# **WEB PROGRAMMING RESEARCH ASSIGMENT**

**QUESTION 1**

Whereas MongoDB is a document-based management system (Database) which leverages BSON (binary JSON) to achieve high throughput, MongooseJS is an [Object Document Mapper](https://dzone.com/articles/era-object-document-mapping) (ODM) that makes using MongoDB easier by translating documents in a MongoDB database to objects in the program.

The three main advantages of using Mongoose versus MongoDB are:

1. MongooseJS provides an abstraction layer on top of MongoDB that eliminates the need to use named collections.
2. Models in Mongoose perform the bulk of the work of establishing up default values for document properties and validating data.
3. Functions may be attached to Models in MongooseJS. This allows for seamless incorporation of new functionality.
4. Queries use function chaining rather than embedded mnemonics which result in code that is more flexible and readable, therefore more maintainable as well.
5. MongoDB is governed by Affero General Public License (AGPL) license, and if we link it with our proprietary code, then we must release our entire source code in the public domain, while in Mongoose, there is no such restriction.

The net result of these is the simplification of database access from applications. Mongoose provides a lot of convenience in the creation and management of data in MongoDB.

**CRUD Operations:**

**Create:**

db.**collection\_name**.*insert*()

**Example:**

db.**student.***insert*({

**RollNo**: "*4388*",

**name**: "*Mehar Fatima*",

**course**: {

**courseName**: "*Web Programming*",

**duration**: "*6 Months*"

},

**address**: {

**city**: "*Lahore*",

**state**: "*Punjab*",

**country**: "*Pakistan*"

}

})

**Read:**

db.**collection\_name**.*find*() //Returns everything there is in the collection

db.**collection\_name**.*find*({"**fieldname**":"*value*"}) //Returns the record where attribute is equal to value

**Example:**

db.student.*find*()

db.**students**.*find*({"**RollNo**":"*4388*"})

**Update:**

db.**collection\_name**.*update*()

**Example:**

db.**student**.*update*({

"**RollNo**": "*4388*"

},

$**set**:

{

"**name**":"*Fatima Khan*"

})

**Delete:**

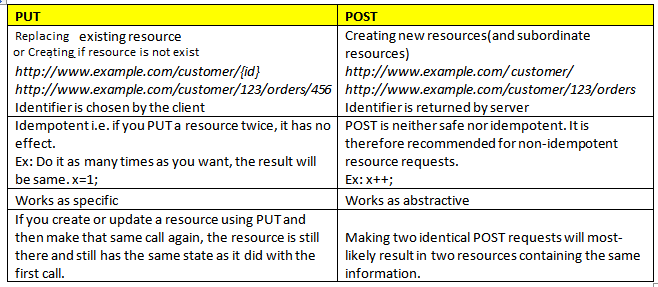
db.**collection\_name**.*remove*({"**fieldname**":"*value*"})

**Example:**

db.**student**.*remove*({"**RollNo**":"*4388*"})

**QUESTION 2:**

Both are used for data transmission between client to server, but there are subtle differences between them, which are:

[](https://i.stack.imgur.com/8D8Z3.png)

In order to update data of employees in AJAX application, we use POST method because AJAX applications use XML Requests which make use of GET and POST method.

**QUESTION 3:**

PUT is considered idempotent. When we PUT a resource, these two assumptions are in play:

1. We are referring to an entity, not to a collection.
2. The entity we are supplying is complete (the *entire* entity).

Examples:

*Entity:*

{ "name": "Mehr Fatima", "Roll No": "16L4388" }

If we POST this document, then we get back an entity such as

## /applicant/16L4388

{

"name": "Mehr Fatima",

"Roll No": "16L4388"

}

If we want to modify this entity later, we choose between PUT and PATCH. A PUT looks like this:

PUT / applicant/16L4388

{

"name": "Mehar Fatima", //updated name

"Roll No": "16L4388"

}

We can accomplish the same using PATCH. That looks like this:

PATCH / applicant/16L4388

{

"name": "Mehar Fatima", //updated name

}

The difference is apparent now. The PUT included all the parameters on the applicant, but PATCH only included the one that was being modified (name).

When using PUT, it is assumed that we are sending the complete entity, and that complete entity *replaces* any existing entity at that URI. In the above example, the PUT and PATCH accomplish the same goal: they both change this applicant's name. But PUT handles it by replacing the entire entity, while PATCH only updates the fields that were supplied, leaving the others alone.

