e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:11/November-2023 Impact Factor- 7.868 www.irjmets.com

IMPLEMENTATION AND COMPARISON OF MERN STACK TECHNOLOGY WITH HTML/CSS, SQL, PHP & MEAN IN WEB DEVELOPMENT

Er. Vikas Goyal[*1], Ashish Kumar Mishra[*2], Daljeet Singh[*3]

[*1]Associate Professor, Department Of CSE, MIMIT Malout, Punjab, India.

[*2,3]Students, Department Of CSE, MIMIT Malout, Punjab, India.

DOI : https://www.doi.org/10.56726/IRJMETS46315

ABSTRACT

This abstract provides a concise overview of the web development and MERN (MongoDB, Express.js, React, Node.js) stack and its comparison with other popular web development technologies. The MERN stack is a full-stack JavaScript framework widely utilized for building modern web applications. In this analysis, we explore the key components of the MERN stack and delve into its benefits. Furthermore, we draw comparisons with alternative web development stacks, such as MEAN (MongoDB, Express.js, Angular, Node.js). By examining factors like scalability, performance, and developer experience, this overview aims to help developers make informed choices when selecting the appropriate technology stack for their projects.

Keywords: MERN Stack, Mongodb, Express.Js, React, Node.Js, Web Applications, HTML, CSS, SQL, Nosql, Front-End, Back-End, PHP And MEAN Stack.

I. INTRODUCTION

Web Technology

Web technology refers to the tools, languages, and frameworks used to create and manage websites and web applications. In this, there are two parts front-end and back-end. It is a dynamic field, constantly evolving with new technologies and trends. It plays a vital role in the digital world, enabling businesses and individuals to establish an online presence and offer interactive and engaging experiences to users.

Implementing web development following steps:

1. Project Planning: Define the goals and requirements of your website or web application. Determine the target audience and the features you need.
2. Choose a Tech Stack: Select the programming languages, frameworks, and tools you'll use. Common choices include HTML, CSS, JavaScript, and popular frameworks like React, Angular, or Vue for the frontend, and languages like Python, Ruby, or Node.js for the backend.
3. Design: Create wireframes and design mockups for your website's layout, look, and feel. We can use design tools like Adobe XD, or Sketch.
4. Frontend Development: Write HTML, CSS, and JavaScript to build the user interface and client-side functionality. Make sure your site is responsive for various screen sizes.
5. Backend Development: Develop the server-side logic that handles requests, processes data, and interacts with databases. Common backend frameworks include Express (Node.js), Ruby on Rails, Django (Python), and Laravel (PHP).
6. Database Setup: Choose a database system (e.g., MySQL, PostgreSQL, MongoDB) and set up the database schema and queries to store and retrieve data.
7. Server Hosting: Deploy the backend on a server. Options include cloud platforms like AWS, Azure, or Heroku, or using a shared hosting service.
8. Domain and Hosting: Register a domain name and choose a hosting service for the website. There are hosting providers that offer domain registration.
9. Testing: Thoroughly test your website or web application. Perform unit testing, integration testing, and user testing to ensure it works as expected.
10. Security: Implement security measures, such as SSL/TLS certificates, input validation, and authentication, to protect your site from common vulnerabilities.
11. Optimization: Optimize your site for performance by minifying files, using content delivery networks (CDNs), and optimizing images and assets.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:11/November-2023 Impact Factor- 7.868 www.irjmets.com

12. SEO (Search Engine Optimization): Optimize your website's content, meta tags, and structure to improve its visibility in search engines.
13. Launch: Once everything is ready and tested, deploy your website to the production server, making it accessible to users.
14. Maintenance: Regularly update and maintain your website to fix bugs, add new features, and keep it secure.
15. Monitoring and Analytics: Implement tools to monitor the website's performance and gather user analytics to make informed decisions for improvements.

Limitations of Web Technology:

1. Performance: Traditional web applications often face performance bottlenecks due to their client-server architecture. This can lead to slower load times and less responsive user interfaces.
2. Scalability: As web applications grow in popularity, they need to scale to handle increased traffic. Scaling can be challenging and costly with traditional technologies.
3. Real-time Updates: Providing real-time updates and interactions can be challenging with technologies like PHP or plain HTML.
4. Code Reusability: Maintaining code and ensuring reusability can be complex in web development, leading to increased development time and potential errors.

