Assignment10

React.js

Q_1 : What is JSX?

Answer : JSX is a syntax extension of JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code.

Q_2 : What is the virtual DOM?

Answer : DOM stands for Document Object Model. The DOM represents an HTML document with a logical tree structure. Each branch of the tree ends in a node, and each node contains objects.React keeps a lightweight representation of the real DOM in the memory, and that is known as the virtual DOM. When the state of an object changes, the virtual DOM changes only that object in the real DOM, rather than updating all the objects. The following are some of the most frequently asked react interview questions.

Q_3 : Why is there a need for using keys in Lists?

Answer : Keys are very important in lists for the following reasons:

- A key is a unique identifier and it is used to identify which items have changed, been updated or deleted from the lists
- It also helps to determine which components need to be re-rendered instead of re-rendering all the components every time. Therefore, it increases performance, as only the updated components are re-rendered

Q_4 : What are forms in React?

Answer : React employs forms to enable users to interact with web applications. Using forms, users can interact with the application and enter the required information whenever needed. Form contain certain elements, such as text fields, buttons, checkboxes, radio buttons, etc
Forms are used for many different tasks such as user authentication, searching, filtering, indexing, etc

Q_5 : What is an arrow function and how is it used in React?

Answer :An arrow function is a short way of writing a function to React.

- It is unnecessary to bind ‘this’ inside the constructor when using an arrow function. This prevents bugs caused by the use of ‘this’ in React callbacks.

Q_6 :What are the components in React?

Answer : Components are the building blocks of any React application, and a single app usually consists of multiple components. A component is essentially a piece of the user interface. It splits the user interface into independent, reusable parts that can be processed separately..

- Functional Components: These types of components have no state of their own and only contain render methods, and therefore are also called stateless components. They may derive data from other components as props (properties).
- Class Components: These types of components can hold and manage their own state and have a separate render method to return JSX on the screen. They are also called Stateful components as they can have a state.

Q_7 : What is the state of React?

Answer :The state is a built-in React object that is used to contain data or information about the component. The state in a component can change over time, and whenever it changes, the component re-renders.
The change in state can happen as a response to user action or system-generated events. It determines the behaviour of the component and how it will render.

Q_8 : What are props in React?

Answer : Props are short for Properties. It is a React built-in object that stores the value of attributes of a tag and works similarly to HTML attributes.

- Props provide a way to pass data from one component to another component. Props are passed to the component in the same way as arguments are passed in a function.

Q_9 : What is a React Router?

Answer : React Router is a routing library built on top of React, which is used to create routes in a React application. This is one of the most frequently asked react interview questions..

Q_10 : Why do we need to React Router?

Answer : It maintains consistent structure and behaviour and is used to develop single-page web applications.
Enables multiple views in a single application by defining multiple routes in the React application.

Q_11 : Why can’t browsers read JSX?

Answer : Browsers can only read JavaScript objects but JSX is not a regular JavaScript object. Thus to enable a browser to read JSX, first, we need to transform JSX file into a JavaScript object using JSX transformers like Babel and then pass it to the browser

Q_12 : What do you understand by refs in React?

Answer : Refs is the shorthand for References in React. It is an attribute which helps to store a reference to a particular React element or component, which will be returned by the components render configuration function. It is used to return references to a particular element or component returned by render(). They come in handy when we need DOM measurements or to add methods to the components.

Q_13 : What is the significance of the useEffect hook in React?

Answer : useEffect is a hook in React that allows side effects to be performed on functional components. It is often used for tasks such as fetching data, subscribing, or manually modifying the DOM. useEffect runs after each render and can be used to manage the lifecycle of the element.

Q_14 : Explain the concept of controlled and uncontrolled components in React ?

Answer : useEffect is a hook in React that allows side effects to be performed on functional components. It is often used for tasks such as fetching data, subscribing, or manually modifying the DOM. useEffect runs after each render and can be used to manage the lifecycle of the element.

Q_15 : What is React.memo()?

