****

**Name:**  Zarfishan Wajid

**Roll No.** L164241

**Section:** A

**Course:** Web Programming

Assignment 2

**Question 1:**

**MongoDB v/s Mongoose.**

Mongodb and Mongoose are two different drivers to interact with MongoDB database.

**MongoDB**

**MongoDB**is an Open Source, NoSQL database management system. It is a document based DBMS which leverages a JSON-style storage format known as **binary JSON** or BSON, to achieve high throughput. BSON makes it easy for applications to extract and manipulate data, as well as allowing properties to be efficiently indexed, mapped, and nested in support of complex query operations and expressions. It is a native driver in Node.js to interact with MongoDB.

**Mongoose**

**Mongoose** is an object data modeling (ODM) library that provides a rigorous modeling environment for your data. Used to interact with MongoDB, it makes life easier by providing convenience in managing data.

|  |  |  |
| --- | --- | --- |
|  | **Mongoose** | **Mongodb native** |
| **Object mapping** | ORM | ODM |
| **Schema** | Mandatory | Not necessary |
| **Performance / processing time** | Not bad | Excellent |
| **Development time** | Fast | Average |
| **Default promise** | No | No |
| **Maintainability** | Easy | Little hard |
| **Learning curve** | Little high | Low |
| **Community** | Good | Good |

**Why are we using Mongoose package instead of MongoDB package?**

  Because we want an object modeling tool to skip some lower level work. Mongoose will make code faster with simple apps with simple db structure. Mongoose allows to read mongodb docs and mongoose docs**. Mongoose** makes returning updated documents or query results easier, provides optional pre and post save operations for data models and allows to create schemas for our data model.

**How would we perform CRUD operations using the MongoDB package? Give code examples for creating, retrieving, updating and deleting MongoDB documents.**

**Create operation:** Create operation or Insert operation are used to add new documents to the collection and if the collection does not exist, it creates one. The command to insert a document on the collection – **db.collection.insert()**

**FOR EXAMPLE:**

db.students.insertOne({

Name:”Zarfishan Wajid”,

RollNo: 4241

})

**Read operation:** This operation reads the documents from the collection. This process is taken place by executing a query.

The command to read the document is – **db.collection.find()**

**FOR EXAMPLE:**

db.students.find({

{Name:”Zarfishan Wajid”},

{RollNo: 4241}

})

**Update operation:** Update operation is used to modify an existing document.

The command that updates a document is – **db.collection.update()**

**db.collection.updateOne()**

**db.collection.updateMany()**

**db.collection.replaceOne()**

**FOR EXAMPLE:**

db.students.update({

{Name:”Zarfishan Wajid”},

{$set: {RollNo: 1234}}

})

**Delete operation:** Delete operation erases the document from the collection.

Following commands performs the operations –

**db.collection.remove()**

**db.collection.DeleteOne()**

**db.collection. DeleteMany()**

**FOR EXAMPLE:**

db.students.remove({

RollNo: 4241

})

db.students. DeleteMany ({

RollNo: 4241

})

**Question 2:**

**POST vs PUT**

The **PUT method** is used to **create or overwrite a resource at a particular URL that is known by the client.** While **POST method should be used to create a subordinate**(or child) of the resource identified by the Request-URI and update it.

Choosing between using PUT vs POST should be based on the action’s Idempotence. *Idempotence is the property of certain operations in mathematics and computer science that can be applied multiple times****without changing the result****beyond the initial application.*

**PUT method is idempotent** because no matter how many times we send the same request, the results will always be the same. On the other hand, the **POST method is not idempotent** since if we send the same POST request multiple times, we will receive various results (i.e. a new subordinate will be created each time).

**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?**

POST method is used to update employee data in the Employees sample AJAX application because post method is not idempotent we can update employee multiple times. But if we use PUT method instead of POST method than all updates requests will give only one result that is not required so we use POST method.

**Question 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?**

HTTP **PUT** method only allows a complete replacement of a document While 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.

In case of updating name in an online university application form we should use PATCH method because we only need to update one field of the resource. **PUT**ting a complete resource representation will utilize more bandwidth and might be cumbersome. So here **PATCH** method is a good choice.

