Simplifying Web Application Development Using MEAN Technology Stack

MUHAMMAD SOHAIL
Shaheed Zulfikar Ali Bhutto Institute of Science & Technology

Abstract- ***The purpose of this research is to understand MEAN stack technology to develop dynamic, robust, scalable, high performance RESTful web application. The term MEAN is an acronym for MongoDB, Express, Angular, and Node — the four key technologies that make up the layers of the technology stack. We shall establish the worth of MEAN stack technology by developing an architecture that will be used as core component to develop a web application. We shall collect the performance metrics by comparing Create, Read, Update and Delete operation both on MEAN and Full stack technologies.***

Indexed Terms- MEAN, JSON, CSS, Web, PHP, HTTP, TCP/IP, SOAP, REST, DOM, CRUD, MVC, MVVM, NPM, CLI, ORM, IDB, AngularJS

I. INTRODUCTION

A. Study Overview

It is crucial to choose the right technology for web development that leads to fast application development, good coding patterns, code reusability, better performance, scalable and acceptable response time. [1] The key reason of this study is to demonstrate the advantages of JavaScript based web application development for developing rapid, scalable, cost effective and high-performance web applications.

Developing web applications consists of two main parts i.e. front end and back-end. The front-end part is also known as client end and backed part is also known as server-side part of the application. Regardless, they are two main parts of the web application. The client part of the web allows its users to interact with applications and perform business operations. The server part of the web deals with handling, defining and maintaining business logic of the application and most importantly interfacing with underlining database to store / retrieve business objects. Front end deals with web technologies like HTML, CSS and JavaScripting. HTML creates the structure of the web page and that structure renders on the browser and user sees the output of that structure on browser. On the other hand, CSS deals with styling the look and feel to make a good impression of what user sees on the page. It enhances the web page visually. [2]

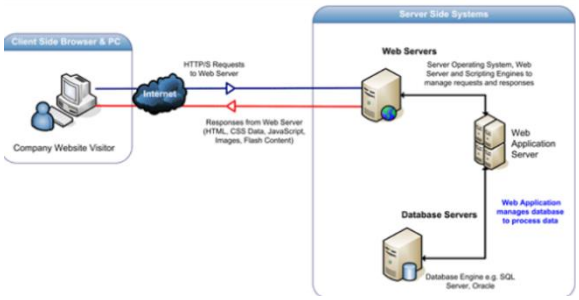
JavaScript provides DOM manipulation, helps to create pop-up messages, performs input validation, provides event driven features, and generates dynamic interface to enhance the user experience. JavaScript runs in the user browsers and is part of the front end. [3]

Backend development or server-side part of the web development consists of developing business logic and code to interact with relational or non-relational database to serve the client with data he requested. It is generally considered that the server-side part of the web application is more important and more critical to the success of the project and huge investment is made on this part by hiring good people and good tools and technologies. This part has been revolving since the day it came into existence and huge investment is being made around this area. Generally, more focus is put on this part to optimize the application performance, scalability, quality, maintainability and security. Primarily languages used on this part of C#, JAVA, PHP and many more.

Web application parts communicate with each other through http protocol on the internet and its responsibility of the protocol to ensure seamless and secure communication. Protocols are nothing but just a set of defined rules for communication and both communication parties adhere to those rules. Some examples of protocols or HTTP, HTTTS, FTP etc. Client prepares and sends http request to the server; server receives that request, interprets the request,

produces the desired results and sends it back to client. [2]

The following is an example of how client communication with server over the internet using protocols.



The above communication has three layers. The client layer has a browser to send and receive the request. The second layer has server-side code that receives and responds to the client request and third layer is nothing but just an operation database that allows to save and retrieve the business objects.

Web development is divided mainly into two key areas i.e. Frontend and Backend. Frontend is developed using HTML, jQuery, AngularJS, CSS and many more open-source libraries e.g. plotting charts, handling input validations and date specific functions. Backend is developed using proprietary software, application programming languages like C#, Java, and PHP and database specifics ORMs. [4] [5]

JavaScript is being used for the client-side programming to build up client part of the application. JavaScript runs in the client browser and jQuery and Angular famous MVVM frameworks were written on top of JavaScripting.

Full stack uses many different technology stacks for the backend part of the web. For example, C#, JAVA, and PHP. Full stack supports different relational databases and mainly uses structured query language to fetch / store data in the databases. Full stack uses a similar set of languages to develop front end part of the application like what is being used in MEAN stack.

