WEB ASSIGNMENT

Question 1:

MongoDB is a NoSQL database system which stores data in the form of BSON documents which **maps to the objects in application code**, making data easy to work with. It is a native driver that interacts with mongodb instances. Whereas, Mongoose is a MongoDB ODM (the NoSQL equivalent of an ORM) for Node. It provides with a simple validation and query API to help you interact with your MongoDB database and plays as a role of abstraction over the database model.

**Using mongoose over mongodb:**

Schema: This even gives a structure to your application code. This doesn’t hamper the scalability feature of mongo; because if in future if your app grows and there is a need to add few more fields, you can modify the schema and work accordingly.

Maintainability & development: Mongoose provides with faster development and high maintainability, much useful for urgently deliverable projects.

**CRUD examples:**

We can perform CRUD in following way:

Insert:

* [db.collection.insertOne()](https://docs.mongodb.com/manual/reference/method/db.collection.insertOne/#db.collection.insertOne)
* [db.collection.insertMany()](https://docs.mongodb.com/manual/reference/method/db.collection.insertMany/#db.collection.insertMany)

db.person.insertOne({

name: "abc",

rollNo: 123

})

Retrieve:

* [db.collection.find()](https://docs.mongodb.com/manual/reference/method/db.collection.find/#db.collection.find)

db.person.find({

{name:"abc”},

{rollNo:123}

})

Update:

* [db.collection.updateOne()](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#db.collection.updateOne)
* [db.collection.updateMany()](https://docs.mongodb.com/manual/reference/method/db.collection.updateMany/#db.collection.updateMany)

db.person.updateOne(

{rollNo: 123},

$set:{name:"xyz"})

Delete:

* [db.collection.deleteOne()](https://docs.mongodb.com/manual/reference/method/db.collection.deleteOne/#db.collection.deleteOne)
* [db.collection.deleteMany()](https://docs.mongodb.com/manual/reference/method/db.collection.deleteMany/#db.collection.deleteMany)

db.person.deleteOne({

rollNo: 123

})

**Question 2:**

POST vs PUT

POSTis used to modify and update a resource whereas, PUT is used to create a resource, or overwrite it.

Using POST to update employee data

Basically, using POST allows us to update as many times as we want and this is actually what we required in employee data that for every request it updates the employee. This would not have been possible if we used PUT request.

Replacing PUT with POST

If we had used PUT instead of POST, we would have got the same response for various same request. But with POST, we get various responses for same requests.

**Question 3:**

PUT vs PATCH

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.

The PATCH method requests that **a set of changes** described in the request entity be applied to the resource identified by the Request- URI.

Partial update:

For making partial updates, we must use PATCH, because if we use PUT it is assumed that you are sending the complete entity, and that complete entity *replaces* any existing entity at that URI,  while PATCH only updates the fields that were supplied, leaving the others alone.

**Question 4:**

Pros of React:

1) *Reusable Components* are wonderful and React is based on them. You start with small things, which you use to build bigger things, which you use to build apps. Each component has its own logic and controls its own rendering, and can be reused wherever you need them.

2) *The Virtual DOM* guarantees a minimum update time to the real DOM, providing higher performance and a cleaner user experience all around.

Cons of React:

1) 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.

2) Different and new libraries like Redux and Reflux are promising to accelerate the work of a library or improve the entire React ecosystem. At the end, developers struggle with integrating these tools with ReactJS. Some members of the community think that React technologies are updating and accelerating so fast that there is no time to write proper instruction.

Advantages of AngularJS:

1) No need to use observable functions; Angular analyses the page DOM and builds the bindings based on the Angular-specific element attributes. That requires less writing, the code is cleaner, easier to understand and less error prone.

2) Quite a number of different ways to do the same things, thus accommodating to particular development styles and tasks.

Limitations of AngularJS:

1) The lifecycle of Angular application is complex, and to master it you really need to read the code. Compile and link are not intuitive, and specific cases can be confusing (recursion in compile, collisions between directives etc.).

2) As the project grows with time, you most likely will need to throw away existing implementations and create new versions using different approaches. Angular implementations scale poorly.

**Question 5:**

Vue.js

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. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with [modern tooling](https://vuejs.org/v2/guide/single-file-components.html) and [supporting libraries](https://github.com/vuejs/awesome-vue#components--libraries).

Vue vs angularJS vs React

* Angular is a framework because it provides you with a good start to build an application with the complete setup. You can simply start building. React and Vue, on the other side, are more flexible and universal than Angular.
* Vue is the cleanest in comparison to these three frameworks. It helps you keep your code efficient with the perfect balance of internal dependencies and flexibilities. It is very simple, straightforward, and easy-to-use JavaScript framework which aims to simplify web development.
* Virtual DOM model is very helpful in terms of performance. Both, React and Vue has a Virtual DOM. Due to a well-built structure, Vue delivers great performance and memory allocation. React, on the other side, works well and Angular is already ahead in the competition.
* Angular uses two-way data binding, React goes for single-data flow, and Vue supports both.
* React and Vue both provide flexibility to select the things which are important for the app. You need to simply add the required libraries to the source code in React and Vue. Angular uses TypeScript and that’s why it is more suitable for SPA rather than microservices. React and Vue, both frameworks offer more flexibility for developing microapps and microservices.

