Web Programming Assignment

Ali Akber

L164081

Q1.

1. **MongoDB vs Mongoose.**

MongoDB is a NoSQL database system which stores data in the form of BSON documents. In terms of Node.js, mongodb is the native driver for interacting with a mongodb instance and mongoose is an Object modeling tool or an Object Document Mapper (ODM) that makes using MongoDB easier by translating documents in a MongoDB database to objects in the program. Mongoose is built upon the MongoDB driver to provide programmers with a way to model their data. Some of the differences are:

1. The first difference between a Mongoose and a native-MongoDB application is that a module containing the schema and model must be created in the models directory.
2. The second major difference, although arguably relative to each developer, is that queries are easier to construct and read in Mongoose than in native-MongoDb.
3. **Why use Mongoose package over MongoDB package**
4. ***Schemas***MongoDB is a denormalized NoSQL database. This makes it inherently schema-less as documents have varying sets of fields with different data types.If one is coming from SQL background, Mongoose defines a schema for data models so documents follow a specific structure with pre-defined data types.
5. ***Validation***Mongoose has built in validation for schema definitions. This saves one from writing a bunch of validation code that you have to otherwise write with the MongoDB driver. By simply including things like “required:true” in your schema definitions, Mongoose provides out-of-the-box validations for your collections (including data types).
6. ***Methods***Mongoose provides optional pre and post save operations for data models. This makes it easy to define hooks and custom functionality on successful reads/writes etc. You can also define custom methods that act on a particular instance (or document). While you can achieve similar functionality with the native MongoDB driver, Mongoose makes it easier to define and organize such methods within your schema definition.
7. ***Returning Results***Mongoose makes returning updated documents or query results easier. A prime example can be found with update queries. While the native driver returns an object with a success flag and the number of documents modified, Mongoose returns the updated object itself so you can easily work with the results.
8. **CRUD operations in MongoDB**
9. *Create*:

Create or insert operations add new documents to a collection. If the collection does not currently exist, insert operations will create the collection.MongoDB provides the following methods to insert documents into a collection:

* db.collection.insertOne()
* db.collection.insertMany()

For example:

db.**student**.insertOne({  
 **regNo**: "3014",  
 **name**: "Test Student",  
 **Course**: “Web development”

)}

1. *Read*:

Read operations retrieves documents from a collection; i.e. queries a collection for documents. MongoDB provides db.collection.find() methods to read documents from a collection.

For example:

db.**students**.*find*({"**regNo**":"*3014*"})

1. *Update:*

Update operations modify existing [documents](https://docs.mongodb.com/manual/core/document/#bson-document-format) in a [collection](https://docs.mongodb.com/manual/core/databases-and-collections/#collections). MongoDB provides the following methods to update documents of a collection:

* [db.collection.updateOne()](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#db.collection.updateOne) *New in version 3.2*
* [db.collection.updateMany()](https://docs.mongodb.com/manual/reference/method/db.collection.updateMany/#db.collection.updateMany) *New in version 3.2*
* [db.collection.replaceOne()](https://docs.mongodb.com/manual/reference/method/db.collection.replaceOne/#db.collection.replaceOne)

Example:

db.**student**.*updateOne*({  
 "**regNo**": "*3014*"   
},  
$**set**:  
{  
 "**name**":"*Viraj*"  
})

1. *Delete:*

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

* db.collection.deleteOne(*)*
* db.collection.deleteMany()

Example:

db.student.deleteOne({“regNo”:”3014”})

Q2.

**POST vs PATCH**

**Post** is used to create document and **Put** is used to create or update document.

The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.

The PUT method requests that the enclosed entity 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 residing on the origin server. If the Request-URI does not point to an existing resource, and that URI is capable of being defined as a new resource by the requesting user agent, the origin server can create the resource with that URI.

We used POST in the the sample AJAX application as AJAX uses xml and we can directly make get, post and head requests. As POST and PUT are quite similar, we used **POST** instead of **PUT** request

Q3.

**PUT vs PATCH.**

PUT handles an entity by replacing the entire entity, while PATCH only updates the fields that were supplied, leaving the others alone. Therefore, it is more preferable to use PATCH instead of PUT in case one only needs to update a single field. PUT cannot be used for partial updates. For instance, in case one needs to update his/her name in university application, one should use PATCH method as the PUT method will require all the fields to be passed whereas by PATCH will only require the name and update it , hence making PATCH method more efficient than PUT method.