Full stack is the process of developing both the frontend and backend of the application. The term stack means the layered structure of data flow through the web.

On the other hand, the MEAN stack consists primarily of the same language used for both front and backend. [6]

The key advantage of MEAN stack technology is its uniformity because there is a single language used throughout the entire stack. Each component of the MEAN stack uses JavaScript or is based on JavaScript. MEAN stack is assumed to reduce the development time, thereby subsequently reducing the overall development cost. MEAN stack does not need people of diverse skill set due its uniformity in data structure and programming languages. [4] [5] [6]

MEAN stack consists of the following components.

MongoDB

MongoDB's internal data storage structure simplifies the integration with other platforms. As an example, YouTube Api response can be stored as JSON in the MongoDB by mapping domain models to database objects.

MongoDB provides features for location-based data i.e. Spatial. This is an important feature for developing location-based applications like location-based event tracking.

MongoDB advanced features like enriched writing queries takes most advantage of indexing to perform nested queries, query on objects and arrays. MongoDB has another important built-in mechanism for writing aggregation queries like MAX, MIN, AVG and GROUP BY that gives good performance boost.

Express.js

Express.js is a wrapper to encapsulate the Node.js functionality that provides the minimal needed features which would otherwise be really challenging and time consuming to directly write Node.js commands. Express.js is used in the backend part of MEAN stack along with Mongoose, MongoDB.

Express.js offers useful features without reinventing the wheel that helps to define application routing with a lot less effort compared to directly working in

Node.js. You write a lot less, managed and better structured code using Express.js.

AngularJS / Angular

Angular is an open-source single page web application framework based on JavaScript. Angular was released originally as AngularJS which later in 2015 was completely redesigned to provide better injections, structure and code prototyping. Angular is useful for both web and mobile application development. Angular offers developers the chance to perform their job in a better and robust way by providing MVVM and MVC architecture. It provides lots of pre-developed and tested components from off-the-shelf that can be used in rapid application development. Since Angular can be embedded in HTML and vice versa so it results in powerful expressive and quick to develop language.

Angular fits in front-end part of the MEAN stack and is mainly used for sending / receiving http calls, interpreting data, showing / hiding data, creating dynamic forms, forms validation and lot more. [7]

Node.js

Node.js is a single-threaded, open-source, cross-platform runtime environment for building fast and scalable server-side applications. It runs on the V8 JavaScript runtime engine, and it uses event-driven, non-blocking I/O architecture, which makes it efficient and suitable for real-time applications. [8]

Node.js signifies JavaScript as a unique web application development as a single programming language for both server and client-side scripts.

	Full Stack	**MEAN stack**
Frontend	JavaScripting, HTML, CSS, AngularJs, jQuery	JavaScripting, HTML, CSS, AngularJs, jQuery
Backend	C#, Java, PHP etc.	JavaScripting, Express.js
Database	Relational (Microsoft SQL Server, MYSQL etc.)	NoSQL (mongoDB)
Protocol	HTTPs	HTTP
Response	JSON	JSON

B. Statement of the problem

The increasing demand for performance, scalability and cost in web technologies has motivated the development of new application development models, architectures, and technology stacks. While the rise of web technologies has helped to ease the application development process, it has also caused misunderstanding among technology specialists in deciding the most suitable technology to achieve their objectives. [2]

The increasing cost of proprietary software's has also motivated software developers to shift their interest to learn, understand and to work with these open-source technologies.

The advancement in open-source technologies has helped developers to shift from low level work to high value work. While your developers could spend their time reinventing wheels that the open-source community has already perfected, it's far preferable to use the world's best wheel, especially when that wheel comes at no cost to you.

C. Objective of the study

To help the software web developers to understand the key concept of MEAN Stack development, its components, and to understand the performance and cost benefits over Full stack development, and to help them out in choosing the right technology stack.

The objective is to highlight the difference in architecture, components, development environment, cost, performance and skills availability between MEAN stack and Full stack development technologies.

This research study will examine in detail the four components of the MEAN stack, how they relate to each other, role of each of component and how well they go together, their benefits as complete stack in web development.

To collect performance metrics, two similar applications shall be developed using MEAN and Full sack to analyze their Create, Read, Update and Delete operations performance.

- Study the MEAN stack and its different components.
- Gather performance and cost metrices.
- Study detailed comparison between MEAN stack and Full stack framework in terms of performance, cost and skills.
- Analyze the strengths of MEAN stack and recommend the right scenario when it should be used?