Example:

var data = { a: 1 }   
var vm = new Vue({  
 data: data  
})   
vm.a == data.a   
vm.a = 2  
data.a   
data.a = 3  
vm.a

Question 7:

Linting:

“Linting” means running a very basic code quality tool which will look at your JavaScript, and tell you where and how to clean it up. It will analyses the code for potential errors. Linting code is already an established part of any (popular) JavaScript project and has a lot of benefits such as:

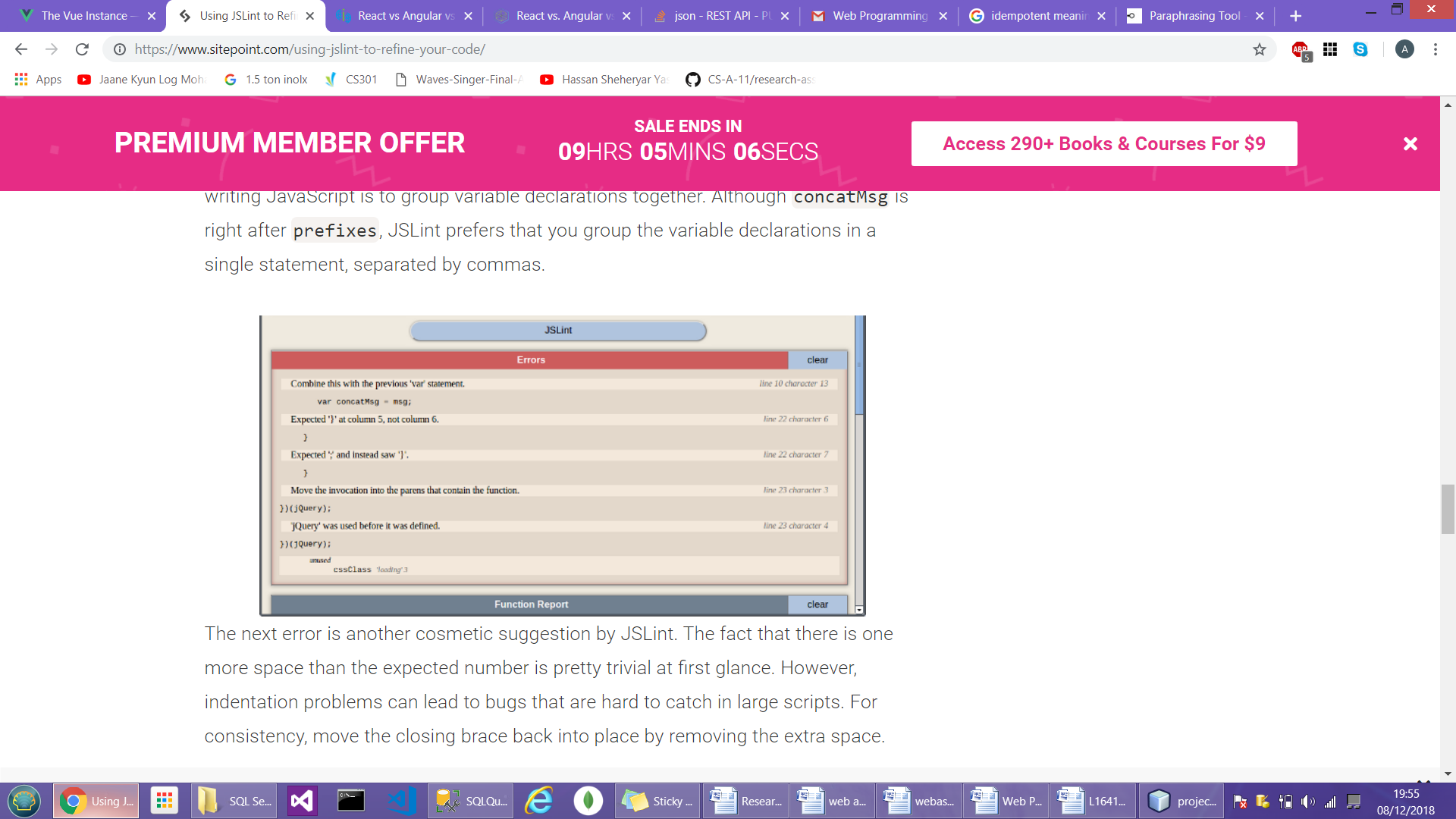
Readability

Pre-code review

Finding (syntax) errors before execution

JSLinting:

