

Review Paper on Web Frameworks, Databases and Web Stacks

Jyoti Shetty¹, Deepika Dash², Akshaya Kumar Joish³, Guruprasad C⁴

¹⁻²Professor, ³⁻⁴UG Student

¹⁻⁴Department of Computer Science and Engineering

¹⁻⁴R V College of Engineering, Bangalore India

Abstract - In today's world, online applications have become an important component of the industry. Businesses can now grow and become easier by using the web apps, and achieve their goals much more quickly. The development of a web application involves identifying product requirements, designing, coding, and testing using frameworks and technologies. Frameworks are libraries and classes that give a rich collection of functionalities. Applications are built on top of frameworks so that the fundamental requirements in developing a web application are already setup. There are various frameworks available for front-end, and back-end development supporting different programming languages. Database is where the web application stores its data for use by back-end frameworks. The most important part of a web development lifecycle is choosing the right combination of front-end framework, back-end framework, and database environment. Web stack is a combination of frameworks, database environment, and an optional operating system environment. This paper discusses a few of the important front-end frameworks, back-end frameworks, databases, and web stacks.

Key Words: Microservices, Framework, Database, Web Stack.

1. INTRODUCTION

In this world of internet, where there are numerous technologies that are available for web application development, choosing the right architecture for application development is a very difficult task. One such architecture is the Microservices based architecture. In this approach, the various collections of services are developed as loosely coupled, highly maintainable, and independently deployable services. From a web application perspective, the application can be broadly divided into two independent services viz. Frontend service and Backend service. These two can interact with each other through API calls. The frontend of the application handles the user interaction while the backend of the application handles the business logic. Having a separate service handling the frontend and backend independently gives a lot of advantages like fast development, easy up-gradation, and high maintainability. But this also comes with its problems which is the risk of running two separate instances for one application. This hampers the continuous integration/continuous deployment pipeline. This paper discusses the various frameworks,

languages, and solution stacks that are available for developing a microservices-based web application.

The rest of the paper is organized as follows: Section 2 focuses on the frontend frameworks and languages, Section 3 is about backend frameworks and languages, Section 4 is about databases followed by Section 5 which describes various web stacks finally Section 6 concludes this review paper.

2. FRONT-END FRAMEWORKS AND LANGUAGES

The component of an app or website that users interact with is called Front End or Client Side of the application. It includes everything that users come across directly including images, buttons, text colours, graphs, tables, etc. HTML and CSS are used for the design and styling while JavaScript is used for validations. The behaviour, design, content, and structure of everything visible on the screen when mobile/web applications are opened, is developed by Front End application developers. Performance and responsiveness are the two main objectives of the front end development. The developers must make sure that the website is responsive i.e. it displays correctly on devices of all sizes. No part of the app/website should behave irregularly irrespective of the screen size.

2.1 AngularJS

AngularJS is a JavaScript framework for developing dynamic single page web applications. It is developed and maintained by Google. It uses HTML as a template language and extends HTML's syntax to express the application's components clearly and concisely. AngularJS's dependency injection and data binding eliminate much of the code which might otherwise be needed to be written and it all happens within the browser, making it apt with any server technology.

2.2 ReactJS

ReactJS is a JavaScript library for developing dynamic User Interfaces. It's developed and maintained by Facebook. React is an efficient, declarative and versatile JavaScript library for building user interfaces. Complex UIs can be composed from small, isolated and reusable pieces of code called "components". ReactJS uses JSX to simplify writing HTML. JSX is a pre-processor that adds XML syntax to JavaScript.

2.3 VueJS

Vue is a progressive framework for building user interfaces and single-page web applications. Vue is designed to be an incrementally adaptable architecture that focuses on declarative rendering and component composition. VueJS is created and maintained by Evan You. Vue extends HTML with HTML attributes called "directives".

Table -1: Comparison of AngularJS and ReactJS

Comparison Parameter	AngularJS	ReactJS
Baseline	Highly performing JavaScript framework	Excellent JavaScript library
Why Choose	If using TypeScript coding	If 'Everything-is-JavaScript' approach
Founders	Maintained by Google	Maintained by Facebook
Learning Curve	Steep learning curve	Fairly simply compared to Angular
Model	Based on Model View Controller	Based on Virtual DOM
Written in	TypeScript	JavaScript
Community Support	Very good community support	Very good community support
Popularity	Used by ~20% developers	Used by ~75% developers

3. BACK-END FRAMEWORKS AND LANGUAGES

Web creation at the Backend consists of countless activities. For example, protecting APIs (Application Programming Interfaces) against external attacks, authenticating users, enabling seamless interaction with databases, and handling user requests to collect and present the required information, etc. The backend frameworks enable all of these activities to developers simple and trouble-free.

These frameworks are evaluated by their programming methods, languages they support, and interfaces. Additionally, frameworks that provide built-in tools and templates that help developers in tracking various web development tasks quickly. Directly put, an advanced backend architecture increases the pace of development, thereby saving development time. These frameworks should allow developers to create interoperable applications that can scale up the workload. A few of the important frameworks are discussed in the following section.