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

**Angular:**

[Angular](https://angular.io/) is a Type Script-based, open-source front-end web application platform. It is an MVC framework created by Google. MVC is an architecture type for frameworks that develop user interfaces. It translates to **M**odel **V**iew **C**ontroller.

**React:**

[React](https://reactjs.org/) is a JavaScript library for building user interfaces. Created by Facebook, Instagram, and the community, React is evolving at an impressively increasing rate. Outside of the pure React library itself, there are 3 distinct versions of React: React-devtools, ReactJS.Net and React Native.

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. Both are opposing approaches and there is no consensus on which is better: mutable/data binding or immutable/unidirectional.

* **Scalability:**  
  Angular is easy to scale because of its design as well as a powerful CLI. React is testable and therefore scalable compared to other frameworks.
* **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.  
  React allows the use of @computed from MobX – achieving the same objective but with a nicer API. Dependency Injection.
* **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. You cannot use large models. However, this has benefits and disadvantages. The Pros are that it makes the code simpler and more testable but the cons are that you need to break down stuff that you normally use and rebuild it again (for example – for server requests). React however gives you the power of choice without the performance penalty. The outcome really depends on whether you’re a good coder or a bad coder.

**Conclusion:**

Both React and Angular work on completely diverse approaches to [front-end application development](https://www.rishabhsoft.com/front-end-development-services) for startup, small and medium enterprises. The technologies are both powerful and flexible and while neither is better or worse, it depends on business application goals and system constraints that make the final choice.

Overall, both frameworks provide a robust set of tools for quality, scalable, reactive [web-based applications](https://www.rishabhsoft.com/web-application). For those who prefer to code in classic JS, React may find more favors, but for those looking at a more mature and sophisticated solution, AngularJS might be your best bet.

**Question 5: What is Vue.js? Where does Vue lie in comparison to React and AngularJS? Give examples**

**Vue.js** is an open-source JavaScript framework for building user interfaces and single-page applications. Vue’s focus is on doing two things well: ease of use and rendering speed.

It excels in both. The library is extremely easy to learn and use, and beats both AngularJS and React when it comes to page rendering. As such, VueJS feels almost like Angular’s fitter little brother as it shares the aspects done well, but at the same time is much lighter, more performant and less opinionated than AngularJS.

Vue.js has gained a lot of popularity due to its gentle learning curve, and because of its scalability. It can be used in single page applications as well as small components in large websites.

* If you like **flexibility**more than other features, use **React**.
* If you love **coding in TypeScript**, go for **Angular**.
* If you are a **JavaScript lover**, use **React**because it is all about JavaScript.
* If you are a**fan of clean code**, use **Vue**in your application.
* Vue provides the easiest learning curve and it’s an ideal option for beginners.
* If you **want separation of concerns** in your application, use **Vue**.
* If you are a fond of **object-oriented programming**, **Angular** is definitely the pick for you.
* Vue is ideal for a small team and a small project. If your app seems to be large and has significant future expansion plan, pick React or Angular.
* **For cross-platform app development**, **React Native is an ideal choice** as it provides modern functions and you can easily find resources. Angular, on the other side, needs a sound knowledge of JavaScript to build large-scale applications.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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 |

**Example:**

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 6:** **How is AngularIO different from AngularJS? Give examples.**

**Different Versions of Angular**

* AngularJS is an open-source, JavaScript-based, front-end web application framework for dynamic web app development. It utilizes HTML as a template language. By extending HTML attributes with directives and binding data to HTML with expressions, AngularJS creates an environment that is readable, extraordinarily expressive and quick to develop.
* Angular is the blanket term used to refer to Angular 2, Angular 4 and all other versions that come after AngularJS. Both Angular 2 and 4 are open-source, TypeScript based front-end web application platforms.
* Angular 4 is the latest version of Angular. Although Angular 2 was a complete rewrite of AngularJS, there are no major differences between Angular 2 and Angular 4. Angular 4 is only an improvement and is backward compatible with Angular 2.

**How Are They Different?**

**1. Architecture**

* **AngularJS**

The architecture of AngularJS is based on model-view-controller (MVC) design. The model is the central component that expresses the application's behavior and manages its data, logic, and rules. The view generates an output based on the information in the model. The controller accepts input, converts it into commands and sends the commands to the model and the view.