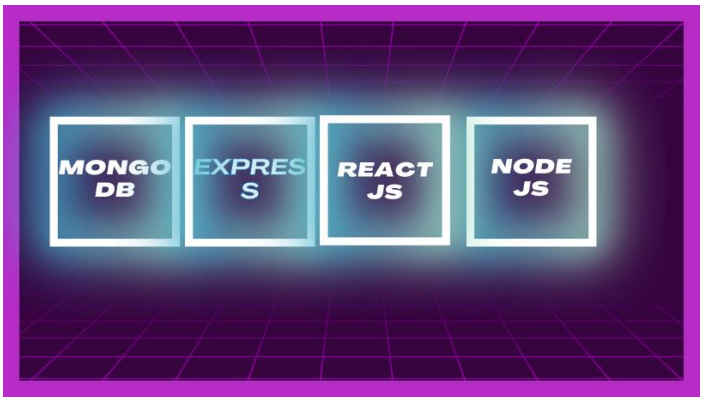

Figure 1: MERN Stack technology

Mern Stack

The MERN stack is a powerful and popular technology stack for building web applications. The word MERN form is M for MongoDB E for Express R for ReactJS and N for NodeJS.

- MongoDB: MongoDB is a NoSQL database that stores data in a flexible way, JSON-like format called BSON. Used for handling unstructured or semi-structured data and for storing the data of web applications.
- Express.js: Express.js is the backend server framework in the MERN stack that provides a robust set of features for building web and mobile applications. It simplifies server-side development and routing, making it easier to create APIs and handle requests and responses.
- React: React is a popular frontend of JS library for building user interfaces. It allows developers to create reusable UI components, making it easy to build interactive and dynamic web applications. Providing a fast and efficient way to develop user interfaces.
- Node.js: Node.js is a JS runtime environment that allows developers to run JavaScript on server-side. It works with Express.js to build the backend MERN applications. Node.js is known for its speed and scalability, for handling server-side logics and data retrieval.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:11/November-2023 Impact Factor- 7.868 www.irjmets.com

II. ARCHITECTURE OF MERN STACK

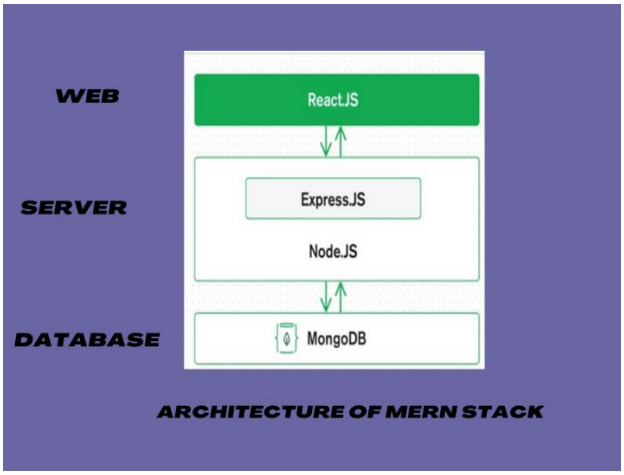

Figure 2: Architecture of MERN stack

The three-tier architecture consists of three parts following are:

1. Web - Frontend is done with the help of the REACT JS javascript library.
2. Server - Backend work with the help of i.e Express Js & Node js.
3. Database - Data storage work is done with the help of cloud MongoDB.

III. IMPLEMENTATION OF MERN STACK

Implementing the MERN (MongoDB, Express.js, React, Node.js) stack for a web application:

Step 1: Set Up Your Development Environment

Install Node.js and npm: Download and install Node.js from the official website.

Step 2: Create a New Project

Create a new directory for the project and navigate to it using the terminal.

Step 3: Backend Setup (Node.js and Express.js)

- Initialize a Node.js project: Run npm init and follow the prompts to set up your project.
- Install Express.js: Run npm install express.
- Create a server file (e.g., server.js) and set up your Express server.
- Define API routes for your application using Express.

Step 4: MongoDB Setup

- Install MongoDB: Set up MongoDB locally or use a cloud-based MongoDB service.
- Install Mongoose: Run npm install mongoose to use Mongoose for MongoDB database interactions.
- Connect the Express application to the MongoDB database using Mongoose.

Step 5: Frontend Setup (React)

- Create a React application: Run npx create-react-app your-app-name to set up a new React project.
- Navigate to your React project directory: cd your-app-name.

Step 6: Connect Frontend and Backend

- Ensure your backend API serves data to your frontend application through HTTP endpoints.
- Use CORS (Cross-Origin Resource Sharing) to handle cross-origin requests.

Step 7: Create React Components

- Create React components to display data and user interfaces.
- Use React Router to handle client-side routing between different pages or views.