Since PUT requests include the entire entity, if we issue the same request repeatedly, it should always have the same outcome (the data we sent is now the entire data of the entity). Therefore, PUT is idempotent.

\*For online university application, PATCH should be used, because it does not require us to enter the complete entity. We will only have to specify the name in order to update it.

**QUESTION 4:**

**About React:** React was released in 2013 by Facebook. It supports JavaScript languages and renders both client and server-side data. Its architecture is component-based and follows a one-way data binding technology. Its DOM is virtual however its learning curve is relatively low.

**Advantages of Reactjs**:

* React allows you to use HTML codes as it comes forward with JSX. You can practice HTML tags and syntax for rendering subcomponents.
* It offers the ability to compound the components of the app in a single time file, and it also promoted the development of machine-readable programs.
* React comes with an ideal setup for app developers and it had led to many developers using React Native for application development.
* React has a prompt rendering feature that gives it a slight edge over the Angular JavaScript. It consists of various approaches to lessen the amount of DOM operation and thereby speeds up the updating process, making it more efficient.
* React has a Virtual DOM which can help developers manage an extensive database.
* React is a purely JavaScript based library product. A primary difference is that Angular is a subset of HTML and React is not.
* React is a choice that you make when you are looking for reliable, intensive and straightforward programming. It is a relatively more advanced language than Angular.

**Disadvantages of Reactjs**

* A traditional MVC framework like Rail needs configuration and integrating Reactjs into it slows down the development time and process.
* If you combine the advantages and disadvantages for React, you will see that the framework might be suitable for specific applications and not work for the rest. Every app is different.

**When to use Reactjs?**

When there is a lot of dynamic content in your application that is when React would be the right choice. Many popular brands like Instagram and Facebook prefer to use ReactJs to base their mobile apps on because of its dynamic nature.

Sometimes, adopting React may make sense over using Angular and vice versa. Here are a few brands which have used Reactjs: Uber, Dropbox, Netflix, Instagram, PayPal, and Flipkart.

**About Angular:** Angular was released in 2010 by Google. It supports JavaScript languages and renders both server and client-side data. Its architecture is component-based, and it is usually used for templating Typescript files. It follows a two-way data binding technology, and it works on a real DOM and has a steep learning curve compared to React is relatively low.

**Advantages of Angularjs**

* Angular is known for its community and developer support services. Angularjs has brought in a rise in demand for its framework because of its support factors.
* The Angular framework runs on all browser environments regardless of its platforms.
* Angular has established itself as a reliable framework because it includes off-the-rack tools and it also has robust components which are evolved in Juxtapose.
* Angular comes with a Bi-directional data binding feature that is, in fact, the primary difference between Reactjs and Angularjs. It disperses influence after every set of data change.

**Disadvantages of Angularjs:**

* Angular does not consist of extensive, all-inclusive documentation or a clear manual.
* It also has a steep learning curve which is one of its main drawbacks.

**When to use Angularjs?**

When you require your app to be compatible with all the browsers that is the best time you use Angular. It even helps your app consistent with browsers older than the IE8. There has also been a recent release of the Angular 5.0 in the market and it has undoubtedly attracted a lot of attention from developers especially with its performance boost features and server-side rending support services. If your mobile application lies between the complexities of low to medium, Angularjs is the best way to go.

**QUESTION 5:**

Here is a comparison between Angular, React and Vue:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Angular** | **React** | **Vue** |
| **Type** | A Framework | Library to build UI | A library |
| **Why Choose** | If you want to use TypeScript | If you want to go for “everything-is-JavaScript” approach | Easy JavaScript and HTML |
| **Founders** | Powered by Google | Maintained by Facebook | Created by Former Google  Employee |
| **Initial Release** | September 2016 | March 2013 | February 2014 |
| **Application Types** | If you want to develop Native apps, hybrid apps, and web apps | If you want to develop SPA and mobile apps | Advanced SPA and started  supporting Native apps |
| **Ideal for** | If you want to focus on large-scale, feature-rich applications | Suitable for modern web development and native-rendered apps for iOS and Android | Ideal for web development and  single-page applications |
| **Learning Curve** | A steep learning curve | A little bit easier than Angular | A small learning curve |
| **Developer-friendly** | If you want to use the structure-based framework | If you want to have flexibility in the development environment | If you want to have separation of concerns |
| **Model** | Based on MVC (Model-View-Controller) architecture | Based on Virtual DOM (Document Object Model) | Based on Virtual DOM (Document Object Model) |
| **Written in** | TypeScript | JavaScript | JavaScript |
| **Community Support** | A large community of developers and supporters | Facebook developers community | Open-source project sponsored through crowd-sourcing |
| **Language Preference** | Recommends the use of TypeScript | Recommends the use of JSX – JavaScript XML | HTML templates and JavaScript |
| **Popularity** | Widely popular among developers | More than 27,000 stars added over the year | More than 40,000 stars added on GitHub during the year |
| **Companies Using** | Used by Google, Forbes, Wix, and weather.com | Used by Facebook, Uber, Netflix, Twitter, Reddit, Paypal, Walmart,  and others | Used by Alibaba, Baidu, GitLab,  and others |