- Understand how Mean tack technology and how it has evolved over time.

Develop closed-ended questionnaire to capture response from people working with two different technology stacks. The purpose of this study research paper is to find out their knowledge of each stack technology, usability, performance limitations, cost comparison, and readiness to switch to alternative stacks, including the challenges they assume in doing so.

D. Hypothesis

MEAN Stack technology offers cost effective, robust, reliable, and rapid application development over Full stack and it is easy to learn.

E. Scope and limitations of the study

The key purpose of this study is to clearly understand the MEAN Stack technology, develop an architecture, develop a sample web application, and collect performance metrics by comparing the CRUD operations both from MEAN Stack and Full Stack based architectures.

F. Significance of the study

This study will help industry professionals, technology experts and decision makers to understand the MEAN Stack, its important components, its usage in developing scalable applications, and its impact on cost and performance.

- A clear understanding of MEAN stack, and its components.
- A clear comparison between MEAN stack and Full stack for cost and performance factors.
- A robust MEAN stack architecture to simplify the web application development.
- Collection of performance and cost metrics to exhibit gain / shortcomings of one stake over another stack.

G. Definition of Terms

Performance

Performance is an important software quality attribute, and it is measured by looking at processing speed, reaction time, resource utilization, throughput, and productivity.

Maintainability

Maintainability is a software quality attribute that deals with how easy it up fix the software issues or how easy it is to upgrade the existing functionality.

Single Page Application

It is a web application that allows users interact by dynamically rewriting page specific contents rather than reloading or rewriting the whole page by retrieving it from server. The purpose of this type of application is to boost up good user experience.

Input Validation

It is the process of validating user data completeness before posting it to the server.

Life Cycle

It means sequence of process steps to accomplish some tasks.

Framework

It is a basic structure for a system or problem. It provides the solution to a given issue or problem.

ECMA Script Standards

ECMA scripts is a set of standards defined by ECMA international defined to write JavaScripting. ECMA scripts standards were mainly developed for JavaScripting.

Relational Database Management System

It is a database management system based on relational schema / model of the underline data.

Streaming

Streaming is the process of delivering the contents to the users of the desktop, computer, and mobile devices through the internet.

Isomorphic

A single language used across the entire stack.

Node Package Manager (NPM)

It is simply a command line program that is responsible for installing and uninstalling Node.js specific packages in the application.

YCSB

The Yahoo! Cloud Serving Benchmark is an open-source specification and program suite for evaluating retrieval and maintenance capabilities of computer programs. It is often used to compare relative performance of NoSQL database management systems.

H. Breif Review of related Literature

MEAN stack technology came to market quite early as an open source that led to the idea of developing single page application and to use NoSQL databases as compared to relational database. All the components of MEAN stack are managed by JavaScript which was initially a client-side web programming language. MEAN stack uses JavaScripting in its framework that makes it extremely easy to understand the entire life cycle and working of each component of MEAN stack.

Node.js is nothing new but just an event driven run time based on JavaScripting that was developed to support scalable, secure, and maintainable applications. For example, live streaming applications and real time applications that are sensitive to internal or external events are the main candidates for Node.js.

Node.js follows a modal that supports or encapsulates the idea of an asynchronous event-driven, non-blocking I/O model. This makes it a lightweight, scalable and high-performance application. Node.js uses NPM for its package management. NPM is responsible for managing all the project dependencies. [2]

There are certain parameters which are important when deciding to choose the correct technology. The technology or framework should offer rapid development, prototyping, code reusability, scalability, maintainability, easy to upgrade, robustness and secure exchange of information over the communication channels.

While the above-mentioned attributes are important, the ease of use of and easy to train new resources, and easy to understand are also equally important parameters. Thus, the idea of isomorphic programming language initiated and evolved over time to overcome and fulfill above mentioned parameters.

The current available technology stacks based on proprietary and multiple tools, technologies and programming languages to be worked with to develop web application. The main tools being used in the modern days are but not limited to: Windows, IIS, ASP.NET, Microsoft SQL, C#, JAVA, and PHP and many more.

JavaScript helped to introduce the idea of having the same language used across whole stack. [6]

The application development efforts are reduced greatly with the invention and use of MEAN stack. MEAN stack has resulted in increasing the productivity of developers thus helping an organization to achieve its goals. MEAN helps to deliver scalable, maintainable, secure and efficient applications to end users.