* **Angular**

In **Angular 2**, controllers and $scope were replaced by components and directives. Components are directives with a template. They deal with a view of the application and logic on the page.

In **Angular 4**, the structural derivatives ng-If and ng-For have been improved, and you can use if/else design syntax in your templates.

**2. Language**

* **AngularJS**

AngularJS is written in JavaScript.

* **Angular**

Angular uses Microsoft’s TypeScript language.

* **Angular 4**

It is compatible with the most recent versions of TypeScript that have powerful type checking and object-oriented features.

**3. Expression Syntax**

* **AngularJS**

To bind an image/property or an event with AngularJS, you have to remember the right ngdirective.

* **Angular**

Angular focuses on “( )” for event binding and “[ ]” for property binding.

**4. Mobile Support**

**AngularJS** was not built with mobile support in mind, but **Angular 2 and 4** both feature mobile support.

**5. Routing**

**AngularJS** uses $routeprovider.when() to configure routing while **Angular** uses @RouteConfig{(…)}.

**Performance**

AngularJS was originally developed for designers, not developers. Although there were a few evolutionary improvements in its design, they were not enough to fulfill developer requirements. The later versions, Angular 2 and Angular 4, have been upgraded to provide an overall improvement in performance, especially in speed and dependency injection.

**1. Speed**

By providing features like 2-way binding, AngularJS reduced the development effort and time. However, by creating more processing on the client side, page load was taking considerable time. Angular2 provides a better structure to more easily create and maintain big applications and a better change detection mechanism. Angular 4 is the fastest version yet.

**2. Dependency injection**

Angular implements unidirectional tree-based change detection and uses Hierarchical Dependency Injection system. This significantly boosts performance for the framework.

## **Advantages and Disadvantages**

Because they are Google products, all Angular versions are trustworthy and enjoy great support from Google engineers and the large community of Angular users and developers. However, each version has its own advantages and disadvantages.

**1. AngularJS**

**Advantages**

* It is unit testing ready.
* It has great MVC data binding makes app development fast.
* Using HTML as a declarative language makes it very intuitive.
* It is a comprehensive solution for rapid front-end development since it does not need any other frameworks or plugins.
* AngularJS apps can run on every significant program and advanced cells including iOS and Android-based phones and tablets.

**Disadvantages**

* It is big and complicated due to the multiple ways of doing the same thing.
* Implementations scale poorly.
* If a user of an AngularJS application disables JavaScript, nothing but the basic page is visible.
* There’s a lagging UI if there are more than 200 watchers.

**2. Angular 2**

**Advantages**

* TypeScript allows code optimization using the OOPS concept.
* It is mobile-oriented.
* It has improved dependency injection and modularity.
* It provides more choice for languages such as Dart, TypeScript, ES5, and ES6 for writing codes.
* It offers simpler routing.

**Disadvantages**

* It is more complicated to set up compared to AngularJS.
* It’s inefficient if you only need to create simple, small web apps.

**3. Angular 4**

**Advantages**

* It enables a fast development process.
* It’s ideal for single-page web applications with an extended interface.
* Full TypeScript support helps in building bulky applications.
* Tests are easy to write.
* An improved View Engine generates less code in AOT mode.
* It has a modularized animation package.

**Disadvantages**

* It’s slow when displaying enormous amounts of data.

**Conclusion:**

Obviously, AngularJS is still useful or else everyone would have migrated to Angular 2 or 4 by now. Each version of Angular has significant benefits, but there is much to gain in being up-to-date with the latest version. Angular is decidedly faster than AngularJS, has a mobile-driven approach, executes better with components, and enables smoother migration from earlier versions.

