**Web Programming Assignment**

**Q1.**

**Mongoose vs. Mongo DB:**

Mongoose is an Object Document Mapper (ODM) that makes using MongoDB easier by translating documents in a MongoDB database to objects in the program. It is used to abstract and simplify some of the boilerplate in interaction with MongoDB, exposed by its native API, including object-modeling and introducing some level of schematic coherence. While on the other hand MongoDB is an Open Source, NoSQL database management system 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.

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

The main advantages of using Mongoose versus native MongoDB are:

* MongooseJS provides an abstraction layer on top of MongoDB that eliminates the need to use named collections.
* Models in Mongoose perform the bulk of the work of establishing up default values for document properties and validating data.
* Functions may be attached to Models in MongooseJS. This allows for seamless incorporation of new functionality.
* Queries use function chaining rather than embedded mnemonics which result in code that is more flexible and readable, therefore more maintainable as well.

The table given below gives a precise comparison between the two:

|  |  |  |
| --- | --- | --- |
|  | 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 |

**CRUD Operations Examples:**

db.users.insertOne({

Name:”Rouel Shafi”,

Age:22

})

Update Operations:  
db.users.updateOne({

{Name:”Rouel Shafi”},

{$set: {Age:22}}

})

Read Operations:  
db.users.find({

{Name:”Rouel Shafi”},

{Age:22}

})

Delete Operations:  
db.users.DeleteMany({

Age:22

})

**Q2.**

**POST vs. PUT**

Post is used to modify and update a resource while PUT method completely replaces whatever currently exists at the target URL with something else. With this method, you can create a new resource or overwrite an existing

**Why was POST used to update employee data in the Employees sample AJAX application?**

POST was used in the employee AJAX app to update records. Because through POST we could update an employee as many times as we wanted. We did not use PUT over there because of its Idempotent nature. We cannot through PUT we cannot update as many times as we want it would only give results the first time.

**What happens if you replace POST with PUT? What difference does it make?**

We can say that the 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

POST and PUT are both popular HTTP methods that may be sometimes confused or used interchangeably. However, it’s important to correctly identify the idempotence of the action at hand in order to determine whether a PUT vs POST method should be used. Otherwise, the misuse of each method may result in the occurrence of unexpected bugs.

**Q3.**

A PUT method replaces or changes the whole document rather than just the specific or required field. Thus, a PUT request always contains a full resource. This is necessary because, a necessary quality of PUT requests is idempotence — the quality of producing the same result even if the same request is made multiple times.

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.

**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?**

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.

**Q4.**

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.

**Advantages of Reactjs**

* JSX is a JS syntax that enables HTML quotes and usage of HTML tag syntax for subcomponents rendering. It promotes building of machine-readable code and provides ability to compound components in one compile-time verified file.
* Prompt rendering is among the best features of React that gives a significant edge over Angular. The technology comprises smart methods to mitigate the amount of DOM operations, optimize and accelerate the updates process. Virtual DOM (Document Object Model) is of great use while handling vast databases.
* The core difference between reactjs and angularjs is that React is JS-centric, while ng2 remains HTML-centric. JavaScript is far more robust, than HTML, that makes React far more simple, focused and consistent.

**Advantages of Angularjs**

* Global community support is one of the factors, that can easily make Angular the best javascript framework. Developers and designers constantly collaborate and contribute to the community, increasing credibility and reliability of the framework.
* It is a full-fledged framework that can run in any browser or platform. Moreover, it is consistent, overwhelmed with ready-made tools, ng components are robust and quite mature, as contrasted with React.
* Two-way data bind is probably the top feature, as it diffuses the impact after every minor data change and does way with the need for further effort with data sync in view and model.

**Disadvantages of angularjs**

* Despite a comprehensive and clear manual, steep learning curve and complexity are named among the main weak points of Angular.js. Like any other client-side rendering technology in javascript framework comparison list, programmers should place special emphasis on security to make apps reliable and safe. Though, with introduction of Angular Universal and pre-rendering option in ng2 this issue was defused.

**Disadvantages of react.js**

* Comparing react vs angular performance, first of all it’s worth mentioning that reactJS is not a full-scale framework and for this very reason integration of the UI library into a common MVC framework requires deeper programming knowledge. It is still young and not mature, considering tutorial volumes, limited ecosystem, etc.
* Apart from pros and cons of reactjs, we should also mention Flux that is frequently applied for adding a structure and architecture to react app. Usage of both technologies can become a challenge for non-experienced programmer, as it lacks structured and comprehensive documentation or guide.

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.