Michael Kennedy shows MongoDB performs much better than SQL server on inserting 10,000 records. Mongo took 2.032 and SQL server took 204.215 seconds. Similarly, reading 50,000 records from MongoDB took 10.4 seconds while SQL server took 28 seconds. [9]

However, for delete operation MySQL performance proved to better compared to MongoDB by a student from University of Edinburgh. The same student further found that MongoDB performs approximately 40% better on long running queries than MySQL in other operations. As per Yahoo Cloud Service Benchmarking (YCSB) benchmark SQL server still performs better over MongoDB, but they also noticed that MongoDB's performance has been improving over time. However, it is believed that NoSQL database compromises the ACID properties of the data to perform better than relational databases.

Expedia delivered MongoDB based application within five months of time to plan travel (with just two months of time for developing the application). During the development time, they had re-write the database schema almost to different way and still it was quick to do and manageable. Similarly, another organization upgraded their existing legacy systems to use MongoDB in just a few days that would otherwise have taken more than few days. [10]

It is assumed that not all the organizations (those could have been ideal candidate to migrate to NoSQL) have migrated to NoSQL. Google and Amazon have developed their own database management system specific to their needs. Facebook and Twitter use a mix of relational and NoSQL databases.

Today, many organizations are thinking of migrating to NoSQL database to have more flexible data structure and to develop the model faster. The benefits of non-relational database over relational database becomes more prominent as the size the web application users increase. The relational database has strict database structure and schema, so it is not easy and quick to change the underline schemas. [1]

The following are the main advantages when using JavaScripting across the whole stack.

- JavaScript is used in Node.js and Express.js
- Angular is writer on top of JavaScripting and use JavaScripting in it.
- MongoDB stores data in JSON format.

The client and server do communication in JSON format.
MEAN stack provides the capability for the database to intercept JavaScript for data exchange in addition to providing JavaScript based server platform.

Use of one language in entire Stack boosts up the overall productivity of developer. Conversely, in traditional development model, a client side (JavaScript) developer needs to be dependent on server-side developer (C# or JAVA as an example) to understand, interpret or debug the code which makes it tough to develop the application rapidly.

MEAN stack takes full advantage of the capabilities of Node.js features like non-blocking I/O, and event driven run time and server side JavaScripting. [11]

MongoDB provides great flexibility towards complex data structure and lets the application developers insert complex data in the form of a single field. MongoDB is most suitable for large scale databases or to the databases where databases schemas must change quite often and continuously. It offers great support and flexibility in dealing with dynamic schemas along with dynamic fields to target object / document or domain model. [1] [12]

MongoDB is not only best suited for non-structured databases, but it also provides better performance and speed to structured databases. There is one of few important week points about MongoDB is that it does not perform well on aggregate functions or on non-key data. [5]

Table 2: Time for writing (time in milliseconds)

Databases	Number of records					
	10	50	100	1000	10000	100000
MongoDB	61	75	84	387	2693	23354
RavenDB	570	898	1213	6939	71343	740450
CouchDB	90	374	616	6211	67216	932038
CassandraDB	117	160	212	1200	9801	88197
Hypetable	55	90	184	1035	10938	114872
Couchbase	60	76	63	142	936	8492
MS SQL Express	30	94	129	1790	15588	216479

Table 3: Time for fetching all keys (time in milliseconds)

Database	Number of records					
	10	50	100	1000	10000	100000
MongoDB	4	4	5	19	98	702
RavenDB	101	113	115	116	136	591
CouchDB	67	196	19	173	1063	9512
CassandraDB	47	50	55	76	237	709
Hypetable	3	3	3	5	25	159
MS SQL Express	4	4	4	4	11	76

Table 4: Databases version

Databases	Version
MongoDB	1.8.5
RavenDB	960
CouchDB	1.2.0
CassandraDB	1.1.2
Hypetable	0.9.6
MS SQL Express	10.50.1600.1

Table 5: Table for reading (time in milliseconds)

