Exploring MERN Stack and Tech Stacks: A Comparative Analysis

Narne Renuka Chowdary[1], Venkata Gayathri Dammala[2], Manjunath Donthineni[3] , Shahid Khan Pathan[4]

[1,2]UG student Dept of CSE (Cyber security, IOT incl. Blockchain Technology), VVIT, Andhra Pradesh, India
[3]UG student Dept of Computer Science and Engineering, VIT, Andhra Pradesh, India
[4] UG student Dept of Computer Science and Engineering, QISCET, Andhra Pradesh, India

---***---

Abstract - *This paper explores the MERN stack, comprising MongoDB, Express.js, React.js, and Node.js, highlighting its superior applications and advantages over alternative web technologies. Examining the stack's architecture and components, we emphasize its versatility in real-time applications and its efficiency in fostering code reusability and modularity. The research advocates for MERN stack adoption, presenting a concise guide for developers and businesses to leverage its unified JavaScript-based framework for modern and efficient web development*

Key Words: **MongoDb, React JS, Node JS, Express JS, MEAN, Python, Java, Libraries, Framework**

1.INTRODUCTION

In today's era the MERN stack was one of the most popular revolutionized concept in web development. Now this research paper provides a comprehensive exploration of the MERN (MongoDB, Express.js, React, Node.js) stack in web development, focusing on its applications and the ease of integration with other technologies. In the contemporary landscape of web development, the MERN stack has emerged as a prominent choice for building robust, scalable, and efficient web applications.

1.1 What is MERN Stack?

MERN Stack is a widely used technology for building frontend and backend applications.

FRONTEND development is creating interactions for the users to use the application for example user interface in web browser.

BACKEND development is the work on the server side. Operations at the backend are not realized by the users. Backend technology primarily aims at databases, servers and API's.

MERN is a short form for MONGO DB, EXPRESS JS, REACT and NODE JS. Each technology in the MERN stack plays an important role in building web-based applications. React is for building frontend interfaces and rest of the three are for backend development. In the MERN stack, react serves as the Presentation Layer, responsible for creating a dynamic and interactive user interface on the web. Node.js, with Express.js, constitutes the Application Layer, managing HTTP requests, routing, and executing business logic on the server. The Business Logic Layer, embedded within the Application Layer, encompasses the rules and processes governing data processing. Consider Fig-1 where MongoDB, employed as the Data Access Layer, facilitates storage and retrieval of data, acting as the underlying database for the application. The advantage of building website with mern stack technology is that it can seamlessly integrate with other emerging technologies like Artificial Intelligence, Machine Learning, IoT (Internet of Things), Blockchain, Cloud Services, etc.
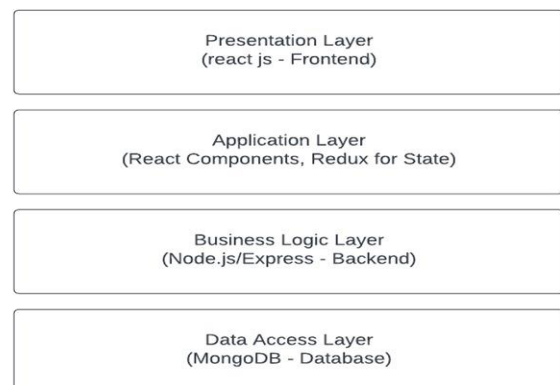

Fig-1: Layered Architecture

1.2 MONGO DB

Mongo DB is a NoSQL database that integrates not only with mern but also with other various technologies. It is a database meant for easier development, scaling of applications, and deployment. Many organizations are using MongoDB for building applications such as: Content Management Systems (CMS), E-commerce Platforms, Mobile App Development.

It's important to note that while MongoDB is well-suited for certain use cases, the choice of a database system depends on various factors, including the specific requirements of the application.

1.3 EXPRESS JS

Express.js is a flexible Node.js web application framework that provides a powerful set of features to develop web and

mobile applications. It is designed to make the process of building web applications and APIs (Application Programming Interfaces) in Node.js much simpler. With middleware and routing, Express JS makes it effortless to organize the functionality of your application.

It will add useful features for Node.js HTTP objects, making it easier to render dynamic HTTP. Express.js allows developers to handle various HTTP methods (e.g., GET, POST, PUT, DELETE) for different URL patterns.

