-end and backend components of a software application. It is a term widely employed within the realm of web development, denoting developers proficient in crafting user interfaces and experiences for the frontend, coupled with a deep understanding of programming languages essential for executing the logic and operations of the application's back-end.

This paradigm constitutes a layered architecture comprising the front-end, the visible interface through which users interact with the application, and the backend, the underlying infrastructure responsible for handling data, logic, and server-side operations. Full

stack developers possess the versatility to seamlessly navigate between these two domains, ensuring the seamless functionality and performance of websites and applications.

Full stack developers, often referred to as "developer generalists," possess comprehensive technical expertise spanning across front-end and back-end development, encompassing various operating systems, programming environments, and tools. Their proficiency enables them to orchestrate complex applications from inception to deployment, leveraging their understanding of how each technical layer interfaces with one another.

The development process in full stack web development typically revolves around three fundamental layers: the logic layer (back-end), responsible for executing the application's operations and business logic; the presentation layer (front-end), facilitating user interaction and experience; and the data layer (database), managing the storage and retrieval of application data.

Several major stacks dominate the landscape of full stack web development, each comprising a combination of technologies tailored to specific needs and preferences:

- 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 (Linux Apache MySQL PHP):** Embracing open-source tools such as **Linux**, **Apache**, **MySQL**, and **PHP**, **LAMP stack serves** as a cornerstone for deploying websites and applications, offering flexibility and cost-effectiveness.

### XAMPP (Cross-Platform Apache MariaDB PHP):

XAMPP provides a cross-platform solution catering to developers' needs for local web server deployment and testing, facilitating streamlined development workflows.

### MEAN (MongoDB Express Angular Node.js):

Emerging as a popular open-source stack, MEAN encompasses MongoDB for NoSQL database management, Express for backend framework, Angular for client-side application development, and Node.js for server-side execution, enabling rapid development of scalable web applications.

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

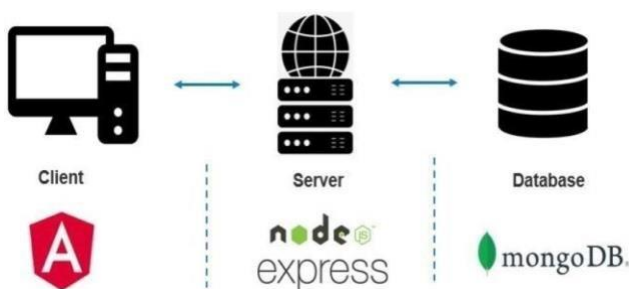


Fig 1. Data Flow Diagram (DFD) -MEAN Stack

**WAMP (Windows Apache MySQL PHP):** Similar to LAMP, WAMP stack caters to Windows users, offering a robust environment for hosting web applications with scalability and flexibility.

**AMPPS (Apache MySQL PHP PERL Softaculous):** AMPPS amalgamates the strengths of LAMP, WAMP, and MAMP (Macintosh) stacks, featuring preinstalled open-source libraries and dependencies, streamlining the development process and enhancing efficiency.

## 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)

In the realm of web development, distinctions between various elements may seem subtle until one delves deeper into their nuances. Visual components such as animations, images, and videos, while enhancing the aesthetics of a website, exemplify the essence of good User Interface (UI) design. Conversely, the creation of an intuitive user experience (UX), where users seamlessly navigate without undue cognitive burden, signifies a hallmark of effective web design. Striking the right balance between UI and UX elements is paramount for crafting engaging and user-friendly websites and applications.

From a programmer's perspective, the front-end of a website constitutes the visible interface that users encounter during their visits. It encompasses aspects of design aimed at enhancing the overall visual appeal and user interaction. Considerations of UX and UI are integral to the development of web programs, pages, as well as applications spanning across diverse platforms such as iOS, Android, Windows, or MacOS.

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

Traditionally, when referring to front-end development, three fundamental languages come to mind: HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets), and JavaScript (JS). These languages remain indispensable in web development due to their extensive scope and versatility within the industry.