3.1 NodeJS

Node.js is an asynchronous event-driven JavaScript runtime designed to build scalable network applications. It supports the handling of many connections concurrently. Call back is triggered for each connection and it sleeps if there is no work to be done. This behavior unlike too many other

common concurrency models, in which operating system threads are used. Moreover, the users of Node.js need not worry about deadlocks, since there are no locks. Since almost no function performs an I/O in Node.js, the process never blocks. These properties make Node.js scalable. JavaScript language is used to write applications in Node.js and run with Node.js runtime environment. The package ecosystem of Node.js called npm (node package manager) has a large number of open-source libraries. Node.js uses Google Chrome's super-fast highly optimized V8 execution engine in JIT (Just in Time) compilation fashion to execute JS code by transforming them into machine language and optimizes through complicated methods such as code inlining, copy emission, etc.

3.2 Django

Django is a web framework based on python programming language. It is ridiculously fast and was designed to help developers transform concept to application as quickly as possible. It takes care of much of the struggle of web application development thereby allowing developers to focus on writing actual business logic of the app without needing to reinvent the wheel. It is incredibly versatile. Django takes care of many necessary tasks of web development such as user authentication, site maps, content administration, RSS (Really Simple Syndication) feeds, etc. Famous web applications such as Instagram, Udemy, Mozilla and Pinterest use Django. Django follows a Model-View-Controller (MVC) architecture. The Model is the logical data structure that acts as a mediator between the website interface and the database. The View component contains the user interface logic. The controller is the main control component and acts as the middleman that binds the view and model together, meaning it enables the data passing from model to view.

3.3 Spring Boot

Spring boot sticks well with microservices but not encouraged for large monolithic applications. The artifacts can be deployed directly into Docker containers. Spring boot eases the development of web applications. In standard code and setup related to web advancement, a cutting-edge web development model that smooths out the improvement of server-side UI applications, REST APIs, and bidirectional, event-based frameworks is provided. Spring's production-ready features such as health status monitoring, metrics data, and tracing helps developers to monitor and get insight into the application. It supports many industry-standard authentication protocols like SAML (Security Assertion Markup Language), OAuth (Open Authorization), and LDAP (Lightweight Directory Access Protocol). It supports various relational and non-relational databases, cloud-based data services, and map-reduce frameworks such as Spark, Flume, etc. Spring supports Java, Kotlin, and Groovy. The development time is considerably more compared to other frameworks because of the programming language and configurations. But this has a use case where there is

concern about enterprise acceptability, standard, integration, and security.

Table -2: Comparison of Node.js and Django

Comparison Parameter	Node.js	Django
Costs	Open Source	Open Source
Scalability	More scalable comparatively	Less scalable
Architecture	Event-driven programming	Model-Template-View
Security	Developers need to need to explicitly make sure the security provided	Provides great security and developers need not worry about basic security concerns
Community	It has a very active community that helps programmers to improve themselves and in improving the framework	It has few communities for support since it is not as famous as Node.js.
Leading	Many countries are using Node.js and it is ahead comparatively	It is new and behind Node.js in usage
Performance	Its performance is better	Its performance is good
Complexity	It is less complex	It is more complex than Node.js

4. DATABASES

When it comes to choosing a database, the main choices are to choose between relational (SQL) and non-relational (NoSQL) databases. Although both databases are viable alternatives, some key differences still remain between the two that users need to bear in mind when making a decision. Each database is built for a specific type of use case and it is very important to choose the right database based on application needs. The relational databases are suitable for small scale data and to store OLTP data. Non-relational databases are preferred when the data is unstructured and the developer cannot stick to a fixed schema.

4.1 PostgreSQL

PostgreSQL is ideal for OLTP (Online Transaction Processing) type of workloads because of its compliance with ACID properties. It is a good choice for its OLTP capabilities. PostgreSQL can act as a highly capable analytical database. Mathematical software such as MATLAB can be easily integrated with PostgreSQL. PostgreSQL is highly scalable and it meets the requirement of modern websites

that require thousands or even hundreds of thousands of requests per second to serve customers. PostgreSQL integrates well with all modern web frameworks such as Django, Hibernate, Node.js, PHP etc. It has excellent replication capabilities which helps in easily scaling out to as many database servers as required. PostgreSQL does not limit itself to just a relational database. It can serve as a NoSQL style data store. In applications that need to leverage both relational and document-oriented databases PostgreSQL is a good choice as the developer can have both in a single product.

4.2 MongoDB

MongoDB is a NoSQL database. It is document-oriented, rather than using tables and rows as within the traditional SQL databases, MongoDB uses collections and documents. Documents contain key-value pairs which are the fundamental unit of data in MongoDB, it is equivalent to records in relational databases. Documents of similar type are grouped to form Collections. MongoDB's data model allows easy representation and storage of hierarchical relationships in the form of arrays, and other complex data structures. MongoDB is very scalable with replication capabilities. BSON is the format used to represent JSON documents in MongoDB. BSON provides additional data types such as Date and binary which are not supported in JSON and also provide ordered fields in order for it to be efficient for encoding and decoding within different languages. MongoDB finds it use case in big data and real-time applications running at multiple different locations. While it resolves network partitions by maintaining consistency, while compromising on availability.