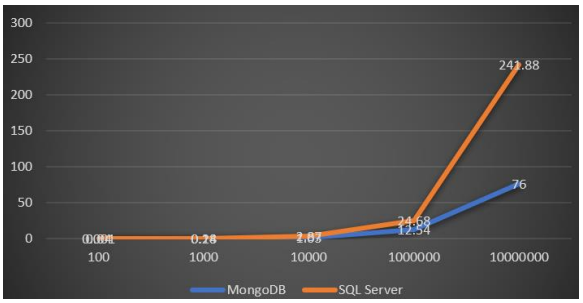
Database	Number of records					
	10	50	100	1000	10000	100000
MongoDB	8	14	23	138	1085	10201
RavenDB	140	351	539	4730	47459	426505
CouchDB	23	101	196	1819	19508	176098
CassandraDB	115	230	354	2385	19758	228096
Hypetable	60	83	103	420	3427	63036
Couchbase	15	22	23	86	811	7244
MS SQL Express	13	23	46	277	1968	17214

Table 6: Time for deleting (time in milliseconds)

Database	Number of records					
	10	50	100	1000	10000	100000
MongoDB	4	15	29	2	35	18688
RavenDB	90	499	809	8342	87562	799409
CouchDB	71	260	597	5945	67952	705684
CassandraDB	33	95	130	1061	9230	83694
Hypetable	19	63	110	1001	10324	130858
Couchbase	6	12	14	81	805	7634
MS SQL Express	11	32	57	360	3751	32741

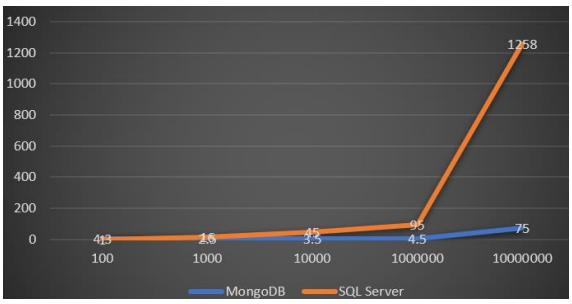
Insert Operations

The following picture shows the time took in milliseconds to insert operations for 100, 1,000, 10,000, 100,000 and 1,000,000 records both in SQL Server and MongoDB. MongoDB gives better performance.



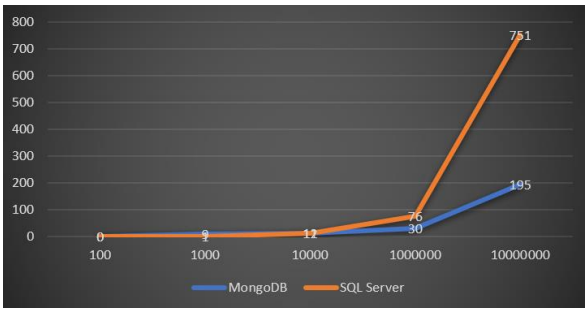
Delete Operations

The following picture shows the time took in milliseconds to delete operations for 100, 1,000, 10,000, 100,000 and 1,000,000 records both in SQL Server and MongoDB. MongoDB gives better performance.



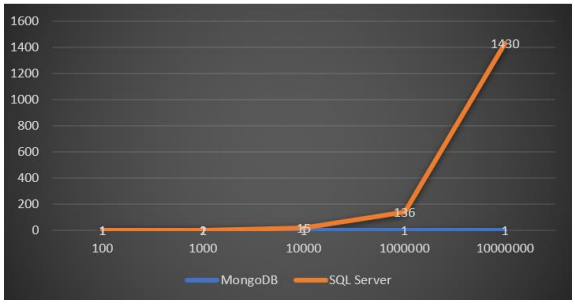
Update Operations

The following picture shows the time took in milliseconds to update records both in SQL Server and MongoDB. MongoDB gives better performance, and it improves even more when the number of records increases.



Select Operations

The following picture shows the time took in milliseconds to select operations for 100, 1,000, 10,000, 100,000 and 1,000,000 records both in SQL Server and MongoDB. MongoDB gives better performance.



Querying in different modes

The literature author went on further analyzing query performance without and with aggregate functions.

Queries without aggregate functions

The author tried different levels of complex queries e.g. multiple joins.

Q1. Name and country of brands that their Units in Stock for this type of product is more than 100 units.

Q2. Name and address of vendors who have customers from Iran.

Q3. Name, address, city and country of vendors who their sold product belongs to Russia.

Q4. Name and telephone number of shipping companies that have not had any instances of product "ABC" for transmission.

By comparing the results obtained from non-aggregate queries in Fig. 8, we can say almost there is no doubt that MongoDB has higher performance than SQL Server in queries with such structure. In the next test we will see how the results will be converse and SQL Server will win against MongoDB in aggregate

operations. Of course, the weakness of MongoDB in such queries has been shown also in.

