A Review Paper on MERN Stack for Web Development

Pragati Bhardwaj [1], Dr Dinesh Kumar [2]

P.G. Scholar, Department of Computer Engineering, SRCEM, Palwal, Haryana, India[1]

HOD & Associate Professor, Department of Computer Engineering, SRCEM, Palwal, Haryana, India[2]

ABSTRACT: Today Developers around the world are making efforts to enhance user experience of using application as well as they are trying to enhance the developer's workflow of designing applications to deliver projects and rollout change requests under strict timeline. Stacks can be used to build web applications in the shortest span of time. The advantage of such JavaScript stacks helps to build an integrated solution by using only JavaScript. This paper provides the introduction to concept and describes the MERN stack open source and explains the meaning of them along with their future scope. This paper also describe that React is preferred more to Angular in front-end development.

KEYWORDS: React , MERN, JavaScript, Express, MongoDB

I. INTRODUCTION

The stacks used in web development are basically the response of software engineers to current demands. They have essentially adopted preexisting frameworks (including JavaScript) to make their lives easier.

While there are many, MEAN and MERN are just two of the popular stacks that have evolved out of JavaScript.

Both of these stacks are made up of open source components and offer an end-to-end framework for building comprehensive web apps that enable browsers to connect with databases. The common theme between the two is JavaScript and this is also the key benefit of using either stack.

You can basically avoid any syntax errors or any confusion by just coding in one programming language, JavaScript. Another advantage of building your next web project with MEAN or MERN is the fact that you benefit from its enhanced flexibility [1].

II. LITERATURE SURVEY

MEAN STACK

The MERN stack is almost similar to MEAN stack except that React used in place of Angular. To understand MERN stack we have to understand firstly React Js. React was born not in the Facebook application that we all see, rather in Facebook's Ads organization. Originally, they used a typical client-side MVC model to start with, which had all the regular two-way data binding and templates. Views would listen to changes on models, and they would respond to those changes by updating themselves. Soon, this got pretty hairy as the application became more and more complex. Cascading updates became difficult to maintain, because there would be subtle difference in the code to update the view, depending on the root cause of the update. That's when they started thinking of building something that's declarative rather than imperative [5].

A. *Mongodb*

Mongodb is an open-source, document-based database. It is also called as a noSQL database program. It makes use of JSON like documents along with schemas. Because of the tremendous increase in the volume and variety of data over the past few years, there has been a need for a non-relational database. Mongodb fulfills these criteria, as it stores

its data in a document-oriented, file structure. Each database can have multiple collections and data in those collections is stored in the form of array of JSON objects.

B. *Express.js*

Express.js is a minimal and flexible Node.js web application framework which provides a robust set of fundamental web application features without obscuring Node.js features. Express makes it easy to create APIs because of access to middleware. Middleware functions are functions which have access to the request and response objects along with a next function, which when invoked, executes the middleware succeeding the current middleware. Express.js provides a similar functionality to that of what Spring offers to Java applications, that is, an easy to use web framework.

C. *React*

A declarative efficient and flexible front end JavaScript library for user interface development. React Js can be used for both client and server side. React anchors the MERN stack. In some sense, this is the defining component of the MERN stack. React is an open-source JavaScript library maintained by Facebook that can be used for creating views rendered in HTML. Unlike Angular JS, React is not a framework. It is a library. Thus, it does not, by itself, dictate a framework pattern such as the MVC pattern. You use React to render a view (the V in MVC), but how to tie the rest of the application together is completely up to you.

D. *Node.js*

Node.js is an event-driven asynchronous JavaScript runtime which was primarily designed to make scalable network applications and APIs. Node Js is the best choice for real time and streaming application. Node Js was built on Chrome V8 Engine which is an open-source JavaScript engine developed on C++ and V8 implements the ECMA Script. Node.js uses an asynchronous event-driven [2], non-blocking I/O model that makes it lightweight, efficient and high-performance [3]. Although the Node is mentioned as the last tool of the MEAN stack, it is the most important one. Node.js' package ecosystem, npm (node package manager), is the largest ecosystem of open source libraries in the world which provides all the modules and dev Dependencies for Angular CLI along with all the Dependencies for Express. It acts as the dependency injector of the stack.

III. PROS OF MERN STACK

- JavaScript Everywhere: The best part about MERN that I like is that there is a single language used everywhere. We use JavaScript for client-side code as well as server-side code. Even if you have database scripts (in MongoDB), you write them in JavaScript. So, the only language you need to know and be comfortable with is JavaScript.
- JSON Everywhere: When using the MERN stack, object representation is JSON (JavaScript Object Notation) everywhere – in the database, in the application server and on the client, and even on the wire.
- Node.js Performance: Due to its event driven architecture and non-blocking I/O, the claim is that Node.js is very fast and a resilient web server.
- The npm Ecosystem: I've already discussed about the huge number of npm packages available freely for everyone to use. Any problem that you face that you think others should have faced too, you'll find that there is an npm package for that. Even if it doesn't fit your needs exactly, you can fork it and make your own npm package.
- Isomorphic: SPAs used to have the problem that they were not SEO friendly. One had to use workarounds like running Phantom JS on the server to pseudo-generate HTML pages, or use Prerender.io services that did the same for you. These introduce an additional complexity.
- It's not a Framework: Not many people like or appreciate this, but I really like the fact that React is a library, not a framework.

IV. CONCLUSION

React gives you freedom and simplicity. We have to write less code to do more in React, also React has better performance than Angular due to React's implementation of a virtual DOM. ReactJS can be used for both client and server side that means you have not to learn two language for server and client side. In React the changes in Application are easy to satisfy the client need. The paper has discussed about the use of mean stack for development along with its definition and features.

V. FUTURE SCOPE

React is just getting started, so it has more future scope of growth. React is very useful for naïve developers as they have to teach only JavaScript Technology. Only time can tell, but MERN certainly has an immense potential. Developing applications using MERN will save you from a whole bunch of learning mistakes and integration issues.

Given how fast React has turned into leading UI development technology, it won't take long before we'll see a full-fledged ecosystem sprouting around it. MERN has a better future for large scale application.

REFERENCES

1. http://digi117.com/blog/choosing-the-right-stack-for-your-next-web-project-mean-vs-mern.html
2. MEAN.io. (2015) MEAN— Full-Stack JavaScript Using MongoDB, Express, AngularJS, and Node.js. [Online]. Available: http://mean.io/
3. S.Tilkov and S. Vinoski, ―Node.js: Using JavaScript to Build High-Performance Network Programs,‖ IEEE Internet Computing, vol. 14, no. 6, pp. 80 – 83, 2010. [Online] Available: http://doi.ieeecomputersociety.org/10.1109/MIC.2010.145
4. Leff and J. T. Rayfield, ―Web-application development using the Model/View/Controller design pattern,‖Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference, pp. 118– 127, 2001.
5. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React - By Vasan Subramanian
6. MEAN vs MERN Stack : https://www.stackchief.com/blog/MEANvsMERNStack