Answer : useEffect is a hook in React that allows side effects to be performed on functional components. It is often used for tasks such as fetching data, subscribing, or manually modifying the DOM. useEffect runs after each render and can be used to manage the lifecycle of the element.

Q_16 : What is React.memo()?

Answer : React.memo is a higher-order component that memorises the rendering of a functional component, preventing unnecessary re-renders if the component's props remain the same..

Q_17 : How does error boundary work in React?

Answer : Error boundaries are React components that catch JavaScript errors anywhere in the component tree below them and log those errors or display a fallback UI. They are implemented using componentDidCatch lifecycle method in class components or using error prop in error boundaries created with the ErrorBoundary component in functional components.

Q_18 : Explain the concept of context in React.?

Answer : Context provides a way to share values (such as themes, user authentication) across the component tree without passing props explicitly. It involves creating a Provider and Consumer to wrap components that need access to the shared data.

Q_19 : Explain the concept of context in React.?

Answer : Context provides a way to share values (such as themes, user authentication) across the component tree without passing props explicitly. It involves creating a Provider and Consumer to wrap components that need access to the shared data.

Q_20 : What is the significance of the useCallback and useMemo hooks?

Answer : useCallback is used to memoize functions, preventing unnecessary re-creation of functions on each render. useMemo is used to memoize values, preventing their recalculation on every render. Both can be useful for optimising performance in certain scenarios.

Node.js

Q_1 : What is Node.js and how does it work?

Answer : Node.js is a virtual machine that uses JavaScript as its scripting language and runs Chrome's V8 JavaScript engine. Basically, Node.js is based on an event-driven architecture where I/O runs asynchronously making it lightweight and efficient. It is being used in developing desktop applications as well with a popular framework called electron as it provides an API to access OS-level features such as file system, network, etc.

Q_2 : How do you manage packages in your node.js project?

Answer : It can be managed by a number of package installers and their configuration file accordingly. Out of them mostly use npm or yarn. Both provide almost all libraries of javascript with extended features of controlling environment-specific configurations. To maintain versions of libs being installed in a project we use package.json and package-lock.json so that there is no issue in porting that app to a different environment.

Q_3 : How is Node.js better than other frameworks most popularly used?

Answer : Node.js provides simplicity in development because of its non-blocking I/O and event-based model results in short response time and concurrent processing, unlike other frameworks where developers have to use thread management.

- It runs on a chrome v8 engine which is written in c++ and is highly performant with constant improvement.

- Also since we will use Javascript in both the frontend and backend the development will be much faster.

- And at last, there are sample libraries so that we don't need to reinvent the wheel

Q_4 : What are some commonly used timing features of Node.js?

Answer : setTimeout/clearTimeout – This is used to implement delays in code execution.

- setInterval/clearInterval – This is used to run a code block multiple times.
- setImmediate/clearImmediate – Any function passed as the setImmediate() argument is a callback that's executed in the next iteration of the event loop.
- process.nextTick – Both setImmediate and process.nextTick appear to be doing the same thing; however, you may prefer one over the other depending on your callback's urgency

Q_5 : What is fork in node JS?

Answer : A fork in general is used to spawn child processes. In node it is used to create a new instance of v8 engine to run multiple workers to execute the code

Q_6 :Why is Node.js single-threaded?

Answer : Node.js was created explicitly as an experiment in async processing. This was to try a new theory of doing async processing on a single thread over the existing thread-based implementation of scaling via different frameworks.

Express.js

Q_1 : What is Express.js?

Answer :Express.js, or simply Express, is a free, open-source, lightweight and fast backend web application framework for Node.js. It is released as open source software under the MIT licence.
It is designed for building single-page, multi-page, and hybrid web applications and APIs. It is said to be the de facto standard server framework for Node.js.

Q_2 : Why do we use Express.js?

Answer : Express.js is an automatically prebuilt Node.js framework that facilitates us to create server-side web applications faster and smarter. The main reason for choosing Express is its simplicity, minimalism, flexibility, and scalability characteristics.

