

1. What is middleware in Node js: is just a function that takes three parameters (req,res,,next). Where next is a function that allows you to chain functions <https://dev.to/getd/express-middleware-is-just-a-fancy-way-of-chaining-a-bunch-of-functions-explained-in-3-mins-43jf#:~:text=A%20middleware%20is%20just%20a,app.>
2. Body-parser
3. How do we connect our database?
Install mongodb ,mongodb
Add url

```
mongoose.connect('mongodb://localhost:27017/Fresh-Corner',{
  useNewUrlParser:true, useNewUrlParser:true
}).then((e)=>{
  console.log('DB compass connected')
  app.listen(port,()=>{
    console.log(`server is running on port: ${port}` )
  })
}).catch(err=>{
  console.log(err)
})
```

4. What are Node Js Module: They are functionalities A. Core module(Http, Url, QueryString, path,fs,util), B. Local Module(are modules we created locally , Third Party Module(node package manager)
5. Write a query to filter all datas that has not middil name
6. Db.find
7. On 28 October 2014, **HTML5 was** released as a W3C Recommendation, bringing the specification process to completion. On 1 November 2016, HTML 5.1 **was** released as a W3C Recommendation. On 14 December 2017, HTML 5.2 **was** released as a W3C Recommendation.
8. How to create custom element in In HTML ? <amele>Hello Worled</amele> let amele= document.createElement('amele')
function myFunction() {
var amele = document.createElement("P");
amele.innerText = "This is a paragraph.";

document.body.appendChild(amele);
}

9. Html% features (nav, section, placeholder, main, audio, Video, article,

11.What are Psuedo-classes? Hover, Visited and Unvisited , focus

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

10. What is Media Query Media query is a CSS technique introduced in CSS3.

It uses the `@media` rule to include a block of CSS properties only if a certain condition is true.

11. What is meta tag in HTML? The `<meta>` tag defines **metadata** about

12. an **HTML** document. **Metadata** is data (information) about data. ... **Metadata** will not be displayed on the page, but is machine parsable. **Metadata** is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

Eg. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

13. What is flexbox: use to design our layout for responsive web page. commonly known as Flexbox, is a CSS 3 web layout model. It is in the W3C's candidate recommendation stage. The flex layout allows responsive elements within a container to be automatically arranged depending upon screen size.

- Flex-direction (row, column)
- Flex-wrap (wrap: unwrap based on screen size)
- Flex-row
- Flex-direction & flex-row
- Flex-flow: row wrap
- Display: flex
- Justify-content: Center, space-between, space-around
- Align-items :center
- Width
- Height:

Box Model All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

- The CSS box model is essentially a box that wraps around every HTML element. It consists of margins, borders, padding, and the actual content. **Content** - The content of the box, where text and images appear

- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

14. Controlled and uncontrolled input?

15. Error boundary: catch JavaScript errors in the component.

16. What is Connect in redux? The **connect()** function connects a React component to a **Redux** store. It provides its **connected** component with the pieces of the data it needs from the store, and the functions it can use to dispatch actions to the store.

17. `useEffect(()=>{},[])` `componentDidMount()`, `useEffect(()=>{[something]})` `componentDidUpdate()`, `useEffect(()=>{ return ()=>{[something]}})` `componentWillUnmount()`

18. Bootstrap? Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

The **Bootstrap Grid** System is used for layout, specifically Responsive Layouts. ...

The **Grid** is made up of groupings of Rows & Columns inside 1 or more Containers.

///JavaScript

19. `Object.freeze()` VS `const` : we can not modified the object and the Value. **freeze()** prevents modification or extension to the existing value of an **object**. ... **const** makes the variable binding immutable but it's value can still be modified. **Object.freeze()** ignores the value modification to an **object** but there is no restriction on the binding

///GitHub

20. What is staging: A **staging** step in git allows you to continue making changes to the working directory, and when you decide you wanna interact with version control, it allows you to record changes in small commits. ... `html` were not logically associated with the first two files and were done in a separate commit.