1.4 REACT JS

For building user interfaces based on components, react.js is a free and open-source front-end JavaScript library. Meta and a community of individual developers and companies are responsible for its maintenance. Virtual DOM (Document Object Model) is a concept and technique used in React.js to improve the efficiency of updating the user interface. It is introduced by react on, a low-level copy of the actual DOM.

When a component's state changes, the react will first update the virtual DOM and then make efficient updates of only the parts that are needed in an actual DOM. With a large library and tools that complement its functions, react has an active ecosystem.

React can easily integrate with various backend technologies, allowing developers to build interactive and dynamic user interfaces that communicate efficiently with server-side APIs. Reactions are characterized by a bilateral data flow, which means that changes occur in the same direction from parent to child components. This allows us to understand the change of data through an application, thus making code propagation predictable and less error-prone.

1.5 NODE JS

Node.js, especially on the server side, has a critical role to play when developing web applications. JavaScript can be used for development on both the client and server-side by Node.js developers. The development is simplified and code reuse is encouraged by this alignment of programming languages throughout the stack.

Node.js is intended to be non-blocking and event-driven, which means it can handle a large number of concurrent connections without becoming slowed by waiting for I/O operations to complete. As a result, Node.js is well-suited for applications requiring high concurrency and real-time responses.

Node.js is frequently used to create RESTful APIs. Its lightweight and efficient design makes it ideal for processing HTTP requests and answers, making it easier to develop backend services for web and mobile apps.

2. DICUSSION ABOUT MERN STACK AND OTHER WEB DEVELOPMENT STACKS

There are various stack technologies such as MEAN Stack, MEVN Stack, Django Stack, Python Flask Stack, and Spring boot stack in the account for creating web-based applications. The efficiency of a technology stack depends on various factors.

The term "full-stack" refers to developers who have expertise in both frontend (client-side) and backend (server-side) development. A full-stack developer is capable of working on all aspects of a web application, from designing user interfaces to managing server-side logic and databases. The technologies used in a full-stack development stack can vary, but here are some common combinations.

2.1 DJANGO STACK

The Django stack revolves around the Django web framework, a high-level Python framework for server-side development. Django ORM (Object-Relational Mapping) facilitates seamless interaction with databases, translating between Python objects and relational database tables. HTML, CSS, and JavaScript are used for frontend development. Django follows the "batteries-included" (The term "batteries-included" is often used to describe frameworks that come with a rich set of built-in tools and features, allowing developers to be more productive and focus on building application-specific logic rather than dealing with the intricacies of setting up basic functionalities.) philosophy, providing a comprehensive set of features, including an admin interface and authentication system, making it a robust choice for rapid development.

2.2 RUBY ON RAILS

Ruby on Rails, often referred to as Rails, is the focal point of this stack. A web application framework written in Ruby, Rails follows the convention over configuration (CoC - A software development principle that suggests a default set of conventions and assumptions to streamline the development process.) and don't repeat yourself (DRY principle advocates for avoiding redundancy in code) principles, emphasizing simplicity and productivity. Active Record serves as the Object-Relational Mapping tool, facilitating interaction with databases. HTML, CSS, and JavaScript are employed for frontend development. The Rails stack is celebrated for its developer-friendly approach and convention-driven development.

2.3 LAMP STACK

The LAMP stack is a classic combination of Linux, Apache, MySQL, and PHP/Python/Perl. Linux serves as the operating system, Apache as the web server, MySQL as the relational database management system, and

PHP/Python/Perl as the server-side scripting language. HTML, CSS, and JavaScript are utilized for frontend development. LAMP has been a popular choice for building web applications, known for its open-source components and widespread support.

2.4 PYTHON FLASK STACK

The Python Flask stack centers around Flask, a lightweight and flexible Python web framework. SQLAlchemy, a SQL toolkit, is used for efficient database interaction. HTML, CSS, and JavaScript handle the frontend development. Flask is known for its simplicity and minimalism, providing developers with the freedom to choose components based on project requirements while maintaining a clean and modular codebase

2.5 SPRING BOOT STACK

Spring Boot, a Java-based framework, is the cornerstone of this stack. Spring Data JPA provides a data access framework for Java, simplifying database interaction. HTML, CSS, and JavaScript are employed for frontend development. Spring Boot is designed to streamline the development of Java applications, emphasizing convention over configuration and offering a wide range of features for building scalable and maintainable systems.