Q_3 : What is the difference between Express.js and Node.js?

Answer : Node.js is an open-source, cross-platform run-time environment used for executing JavaScript code outside of a browser. Node.js is not a framework or a programming language; it is a platform that acts as a web server. Many big companies such as Paypal, Uber, Netflix, Walmart, etc., are using this. On the other hand, Express is a small framework based on the functionality of Node.js.

Q_4 : Which are the arguments available to an Express JS route handler function?

Answer : Following are the arguments that are available to an Express.js route handler-function:

- **Req:** the request object
- **Res:** the response object
- **Next (optional):** It is a function employed to pass management to one of the above route handlers.

Q_5 : What is Middleware in Express.js? What are the different types of Middleware?

Answer : Middleware is a function invoked by the Express routing layer before the final request handler.Middleware functions are used to perform the following tasks:

- It is used to execute any code.
- It is also used to make changes to the request and the response objects.
- It is responsible for ending the request-response cycle.
- It can call the next middleware function in the stack.

Q_6 : What is CORS, and how can it be handled in Express.js?

Answer : CORS (Cross-Origin Resource Sharing) is a security feature implemented by web browsers that restricts web pages from making requests to a different domain than the one that served the web page. In Express.js, you can handle CORS using the cors middleware or by manually setting appropriate headers.

MongoDB

Q_1 : What is a Document in MongoDB?

Answer : A Document in MongoDB is an ordered set of keys with associated values. It is represented by a map, hash, or dictionary. In JavaScript, documents are represented as objects

Q_2 : What is a Collection in MongoDB?

Answer : A collection in MongoDB is a group of documents. If a document is the MongoDB analog of a row in a relational database, a collection can be thought of as the analog of a table.
Documents within a single collection can have any number of different "shapes", such that collections have dynamic schemas.

Q_3 : What are Databases in MongoDB?

Answer : MongoDB groups collections into databases. MongoDB can host several databases, each grouping together collections.
Some reserved database names are as follows:
Admin , local , config

Q_4 : What is the Mongo Shell?

Answer : It is a JavaScript shell that allows interaction with a MongoDB instance from the command line. With that one can perform administrative functions, inspecting an instance, or exploring MongoDB.

Q_5 : What are some features of MongoDB?

Answer : Indexing: It supports generic secondary indexes and provides unique, compound, geospatial, and full-text indexing capabilities as well.

- Aggregation: It provides an aggregation framework based on the concept of data processing pipelines.
- Special collection and index types: It supports time-to-live (TTL) collections for data that should expire at a certain time
- File storage: It supports an easy-to-use protocol for storing large files and file metadata.
- Sharding: Sharding is the process of splitting data up across machines.

Q_6 : How to add data in MongoDB?

Answer : The basic method for adding data to MongoDB is “inserts”. To insert a single document, use the collection’s insertOne method : db.books.insertOne({"title" : "Start With Why"})

Q_7 : How to perform queries in MongoDB?

Answer : The find method is used to perform queries in MongoDB. Querying returns a subset of documents in a collection, from no documents at all to the entire collection. Which documents get returned is determined by the first argument to find which is a document specifying the query criteria.
Example:
> db.users.find({"age" : 24})

Q_8 : What are the data types in MongoDB?

Answer : MongoDB supports a wide range of data types as values in documents. Documents in MongoDB are similar to objects in JavaScript. Along with JSON’s essential key/value–pair nature, MongoDB adds support for a number of additional data types. The common data types in MongoDB are:

Null
{"x" : null}
Boolean
{"x" : true}
Number
{"x" : 4}
String
{"x" : "foobar"}
Date
{"x" : new Date()}
Regular expression
{"x" : /foobar/i}
Array
{"x" : ["a", "b", "c"]}
Embedded document
{"x" : {"foo" : "bar"}}
Object ID
{"x" : ObjectId()}
Binary Data
Binary data is a string of arbitrary bytes.
Code
{"x" : function() { /* ... */ }}