Below is a code comparison between React and Vue:



**QUESTION 6:**

AngularJS:

A JavaScript-based open-source front-end web application framework led by the Angular Team at Google and by a community.

* v1.x.x is still maintained which has its own differences and new features such as from controllers to components and so on.
* Stable version of Angular JS is v1.5.6.

Angular2+:

A complete rewrite from the same team that built AngularJS. Typescript -based open-source front-end web application framework and support Dart as well.

* v2.x.x is more focusing on TypeScript instead of JavaScript alternatively Dart, too.
* Major break changes in Angular as it is ground-up rewrite.
* New style guide, new project structure (component-based architecture).
* Following semantic versioning which will bring new version of Angular in every six months tentatively.
* Latest/Upcoming angular versions i.e. v4.x.x, v5.x.x, v6.x.x and so on are/will be having backward compatibility with v2.x.x after very minor changes.
* These latest versions are the enhancements as well as new extended feature in angular framework.
* If one knows Angular v2.x.x or above will easily be comfortable with latest angular version as it is not rewrite but further enhancement.

**A few differences between AngularJS(v1.x.x) and Angular (v2.x.x or above):**

* JS to TS
* ng-app to bootstrap
* Controller to Component
* Filter to pipes
* service/provider/factory to service

**QUESTION 7:**

**Linting** is the process of checking the source code for Programmatic as well as Stylistic errors. This is most helpful in identifying some common and uncommon mistakes that are made during coding.

A lint or a linter is a program that supports linting (verifying code quality). They are available for most languages like JavaScript, CSS, HTML, Python, etc.

Some of the useful linters are [JSLint](http://www.jslint.com/), [CSSLint](http://csslint.net/), [ESLint](http://jshint.com/), [Pylint](http://www.pylint.org/) etc.

**JSLint** is a static code analysis tool used software development for checkin 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 based on the premise of strict mode, which is available in the fifth edition of the ECMAScript standard. With strict mode, you are making your code run with a set of more restrictive rules than normal. JSLint is the oldest linter there is.

**ESLint** is open source and started in 2013. The primary reason ESLint was created was to allow developers to create their own linting rules. ESLint is designed to have all rules completely pluggable. It ships with a basic set of rules to get started with, then it’s up to you what rules you want to change or implement. This is the linter for the person who needs to add and change linting rules. It’s a tool for identifying and reporting on patterns found in JavaScript code, with the goal of making code more consistent and avoiding bugs.

EXAMPLE OF A LINTER (Left is the original code & Right is the linter code):

#### JSLint

**QUESTION 8:**

AJAX stands for Asynchronous JavaScript and XML, and is a way that a webpage can use JavaScript to send and receive data from a server without refreshing a webpage.

A sample AJAC call might work like this:

* Client requests page from server
* Server responds to request and sends page
* Client makes AJAX call to the server and requests more data
* Server sends that data
* Client updates the page using that data without refreshing

The Asynchronous part refers to the fact that when the JavaScript makes the AJAX call to the webserver, it continues to work until the response; it doesn’t “block” and stop while the data is being processed server-side.

One great application of AJAX is that when a user is signing up for a website, AJAX calls immediately tell the user whether the username is available or not. This way, the user doesn’t have to wait for the whole page to reload.

AngularJS is a full frontend MVC framework for JavaScript web applications. It was built at Google and provides a wat to quickly build large single-page applications. Angular contains methods that make using AJAX easier.

The AngularJS $http service makes a request to the server and returns a response, thereby making use of AJAX easier:

<div ng-app="myApp" ng-controller="myCtrl">   
  
<p>Today's welcome message is:</p>  
<h1>{{myWelcome}}</h1>  
  
</div>  
  
<script>

var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope, $http) {  
    $http.get("welcome.htm")  
    .then(function(response) {  
        $scope.myWelcome = response.data;  
    });  
});

</script>