Step 8: Styling and UI

- Style your React components using CSS, Sass, or a CSS-in-JS solution.
- You can use CSS frameworks like Material-UI or Bootstrap for pre-designed components.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:11/November-2023 Impact Factor- 7.868 www.irjmets.com

Step 9: User Authentication and Authorization

- Implement user authentication using Passport.js and JWT (JSON Web Tokens).
- Create routes and middleware for authentication and authorization.

Step 10: Deployment

- Deploy your MongoDB database to a cloud service or host it locally.
- Deploy your Express.js backend to a server, such as Heroku, AWS, or a VPS.
- Deploy your React frontend to a static file hosting service like Netlify or Vercel.

Step 11: Testing and Debugging

- Write unit tests for your backend and frontend components.
- Use debugging tools and techniques to identify and fix issues in your application.

Step 12: Continuous Integration/Continuous Deployment (CI/CD)

Set up CI/CD pipelines for automating testing and deployment processes.

Step 13: Scaling and Optimization

Optimize your application for performance and scalability as needed.

Step 14: Monitoring and Error Handling

Implement logging and monitoring to track errors and performance.

So, these steps provide a comprehensive guide to implementing the MERN stack for the web application.

Figure 3: HTML/CSS VS REACT

IV. COMPARISON OF HTML/CSS WITH REACT

HTML and CSS are foundational technologies for building web pages, while React is a JavaScript library for building user interfaces. Here's a comparison of these technologies:

1. Purpose: HTML (Hypertext Markup Language) is used to structure the content and define the elements on a web page, such as headings, paragraphs, images, and links.CSS (Cascading Style Sheets) is used to control the presentation and styling of web page elements, including layout, colors, fonts, and animations.

React is a JavaScript library for building interactive and dynamic user interfaces. It allows you to create reusable UI components and manage the state and behavior of your web applications.

2. Language: HTML and CSS are not programming languages; they are markup and style sheet languages, respectively.

React is built on JavaScript, which is a programming language.

3. Structure: HTML defines the structure and content of a web page using tags and elements.CSS is used to style and format the elements defined in HTML.

React uses JSX (JavaScript XML) to define the structure and components of a user interface.

4. Interactivity: HTML and CSS are static; they describe the structure and styling of a web page, but they don't provide interactivity on their own.

React allows you to create interactive web applications by handling user events, managing the state, and updating the UI in response to changes.

5. Reusability: HTML and CSS are not inherently designed for reusability, although CSS classes can be reused to style multiple elements.

React promotes the creation of reusable components, making it easier to build and maintain complex user interfaces by composing smaller, self-contained pieces.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:11/November-2023 Impact Factor- 7.868 www.irjmets.com

6. DOM Manipulation: In traditional HTML-based development, you might need to manipulate the Document Object Model (DOM) directly to create dynamic UI updates.
React abstracts DOM manipulation by using a virtual DOM to efficiently update the actual DOM when the state of the application changes, making UI updates more efficient.

Figure 4: MongoDB VS SQL

V. COMPARISON OF SQL WITH MONGODB (NO SQL)

SQL (a relational database) with MongoDB (a NoSQL database):

1. Data Model: SQL databases use a structured, tabular data model with predefined schemas. Data is organized in tables with rows and columns.
MongoDB uses a document-oriented data model where data is stored in JSON-like documents. It's schema-less and flexible.
2. Schema: SQL databases require a fixed schema, meaning you need to define the structure of tables and relationships in advance.
MongoDB is schema-less, allowing you to insert data without a predefined schema, and you can evolve the schema as needed.
3. Query Language: SQL databases use SQL (Structured Query Language) for querying, which is well-suited for complex queries and reporting.
MongoDB: MongoDB has its query language and supports rich querying capabilities, particularly for document-based data.
4. Scalability: Scaling SQL databases can be challenging, often involving vertical scaling (adding more resources to a single server) or complex sharding setups.
NoSQL databases like MongoDB are designed for horizontal scalability, making it easier to distribute data across multiple servers.
5. Consistency: SQL databases provide strong consistency with ACID (Atomicity, Consistency, Isolation, Durability) transactions, ensuring data integrity
NoSQL databases, including MongoDB, often prioritize availability and partition tolerance over strong consistency and use the BASE (Basically Available, Soft state, Eventually consistent) model.
6. Development Speed: Setting up an SQL database can be time-consuming due to schema design and migrations.
MongoDB can accelerate development as it allows for flexible data structures, making it easier to adapt to evolving requirements.