However, despite their utility, these languages present certain drawbacks, notably in terms of verbosity and time-consuming nature. Yet, their simplicity in writing and coding, coupled with modern tools such as Visual Studio Code (VSC), Atom, Notepad, and Notepad++, renders them indispensable in the development workflow.

To address the limitations inherent in traditional languages, the evolution of frameworks has played a pivotal role in accelerating development processes. Frameworks such as React, Angular, Vue.js, Bootstrap, Node.js, Meteor, Flutter, GitHub, Chrome Dev Tools, Grunt, and HTML5 Boilerplate have emerged over time,



Fig 2. FSWD components

offering developers a repertoire of ready-made components to expedite development cycles. These frameworks continue to undergo refinement and upgrades, empowering developers with the tools necessary to build applications efficiently and effectively.

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

(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 –

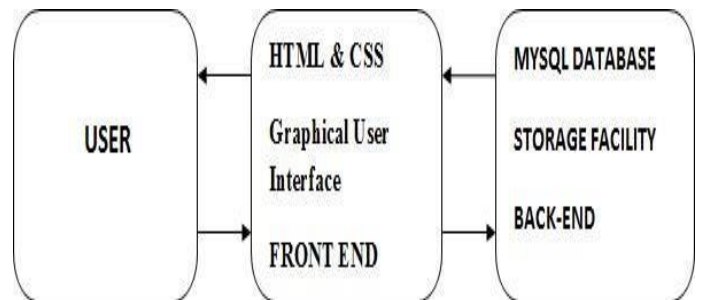
- The server
- The application

The **server** plays a pivotal role in managing incoming requests generated by an application's codebase. This interaction between the application's source code and the server necessitates continuous synchronization to ensure seamless operation. Various components within the codebase, such as asynchronous operations (async-await), error handling (try-catch), and synchronous blocks, exemplify the intricacies of this relationship. The server meticulously handles these components based on the nature of the requests, facilitating responses to either the client-side (frontend) or server-side (back-end) as dictated by the presence of specific callbacks within the code.

The **application** serves as a crucial intermediary bridging communication between the server and the database, facilitating the exchange of data and requests necessary for generating the desired output. It

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.



encompasses all the logical and functional code designed to operate on the server or host of the website or application, tailored to meet the specific requirements of the product or platform.

A plethora of technologies abound on the internet, offering robust solutions for developing the backend infrastructure of websites and applications. Examples include PHP, Node.js, Angular 2, Ruby on Rails, Java (primarily for mobile application development), and Laravel (recognized as a prominent PHP framework). These technologies are at the forefront of backend development, empowering developers to build dynamic and scalable web applications, websites, mobile apps, and more.

Fig 3. FSWD -Workflow Process

(c.) The **Database** serves as the foundational storage component within the realm of full stack web development, housing all pertinent data in structured formats such as tables comprising rows (tuples) and columns (attributes). Upon receiving requests, the database securely transmits data through channels, enabling the application to dynamically process and manipulate vast amounts of information while facilitating simultaneous sending and receiving of data.

(d.) A **variety of database** technologies are available to meet diverse application requirements, including **MY-SQL, MongoDB, CouchDB, MS-SQL**, among others. These databases play a crucial role in underpinning the backend infrastructure of robust applications, providing the necessary support for handling complex data operations and ensuring seamless functionality.

- Local Version Control System
- Centralized Version Control System

- Distributed Version Control System

**Local Version Control:**

Local version control systems operate directly on the user's system, allowing storage of every file as a repository. Users can access the last version of a file by traversing through the repository's version history. However, a significant drawback of this system is its inherent vulnerability to data loss. Since all data resides locally, any mishap with the local database or a single version could result in the loss of patches and changes. Additionally, collaboration with other developers becomes challenging due to the localized nature of the system.