2.5 MERN VS MEAN VS MEVN STACK

Angular.js (Frontend Framework) is used for building dynamic user interfaces. It is a comprehensive framework maintained by Google, designed for creating powerful single-page applications (SPAs). Developers widely recognize Vue.js as a progressive JavaScript framework appreciated for its simplicity and flexibility, enabling them to incrementally adopt its features and making it suitable for both small and large-scale projects. React.js (JS Library) is used as a frontend library for building user interfaces. React is developed and maintained by Facebook, focusing on creating reusable UI components and providing a virtual DOM for efficient updates.

3. EFFICIENCES OF STACKS:

The efficiency of a technology stack depends on various factors, including the specific requirements of the project, the expertise of the development team, scalability needs, and performance considerations. Different technology stacks have their strengths and are suitable for different use cases.

MEAN/MERN/MEVN:
Strengths:
JavaScript throughout the stack for seamless development.
Single language (JavaScript) simplifies development for full-stack developers.

Strong support for building modern, dynamic, and scalable web applications.
Considerations:
May have a steeper learning curve for beginners.

DJANGO:
Strengths:
Rapid development with a "batteries-included" philosophy.
Clean and pragmatic design.
Excellent documentation.
Considerations:
This may be more opinionated, which can limit flexibility.

RUBY ON RAILS:
Strengths:
Convention over configuration for increased developer productivity.
Elegant and concise code.
Considerations:
May have a learning curve for developers new to Ruby.

LAMP:
Strengths:
Long-established and well-supported stack.
Wide adoption with a large community.
Considerations:
May be perceived as less modern compared to newer stacks.

PYTHON FLASK:
Strengths:
Lightweight and flexible.
Well-suited for small to medium-sized projects.
Considerations:
Less opinionated compared to larger frameworks.

SPRING BOOT:
Strengths:
A comprehensive ecosystem with a strong emphasis on enterprise-level applications.
Excellent support for building scalable and robust systems.
Considerations:
Java-based, which might have a steeper learning curve.

Considerations for Efficiency:
Project Size and Complexity:
Larger frameworks like Django or Spring Boot may be suitable for complex enterprise-level applications.
Smaller frameworks like Flask might be more efficient for smaller projects.
Developer Experience:
Consider the familiarity and expertise of your development team with a particular stack.
Scalability Requirements:
Some stacks are designed with scalability in mind and might be more efficient for high-traffic applications.

Community Support: The size and activity of the community can impact the availability of resources, libraries, and support.

Specific Technology Requirements: Consider whether specific technologies or features are necessary for your project.

Ultimately, the choice of a technology stack should align with the specific needs and goals of your project. There's no one-size-fits-all answer, and the efficiency of a stack depends on how well it meets the particular requirements of the application being developed.

4. APPLICATIONS

Currently, numerous websites employ the MERN stack in their applications. Some notable examples include Netflix, Uber Eats, Airbnb, Instagram, Walmart, and many others. Let's now focus on a specific application, such as an e-learning platform. This platform primarily serve students to enhance their knowledge and come in various forms, including those provided by universities, private organizations, and governments. The e-learning platform we're discussing is built using the MERN stack.

This platform contains comprehensive materials required for an engineering curriculum spanning four years which are stored in mongodb. In addition to these resources, it features blogs authored by both students and faculty members. Mongoose can be simplify the interactions with database.

These blogs can cover various topics, particularly emerging technologies, enabling users to stay updated. Moreover, the platform offers students quick access to their results whenever they are published or needed. These Results can be handled through redux.

Faculty members can access detailed student information, organized by department. Charting libraries like Chart.js or D3.js can be used to create interactive charts and graphs to represent student performance and results visually.

