1. **What is React? (*E.g. Consider: what is it? What is the benefit of using it? What is its virtual DOM? Why would someone choose it over the standard HTML / CSS stack?)***

ReactJS is an open-source, component based front end library responsible only for the view layer of the application. It is maintained by Facebook.

It developed a new way of rendering websites, evolving the responsive nature of web pages.

The React allows us to develop the individual parts of the application on both the client side and server side. It makes the development process fast.

By using React we can save the time and cost and easier to maintain. Which consists a modular structure.

The main benefits of ReactJS is to reuse the components. It saves the time for the developers to rewrite the code. If we do any changes to the one component only that change will be applied to only to that particular components.

By using ReactJS we can build rich user Interfaces and also it offers rich in rendering

The virtual DOM works fast owning to the fact that it only changes individual DOM elements instead of reloading complete DOM every time.

Doing changes virtual DOM is faster thenactual(Real) DOM.

Using ReactJS over the HTML/CSS we can develop the dynamic webpages which can hold large no of data.

We can build more organized websites with components.

We can build applications in a faster manner.

1. **What are Props? What is State? What is the difference between them?**

State is defined as an object used to store the data or information about the component.

It is defined as an object to store the key and value pairs.

The state can be changed based on the user action.

The state object can store the multiple values.

The state object is initialized using a constructor

Props are a way to pass information into a React component, they can have any type including functions - sometimes referred to as callbacks.

The props components are read only components and cannot be modifed

In ReactJSthe data can be passed from one component to another component using the props just like a arguments passed to a function.

Differences between State and Props:

Props are read only we cannot changed.

State can be changed

Props can be used in Class components and Functional components.

State can be used in only in Class Components.

Props will make components reusable

State cannot make components reusable

1. **What are React Hooks? How do they differ from existing lifecycle methods?**

React Hooks makes the functional components behave like a Class component.

By using React Hooks we can add the state to the Functional component

We can use Functional components instead of Class Components.

We can implement React State and life cycle methods without writing Classes.

When we create a component using Class based component we need to create the life cycle methods in class.

But using Functional components we can implement the life cycle of a React using Hooks

Using use Effect() we can replace the life cycle methods like component Did Mount and component Will Mount and component Did Update and component Will Update

1. **Design the perfect door - what should it look like, what are the components for it? What design heuristics should it follow, and how does your design match? What made you choose this design?** 
   1. ***Consider in particular (likely need to do independent learning*): *who are your stakeholders? What is their personas? What is the doors requirements and how will your stakeholders benefit from your solution?***

Stakeholders are grouped into active supporters (upper right quadrant), passive supporters (lower right quadrant), active nonsupporters (upper left quadrant), passive nonsupporters (lower left quadrant). By connecting stakeholders with positive relationships, you build a Stakeholder Support Network.

First, you need a clear objective, your expected outcome. For example, Ed is an active nonsupporter; realistically you want Ed to accept your idea and not campaign against it. Second, anticipate Ed's expected outcome. Ed probably wants you to agree with him and either abandon your idea or change it based on his perception. Third, you need to understand why Ed does not agree with your idea, his objections. Ed may feel that your idea will cost more than you anticipate or cause delays in the project. Fourth, envision an outcome that both you and Ed can live with, consensus. If cost is the primary issue, identify ways to minimize the cost without minimizing the benefits of your idea. Answer the questions in writing to solidify your thoughts.

1. **What is Angular, and how does it differ from React? *You may need to conduct independent research and learning for this***

Angular is an client based open source and java script based, component based frame work for building the large scale and powerful and easy to serve web applications.it consists the collections of libraries which include routing, forms, client server communication and many more

It is used to build the single page client applications using HTML and Typescript.

Angular is afull fledged model view controller architecture and it provides guidance on how the application should be structured

React is a library whereas Angular is a component based frame work.

React uses Virtual DOM and Angular uses RealDOM

React uses only one way data Binding whereas Angular uses Two way Data Binding

The Bundle Size is smaller compare to the bundle size of Angular.

