**Web Assignment 2**

**Ali Haider Rizvi**

**16L-4267**

**Question 1:**

**MongoDB vs Mongoose. Why are we using Mongoose package instead of MongoDB package? How would we perform CRUD operations using the MongoDB package? Give code examples for creating, retrieving, updating and deleting MongoDB documents.**

One of the MongoDB’s library is Mongoose which is a JavaScript library for a database that is used to abstract and simplify some of the boilerplate in interaction with MongoDB, shown by its native API, including object-modeling and introducing some level of schematic coherence. In particular, MongoDB is the software that manages durable storage and efficient retrieval & processing of data. Moreover, Mongoose is a set of high-level APIs for interaction with MongoDB, using JavaScript as a language.Taking example of an Ad Schema: -

* CREATE: -

let newAd = new Ad({

vendorUserName: “alihaider001”,

title: “A new place to live!”,

category: “BOYS”,

total\_capacity: 4,

current\_capacity: 2,

description: “wifi: yes”,

price\_per\_person:4000

});

newAd.save((err,Ad)=>{

if(err){

res.json(err);

}

else{

res.redirect('/'+newAd.\_id);

}

});

* READ:-

db.Ad.find({"category":"BOYS"})

* UPDATE:-

Ad.findOneAndUpdate({\_id: 1},{

$set:{

title: “edited title”,

category: “BOYS”,

total\_capacity: 4,

current\_capacity:2,

description: “wifi: yes”,

price\_per\_person: 5000

}

},

function(err,result){

if(err){res.json(err);}

else{res.json(“updated”);}

})

* DELETE:-

Ad.deleteOne({\_id: 1},function(err,result){

if(err){res.json(err);}

else

{

res.redirect('/');

}

});

**Question 2:**

**POST vs PUT. Why was POST used to update employee data in the Employees sample AJAX application? What happens if you replace POST with PUT? What difference does it make?**

PUT VS PATCH

Both PUT and POST can be used for creating.

Initially, to distinguish between the two, we should ask a question "what are you performing the action to?" Now let’s consider you're designing an API for asking questions. If you want to use POST then you would do a list of questions. If you want to use PUT then you would do that to a particular question.

Since both can be used so now the main question is which one should we use in a RESTful design:

You do not need to support both PUT and POST.

Which is used is left up to you. But just remember to use the right one depending on what object you are referencing in the request.

Some considerations:

Do you name your URL objects you create explicitly, or let the server decide? If you name them then use PUT. If you let the server decide then use POST.PUT is idempotent, so if you PUT an object twice, it has no effect. This is a nice property, so I would use PUT when possible.

You can update or create a resource with PUT with the same object URL. With POST you can have 2 requests coming in at the same time making modifications to a URL, and they may update different parts of the object.

**Question 3:**

**PUT vs PATCH. Can PUT be used for partial updates e.g. in case of updating your name in an online university application form what method should be used? PUT or PATCH or anyone? Why?**

The PUT method ensures that the enclosed entity should be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource, the enclosed entity should be considered as a modified version of the one originally residing on the origin server. Otherwise, if the URI is capable of being defined as a new resource by the requesting user agent, the origin server can create a new resource with that URI.

PUT is considered idempotent. When you PUT a resource, these two assumptions are considered:

1. You are referring to an entity and not to a collection.
2. The entity you are supplying is complete.

PATCH:

The PATCH method requests that a set of changes described in the requested entity be applied to the resource identified by the Request- URI.

The real difference between PUT and PATCH is that the PUT might be used to replace entire entity under specific resource, so one should send the full entity instead of set of attributes as in the case of PATCH.

Patch should be used if only the name is to be changed or updated in an online form because Put is used to replace the whole form.

**Question 4:**

**Where does React lie in comparison to AngularJS? Compare with respect to advantages and disadvantages.**

ReactJS is an open-source, isomorphic JavaScript library but not a framework that is developed by Facebook as seen on Instagram in 2012. It provides the possibility to create apps that are updated from time to time without the need to reload the page. The main purpose is to create high-performance solution that are updated frequently without having to reload the page. This partial update technology is quite appropriate for single page apps developers. It uses one-way data binding with immutable data structures and give us both server-side and client-side rendering to give it a performance edge over competing technologies.