Queries with aggregate functions

The trial against the previous experiments has obtained interesting results that indicate superiority of SQL Server in applying three aggregation functions. We performed three queries in this experiment using three aggregation functions separately, i.e. count, sum and avg. The goal of this experiment is evaluation of the two databases in terms of aggregation operations. Queries raised in this experiment are as follows:

Q1. The aggregate number of workers of the exchanging framework that have Iranian country.

Q2. The aggregate number of items in stock that they have been requested is more than 500 units.

Q3. Normal refunds are given to various requests for items, for every item independently.

Times that appeared in Figure. 5, demonstrate the consequence of 100 times running each inquiry, that we have demonstrated the normal of them.

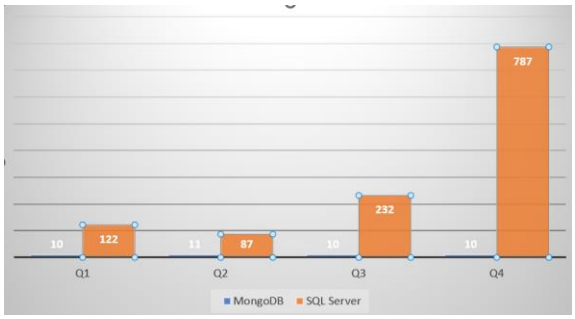


Figure 5: Processing time comparison between two databases (millisecond)

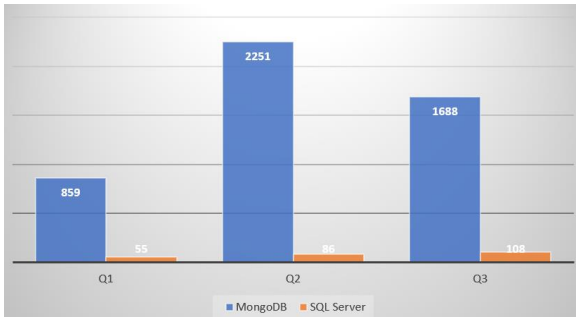


Figure 6: Pressing time comparison of aggregate queries (millisecond)

I. Synthesis of Reviewed Literature

MEAN provides an excellent way of developing web services because the entire stack uses JSON data format and single language i.e. JavaScript. MEAN stack-based web services are robust and highly scalable due to the great benefits of both Node and MongoDB. Because of developing scalable web services offers smooth software updates for multiple clients.

II. MATERALS AND METHDOS

A. Appropriate research techniques

Review Existing Literature

We took the following steps for literature review.

Refined Topic

The research topic was refined to Simplify the web application development using MEAN stack technology.

Design Research

We designed research study into three main parts.

- Literature Review (Scholarly Journals)
- Understood the MEAN technology stack.
- Developed MEAN stack architecture.
- Collect the performance metrics.
- Compare performance metrics.

Locate the scholarly journals.

We studied approximately 10 scholarly journals considering the 2 to 3 weeks' time factor.

Read and take notes on the report found.

We referred to the conclusion section of the scholarly journals to find important notes about the subject.

Organize and write the review.

We discussed and concluded the main findings with a clear picture in mind of how all the information from various research reports fit together and meets the objective.

Create a reference list.

We added references at the end of the paper.

We first reviewed the existing literature on MEAN stack and Full stack technologies, understood their components, understood the role of each component, understood performance weaknesses, find the strength

and weakness of each stack and then draw a detailed comparison between them in terms of different quality attributes, and how each technology evolved over the time.

Quantitative Analysis

This study is based on quantitative technique to collect performance metrics in numerical form those are further measured and analyzed in units of measurement (UOM) in milliseconds (MS). This data helped to construct graph and tabular data to show the performance metrics outcome.

B. Appropriate Research Instruments or Methodology

The research is based on developing two similar applications; one using MEAN stack and second using Full stack, to collect performance metrics to reach out the decision.

III. Result and Findings

A. Key findings of the study

As a part of this study, the following metrics measurement were collected to analyze the performance both in MEAN Stack and Full stack technologies.

- Time (millisecond) took to perform Insert operation.
- Time (millisecond) took to perform delete operation.
- Time (millisecond) took to perform to retrieve operation.
- Time (millisecond) took to perform update operation.
- Time (millisecond) took to show 100 records.
- Time (millisecond) took to show 200 records.
- Time (millisecond) took to show 300 records.
- Time (millisecond) took to show 500 records.
- Time (millisecond) took to show 1000 records.