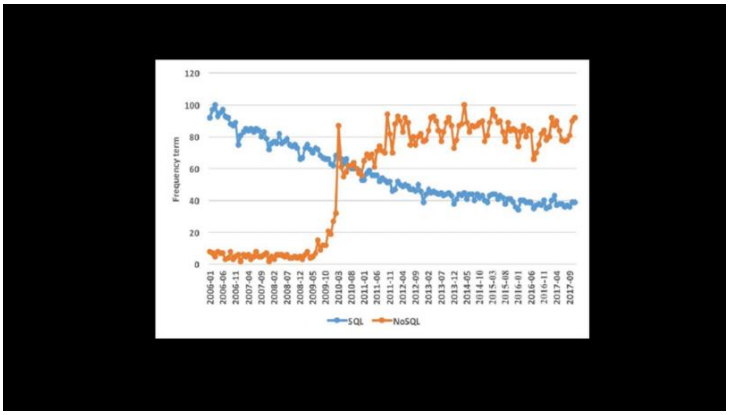

Figure 5: NOSQL and SQL growth

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:11/November-2023 Impact Factor- 7.868 www.irjmets.com

So MongoDB is a strong choice over SQL. The coming future will replace SQL completely with MongoDB

Figure 6: Nodesjs VS PHP

VI. COMPARISON OF NODEJS WITH PHP

Node.js and PHP are both popular server-side scripting languages, but they have different characteristics and use cases. Here's a comparison between Node.js and PHP:

1. Architecture: Node.js is built on a non-blocking, event-driven architecture. It uses JavaScript and the V8 engine to execute server-side code asynchronously.

PHP is a synchronous, blocking language. Each request typically waits for the previous one to complete before handling the next request.

2. Language: Node.js uses JavaScript, which is a widely used and versatile language known for its speed and scalability.

PHP has its scripting language specifically designed for web development.

3. Performance: Node.js is known for its high performance and scalability, making it suitable for building real-time applications and APIs.

PHP performs well for traditional web applications but may require additional tools or extensions for high-performance tasks.

4. Ecosystem: Node.js has a large ecosystem of packages and libraries available through npm, making it easy to extend and build applications.

PHP also has a rich ecosystem with a wide range of pre-built libraries and frameworks like Laravel and Symfony.

5. Concurrency: Node.js excels at handling a large number of concurrent connections due to its non-blocking architecture.

PHP may struggle with handling a high number of concurrent requests, but this can be mitigated with additional configurations or web server setups.

Figure 7: Nodesjs and PHP interest graph

In summary, the choice between Node.js and PHP. Node.js is a strong choice for real-time and scalable applications.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:11/November-2023 Impact Factor- 7.868 www.irjmets.com

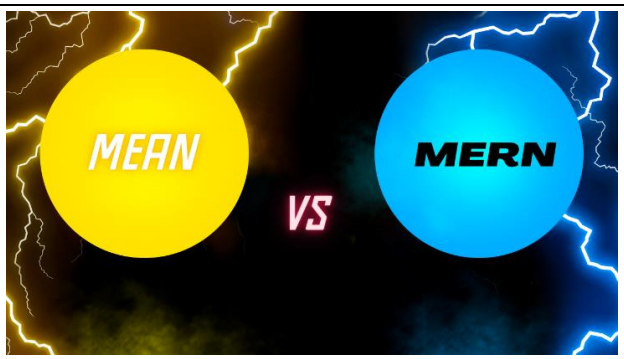

Figure 8: MEAN VS MERN

VII. COMPARISON OF MEAN STACK WITH MEAN

1. Flexibility-MERN: MERN is often considered more flexible, allowing developers to choose from a wide range of libraries and packages for various tasks, which can result in a more customized development approach.

MEAN: MEAN has a more opinionated structure with Angular, which can lead to a more standardized and streamlined development process but with less flexibility in choosing components.

2. Learning Curve-MERN: React has a shorter learning curve, making it easier for developers to get started quickly, especially if they have prior experience with JavaScript.

MEAN: Angular has a steeper learning curve due to its comprehensive framework and TypeScript usage, which can take more time to master.

3. Community and Ecosystem-MERN: React and the MERN stack have a large and active community, which means abundant resources, third-party libraries, and a vibrant ecosystem.

MEAN: Angular has a strong community and ecosystem as well, although it may not be as large as React's.

4. Performance-MERN: React's virtual DOM and efficient rendering mechanisms often result in excellent front-end performance.

MEAN: Angular is also performant, but its size and complexity may lead to larger bundle sizes and slightly slower initial loading times.

