

4ME305 Web 2.0 Development

Annamária Váradyová

School of Computer Science, Social Media
and Web Technology, Linnaeus University, Sweden

av22xc@student.lnu.se

Implementation Details

1. Introduction of selected frameworks

1.1. Front-end - React

React is a popular JavaScript library for creating front-end user interfaces. It offers a quick and effective method for developing versatile, complex UI components. The real DOM update process is optimised by React using a virtual DOM, leading to quicker and more effective updates. [Chiarelli, A. (2018).]

React adheres to the concept of components, which makes it simple to reuse code and assemble complex UI structures. The development process is streamlined and code clarity is maintained thanks to this modular approach. [Chiarelli, A. (2018).]

React is also compatible with React Native, allowing programmers to create native mobile applications with the same codebase as their web applications. React has become a flexible option for many enterprises thanks to its cross-platform functionality. [Chiarelli, A. (2018).]

For front-end development, React has a lot of advantages, including quickness, effectiveness, and simplicity. Its enduring popularity and extensive use by businesses across the globe show how valuable a frontend tool it is. [Chiarelli, A. (2018).]

1.2. Back-end – Express, Node.js

Popular server-side web development tools include Express and Node.js. Based on Chrome's V8 JavaScript engine, Node.js is a JavaScript runtime, while Express is a lightweight and adaptable web application framework. [Shah, H. (2017).]

Express enables programmers to handle requests and answers, middleware, and HTTP routes with ease. It offers a quick and orderly method for creating strong backends for web applications. The event-driven, non-blocking I/O architecture provided by Node.js, on the other hand, makes it the ideal choice for real-time applications and high speed web apps. [Shah, H. (2017).]

Backends may be made that are effective, scalable, and quick by combining Express with Node.js. They provide an adaptable and open-source server-side programming solution, making them a popular option among developers and businesses. [Shah, H. (2017).]

For creating backend systems, two potent technologies are Express and Node.js. They are a desirable option for web development because of their simplicity and flexibility as well as the performance and scalability provided by Node.js. [Shah, H. (2017).]

1.3. Database – MongoDB

A popular NoSQL database management solution is MongoDB. Data is kept as documents in a collection rather than in a table with rows and columns as in conventional relational databases since it is a document-oriented database. [Satheesh, M., D'mello, B. J., & Krol, J. (2015).]

Scalability is one of MongoDB's primary advantages since it makes it simple to handle massive volumes of data and the rising needs of contemporary online applications. Because the organisation of documents may be quickly changed to meet the changing requirements of an application, it also provides excellent performance and flexible data modelling. [Satheesh, M., D'mello, B. J., & Krol, J. (2015).]

MongoDB is renowned for being user-friendly, having a straightforward and understandable syntax for data queries. This makes it a fantastic solution for a range of use cases, including real-time analytics, content management, and e-commerce, in addition to its strong indexing and aggregation features. [Satheesh, M., D'mello, B. J., & Krol, J. (2015).]

MongoDB is a useful tool for managing databases. Many businesses and developers favour it due to its scalability, performance, and usability. MongoDB is a great option for your database needs whether you're developing a straightforward web application or a sophisticated solution. [Satheesh, M., D'mello, B. J., & Krol, J. (2015).]

1.4. Full stack – MERN

The MERN stack, which consists of MongoDB, Express, React, and Node.js, is a strong full-stack development framework that offers a comprehensive solution for creating dynamic and interactive online applications. Complex UI components can be easily and quickly built using React on the frontend, while Express and Node.js on the backend make it simple to handle HTTP routes and process data in real-time. The database used is MongoDB, which provides a scalable and adaptable way to store data. [Subramanian, V. (2017).]

JavaScript is the only language used by the MERN stack, which simplifies development and eliminates the need for several technologies. It is perfect for a wide variety of web applications thanks to its highly flexible and scalable architecture. The MERN stack's components are continuously updated and refined, making it a cutting-edge framework for full-stack development. [Subramanian, V. (2017).]

2. Sketch of web application architecture

The term "web application architecture" describes the overall layout and organisation of a web application, including all of its elements, modules, and connections. It is focused on how the various components of a web application communicate with one another as well as with external systems like databases and APIs. [Subramanian, V. (2017).]