**Centralized Version Control:**

Centralized version control systems revolve around a singular central server housing all file versions. Multiple clients can work concurrently, accessing files from the central server, pulling them to their local machines, and pushing changes back to the server. This setup facilitates seamless collaboration among developers or teams on a project. However, a critical downside is the reliance on the centralized server; any disruption or data loss on the server could jeopardize the entire project's integrity and collaboration efforts.

**Distributed Version Control:**

Distributed version control systems take a different approach, wherein clients not only check out files from the server but fully clone the repository along with its complete history. Consequently, each collaborator possesses a local copy of the entire project. This model offers resilience against server failures, as any client can restore the project by sending a copy of the version to another client or back to the server when it becomes available again. Notably, the distributed version control system is exemplified by Git/GitHub, providing robust and decentralized collaboration capabilities.

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 possess comprehensive knowledge and expertise in web programming

languages, enabling them to introduce innovative features to projects and propel them into the market with contemporary trends. Proficient full stack developers excel in managing the entire development stack from inception to deployment, demonstrating their ability to execute projects from scratch. **b) Versatility**

Full stack developers exhibit remarkable versatility by adeptly navigating both front-end and back-end development realms. Back-end development entails crafting application code to establish connections between websites and various management systems, while front-end development involves coding for design and interaction within web browsers. Full stack developers seamlessly handle both aspects of development, showcasing their adaptability and proficiency across diverse domains. **c) Experience**

One of the most significant advantages of full stack web developers lies in their extensive experience in web and app development. With their comprehensive exposure to various facets of web development, full stack developers possess in-depth knowledge of the field, contributing significantly to the performance and competitiveness of products, websites, web applications, and interfaces. Their vast experience and familiarity with evolving market trends enable them to stay abreast of technological advancements and enhance product performance accordingly.

**d) Jack of all trades**

Full stack developers are often likened to "Jack of All Trades," a testament to their ability to seamlessly operate at various levels of a project and provide comprehensive design and structural solutions throughout the development lifecycle. Their versatility extends beyond initial development stages to encompass maintenance and optimization of existing systems, ensuring continuous alignment with evolving project requirements and industry standards.



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



Fig 4. Advantages of FSWD

#### e) Smart upgraders

Research indicates that 90% of individuals cite website appearance as a primary factor influencing their acceptance or rejection of a site. In this context, full stack developers play a crucial role in enhancing userfrendliness and staying abreast of the latest technologies and tools. Their proficiency enables them to implement smart features that enhance responsiveness and interactivity, thus ensuring the creation of top-tier webpages, websites, and apps tailored to clients' needs.

#### f) Troubleshooting problems

Full stack developers possess extensive experience in troubleshooting issues within the web development sphere. Equipped with a diverse skill set, they adeptly identify and address challenges encountered during project creation, leveraging the necessary tools to implement effective solutions. Their adeptness in problem-solving makes them invaluable assets in resolving issues and optimizing websites or applications for optimal performance. **g) Convenient web developers**

Entrusting full-stack developers with project requirements is a prudent decision owing to their accountability and proficiency in software development. Capable of overseeing web development from inception to completion, they seamlessly adapt to various roles and contribute to project success. Additionally, they ensure project quality and design

adherence, instilling confidence in project stakeholders and paving the way forward.

#### h) Handling both server-side and client-side

Full stack developers empower websites and applications to stand out in the competitive technology market by providing the necessary edge. In the modern era, where cutting-edge technology is paramount, full stack developers emerge as ideal candidates due to their ability to manage projects comprehensively. Their proficiency in handling both server-side and client-side aspects of development ensures fast-track delivery and positions projects for success in the dynamic development industry.

## IV. CHALLENGES AND OBSTACLES TO FSWD

#### a) Bugs & Broken Codes

One of the most common challenges in full stack web development is dealing with bugs and broken codes. Oftentimes, incomplete scoping of project requirements leads to rushed development efforts, resulting in missing semicolons or misplaced blocks of code.