**AngularJS analysis**

Regarding AngularJS, it is very popular and robust framework with open source code for web applications that consist of one HTML-pages with CSS JavaScript named (SPAs). Also, AngularJS is a huge framework which include these three structures: Models, Views and Controllers, as we all known as MVC design pattern. Although views in AngularJS has a very compact shape in terms of handling large amounts of data it loses in Angular vs React fight to the latter.

**Question 5:**

**What is Vue.js? Where does Vue lie in comparison to React and AngularJS? Give examples.**

Evan You is the founder of Vue who created it after working for Google using AngularJS in numerous projects. He later revealed his thought process by saying that "I figured, what if I could just extract the part that I really liked about Angular and build something really lightweight." It was originally released in February 2014.Even though Vue is a relatively new technology, it is already being used in several large-scale projects mostly in China. The websites of Alibaba, Xiaomi, Baidu, and Tencent all use Vue.js as a central frontend technology.

Vue can be used for many different types of applications. Due to its compatibility with other JavaScript libraries and ability to add more complex logic to the existing apps, it can provide a perfect solution for nearly all type of projects. Primarily Vue.js is suitable for the following requirements:

1. Dynamic high-performance applications: Similar to React, Vue.js is a good choice for dynamic applications, however thanks to virtual DOM it offers higher performance, which is beneficial for complex apps.
2. Single page apps: Vue.js allows changing the content quickly without reloading the page, which makes this framework perfect for SPAs.

**Question 6:**

**How is Angular IO different from AngularJS? Give examples.**

AngularJS is a popular JavaScript FrontendFramework for creating web application which was released by **Google** in 2010 and in a short time became quite popular because of its rich features and built-in functionality to create reactive and responsive web apps.

**Problems with AngularJS**

In the years **2012–14** frameworks like **ember**.js and **react**.js (developed by Facebook) popped in with a better benchmark results and performance, hence stressing upon the AngularJS drawbacks to the developer community.

**Inception of the new Angular (Version 2)**

The angular team therefore decided to create a new framework instead of upgrading the old AngularJS by incorporating all their hard-learned lessons from AngularJS. Hence, Angular 2 was released in Sept’ 2016 which is a complete rewrite of AngularJS.

**Confusion because of names and versions**

However, the team decided to continue with the same name except appending ‘JS’ in the end. So, the improved version is called angular and follows a semantic version approach for versioning. According to it, a new major version of angular will be released in every six months (upgraded version, not a rewrite i.e. Angular version 2/4/5 are all same and should be referred as ‘*angular’* only). The current version is 5.

**Question 7:**

**What is Linting? What is the use of JSLint? What is the use of ESLint? Give examples.**

Originally, a particular program that flagged some suspicious and non-portable constructs (likely to be bugs) in C language source code was called lint. However the term is now used generally for tools that flag suspicious usage in software written in any computer language.

Linting code is already an established part of any JavaScript project and, have a lot of benefits such as:

1. Readability

2.Pre-code review

3.Finding (syntax) errors before execution

On the other hand, ESLint is an open source. It is a JavaScript linting utility originally created by Nicholas C. Zakas. Code linting is a type of static analysis that is frequently used to find problematic patterns or code that doesn’t follow certain style guidelines. There are code linters for most programming languages, and compilers can sometimes incorporate linting into the compilation process.

For further explanation, following simple script has been used to generate an AST for the given JavaScript code.

var espree = require('espree');

var fs = require('fs');var code = `let array = [3,4,'z'];var ast = espree.parse(code, {

ecmaVersion: 6

});console.log("writing ast to ast.json")

fs.writeFile("ast.json", JSON.stringify(ast, null, 4), function(err) {

if(err) return err;

});

**Question 8:**

**Give an example where you would prefer to use AngularJS over AJAX and vice versa.**

Ajax is the feature which allows you to update the part of the page without update or refresh the page while AngularJS is one of the JavaScript framework, client side MVC framework.

Even before AngularJS ajax calls were used to update the particular part of a page and now you can use Angular too for Ajax calls. One is a way to attain a certain functionality while other is a big framework which include many other functionalities along with Ajax.

You can think of AJAX as the ability to get data from a server without the need to refresh a webpage.

However, Angular extends this idea with two-way data binding. So the HTML elements on your front and end page are in constant communication with your back end server and vice versa. In addition to this, Angular can achieve other things and is also useful for separating your data, the functions performed on that data, and how the user sees the data.