Table -3: Comparison of PostgreSQL and MongoDB

Comparison Parameter	PostgreSQL	MongoDB
Architecture	Monolithic	Distributed
Scaling	No built-in mechanism to horizontally scale by partitioning	Sharding is used horizontally scale-out database. Commodity hardware is used for this purpose
Availability	Does not have integrated mechanisms for handling automatic failover and recovery between replicas	Maintains enough number of replicas to achieve availability and has automatic failover handling mechanisms
JSON Data handling	Drivers lack the ability to accurately convert JSON to useful datatype used in application	BSON and its drivers support more complex and advanced data types which are not supported by JSON

Data Governance	Does not have any in-built mechanism for JSON schema validation	Supports schema validation through the IETF standard, thereby enforcing schema structure on database collections
Schema	Strongly typed	Flexible
Transaction	ACID	Multi-document ACID Transactions with snapshot isolation
Consistency	Achieves Immediate Consistency	Supports both Immediate Consistency and Eventual Consistency which can be individually decided for each write operation

5. WEB STACK

A web stack also known as web application stack is a collection of software applications needed for web development, specifically for developing web applications and implementing websites. A web stack is a type of solution stack, which is a collection of software for performing specific tasks. Web stacks are critical components for web applications as well as websites. At a minimum, a web stack should contain an operating system, database software, programming language, and the webserver.

5.1 LAMP/LEMP

Linux, Apache/Nginx, MySQL, PHP

This is a popular web stack which involves an operating system, Linux, a web server, either Apache or Nginx, a relational database management system, MySQL and a scripting language well suited for web development, PHP. The scripting language can also be Perl or Python but, PHP is the popular choice. This stack is well suited for developing dynamic web sites and web applications. This is a layered structure and each layer builds on the previous layer. Linux is the base layer with the operating system. Apache/Nginx is the web demon that sits on the OS. The DB layer is used to store the data and the scripting language is used to drive and display the data and enable user interfaces.

5.2 MEAN/MERN/MEVN

Mongo, ExpressJS, AngularJS/ReactJS/VueJS, NodeJS

This is a highly popular JavaScript based software stack used to build dynamic websites and web applications. The major advantage of this stack is that both the client side and server-side environments can be developed in one language.

MongoDB is a NoSQL database and is used for data storage. ExpressJS is a JavaScript library used as a reverse proxy for an application server like Apache or Nginx. Angular, React and Vue are popular JavaScript frameworks for developing the client-side application framework. NodeJS is a JavaScript runtime environment that can execute JavaScript outside a web browser. This is where the server-side scripts will execute.

5.3 SPRING STACK

Using Java Spring as the base calls for the Spring Stack. This stack is used to build dynamic websites and web applications. Spring Boot provides various starter packs to start development very quickly without doing any configurations for application servers. It is very easy to build REST services using it. Database connectivity can be greatly simplified by using JPA and Hibernate. The frontend framework can be chosen from a wide variety of choices like ReactJS, AngularJS, JSP etc. Spring Boot also provides an embedded application server when required.

6. CONCLUSION

Every web framework, database and web stack has its own set of advantages and disadvantages. The choosing of web stack should be done based on few critical factors like time needed to take the product to market, long term scalability and maintenance, ease of programming, developer's proficiency with the technologies etc. Database has to be chosen based on the type of data the application is dealing with such as structured, unstructured or semi-structured and the consistency, availability and partition tolerance required by the application. All these factors are important to be considered while building modern web applications.

REFERENCES

- [1] Angular.io. 2020. *Angular*. Accessed on 29 February 2020 [online] Available at: <<https://angular.io/docs>>
- [2] Reactjs.org. 2020. *React* Accessed on 06 March 2020 [online] Available at: <<https://reactjs.org/tutorial/tutorial.html>>
- [3] P. Di Francesco, "Architecting Microservices," 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, 2017, pp. 224-229.
- [4] Jovanovic, Zeljko & Jagodic, Dijana & Vujcic, Dejan & Randić, Siniša, "Java Spring Boot Rest WEB Service Integration with Java Artificial Intelligence Weka Framework" 2017 INTERNATIONAL SCIENTIFIC CONFERENCE, GABROVO
- [5] Dunka, Bakwa & Emmanuel, Edim & Oyerinde, Yinka. (2018). Simplifying Web Application Development Using-Mean Stack Technologies. 04.

[6] Mongoddb.com. 2020. *MongoDB*. Accessed on 07 March 2020 [online] Available at: <<https://www.mongodb.com/compare/mongodb-postgresql>>

[7] Nodejs.org. 2020. *Nodejs*. Accessed on 28 April 2020 [online] Available at: <<https://nodejs.org/>>

[8] Docs.djangoproject.com. 2020. *Django*. Accessed on 26 April 2020 [online] Available at: <<https://docs.djangoproject.com/en/3.0/>>

[9] www.techopedia.com. *Web Stack*. Accessed on 01April 2020 [online] Available at: <www.techopedia.com>