Q4.

**React vs AngularJS**

1. **High Scalability**

You can easily gamut Angular. How? All the credit goes to its simple and unique design as well as a solid CLI. While, React is provable that it is reliable and efficient for better workflow. Hence, this framework is very scalable comparatively. And according to Js researches, ReactJs is hits the list when respondents were asked about their satisfaction levels.

1. **Overall Performance**

If we talk about performance, simple recipients in Angular are no more in the picture because these people are called on each service delivery. Thus, it is advisable to optimize BehaviorSubject from ReactJs as it efficiently assists the idea of businesses.

1. **Confidence dose**

From confidence dose I mean dependency injection, which is the heart of debate as it is clashing React model of serviceable programming and stability. However, some sort of dependency injection is inevitable in data binding atmosphere as it assist in detachment where there is no split data coating architecture. More so, Angular serves DI as well and the greatest benefit of the same is it has the capability to include diverse lifecycle for multiple stores.

Well, some of the React models organize some kind of comprehensive app situation that draw to distinct workings, however, it is favorable to the establishment of bugs while vanishing the global situation

1. **DOM Usage**

DOM is the Data Object Model of a web app. Angular uses the browser's DOM, while React uses a virtual DOM which help developers manage an extensive database. By using a virtual DOM, you can change any element very quickly and without needing to render the whole DOM. It drastically changes the performance from good to excellent.

Q5.

Angular and Vue are both frameworks while React is a library to user interfaces(UI). Vue uses easy javascript and html so the learning curve is smaller than angular and react. AngularJS is used to develop Native apps, hybrid apps, web apps and focus on large-scale, feature-rich applications. React is used to develop Single Page and mobile applications and is very suitable for modern web development and native-rendered apps for iOS and Android. Vue is used to develop Advanced SPA and started supporting Native apps and is ideal for this purpose. Lastly, Angular is based on MVC model, while React and Vue are based on Virtual DOM(Document Object Model).

Example:

Javascript for printing hello in vue:

new Vue({

el: '#editor',

data: {

input: '# hello'

},

computed: {

compiledMarkdown: function () {

return marked(this.input, { sanitize: true })

}

},

methods: {

update: \_.debounce(function (e) {

this.input = e.target.value

}, 300)

}

})

Q6.

**AngularIO vs AngularJS**

AngularIO is simply put just a website containing all the documentation needed for Angular framework. Angular.io contains all new updates and latest versions of angular and keeps users up to date with new functions and directives. AngularJS as mentioned in the previous answers is just a javascript framework that makes it easier to design Single page applications. AngularJS makes routing very easy with the use of ng-controller directive that binds each html element to an ng-controller successfully making it extremely easy and highly efficient for MVC apps.

Q7.

**Linting:**

Linting is the process of running a program that tests the code and points out programming errors, bugs, stylistic errors, and suspicious constructs. As coding is 80% debugging, linting seems to be a much needed process of the workflow of an efficient programmer. Now a days, facebook, amazon, twitter etc are using linting for debugging the codebase.

**Use of JSLint:**

JSLint is a static code analysis tool used in software development for checking if JavaScript source code complies with coding rules. It is provided primarily as a web application through jslint.com, but there are also command-line adaptations. JSLint is used for faster and safer coding.

**Use of ESLint:**

ESLint is a tool for identifying and reporting on patterns found in JavaScript code, with the goal of making code more consistent and avoiding bugs. In many ways, it is similar to JSLint and JSHint with a few exceptions. ESLint is completely pluggable, every single rule is a plugin and you can add more at runtime.

Q8.

**Angular and AJAX**

AJAX is basically the underlying logic behind Angular’s responsive design. AJAX and AngularJS both have the same purpose, to perform asynchronous logic and design efficient and fast SPAs.

**Where to use AngularJS instead explicit AJAX methods?**

As AngularJS lets the website have a responsive layout and is highly efficient for making interactive user interfaces, instead of making using AJAX methods explicity through out a website, one should use AngularJS framework for better readability and scalability. If the whole website is to be designed as an SPA ,I would prefer to use AngularJS to AJAX as it makes it easier for me to read and understand routes and makes error handling much more simpler.

**Where to use AJAX methods instead of AngularJS?**

If one is making a website where there needs to be just one instance of SPA, one may use AJAX to Angular. For instance, an auto completion implemented with AJAX saves more memory. However if the whole website is to be designed as an SPA, AngularJS is definitely easier and hence, my preference.