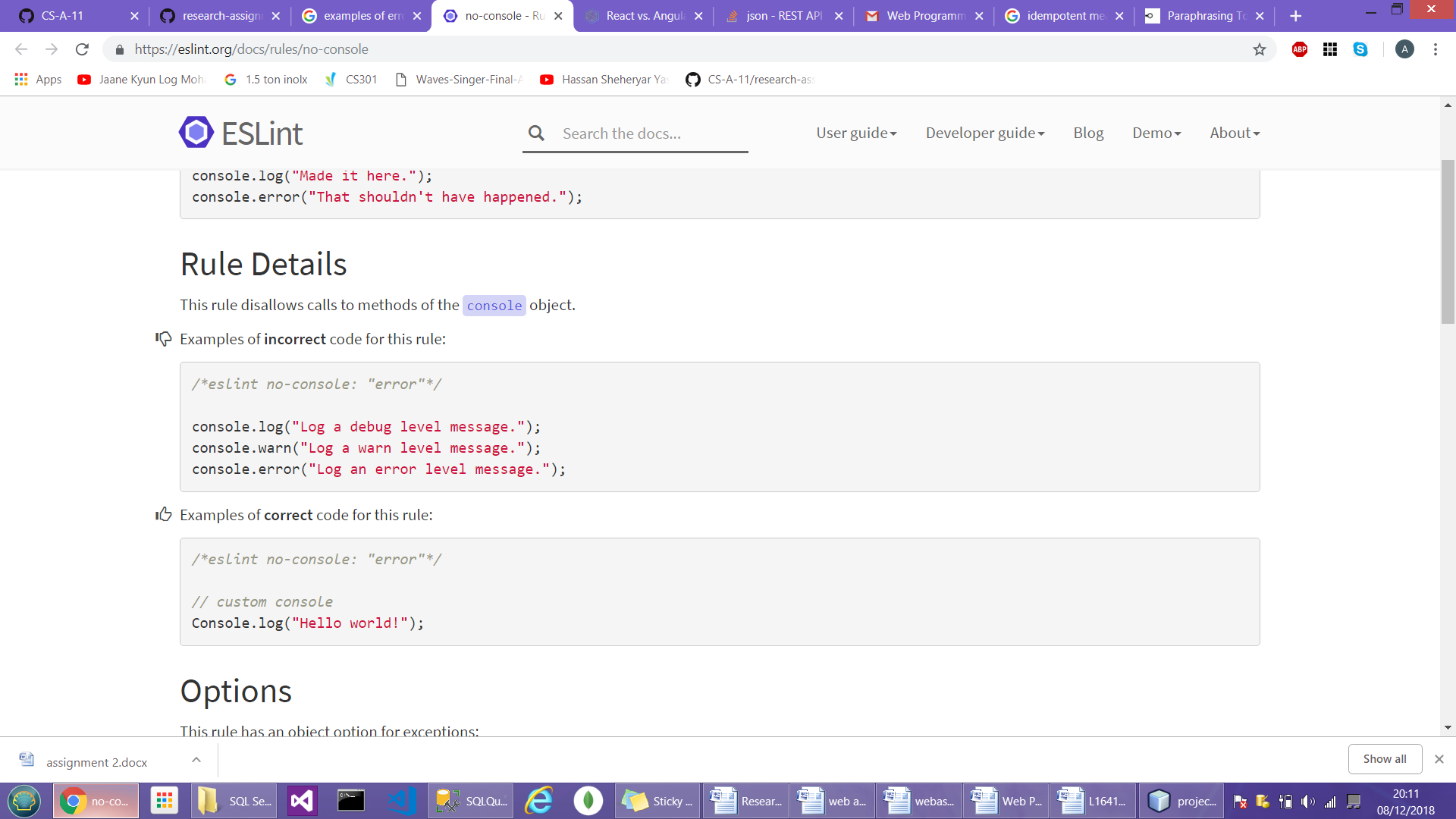
JSLint (or ESLint) is a static code analysis tool for checking whether your JS code is compliant with good coding practices. It will warn you when your code uses any questionable language feature or any feature in a questionable manner.



ESLinting:

[eslint](https://eslint.org/) is the dominant tool for linting Node.js packages, and can be configured to enforce multiple coding styles. ESLint is a tool for identifying and reporting on patterns found in JavaScript code, with the goal of making code more consistent and avoiding bugs. In many ways, it is similar to JSLint and JSHint with a few exceptions.

Example:



Question 8:

Angular vs AJAX

AJAX over Angular:

For asynchronous interaction between the user and the application we use AJAX. With AJAX, when you request, JavaScript will hit the server, and update the current screen with required response without loading the page.

Angular over AJAX:

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

Question 6:

1) Angular is based on TypeScript while AngularJS is based on JavaScript.

2) AngularJS has many directives and every developer can also specify custom new directive. Angular also has standard directives, but they are used in a bit different way. For example: **ng-model** in AngularJS means that you want to create two-way binding. If you want to create one-way binding, you should use **ng-bind**.

3) AngularJS has an injector subsystem that’s responsible for creating components, injecting dependencies and resolving the list all dependencies. Angular has a newer dependency injection system that’s different from that of the older DI pattern. Angular’s dependency injection is managed through the @NgModule array that comprises providers and declarations.

4) AngularJS follows the traditional MVC architecture that comprises a model, a view and a controller. Angular, on the other, hand has a component-based architecture. Every Angular application has at least one component known as the root component. Each component has an associated class that’s responsible for handling the business logic and a template that represents the view layer.