21.Redux can be described in three fundamental principles:

22.

What is Redux?

It helps to understand what this "Redux" thing is in the first place. What does it do? What problems does it help me solve? Why would I want to use it?

Redux is a pattern and library for managing and updating application state, using events called "actions". It serves as a centralized store for state that needs to be used across your entire application, with rules ensuring that the state can only be updated in a predictable fashion.

Single source of truth

The [global state](#) of your application is stored in an object tree within a single [store](#).

State is read-only

The only way to change the state is to emit an [action](#), an object describing what

happened. **Changes are made with pure functions**

To specify how the state tree is transformed by actions, you write pure [reducers](#).

CONTROLLED COMPONENT

In a controlled component, form data is handled by a React component. Input data is stored inside the component state;

UNCONTROLLED COMPONENT

The alternative is uncontrolled components, where form data is handled by the DOM itself. the input data is stored in the DOM. And we have to grab a reference to manipulate the state; What is state lifting in react?

In React, sharing state is accomplished by moving it up to the closest common ancestor of the components that need it.

This is called "lifting state up".

WHAT IS WEBPACK

* Webpack is a tool that lets you compile JavaScript modules, also known as module bundler. Given a large number of files, it generates a single file (or a few files) that run your app
what is StrictMode in React?

it is a tool for highlighting potential problems in an application

WHAT IS MIXINS?

a mixin is a class containing methods that can be used by other classes without a need to inherit from it.

ERROR BOUNDARIES

Error boundaries are React components that catch JavaScript errors anywhere in their child component tree, log those errors, and display a fallback UI instead of the component tree that crashed. Error boundaries catch errors during rendering, in lifecycle methods, and in constructors of the whole tree below them.

Note

Error boundaries do not catch errors for:

Event handlers (learn more)

Asynchronous code (e.g. `setTimeout` or `requestAnimationFrame` callbacks)

Server side rendering

Errors thrown in the error boundary itself (rather than its children)

*Only class components can be error boundaries.

*A class component becomes an error boundary if it defines either (or both) of the lifecycle methods `static getDerivedStateFromError()` or `componentDidCatch()`. Use `static getDerivedStateFromError()` to render a fallback UI after an error has been thrown. Use `componentDidCatch()` to log error information.

1. virtual dom
2. class and functional based component
functional components
3. what is hooks
4. how do you define a variable in functional component
`const[name, setName] =`
5.
what is redux?
6.
how pass data to child ?
7. from parent to child ?
8. how workers react?
9. what is node js
10. how do you connect your react with node js
middle ware
body-parser
11. how do you connect database
which module you used
first name
lname
12. what are modules
core
3rd party
13.
new html features

flex-box
box model
display:flex;
box model

14. can you explain the projects you did in react.
what challenges you faced working with react.
controlled and uncontrolled components ?
error boundaries in react?
how do we implement them?
methods supported?
componentDidCatch()
other method
react lifecycle added in new version 15 +
willmount deprecated
what is redux flow?
how simple counter work?
are you aware of connect method?
are you aware of routing ?
nested routing?
what is the efficient way of implementing nested routing in react using query
useEffect different scenarios
code optimization rules we should follow in react?
lazy loading
minification
scripting
15. strategies
16. what is deployment state?
html
is it possible to create custom element in html
meta tag
the use of meta tag
newly added features of html5
css3
flex-box properties
what are the properties
we use them for
pseudo class?
media query?
17.
ES6
recent features
arrow function
destructuring
spread operators
const and let

18. object.freeze
difference between object.freeze
19. have you ever used jquery?
bootstrap
grid-class in bootstrap
git
what is staging in git?
what is stashing?
are you agile?
test cases
how to configure enzyme
how to configure react
how to use it
20. node.js
21. what is changing in node.js?
callback hell?
how to avoid ?
22. promises.all?
what is RPL
23. What is CORS: cross origin-resource-sharing: is a mechanism that enables different resources on a web page to be requested from another domain outside the domain from which the request originated