Technology	Mean Stack	Full Stack
Create	1	258
Read	554	301
Update	2	232
Delete	3	216
Retrieve 100	538	4435
Retrieve 200	545	8054
Retrieve 300	552	11614
Retrieve 500	580	17461
Retrieve 1000	646	49490

Table 7: Performance metrics comparison (time in milliseconds)

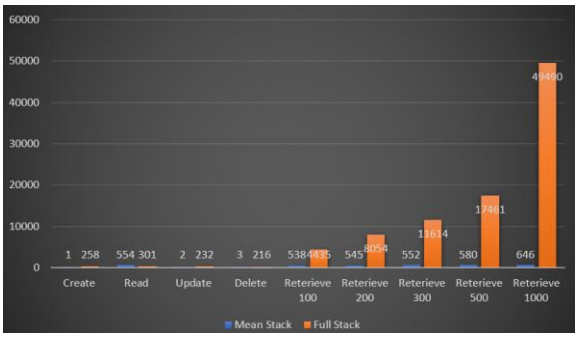


Figure 36: Performance metrics comparison (time in milliseconds)

B. Relevance to the objectives

The findings are useful for the industry professionals, software developers, architects and managers to decide which technology they should use for the application under development. This paper's findings will help all of those to compare both technology from performance, cost and ease of use tradeoffs.

C. Relevance to the Industry

Node.js is one the most critical parts of the MEAN technology stack. Some prominent companies had / have been using Node.js to produce the following quality products.

- LinkedIn found it is 20% faster and consumes less resources.
- Netflix improved app startup performance by 70%.
- PayPal re-engineered existing products to Node.js and
- found the new architecture uses less space and fewer lines of code.
- Uber improved the overall performance of the application.

Many startups small scale organization are putting more efforts towards Node.js to develop small, fast and robust applications.Node.js offers rich scalability and offers better performance when handling large http traffic and it is gradually become most powerful and widely used framework.

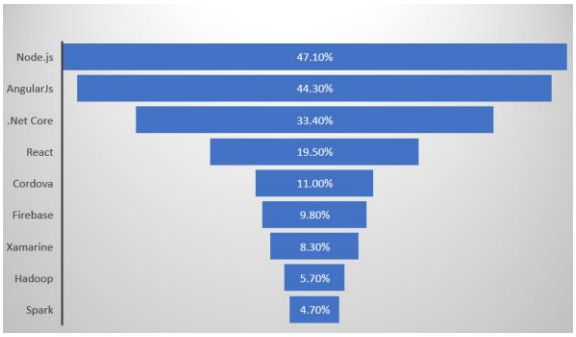


Figure 36: Language popularity comparison

As per graph showing above, the three-top web application development frameworks are JavaScript based that clearly shows how good JavaScript and everyone choice. Almost every company prefers JavaScripting in different ways. JavaScripting and Angular are two main pillars of MEAN stack. Application web developers who work on MEAN stack develop strong skills in JavaScripting because of the isomorphic programming language. A lot of research is being made to enhance JavaScript to meet the latest challenges and market needs. There are lots of open source based on JavaScript available to be used to fulfill specific project needs.

IV. SUMMARY OF FINDINGS, CONCLUSION, FUTURE WORK AND RECOMMENDATIONS

A. Summary of Findings

Fast switching between client and server

MEAN offers simple and fast application development because there is only JavaScript used in all layers i.e. from back to front end. Therefore, a developer who is expert at JavaScript can handle and manage the who project development compared to Full Stack where a developer must be master in multiple languages i.e. JavaScript, C# and many more. Another cool thing is Node.js that lets developers deploy applications quickly and directly on server without preparing a separate standalone server.

Isomorphic coding style

An isomorphic coding is the one where a single language can be run both on client and server side. The main advantage of isomorphic framework is to reduce the training and skillsets required to develop an application end to end because there's one less language to learn.

Mean is highly flexible.

It lets application testers run and test the application hosted on cloud platforms i.e. Amazon or Azure quickly and easily. MEAN stack applications are quick to build and develop on cloud. The design of MongoDB really suits cloud computing.

