**Waneed Parkash**

**L12-4097**

**Question# 1**

**MongoDB vs Mongoose**

MongoDb is node package which is native mongoDB driver which is the lowest API for interacting with a mongoDb server from node js.

Mongoose is built upon mongoDb driver to help with a better way to model data.

Mongoose is used to help design the schema for the documents in a particular collection. However, if the schema of collection is unpredictable then MongoDb driver is better option but this will require a large amount of code for validating the data and risk of error is higher.

Another difference is if we want to access to access multiple databases then MongoDb is more helpful whereas in Mongoose various workarounds like other libraries will be needed which have various drawbacks.

**Crud Operations**

**Create Db Entry:**

var mongodb = require('mongodb');

var url = "mongodb://localhost:27017/test";

router.get("/insert\_data", function (req, res) {

mongodb.connect(url, function (err, db) {

var user = { name: "Waneed Parkash", phone: "009231311313" };

db.collection("my-table").insertOne(user, function (err, res) {

if (err) throw err;

console.log("1 document inserted");

db.close();

});

});

});

**Retrieve Db data:**

var mongodb = require('mongodb');

var url = "mongodb://localhost:27017/test";

router.get("/get\_data", function (req, res) {

mongodb.connect(url, function (err, db) {

if (err) throw err;

db.collection("my-table").find({}, function(err, result) {

if (err) throw err;

console.log(result);

db.close();

});

});

**Delete Db data:**

var mongodb = require('mongodb');

var url = "mongodb://localhost:27017/test";

router.get("/delete\_data", function (req, res) {

mongodb.connect(url, function (err, db) {

if (err) throw err;

var myquery = { name: "Waneed Parkash" };

dbo.collection("my-table").deleteOne(myquery, function(err, obj) {

if (err) throw err;

console.log("1 document deleted");

db.close();

});

});

});

**Update db data:**

var mongodb = require('mongodb');

var url = "mongodb://localhost:27017/test";

router.get("/update\_data", function (req, res) {

mongodb.connect(url, function (err, db) {

if (err) throw err;

var myquery = { name: "Waneed Parkash" };

var newValues = { **$set:** { phone: "009231314413" } };

dbo.collection("my-table").updateOne(myquery, newValues, function(err, obj) {

if (err) throw err;

console.log("1 document updated");

db.close();

});

});

});

**Question# 2**

The Post method is used to send a file to a specific URI. It is also expected that the source has a URI to handle that request. It is cacheable. Whereas PUT puts a file or resource at a specific URI, and exactly at that URI. If there's already a file or resource at that URI, PUT replaces that file or resource. If there is no file or resource there, PUT creates one. Put is non-cacheable.

Both PUT and POST can be used for updating data. For PUT method exact URI must be known whereas in POST method the URI identified as the resource that will handle the enclosed entity. That resource might be a data-accepting process, a gateway to some other protocol, or a separate entity that accepts annotations.

So, if POST is used for updating data of employ then it might be gateway to some other protocol or some other Resource to be performing the operation of updation.

**Question# 3**

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.

In case to update name in an online university the PATCH method is to be used because it for PUT method complete data is required for update if whole data is not provided then it might remove the already present values.

**Question# 4**

1. React.Js is library mainly dealing with views while Angular fully fledged framework. Many libraries like Redux, Router etc. are added to React which than turns it to framework.
2. Angular do data binding bundled in do state management. In React Redux or MobX library is used for state management.
3. For injection Angular uses directives via scope. A scope in an Angular world is an object which contains various data controls like directives and controllers. On top of all that is the root scope. Whereas in React data is injected at construct time. When root is created or via a nested view.
4. In Angular event handling is assigned to a various portions of a view via directives. They are particularly used for everything. The ng-click directive is an attribute that can be placed on HTML element that hooks up a click handler that already exists on the local scope. In React most of the event handlers are set on the HTML elements that are being built.

**Question# 5**

**What is Vue?**

Vue is used to front-end framework to build user interfaces using Javascript and Html. It is well suited for making single page applications. It is extremely small framework.

**Angular vs Vue:**

1. Vue is of much less complexity as compared to angular. It is much simpler in term of design and API.
2. The integration part is easy for Vue that angular.
3. Angular uses 2-way data binding between scopes but if we want to connect with asynchronous services we need to have third party developed components to integrate with application. Vue uses on-way data flow between the components which data flow easier and leads to develop non trivial application in less time.
4. Later versions of Angular are based on typescript whereas Vue does not uses typescript.

**Question# 6**

**Difference between Angular.Js and later Angular versions**

1. Angular.Js is also referred as Angular 1 and it is based on javascript while Angular is based on Typescript.
2. Angular uses a term of scope and controller. To scope a variable you can add many variables that will be visible in view as well as controller. Angular does not have a concept of scope or controllers. Instead Angular uses a hierarchy of components as its main architectural concept.
3. Angular is being the later version is much improved version that AngularJs, so it is modular which means much of core functionality was moved to different modules which caused faster core and dynamic loading.
4. Angular has advantage of reactive programming over Angular.Js.

**Question# 7**

**What is Linting?**

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

**Use of EsLint:**

ESLint is an open source project originally created by [Nicholas C. Zakas](http://nczonline.net/) in June 2013. Its goal is to provide a pluggable linting utility for JavaScript. It requires a Node.Js and works on window, mac, linux

**Examples:**

1. Identifies whether a variable out of scope is used. If a global is variable is declared in a function or in some local scope it gives warning.
2. If a library is not installed in node module and it is imported that Eslint gives warning.

**Question# 8**

AngularJS is a full fledged, front end MVC framework which does a lot more. It extends the above ($http module) with a lot of neat features such as 2-way data binding, templating, filters and directives etc.

Let’s consider a scenario where a user at screen A sends a request and before we were able to process the request the user moves to screen and we want to cancel the request from front-end which is not possible using Ajax but with angular it is possible with **ngOnDestroy**, which destroys the life cycle of component.

**References**

1. <https://stackoverflow.com/questions/107390/whats-the-difference-between-a-post-and-a-put-http-request>
2. <https://stackoverflow.com/questions/630453/put-vs-post-in-rest>
3. <https://gorrion.io/blog/angularjs-vs-angular>
4. <https://eslint.org/>