**NAME: AFRASIYAB  
ROLL\_NO: L16-4285**

**SECTION-A**

**WEB PROGRAMMING RESEARCH ASSIGNMENT**

**Question1**

MongoDB is an [open source](https://whatis.techtarget.com/definition/open-source) database management system (DBMS) that uses a document-oriented database model which supports various forms of data. Instead of using [tables](https://whatis.techtarget.com/definition/table) and [rows](https://searchoracle.techtarget.com/definition/row) as in [relational databases](https://searchdatamanagement.techtarget.com/definition/relational-database), the MongoDB architecture is made up of collections and documents. A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to [JavaScript Object Notation](https://searchwindevelopment.techtarget.com/definition/JSON-Javascript-Object-Notation) objects but use a variant called Binary JSON (BSON) that accommodates more data types. However, Mongoose is built on top of the Mongoose Embedded Library which can be used for the implementation of [RESTful services](https://en.wikipedia.org/wiki/Representational_state_transfer" \o "Representational state transfer), to serve Web GUI on embedded devices, create [RPC](https://en.wikipedia.org/wiki/Remote_procedure_call) frameworks (e.g. [JSON-RPC](https://en.wikipedia.org/wiki/JSON-RPC)). Mongoose is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application that can be used on Windows, Macintosh OS, Linux, Free RTOS, Android and iOS. Mongoose contains many different functions that allow you to validate, save, delete, and query your data using common MongoDB functions.

**CREATE:**

Create is the insertion of new data in the collection.

db.user.insert({

cell\_no: "03214023321",

name: "Bilal Ashfaq",

email: “someone@x.com”

address: {

city: "Lahore",

state: "Punjab",

country: "Pakistan"

}

})

**READ:**  
Read is used to get data stored in the collection.

db.users.find({})

db.users.find({cell\_no: “03214023321”})

**UPDATE:**

UPDATE is used to update the previous data stored in the collections.

db.user.update({ "email": "someone@x.com" },

$set:

{"name":"Bilal Zameer"})

**DELETE:**

DELETE is used to remove the existing data, it takes some specified query otherwise it deletes all data.

db.users.remove({"email":" someone@x.com "})

**Question2**

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.

**Question3**

Responses to PUT requests are not cacheable. If a PUT request finds a response in a cache infrastructure, that response (cache entry) should be treated as stale. However, 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. Since PATCH is not idempotent, failed requests are not automatically re-attempted on the network. Also, if a PATCH request is made to a non-existent url e.g attempting to replace the front door of a non-existent building, it should simply fail without creating a new resource unlike PUT, which would create a new one using the payload.

Patch should be used if only the name is to be changed or updated in an online form rather than PUT which replaces the whole form.

**Question4**

React and AngularJS are both advanced, widely adopted JavaScript (JS) technologies that we use to create interactive single-page applications (SPAs). The number of JS tools for fast single-page application development is constantly growing, making the choice of which technology to rely on more challenging for us as web developers. React is even more widely used by JavaScript programmers, although it’s actually a library, not a framework: the React library only has a View, but lacks Model and Controller components. AngularJS is monolithic frameworks which include these three paradigms: Models, Views and Controllers, as we all known as MVC design pattern. Both React and Angular work on completely diverse approaches to front-end application development for startup, small and medium enterprises. The technologies are both powerful and flexible and while neither is better or worse, it depends on your business application goals and system constraints that make the final choice.

**Question5**

Vue is a **progressive framework** for building user interfaces. Unlike other monolithic frameworks, 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. Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

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)  
 }  
})

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).

**Question6**

Contrasts among AngularJS and AngularIO:

Precise JS is an open source JS-based, frontend web application structure for dynamic web application improvement though, AngularIO (rakish 2 and precise 4) are open-source, Typescript based frontend web application stages.

The design of angularJS depends on MVC structure i.e. Models, Views and Controllers. This in angularIO, was first supplanted by controllers and orders, which were later on moved forward

Precise JS is written in JavaScript. Rakish utilizations Microsoft's TypeScript dialect, which is a superset of ECMAScript 6 (ES6). This has the consolidated focal points of the TypeScript highlights, similar to type presentations, and the advantages of ES6, as iterators and lambdas. Rakish 4 is perfect with the latest variants of TypeScript that have ground-breaking type checking and question situated highlights.

By giving highlights like 2-way authoritative, AngularJS decreased the advancement exertion and time. Be that as it may, by making additionally handling on the customer side, page stack was taking impressive time. Angular2 gives a superior structure to all the more effectively make and keep up enormous applications and a superior change discovery system. Precise 4 is the quickest form yet.

**Question7**

**LINTING:**

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

**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.

Example: Tolerate conversion operators

**For example:**

/\*global

ADSAFE, report, jslint

\*/

instructs JSLint to not give warnings about the global variables ADsafe, report, and jslint. However, if any of those names are expected to be supplied by other files and those other files fail to do so, then execution errors will result. It is usually better to use top-level variable declarations instead:

var ADSAFE;

var report;

var jslint;

**ESLINT:**

**ESLint** is designed to be completely configurable, **meaning** you can turn off every rule and run only with basic syntax validation, or mix and match the bundled rules and your custom rules to make **ESLint** perfect for your project.

**Question8**

The fundamental motivation behind Ajax is to enhance the speed, execution and ease of use of a web application. An incredible case of Ajax is the motion picture rating highlight on Netflix. The client rates a film and their own rating for that motion picture will be spared to their database without trusting that the page will revive or reload. These motion picture evaluations are being spared to their database without posting the whole page back to the server. Facebook, Gmail, and Pinterest are instances of locales that utilization a great deal of AJAX.

The "Nonconcurrent" part alludes to the way that when the JavaScript makes the AJAX call to the webserver, it keeps on working until the reaction – it doesn't "square" and stop while the information is being prepared server-side.

Angularjs is best utilized for single page applications which give the customer a liquid ordeal since all the code is recovered in advance or progressively stacked as important to make an affair that feels like a work area application than a customary, multi-page site