DALHOUSIE
UNIVERSITY

CSCI 5709

Advanced Web Services

Tutorial 1: Application Frameworks

Group 10

Prepared By:

Aharnish Solanki (B00933563)
Bhavisha Oza (B00935827)
Dharven Doshi (B00925391)
Keyur Khant (B00935171)
Parth Mehta (B00931931)
Riya Patel (B00926204)

Table of Contents

Frontend Framework

We select **React** for our **frontend** due to its vast community support and smooth library integration, ensuring access to an extensive array of resources and tools. React's streamlined architecture and ease of use will significantly enhance our development speed and agility, making it the optimal choice for our project's frontend needs [1]. In contrast to other frameworks like Angular, React offers a more lightweight and flexible approach. While Angular is known for its comprehensive feature set, it also introduces complexity and a steeper learning curve due to its opinionated nature and large array of built-in functionalities [8].

Backend Technologies

We select **Express** and **Node.js** over other backend technologies due to their seamless integration and performance advantages. Node.js excels in handling asynchronous operations and high-traffic scenarios, crucial for real-time applications [2]. Its JavaScript base ensures smooth integration with MongoDB and React, offering a consistent development experience. Express.js enhances this with its simplicity, middleware support, and efficient RESTful API development capabilities [3]. They provide a cohesive, developer-friendly environment, making it a practical choice for efficient, scalable backend development in our project.

Database Solution

We selected **MongoDB** for our database over other SQL [10] and NoSQL alternatives [11] due to its compatibility with the MERN stack, ease of use, and efficiency in handling large volumes of unstructured data. Its document-oriented structure is highly efficient for storing and querying JSON-like data, which seamlessly integrates with JavaScript in Node.js and React. This leads to a more intuitive development process, reducing the complexity of data handling. MongoDB's scalability and adeptness with large, unstructured data sets give it an edge over traditional SQL databases and other NoSQL alternatives.

Hence, we decided to build our project on **MERN (MongoDB, Express, React.js, Node.js)** stack [4].

Some of the key benefits of MERN stack that we have taken in consideration are:

1. Using JavaScript across the stack can lead to more consistent coding practices and easier maintenance, as the same language paradigms and structures are used throughout the application [5].
2. MERN allows us to use JavaScript, both React (for frontend) and Node.js (for backend) streamlining development and reducing the need to context switch between languages [7].
3. Node.js is known for its non-blocking I/O and event-driven architecture, making it suitable for building scalable and high-performance applications. React's virtual DOM and efficient update mechanism make it fast for rendering dynamic user interfaces [8].
4. React and Node technologies benefit from the vast npm ecosystem, providing access to countless libraries and tools, streamlining the development process, also it works well with json object, so that NoSQL can be used and could be more beneficial for the project [6].
5. Being open-source and supported by a large community, the MERN stack can be a cost-effective solution. Both React and Node.js have large, active communities. This means ample number of resources, support, and best practices are available, making the development process more efficient and up to date [5,7].
6. MongoDB's schema-less nature allows for flexibility in handling data. Combined with Node.js, the stack is highly scalable, catering to the needs of growing applications [9].

References

[1] "React," *React.dev*. [Online]. Available: https://react.dev/. [Accessed: January 21, 2024].

[2] "About node.Js®," *Nodejs.org*. [Online]. Available: https://nodejs.org/en/about. [Accessed: January 21, 2024].

[3] "Express - Node.js web application framework," *Expressjs.com*. [Online]. Available: https://expressjs.com/. [Accessed: January 21, 2024].

[4] "What is the MERN stack? Introduction & examples," *MongoDB*. [Online]. Available: https://www.mongodb.com/mern-stack. [Accessed: January 21, 2024].

[5] D. Ó. Tuama, "Advantages of JavaScript," *Code Institute Global*, 03-Jan-2023. [Online]. Available: https://codeinstitute.net/global/blog/advantages-of-javascript/. [Accessed: January 21, 2024].

[6] J. Friesen, "The incredible advantages of full stack JavaScript - Jacob Friesen," *Medium*, 07-Apr-2019. [Online]. Available: https://medium.com/@jacobfriesen/the-incredible-advantages-of-full-stack-javascript-ba8a2d42626f. [Accessed: January 21, 2024].

[7] A. Rudenko, "MongoDB, express.Js, react.Js, node.Js," *ADCI Solutions*, 17-Aug-2021. [Online]. Available: https://www.adcisolutions.com/knowledge/mongodb-expressjs-reactjs-nodejs-what-makes-technology-stack-one-best. [Accessed: January 21, 2024].

[8] O. Green, "Node.Js vs. React vs. Angular vs. Vue: A comprehensive comparison," *Medium*, 24-Aug-2023. [Online]. Available: https://medium.com/@greennolgaa/node-js-vs-react-vs-angular-vs-vue-a-comprehensive-comparison-6ba6165ef8f. [Accessed: January 21, 2024].

[9] "Why use MongoDB and when to use it?," *MongoDB*. [Online]. Available: https://www.mongodb.com/why-use-mongodb. [Accessed: January 21, 2024].

[10] "Comparing the differences - MongoDB vs MySQL," *MongoDB*. [Online]. Available: https://www.mongodb.com/compare/mongodb-mysql. [Accessed: January 21, 2024].

[11] A. Kowalski, "Top MongoDB alternatives — features, use cases, pros and cons," *Softkraft.co*. [Online]. Available: https://www.softkraft.co/top-mongo-db-alternatives/. [Accessed: January 21, 2024].