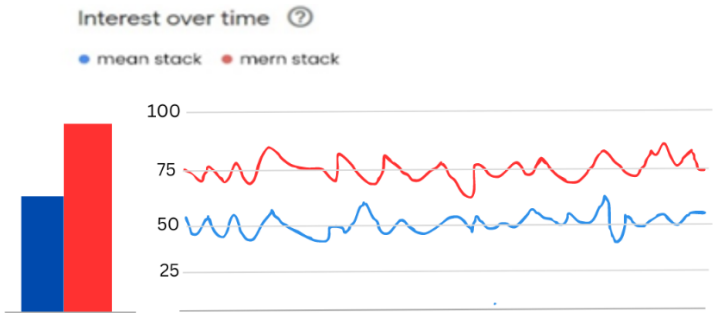

Figure 9: MERN and MEAN interest graph

In summary, the choice between MERN and MEAN. MERN is a strong choice for real-time and scalable applications.

VIII. BENEFITS OF MERN STACK OVER OTHER TECHNOLOGIES

MERN Stack Overcomes these Limitations:

1. HTML/CSS:

Limitation: HTML and CSS are fundamental for web structure and styling, but they are static and lack interactivity.

MERN Solution: React, part of the MERN stack, allows for dynamic and interactive user interfaces, enabling components to be updated without a full page refresh.

2. PHP:

Limitation: PHP is a server-side scripting language known for slower execution and scalability challenges.

MERN Solution: Node.js, which is part of MERN, provides faster, non-blocking I/O, making it well-suited for high-performance server-side scripting.

3. MySQL:

Limitation: Traditional relational databases like MySQL can be less flexible and scalable for certain web applications.

MERN Solution: MongoDB, a NoSQL database used in the MERN stack, offers scalability and flexibility, making it ideal for handling the dynamic data requirements of modern web apps.

4. jQuery:

Limitation: jQuery, while helpful for DOM manipulation, can lead to complex and hard-to-maintain code.

MERN Solution: React simplifies UI development with its component-based architecture, promoting cleaner and more maintainable code.

5. AJAX:

Limitation: Asynchronous JavaScript and XML (AJAX) can be challenging to implement and maintain for real-time web applications.

MERN Solution: MERN stack leverages WebSocket technology and libraries like Socket.io for efficient real-time communication.

6. LAMP Stack (Linux, Apache, MySQL, PHP):

Limitation: LAMP stack can be less modular and require manual configuration, making it harder to adapt to modern web application needs.

MERN Solution: MERN stack provides a more flexible and modular architecture, allowing for rapid development and easier adaptation to evolving requirement.

IX. CONCLUSION

This research paper concludes and examines various factors like scalability, performance, and developer experience, aims to help developers make informed choices when selecting the appropriate technology stack for their projects. Research paper shows the Mern stack Technology is better than all other Technology to learn to build projects.

X. REFERENCES

[1] Yogesh Baiskar, Priyas Paulzagade, Krutik Koradia, Pramod Ingole, Dhiraj Shirbhate, “MERN: A Full-Stack Development”, International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653.

[2] Sumangala A. Bafna, Pratiksha D. Dutonde, Shivani S. Mamidwar, Monali S. Korvate, Prof. Dhiraj , “Review on Study and Usage of MERN Stack for Web Development” International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653.

[3] Sourabh Mahadev Malewade, Archana Ekbote Information Technology Department VCET Vasai, Palghar, India “Performance Optimization using MERN stack on Web Application” International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 IJERTV10IS060239 Vol. 10 Issue 06, June-2021

[4] Dr. Santosh Kumar Shukla, Shivam Dubey, Tarun Rastogi, Nikita Srivastav "Application using MERN Stack" International Journal for Modern Trends in Science and Technology, 8(06): 102-105, 2022 ISSN: 2455-3778.

[5] Avnish Kumar Sharma. BIG BUY(E-COMMERCE) by using MERN. International Journal for Modern Trends in Science and Technology 2022, 8(06), pp. 106-111.

[6] Dharavath Ramesh, Shankar Nayak Bhukya "Inclusion of e-Commerce Workflow with NoSQL: DBMS MongoDB Document store" 978-1-5090-0612-0/16/$31.00 ©2016 IEEE.

[7] Karishma Arora, Jai Nagpal, Vaishnavi "Implementation of MERN: A Stack of Technologies to Design Effective Web Based Freelancing Applications" International Journal of Scientific Research in Computer Science Engineering and Information TechnologyDOI:10.32628/CSEIT23902104