FRAMEWORK REPORT

MERN STACK

Prepared by Group 1

Ha Noi, 3-29-2024

Table of Contents

I. Introduction

In order to make a website that can meet the needs of both users and experts, MERN (MongoDB, Express.js framework, ReactJS library, NodeJS platform) is one of the powerful stacks that can help us to develop an appointment web application.

MERN stack developers are web developers who are certified and have hands-on experience using different frameworks such as MongoDB, Express JS, Node JS, and React JS. They can understand and develop the logic and front end. They use MongoDB and Express JS to maintain the data and React and Node JS to create the UI components and handle the requests sent to the servers by clients. The front end will answer the client requests and the back end will store the data and manage it (such as websites and application data) so that the front end can access the information quickly.

II. Components

MERN is an acronym representing a full-stack web development framework, where each letter stands for a key component of the technology stack:

- MongoDB: A versatile and scalable NoSQL database, MongoDB stores and manages the data of web applications in a flexible, JSON-like format.
- Express.js: A robust and minimalist Node.js web application framework, Express.js simplifies the creation of server-side applications and APIs.
- React: A powerful JavaScript library for building dynamic user interfaces, React streamlines the development of interactive and responsive front-end components.
- Node.js: An open-source, server-side JavaScript runtime environment, Node.js executes JavaScript code on the server, enabling high-performance, non-blocking, and event-driven applications.

III. Usage

The MERN stack is a set of strong technologies used to create scalable web applications. With JSON and JavaScript, the MERN architecture makes it easy to build 3-tier systems, covering front-end, backend, and database components.

1. Front-end: React JS

At the core of the MERN stack is React JS, a JavaScript framework for creating client-side HTML applications. It simplifies the creation of intricate interfaces that connect to data on a backend server and present it as HTML using straightforward elements.

React is robust and intuitive, handling data-driven interfaces with minimal code—a distinct advantage of the MERN stack. It efficiently supports forms, lists, events, and error handling, all in one place.

2. Server Tier: Node JS and Express JS

One great thing about the MERN stack is Express JS, the server-side framework that works on a Node JS server. Express JS is a fast, unbiased, and lightweight web framework for Node JS. It has modules that help link incoming URLs to server functions, handling HTTP requests and responses.

To make your application better, you can use XML HTTP requests or React JS front-end POSTs to connect to Express JS functions. These functions use Node.js drivers for MongoDB to read and update data in your MongoDB database.

3. Database Tier: MongoDB

One of the advantages of the Mern Stack is its ability to collect data for content, user profiles, uploads, events, and comments. The Most preferable situation is to work along with the Node, Express, and React.

React JS produces the JSON files on the front end, and then it is sent to the Express JS server for processing and instant archival, which are available in MongoDB for recovery. Moreover, it is possible to get more out of the MERN stack by making a web application that depends on the system’s functional requirements.

IV. Benefit

JavaScript Throughout: MERN allows developers to use JavaScript for both front-end and back-end development. This language consistency simplifies the development process and reduces the need to switch between different programming languages.

- Reusability: React components are highly reusable, which speeds up development by allowing developers to build modular UI elements that can be used across the application. This reusability also ensures a consistent and maintainable codebase.
- Efficiency: Node.js, a part of the MERN stack, is known for its non-blocking and event-driven architecture. This means that applications built with MERN can handle a large number of simultaneous connections efficiently, making them well-suited for real-time applications and scalability.
- Rich User Interfaces: React, with its component-based architecture, enables the creation of interactive and dynamic user interfaces. This is particularly useful for building single-page applications (SPAs) that offer a smooth and responsive user experience.
- Cross-Platform Compatibility: Applications developed using the MERN stack are highly adaptable and can run on various platforms, including web browsers, mobile devices, and desktops, with minimal modification.
- Real-Time Capabilities: Node.js, in combination with React, is well-suited for building real-time applications, such as chat applications, notifications, and live updates, enhancing user engagement and interaction.
- Open Source: All components of the MERN stack are open-source, which not only reduces development costs but also allows developers to tap into a vast community of contributors and stay up-to-date with the latest innovations.
- Flexibility: The modular nature of the MERN stack allows developers to choose the components and tools that best suit their project's requirements, promoting flexibility and customization.
- Testing and Debugging: The MERN stack supports various testing and debugging tools, making it easier to identify and resolve issues during development, resulting in more robust and reliable applications.

V. Limitations

While the MERN (MongoDB, Express.js, React, Node.js) stack is a popular and powerful choice for building web applications, it does have some disadvantages that developers should be aware of. Here are some potential drawbacks of the MERN stack:

- Learning Curve: Learning the entire MERN stack can be challenging for beginners, especially those who are new to JavaScript and web development. Each component of the stack has its own set of concepts and practices, making it a steep learning curve for someone starting from scratch.
- Full Stack Development Skills: Building a MERN stack application requires proficiency in both front-end (React) and back-end (Node.js, Express.js) development. Finding developers who are skilled in both areas can be more challenging and might limit the pool of available talent.
- JavaScript Everywhere: While using a single language (JavaScript) for both front-end and back-end development can be seen as an advantage, it can also be a disadvantage. Developers who prefer to use different languages for different parts of the stack may find this limitation frustrating.
- Scalability Challenges: MongoDB, being a NoSQL database, may face challenges with scalability when dealing with complex relationships between data. Developers need to carefully design the schema and structure data to ensure efficient scaling. Security Concerns:
- Security is a critical aspect of web development, and improper configurations or coding practices can lead to vulnerabilities. Developers need to be vigilant in securing both the front-end and back-end components, and the flexibility of the MERN stack can make it easier to overlook security best practices.
- SEO Challenges: Single Page Applications (SPAs) built with React may face challenges with search engine optimization (SEO) since search engines may have difficulty crawling and indexing content generated dynamically through JavaScript.
- Community Support: While the MERN stack has a large and active community, it might not be as extensive as some other stacks. This can lead to slower adoption of new features or technologies and potentially fewer third-party libraries and tools compared to more established stacks.
- Real-Time Features: While Node.js is well-suited for real-time applications, implementing complex real-time features may require additional libraries or tools. Ensuring smooth real-time communication can add complexity to the development process.
- No Built-in Authorization and Authentication: The MERN stack does not provide built-in solutions for user authentication and authorization. Developers need to rely on third-party libraries or implement these features from scratch,

which can be time-consuming and may introduce security risks if not done correctly.

- Vendor Lock-In: Using the MERN stack can lead to a certain degree of vendor lock-in, especially with MongoDB as the database. If there's a need to switch to a different technology or database, migrating the entire stack can be challenging.

VI. Conclusion

The MERN stack, composed of MongoDB, Express.js, React.js, and Node.js, has proven to be a comprehensive and powerful solution for modern web application development. Each component of the MERN stack contributes significantly to projects, from managing flexible data storage with MongoDB, handling application logic and APIs with Express.js, creating dynamic and responsive user interfaces with React.js, to running JavaScript on the server with Node.js. This combination not only provides a seamless and efficient workflow for developers but also optimizes the end-user experience.