JSON

JSON is light weight and can be quickly loaded in JavaScript. Transferring data over HTTP in Json format has less messages size. JSON can easily distinguish the difference between 1 and 1. JSON are the best suited for JavaScript to consume. JSON is used everywhere in MEAN i.e. from Node.js to Angular. MongoDB saves data in JSON format. Therefore, there is no need to do extra work encode / decode messages sent over the http.

MEAN is less costly.

MEAN offers less cost to develop the application. You don’t need to hire multiple developers with different skills sets. You don’t need to pay for propriety tools & technology.

Node is fast

Node.js is a fast response runtime engine because of its non-blocking architecture. Angular is an open-source JavaScript framework that offers maintenance, testability, and reusability. Node.js offers fast execution of JavaScript. To scale to large number of connections, all I/O operations in Node.js are performed asynchronously.

MEAN stack is fully open source.

Each technology used in MEAN stack is open source and is available as an open source and is free to use. It assists the entire application development process using open-source libraries and public repositories

which immensely reduces the overall development time.

Cloud Compatible

Mean stack technologies help to deploy cloud functionalities within the app by reducing the disk space cost.

B. Conclusions

- It uses a single language (JavaScript) for both server side and client-side execution environments.
- You will find JSON everywhere, which is a good thing.
- It has immense support from industry leaders.
- Node.JS - Amazon Web Services and Azure
- Angular.JS - Google
- And the best of all it is open source.

C. Recommendations

It is recommended to use MEAN technology stack because it is quick to learn and quick to develop applications and can easily integrate with other technologies.

D. Future Work

The following points have been left for future work.

- Collect metrics based on complex queries.
- Bulk deletion and monitoring the performance of the
- underline technology.
- Collect the performance metrics after deploying the backend part on the remote server or cloud.
- This study does not explain the step-by-step process for developing MEAN stack web application.

REFERENCES

[1] M. R. M. M. N. G. Seyyed H.A., "Performance evaluation of SQL and MongoDB databases for big e-commerce data," *2015 International Symposium on Computer Science and Software Engineering (CSSE)*, 2016.

[2] M. K. a. B. V. M. Kaluža, "A COMPARISON OF BACK-END FRAMEWORKS FOR WEB APPLICATION DEVELOPMENT," *SRCE*.

[3] H. Kienle, " It's about Time to Take JavaScript (More) Seriously," *Software, IEEE*, vol. 27, 2010.

[4] D. E. A. E. D. O. O. Bakwa, "Simplifying Web Application Development Using-Mean stack Technologies," *International Journal of Latest Research in Engineering and Technology (IJLRET)*, vol. 4, no. 1, pp. 62-76, 2018.

[5] D. D. K. Ragati B., "A Review Paper on MERN stack for Web Development," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 6, no. 4, 2018.

[6] P. S. Ninaad N., "The MEAN stack," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 5, 2017.

[7] W. C. a. T. Senivongse, "A model-driven development of web applications using AngularJS framework," *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 2016.

[8] A. C. M. G. A. K. Varsha Sahni1, "Mern(Mongodb , Express-Js, React-Js, Node-Js) Stack Web-Based Themefied Education Platform For Placement Preparation," *Educational Administration: Theory and Practice*, 2024.

[9] D. N. a. F. Y. H. Ahmed, "MEAN stack to enhance the advancement of," *Journal of Physics: Conference*, 2019.

[10] MongoDB, "https://www.mongodb.com/solutions/customer-case-studies/expedia," MongoDB. [Online].

[11] S. G. M. D. A. I. S. L. Bangare, "S. L. BanUsing Node.Js to Build High Speed and Scalable Backend Database Server,," *International Journal of Research in Advent Technology* , 2016.

[12] W. A. J, A comparison of the performance and scalability of relational and document-based web-systems for large scale applications in a rehabilitation context, UNIVERSITY OF HERTFORDSHIRE.

[13] Y. S. B. K. Swathi, "Ease of MEAN stack in Web Development," *International Journal of Current Trends in Engineering & Research (IJCTER)*, vol. 2, pp. 217-219, 5 May 2016.

[14] S. J. J. S. J. C. Andrew Jhon Poulter, "Using the MEAN stack to implement a RESTful service for an Internet of Things application," *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2016.

[15] D. N. a. F. Y. H. A. 2. J. P. C. S. 1. 012075, "MEAN stack to enhance the advancement of parking application: A narrative review," *Journal of Physics: Conference,* p. 19, 2019.

[16] 9, "MEAN stack to enhance the advancement of parking application: A narrative review," *Journal of Physics: Conference.*