The database in the MERN stack is MongoDB, which stores data in an adaptable and scalable document-oriented manner. To manage routes and process HTTP requests, Express is utilised as the web framework, and Node.js provides the server-side runtime environment.

React is used to build sophisticated UI components on the frontend, offering a quick and easy method for developing dynamic and interactive user interfaces. To access and alter data stored in MongoDB, the React components interact with the Express server. [Subramanian, V. (2017).]

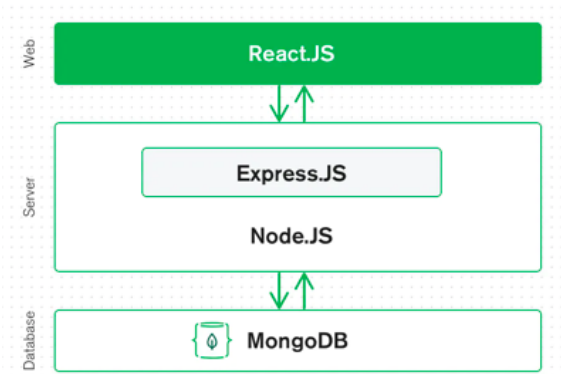


Figure 1 - Web architecture (MERN)

The MERN (MongoDB, Express, React, and Node.js) stack's potential architecture for a registration and login application is as follows:

1. *MongoDB: MongoDB, a NoSQL database that stores data in an adaptable and scalable document-oriented format, is used to create the database layer. The database keeps track of user details, login information, and any other pertinent information.*
2. *Express: Express, a web framework based on Node.js, is used to create the application layer. It manages the application's logic and processing, including data validation, user authentication, and interaction with the MongoDB database.*
3. *React: React, a frontend JavaScript toolkit for creating dynamic and interactive user interfaces, is used to build the presentation layer. The creation of UI components and control of data flow are both made easy and effective by React.*
4. *Node.js: Node.js offers the server-side runtime environment. This makes it possible for the Express application to manage HTTP requests and reply to user actions.*

Using the MERN (MongoDB, Express, React, and Node.js) stack, an application to submit photos to a social media site might have the following architecture:

1. *MongoDB: MongoDB, a NoSQL database that stores data in an adaptable and scalable document-oriented format, is used to create the database layer. Information about users, posts, and any other pertinent data are all stored in the database.*
2. *Express: Express, a web framework based on Node.js, is used to create the application layer. It manages the application's logic and processing, including data validation, user authentication, and interaction with the MongoDB database.*
3. *React: React, a frontend JavaScript toolkit for creating dynamic and interactive user interfaces, is used to build the presentation layer. The creation of UI components and control of data flow are both made easy and effective by React.*
4. *Node.js: Node.js offers the server-side runtime environment. It enables HTTP queries, file uploads, and image processing for the Express application.*

Discussion

Before the assignment, I had a short experience with front-end React. Not a big one, but I tried to make a small project in it during which I got to know the given front end. I chose Node.js and Express as a back-end, which complements each other, I have been learning this environment since the beginning of my studies and I have done all the projects through them. I chose MongoDB as the database, and also because I use it most often.

For the assignment, I chose the full-stack MERN, which consists of MongoDB, Express, React and Node.js, which can be used to create various applications. I have never connected front and back before, so this task was a big challenge. The biggest problem with the 'post' and

'get' methods. Therefore, it is necessary to receive information from the user, which the server will then process and display again to the user.

This challenge was difficult for me, but I did as much as I could. It was good to discover the overall full stack MERN, I want to continue with it, there is a lot of variety in which I want to further my education. There are many shortcomings in the created code that should be improved, but I mainly focused based on the requirements that the given application should contain.

Source code: <https://github.com/Anna8295/scheduler>

References

- Subramanian, V. (2017). *Pro MERN Stack* (Vol. 13). Apress.
- Chiarelli, A. (2018). *Beginning React: Simplify your frontend development workflow and enhance the user experience of your applications with React*. Packt Publishing Ltd.
- Shah, H. (2017). Node.js challenges in implementation. *Global Journal of Computer Science and Technology*, 17(E2), 73-84.
- Satheesh, M., D'mello, B. J., & Krol, J. (2015). *Web development with MongoDB and NodeJs*. Packt Publishing Ltd.