The Execution Speed of React is faster then Execution speed of Angular Applications.

1. **Please describe Redux in as much detail - especially consider: why would someone use it? What is it? What's the benefit of using it? Are there any potential drawbacks to using it? How can it be added to a project? What is dispatch, provider, actions, etc?**

Redux is a state container for javascript applications.It is a state management tool in React. It is read only.

Whenever we develop an application as the application grows it becomes difficult to maintain the data flow and to keep it organized.

State transfer between the components is somewhat difficult in React since it is hard to keep track of which component the data is coming from. IT becomes really complicated if users are working with a large number of states within the application.

Redux will solve the state transfer problem by storing all the states in a single place called a store.

Every Component in the application can then directly access the required state from that store.

To add the Redux we need to install the modules react-redux and redux

Dispatcher is the center point of the data flow in a application.it is used to dispatch the actions and trigger state changes to the store.

The provider component makes the Redux store available to any nested components that need to access the Redux store

An action is a plain object that represents an intention to change the state. They must have a property to indicate the type of action to be carried out.

Actions must have a type field, indicating the type of action to be performed.

Any data whether from UI events needs to eventually be dispatched as actions

1. **Please describe Linux in as much detail as possible (feel free to use notes made during lessons, or draw from the lesson directly!). Especially consider: *what is its history? Why would someone use it over other existing operating systems? How does Windows and Mac OSX differ to Linux? How does Linux function, what are some unique features to it? How can it be installed today?***

Linux is a family of open source Unix like operating system based on Linux kernel.

An Operating System kernel first released on September 17, 1990, by Linus Torvalds

Linux packaged in a Linux distribution. It include the Linux kernel and supporting soft wares and libraries

Popular Linux distributions include Debian, Fedora Linux and Ubuntu

The most important advantage of Linus over the other operating systems is its security. Which u cannot find in the Windows and Mac.

Linux is with very low risk of viruses and malware.

There is no flavors of windows operating system. But in Linux we find tons of Linux Distributions catered for a different set of needs

Linux is accessible to the public for free.

Linux helps you to use or utilize your old and outdated computer systems as a firewall, router, backup server or file server and many more

Linux systems are known for consuming fewer system resources (RAM, disk space, etc.) when compared to Windows.

It is a Free and OpenSource. It consists shell /command line interface

The Linux environment allows users to encrypt their data.

Linux is available in various languages, it is simple to use it worldwide. As a result, you can change the language on your keyboard as per your preference.

Software updates are controlled by the users in Linux.

Linux is a highly lightweight operating system. Linux has far fewer prerequisites than any other operating system, has a smaller memory footprint, and uses less storage space.

1. **What are they, and which is better between Class components and Functional components? Provide a discussion. Consider: Go deep - how does each one work? What is the unique properties or behaviours to each one? Why would someone use one over the other? What are the advantages and disadvantages of each one? Who benefits from these advantages and disadvantages, who is it suitable for?**

A functional Component is a plain java script function that accepts props as an argument and returns a React element.

Functional Components are very easier to read and test because they are plain java script functions without state.

Functional componentsdo not have to access to dedicated variables like class based components.

The only state that a functional component effectively has access to are the props passed to it from its parent component.

Functional components do not access to the life cycle functions like a class based components.

**Ex:**

function Example1(props)

{

return(

<h1> Hello {props.name}</h1>

)

A Class based Component is a component which is a class which extends from React.Component Class creates a render function which returns a React element.

Class based Components are used when we

1. Managing the State of a component
2. Adding the Life cycle of the Component
3. Need to write the Logic for Event Handlers

**Ex:**

class Example1 extends React.Component

{

render()

{

return

<h1> Hello {this.props.name}</h1>

}

Functional component are easier to write and test compared to class based components.

Functional component syntax is not complex when we comaper to syntax of Class Components.

The performance of functional components is better then class components.

To accept the data and display the data we use functional components

In functional Component we cannot use a render method as we use in class components.