Web Programming Assignment

Question no 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.

Why are we using mongoose?

We are using mongoose because we can define the schema for the documents in a particular collection. Also, It provides a lot of convenience in the creation and management of data in MongoDB. It is easy to query in mongoose. We have to write less code and chances of error are also less.

CRUD operations in MongoDB

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”

)}

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*"})

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*"  
})

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”})

Question no 2:

Post vs Put:

Post is used to create document, 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."

POST was used to update employee data in the Employees sample AJAX application because in ajax we were using xml and through that we can directly make get, post or head request. As two of them are quite similar so we used post in place of put.

Question no 3:

Put vs Patch:

When a client needs to replace an existing Resource entirely, they can use PUT.

A PATCH request on the other hand, is used to make changes to part of the resource at a location. That is, it PATCHES the resource — changing its properties. It is used to make minor updates to resources and it’s not required to be idempotent.

Can PUT be used for partial updates:

No. it is used for updating the row or document in table completely. In case of updating our name in an online university application form, **Patch** should be used because here we are partially updating the information so using Put for this purpose will be inefficient and won’t work in some cases.

Question no 4:

React vs AngularJS:

The primary difference between AngularJS and ReactJS lies in state management. Angular has data-binding bundled in by default, whereas React is generally augmented by Redux to give unidirectional data flow and work with immutable data. Some of the other differences are

* **Scalability:** Angular is easy to scale thanks to its design as well as a powerful CLI. React is testable and therefore scalable compared to other frameworks like Vue.
* **Computed Properties:** As far as performance is concerned, plain getters in Angular are out of the scenario because they get called on each render. It is however possible to use BehaviorSubject from RsJS, as it serves the purpose.
* **Simplicity + Code length:** React is quite easy and simple to understand but it takes quite some time to set up a project in React. Angular on the other hand, is not simple by any means. Its inherent complexity sometimes causes confusion and Angular specific 3rd party libraries and syntax.
* **Model Complexity:** Angular’s performance is sensitive in terms of scope because of copy-n-compare**.** React however gives you the power of choice without the performance penalty.

Question no 5:

Vue.js:

Vue is a progressive framework for building user interfaces. Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

Comparison:

Angular and Vue are frameworks and React is a library to build UI. Vue uses easy javascript and html so it is easier to learn than the other two. AngularJS is used to develop Native apps, hybrid apps, web apps and focus on large-scale, feature-rich applications, React is used to develop SPA and mobile apps and Suitable for modern web development and native-rendered apps for iOS and Android and 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, 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)  
 }  
})

Question no 6:

Angular IO vs AngularJs:

1. First of all, Angular is based on TypeScript while AngularJS is based on JavaScript.
2. AngularJS uses terms of **scope** and **controller**. To scope a variable you can add many variables that will be visible in View as well as in Controller. AngularJS has also a concept of **rootScope**. Variables in rootScope are available on all throughout application. Angular does not have a concept of scope or controllers. Instead of them it uses a hierarchy of components as its main architectural concept. Component is a directive with a template.
3. AngularJS has many directives and every developer can also specify custom new directive. Angular also has standard directives, but they are used in a bit different way. For example: **ng-model** in AngularJS means that you want to create two-way binding. If you want to create one-way binding, you should use **ng-bind**.
4. Angular has some advantages over AngularJS i.e. modularity, dynamic loading and reactive programming.

Question no 7:

Linting:

Linting is the process of running a program that will analyze source code to flag programming errors, bugs, stylistic errors, and suspicious constructs. This is most helpful in identifying some common and uncommon mistakes that are made during coding. Now a days, facebook, amazon, twitter etc are using this in their programmes.

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.

Question no 8;

Angular is a framework. It allows the developer to produce reusable UI components. Thus your development time gets reduced and your source code becomes maintainable to a great extent. But, eventually it’s your AJAX that is doing the work below all these layers (in scenarios where you need to communicate with the server from your Angular code).

We can use ajax when we need to send requests to the server e.g. dynamic injection. On the other hand angular is a complete framework, it can be used for building the whole UI so we can use it wherever we want e.g. automatic completion.