**QUE 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. (8)**

**ANSWER :**

MongoDB is a NoSQL database system which stores data in the form of BSON documents. Your question, however is about the packages for Node.js.In terms of Node.js, mongodb is the **native driver** for interacting with a mongodb instance and mongoose is an **Object modeling tool** for MongoDB.

Mongoose is built upon the MongoDB driver to provide programmers with a way to model their data. Using Mongoose, a user can define the schema for the documents in a particular collection. It provides a lot of convenience in the creation and management of data in MongoDB. On the downside, learning mongoose can take some time, and has some limitations in handling schemas that are quite complex.

However, if your collection schema is unpredictable, or you want a Mongo-shell like experience inside Node.js, then go ahead and use the MongoDB driver. It is the simplest to pick up. The downside here is that you will have to write larger amounts of code for validating the data, and the risk of errors is higher

For performing CRUD we need to follow these steps first.

1.Install MongoDb, NodeJs and dependencies

2.Create Server.

3.Connect to server

4.Create middle-ware route.

CREATE CRUD FUNCTION :

db.inventory.insertMany([

{ item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },

{ item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },

{ item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }

])

Read CRUD FUNCTION :

db.inventory.insertMany( [

{ item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 } ] },

{ item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] },

{ item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse: "B", qty: 15 } ] },

{ item: "planner", instock: [ { warehouse: "A", qty: 40 }, { warehouse: "B", qty: 5 } ] },

{ item: "postcard", instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 } ] }

]);

db.inventory.find( { "instock": { warehouse: "A", qty: 5 } } )

db.inventory.find( { 'instock.qty': { $lte: 20 } } )

Update CRUD FUNCTION

db.collection.updateOne(<filter>, <update>, <option>)

db.collection.updateMany(<filter>, <update>, <options>)

db.collection.replaceOne(<filter>, <update>, <options>)

Delete CRUD

db.inventory.deleteMany({})

**QUE 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? (2)**

**ANSWER :**

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.PUT and POST are both unsafe methods. However, PUT is idempotent, while POST is not.

**QUE 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?**

**ASNWER :**

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

Now lets take a look at a real example. When I do POST to /users with the data {username: 'skwee357', email: 'skwee357@domain.com'} and the server is capable of creating a resource, it will respond with 201 and resource location (lets assume /users/1) and any next call to GET /users/1 will return {id: 1, username: 'skwee357', email: 'skwee357@domain.com'}

**QUE 4 :**

**Where does React lie in comparison to AngularJS? Compare with respect to advantages and disadvantages. (4)**

**ANSWER :**

Angular allows two-way data binding while React allows one-way data binding.

DOM is the Data Object Model of a web app. You can either use a regular DOM or create a virtual DOM.

**Angular** uses the browser's DOM, while **React** uses a virtual DOM.

Angularis a JS framework by nature, but is built to use TypeScript. **React,** on the other hand, is a JavaScript library as well, but recommends using JSX.

This will differ from individual to individual based on skill and experience. On average, TypeScript is considered harder to learn than JSX, in turn increasing the learning curve with Angular as compared to React.

**Que 5 :**

**What is Vue.js? Where does Vue lie in comparison to React and AngularJS? Give examples.(4)**

**ANSWER :**

Vue.js is a **progressive framework** for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

The tool itself is not as powerful as the design patterns and school of thought it promotes. React’s API is minimal, and it really allows you to do whatever you like. — It is just the platform upon which you stand your application, 80% philosophical, 20% “physical” API.

You’re not going to build anything with any of these tools that you could not build yourself. Using these tools offers better structure and more maintainable code, but they do not enable you to do anything that you are not capable of already.

You could use Riot and build a more powerful, performant application than many React applications. You can write better programs in Angular 1 than many people do with the latest React. The answer to your question is yes. Vue is absolutely comparable to React and Angular. Vue is a great tool.

**QUE 6 :**

**How is AngularIO different from AngularJS? Give examples.(4)**

**ANSWER**

First of all, Angular is based on TypeScript while AngularJS is based on JavaScript. TypeScript is a superset of ES6 and it’s backward compatible with ES5. Angular has also benefits of ES6 like: lambda operators, iterators or reflection’s mechanism.   
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. That is a similar approach as in **ReactJS** – another library used for building user interfaces.

The newer version of Angular is a popular solution. Moreover, mostly it will be a better choice. Anyway, before you choose one of them to your new project, some questions should keep in mind :   
What libraries would you like to use?   
Are they compatible with Angular?   
What web browsers we want to support? (this one is really important)  
If you choose only new browsers, then Angular is the best. It’s because it is a forward-looking library, but don’t forget about other browsers. For example: IE8 for which you should definitely use AngularJS from version 1.3 that supports IE8 and higher.

**Q7 :**

**What is Linting? What is the use of JSLint? What is the use of ESLint? Give examples. (4)**

**ANSWER :**

Linting is the process of running a program that will analyse code for potential errors.

**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. It was created in 2002 by Douglas Crockford.

ESLint is an open source JavaScript linting utility originally created by Nicholas C. Zakas in June 2013. Code linting is a type of static analysis that is frequently used to find problematic patterns or code that doesn’t adhere to certain style guidelines. There are code linters for most programming languages, and compilers sometimes incorporate linting into the compilation process.

**Q8 :**

**Give an example where you would prefer to use AngularJS over AJAX and vice versa. (2)**

**ASNWER :**

$http is the service which you can inject in your angularjs controller and make the requests but if you want to use ajax features like async promises and others it will be time consuming for you to learn all that so in that case you can also simply put the ajax post method in the js of controller it will work fine.

**REFERRENCES**

https://dzone.com/articles/differences-between-react-and-angular

<https://www.quora.com/Is-Vue-js-really-so-powerful-that-it-can-be-compared-to-ReactJS-or-Angular>

<https://vuejs.org/v2/guide/>

<https://gorrion.io/blog/angularjs-vs-angular/>

https://stackoverflow.com/questions/8503559/what-is-linting