The website also provides login credentials for different user roles, including faculty, students, data entry officer (DEOs), and administrators. The management of these login credentials is meticulously handled using React.js libraries such as bycrypt, jwt tokens etc.
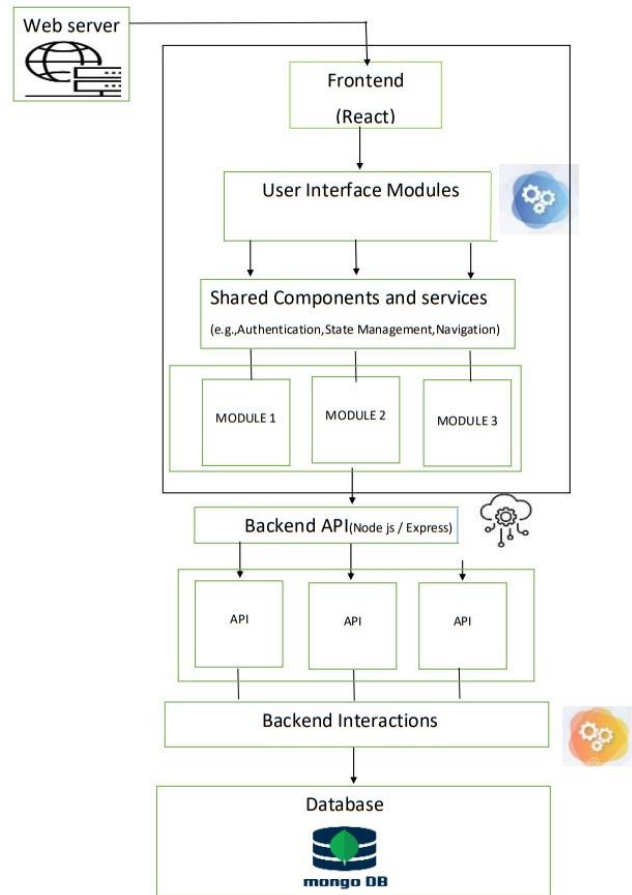

Fig-2: Architecture of MERN Stack

Table -1: Evolution of MERN

YEAR	MONGO DB	EXPRESS JS	REACT JS	MONGO DB
2010	Initial Release	Initial Release	____	Initial Release
2011	____	____	____	____
2012	____	____	____	____
2013	____	____	____	____
2014	____	____	____	____
2015	WiredTiger Engine	____	Initial Release	____
2016	____	____	Major Improvements	____
2017	____	____	Fiber Architecture	Initial LTS
2018	Transactions	____	Context API	LTS
2019	Atlas Data Lake	____	Hooks	LTS
2020	Realm	____	Improved JSX	LTS
2021	Native Time Series	____	Experimental	LTS
2022	New Query Language	Express 5	Continued Updates	Expected LTS

5. CONCLUSIONS

In the MERN stack, consisting of MongoDB, Express.js, React.js, and Node.js, stands out as a powerful and versatile framework for modern web development. This research delves into the stack's architecture, individual components, and its applications, emphasizing its superiority in real-time and scalable web applications.

Comparative analysis with other stacks such as MEAN, Django, Ruby on Rails, LAMP, Python Flask, and Spring Boot reveals that the choice of a technology stack depends on factors like project size, developer experience, scalability requirements, and specific technology needs. While each stack has its strengths, the MERN stack's adoption by industry giants like Netflix, Uber Eats, and Airbnb showcases its widespread acceptance and success in building robust applications.

The applications section illustrates a practical use case of the MERN stack in developing an e-learning platform, showcasing its capabilities in handling complex functionalities, data management, and user roles. In essence, the MERN stack emerges as an efficient and modern choice for developers and businesses seeking a unified JavaScript-based framework for streamlined and scalable web development.

REFERENCES

[1] Shetty J & Dash D, "Review Paper on Web Frameworks, Databases and Web Stacks" Science, e-ISSN: 2395-0056 Volume: 07 Issue: 04 Apr-2020 p-ISSN: 2395-0072

[2] Baiskar Y, "MERN: A Full-Stack Development" https://doi.org/10.22214/ijraset.2022.39982 ISSN: 2321-9653 ESTD: 2013

[3] Dutonde," Website Developmemt Technologies: A Review" ISSN: 2321-9653 Volume:10 Issue:01 Jan 2022

[4] Gurjeet, "Full Stack Web Development: Vision, Challenges and Future Scope" e-ISSN: 2395-0056 p-ISSN: 2395-0072 Volume:9 Issue:04 Apr-2022

[5] Kadam, "Introduction to MERN Stack & Comparison with Previous Technologies" doi: 10.48047/ecb/2023.12.si4.1300

[6] Vedhapriya, "Full Stack Development – A New Horizon in Technologies" Volume:05 Issue:06 June-2023 e-ISSN: 2582-5208