#### b) Premature Optimization

The temptation to prioritize speed over code cleanliness and readability poses a significant challenge for developers. Premature optimization, while aiming to enhance code efficiency, often results in code that is difficult to understand and maintain over time. This can impede collaboration within development teams and hinder productivity, particularly when new team members struggle to decipher complex code.

#### c) Stuck with Complicated New Technologies

Full stack developers encounter the challenge of staying updated with emerging technologies, which promise increased efficiency and flexibility in production. However, mastering these technologies requires significant time and effort, from installation to testing. Moreover, the complexity of navigating new technologies, particularly in areas such as database storage, underscores the need for effective leadership in managing software teams.

#### d) Too much or Less Abstraction

Balancing abstraction levels in code presents a challenge for full stack developers. Overly abstracted codebases can lead to performance issues and development slowdowns, while excessively verbose code may become difficult to comprehend and maintain. Finding the right balance is crucial to ensuring code clarity, efficiency, and scalability. e)

#### Overlooking the little things

Full stack developers must remain vigilant to avoid overlooking minor details that can have significant repercussions on project outcomes. Failure to properly structure code sections or provide logical coherence can lead to confusion and inefficiencies during development. Neglecting these small yet critical aspects of coding can result in major problems down the line, highlighting the importance of thorough code review and attention to detail.

### V. CONCLUSION AND FUTURE SCOPE

In the contemporary digital landscape, the demand for full stack developers continues to soar, driven by the proliferation of websites and e-commerce platforms. Companies prefer hiring full stack developers for their multifaceted skill set, allowing them to seamlessly navigate both front-end and back-end environments concurrently. This efficiency translates into significant cost savings for companies, as fewer professionals can accomplish a broader scope of work compared to hiring multiple developers for different aspects of website or application development. The remuneration for full stack web developers is on an upward trajectory globally, reflecting the escalating demand for their unique skill sets. Companies are willing to offer competitive salaries to talented full stack developers who possess diverse stack development abilities.

The future prospects for full stack web developers appear exceedingly promising, with demand projected to continue rising at a rapid pace.

**Smaller Teams-** The presence of full stack developers enables companies to maintain leaner, more agile teams, which offer numerous advantages over larger teams. Smaller teams require less communication and resources, fostering efficiency and flexibility in project execution.

**Adaptability-** Web development is inherently dynamic, with frequent changes in project

requirements, technologies, and team compositions. Full stack developers exhibit remarkable adaptability, swiftly adjusting to evolving circumstances and ensuring seamless project continuity in the face of challenges.

**New Technologies-** The web development landscape constantly evolves, with tools, software, and programming languages receiving regular updates to enhance performance and functionality. Full stack developers, equipped with a comprehensive understanding of various technologies, can effortlessly stay abreast of these advancements. Their innate familiarity with different technologies enables them to integrate new updates seamlessly, ensuring continued relevance and demand in the ever-evolving tech industry.

In essence, the versatility, adaptability, and proficiency of full stack web developers position them as indispensable assets in the modern development ecosystem. Their ability to navigate diverse environments and stay abreast of emerging technologies ensures their enduring relevance and prominence in the future of web development.

### REFERENCES

- [1] Valerio De Sanctis, Full Stack Web Development
- [2] Third Edition, 2020, Publisher: Packt Publishing.
- [3] Chris Northwood, Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Developer, Publisher: Independently published, 2020.
- [4] Frank Zammeti, Modern Full Stack Development: Using Typescript, React, Node.js, Webpack, and Docker, Publisher: Independently published, 2019.
- Juha Hinkula, Hands-on- Full Stack Development with Spring Boot 2 and React, Second Edition, Publisher: Packt Publishing, 2021.
- [5] Hassan Djirdeh, Nate Murray, Ari Lerner, Full Stack Vue: The Complete Guide to Vue.js, Publisher: Independently published, 2017.

- [6] 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", Volume 12.
- [7] 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, Pages: 4341-4353.