**Q5.**

Vue.js is an open-source JavaScript framework for building user interfaces and single-page applications. Vue.js features an incrementally adoptable architecture that focuses on declarative rendering and component composition. Advanced features required for complex applications such as routing, state management and build tooling are offered via officially maintained supporting libraries and packages.

**Benefits of Vue.js:**

* Empowered HTML. This means that Vue.js has many similar characteristics with Angular and this can help to optimize HTML blocks handling with a usage of different components.
* Detailed documentation. Vue.js has very circumstantial documentation which can fasten learning curve for developers and save a lot of time to develop an app using only the basic knowledge of HTML and JavaScript.
* Adaptability. It provides a rapid switching period from other frameworks to Vue.js because of the similarity with Angular and React in terms of design and architecture.
* Awesome integration. Vue.js can be used for both building single-page applications and more difficult web interfaces of apps. The main thing is that smaller interactive parts can be easily integrated into the existing infrastructure with no negative effect on the entire system.
* Large scaling. Vue.js can help to develop pretty large reusable templates that can be made with no extra time allocated for that according to its simple structure.
* Tiny size. Vue.js can weight around 20KB keeping its speed and flexibility that allows reaching much better performance in comparison to other frameworks.

**Drawbacks of Vue.js:**

* Lack of resources. Vue.js still has a pretty small market share in comparison with React or Angular, which means that knowledge sharing in this framework is still in the beginning phase.
* Risk of over flexibility. Sometimes, Vue.js might have issues while integrating into huge projects and there is still no experience with possible solutions, but they will definitely come soon.
* Lack of full English documentation. This leads to a partial complexity on some stages of development, nevertheless, more and more materials are being translated into English.

**Example**

|  |
| --- |
| <ul id="demo">  <li v-repeat="items" class="item-{{$index}}">  {{$index}} - {{parentMsg}} {{childMsg}}  </li></ul>  var demo = new Vue({  el: '#demo',  data: {  parentMsg: 'Hello',  items: [  { childMsg: 'Foo' },  { childMsg: 'Bar' }  ] }}) |

**Q6.**

First of all, Angular is based on TypeScript while AngularJS is based on JavaScript.

AngularJS uses terms of scope and controller while Angular does not have a concept of scope or controllers.

There are other advantages such **as modularity**. Much core functionality was moved to different modules. That caused lighter and faster core, **dynamic loading**, asynchronous template compilation and added support for **reactive programming**.

|  |  |
| --- | --- |
| **Angular JS** | **Angular** |
| Bindings/interpolation  Your favorite hero is: {{vm.favoriteHero}} | Bindings/interpolation  Your favorite hero is: {{favoriteHero}} |
| Filters  <td>{{movie.title | [uppercase](https://angular.io/api/common/UpperCasePipe)}}</td> | Pipes  <td>{{movie.title | [uppercase](https://angular.io/api/common/UpperCasePipe)}}</td> |
| Local variables  <tr ng-repeat="movie in vm.movies">  <td>{{movie.title}}</td>  </tr> | Input variables  <tr \*ngFor="let movie of movies">  <td>{{movie.title}}</td>  </tr> |
| ng-click  <buttonng-click="vm.toggleImage()">  <buttonng-click="vm.toggleImage($event)"> | Bind to the click event  <button (click)="toggleImage()">  <button (click)="toggleImage($event)"> |

**Q7.**

A linter or lint refers to tools that analyze source code to flag programming errors, bugs, stylistic errors, and suspicious constructs.

**JSLint with Example**

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.

|  |
| --- |
| (function ($) {  $.fn.loading = function(msg, type, cssClass){  var prefixes = {  warning: 'Warning: ' + msg,  error: 'Error: ' + msg,  info: 'Info: ' + msg,  warning: 'Caution: ' + msg,  };  if (type) {  concatMsg = prefixes[type];  } else {  concatMsg = msg;  }  $(this).each(function() {  var tis = $(this)  if (msg == false) {  tis.html('');  } else {  tis.html(concatMsg);  }  });  }})(jQuery |

**ESLint With Example**

ESLint is an open source JavaScript linting utility. 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.

|  |
| --- |
| {  "parserOptions": {  "ecmaVersion": 6,  "sourceType": "module",  "ecmaFeatures": {  "jsx": true  }  },  "rules": {  "semi": 2  }  } |

**Q8.**

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. XML is a kind of markup language – like HTML, which people sometimes use for sending data across the internet.

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

**Example to choose AJAX over Angular**

If you want to have asynchronous, better faster and better interaction between the user and the web application. For example with AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server. Basically when you don’t want the page to reload again and again.

**Example to choose Angular over AJAX**

AngularJS is best suited for the web application development as it works on the HTML code and JSON data which helps in developing for interactive and robust applications but using the same for a simple website development results in slow loading and quite erratic websites.