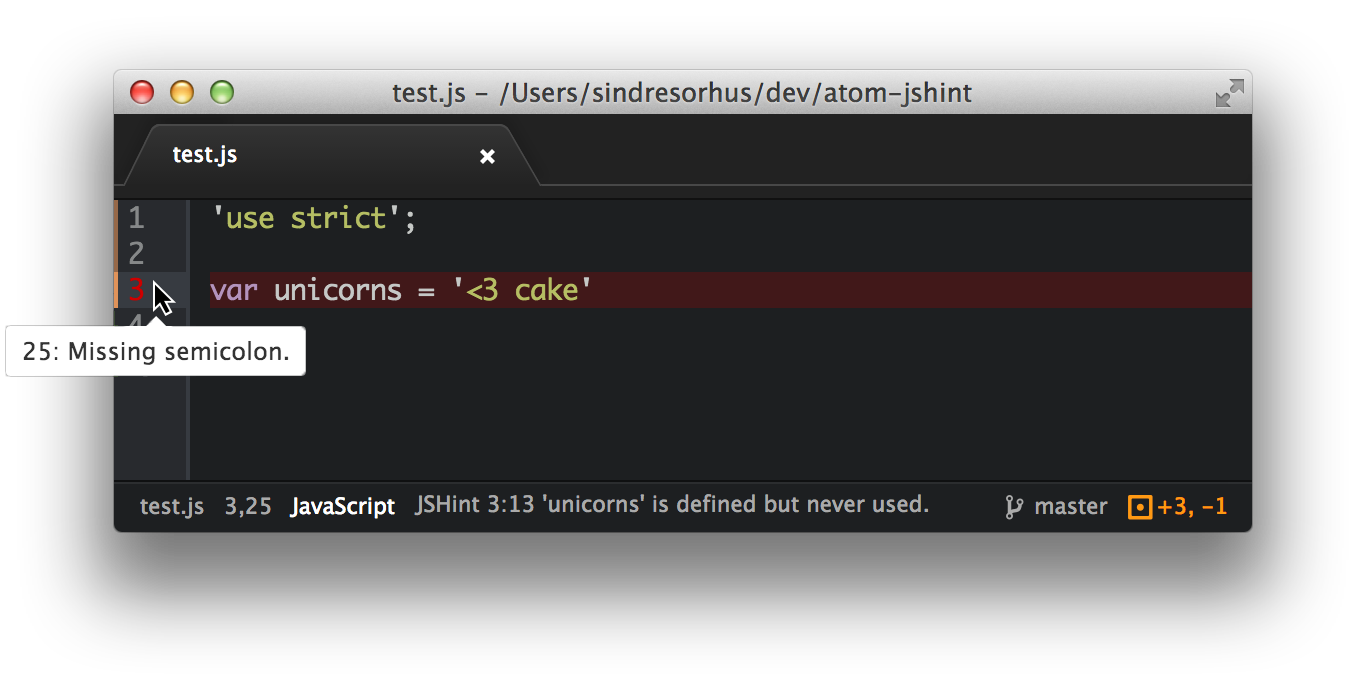
**Question 7: What is Linting? What is the use of JSLint? What is the use of ESLint? Give examples.**

**Linting**:

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

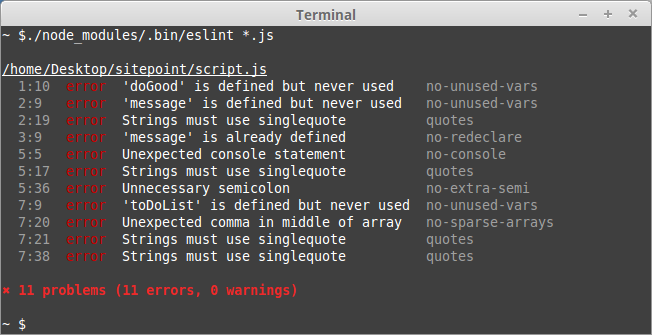
**JSLint:**

It 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:**

It is a pluggable and configurable linter tool for identifying and reporting on patterns in JavaScript. Maintain your code quality with ease.



**Question 8: Give an example where you would prefer to use AngularJS over AJAX and vice versa.**

**AngularJS** is a framework for developing Frontend MVC application whereas **Ajax** is a JavaScript method to get data from urls without reloading the page. ... As it evolved, **Ajax** came to mean any asynchronous request sent to a server from a JavaScript application. Typically the response is JSON, or HTML fragments.

**Ajax** only lets you make HTTP calls to server while Angular is complete framework including Ajax. If you just want to add HTTP requests capability to your app **just use Ajax**. If you want to build complete Single Page Application **use Angular** (or other web frameworks).