PART II -- Technology Requirements

A-Technological Requirements

As the APlate application developers, we have decided to use the MERN stack. While developing and implementing the application, we have chosen to use MongoDB to store data in the database, ExpressJS for backend framework, NodeJS as a runtime environment, and ReactJs for the user interface.

We agreed that we should use AWS as the service provider for the APlate web application.

Technologies
1- MongoDB technology for database
2- ExpressJS technology for framework
3- ReactJS technology for frontend
4- NodeJS technology for backend

1-MongoDB

MongoDB is a document-oriented database system that uses the concept of NoSQL. Easy usability together with automatic scaling provides high performance. Easy to learn in short period of time for all team members.

Advantages

- It's a schemeless database, so we don't need to specify the number or type of columns before adding our data.
- MongoDB can easily handle heavy traffic flow for our website.
- It provides high performance data persistence.
- It uses internal memory to store running datasets and provides faster access to data. It also optimizes schemas for the most common use cases.

Disadvantages

- No joins Unlike relational databases, more than one query may be required if there is a need for aggregation that needs to be done manually.
- By storing a key name for each document, it needs higher memory space. Therefore, it may be necessary to deal with quite a lot of duplicate data.

2-Express.js

It is a Web Framework written for Nodejs. It can be installed as a package via npm. URL parse operations can be easier, and we can do our static file management works more easily. In short, Express provides with all the infrastructure necessary to make a website/application with NodeJS.

Advantages

- It provides the opportunity to develop our web application easily and quickly,
- Comprehensive applications can be developed with middleware modules,
- Web requests can be managed with the Route method,
- MongoDB can be easily used in the application.
- It allows to create a REST API,
- Static files are easy to manage,

Disadvantages

- As we have worked and learned other programming languages, we might find Express.js difficult to understand callback nature at the beginning

3-React Js

React is a component-based frontend library that uses the Virtual DOM (Virtual DOM) architecture, and its biggest task is to do all the operations on the interface in the most logical, easiest, cost-effective, and most efficient way.

Advantages

- Easy interface design
- Faster updates
- Perfect for JS errors
- It is purely component-built architecture
- Code reuse
- Rather than fully syncing changes, it creates a minimal list of updates sections to the actual DOM.

Disadvantages

- Some configuration is required to integrate into traditional MVC framework
- It's different than from what we are using in other languages

4-Node.js

Data transfers, form validations, database CRUD operations, interaction with DOM elements. The template engines usage, it is very enjoyable to write all of them with JavaScript and create an application.

Advantages

- It can distribute the density thanks to its flexible structure and asynchronous feature.
- Since we know JavaScript, we can analyze all our works with JavaScript without using other server-based languages.
- As Nodejs developers are a growing community, we will have ways to solve our bugs

Disadvantages

- Not being able to check the data type during compile time can be tiring in mathematical calculations.
- Finding the appropriate package for our application via NPM and adding it can be as simple as it is complex. This package may not be stable or work in harmony with each other. In this case, many bugs need to be debugged and fixed.
- Nested callback functions can get cluttered

B- Learning Plan

In order to achieve the goals of the technologies that decided to be used for our web application, we will read, watch or listen some tutorials especially the subjects we learned in our Full stack class.

The knowledge level of each group member and the level table we want to reach for MERN are as follows. Each group member will improve himself by paying attention to this learning process in order to achieve to develop the “APlate” website.

Yuksektepe Resul	**Current Knowledge**	**%70**	**%80**	**%90**
MongoDB	%70		5 Jan 2022	15 Jan 2022
ExpressJS	%60	20 Dec 2021	5 Jan 2022	15 Jan 2022
ReactJS	%50	25 Dec 2021	5 Jan 2022	15 Jan 2022
NodeJS	%60	25 Dec 2021	5 Jan 2022	15 Jan 2022
AWS	%50	25 Dec 2021	5 Jan 2022	15 Jan 2022

Ozduman, Kursad	**Current Knowledge**	**%70**	**%80**	**%90**
MongoDB	%60	25 Dec 2021	5 Jan 2022	15 Jan 2022
ExpressJS	%60	25 Dec 2021	5 Jan 2022	15 Jan 2022
ReactJS	%60	25 Dec 2021	5 Jan 2022	15 Jan 2022
NodeJS	%60	25 Dec 2021	5 Jan 2022	15 Jan 2022
AWS	%50	25 Dec 2021	5 Jan 2022	15 Jan 2022

Safa, Kazi Hasanus	**Current Knowledge**	**%70**	**%80**	**%90**
MongoDB	%70		5 Jan 2022	15 Jan 2022
ExpressJS	%70		5 Jan 2022	15 Jan 2022
ReactJS	%70		5 Jan 2022	15 Jan 2022
NodeJS	%70		5 Jan 2022	15 Jan 2022
AWS	%50	25 Dec 2021	5 Jan 2022	15 Jan 2022

Koc Hamza	**Current Knowledge**	**%70**	**%80**	**%90**
MongoDB	%80			15 Jan 2022
ExpressJS	%70		5 Jan 2022	15 Jan 2022
ReactJS	%80		5 Jan 2022	15 Jan 2022
NodeJS	%80		5 Jan 2022	15 Jan 2022
AWS	%50	25 Dec 2021	5 Jan 2022	15 Jan 2022