# Technical Questions:

## Q1: Describe React props and state - how do they work, how are they different, when should we use each one

**React props** – props are arguments passed into react component. Props are used to share same data across the components. It flows from parent component to child components(unidirectional flow).

However with callback function, props can be passed back from child to parent components.

Props are readonly.

Props can be passed to any component just like we use attributes in any HTML tag.

Ex- **<User name = ”nidhi” />**

In above example User is a react component and name is a props passed to this component.

**class UserList extends React.Component{**

**render(){**

**return(**

**<ul>**

**<li> <User name = “nidhi” />**

**</ul>**

**)**

**}**

**}**

UserList is parent component and User is child component and name is props passed from parent to child.

Now, User component will look something like below-

1. Function component –

We are giving props as an argument to function

**const User =(props) =>{**

**return <p>{props.name}</p>**

**};**

1. Class component –

In class components , props are received as a parameter of class constructor.

**class User extends React.Component{**

**constructor(props){**

**super(props);**

**}**

**Render(){**

**return(**

**<div>{this.props.name}</div>**

**)**

**}**

**}**

In both ways, component is returning div containing **nidhi**.

**React State –** State is a built-in object of react. It allows component to create and manage its own data. It is private to a component and not accessible to any other component.

Example –

**class Number extends React.Component{**

**constructor(){**

**this.state = {**

**counter : 1**

**}**

**}**

**this.setState({**

**counter : 8**

**})**

**Render(){**

**return(**

**<div>**

**<p>{this.state.counter}</p>**

**</div>**

**)**

**}**

**}**

Initial information of state is always defined in constructor. One should not modify state directly. It should always be updated with method called setState() as shown in above example.

State can be changed based on user input or by triggering any event.

Whenever state of any component changes, react re-renders that component with updated state in DOM. setState() method triggers the re-rendering process.

Classes are stateful component and functions are stateless components.

But after React hook introduction , state can be used in function as well.

**Difference between state and props-**

1. Props are used to pass data from parent to child components whereas state is used to create and manage component’s own data.
2. Props are read-only. State data can be modified by it’s own component using setState() method.
3. Modifying state triggers re-render event for that component. Props can not be modified.

**When to use state and props -**

States are mainly used when you need to re-render components based on user input or on trigger of any event.

Props are used when you need to pass data from parent components to child components.

## Q2: What will cause the render() function of a React component to be executed –

When a component is called , it calls the render method by default. Render() must return something – HTML or null.

setState() method is used to re-render the component. That is the reason we should never use setState() in render() method.

## Q3: Will React or raw JS/HTML deliver better performance for a single page web application, and why

Single Page Application is faster because instead of waiting for the client server communication to happen and wait for the browser to re-render the page, we can have instant response.

If we use Javascript or any other JS framework for developing application, server will consume less resources. Now when some part of the content on a page has to be altered, browser sends an AJAX call to the server. The server in return sends the data browser requires to update the page.It also enables to provide local caching and offline experience.

But downside of SPA is, we need to write more code to handle states, implement navigation and performance monitoring. And that is why we opt for JS framework.Like for navigation, we have ReactRouter.

Data needed everytime we change the page is requested by the application asynchronously using API calls.(XMLHttpRequest, Jquery.ajax, fetch)

## Q4: Describe the Observer pattern (in the context of UI development), what benefit it provides, how it is used, and a method of implementing in JS

**Observer Pattern –**It is a design pattern in which an object(Subject) maintains a list of objects depending on it(observers). Whenever there is a change in state of subject , all observers will be notified.

When a subject needs to notify observers about some changes, it broadcasts a notification to observers.

When we no longer want to notifiy any particular observer, the subject can remove it from the list of observers.

Subject –

1. Add(Observer)
2. Remove(Observer)
3. Notify()

Observer –

1. Update()

ConcreteSubject –

1. stateOfSubject
2. GetState()

ConcreteObserver –

1. stateOfSubject
2. Update()

Benefits – In JS, we need to update sections of page when there is some user input or an event triggers. In this type of scenario , we can use Observer pattern.

It enables one to many data binding between elements.

Class Subject{

Constructor(){

This.observers =[];

}

addObserver(observer){

this.observers.push(observer);

}

removeObserver(observer){

const index = this.observers.findIndex(o => {return observer === o});

if(index != -1){

this.observers = this.observers.slice(index, 1);

}

}

notify(){

if(this.observers.length > 0){

this.observers.forEach( o=> o.Update());

}

}

}

Class Observer{

Update(){

// action

}

}

## Q5: Explain an approach for adding zoom and pan interaction to the display of a large floorplan image in a web application

There are libraries available to add zoom and pan interaction to a large image. Ex- leaflet.js

1. Need to define area in which we want to show our image.
2. Exact size of image.
3. Initital state of zoom
4. Number of level of zoom. Min and max value of zoom.

With help of leaflet.js file, we create a map having minZoom,maxZoom, defaultzoom level.

Then define the mapping between screen pixels and its internal longitude/latitude co-ordinates.

Calculate boundaries of the image in co-ordinate system which is then used to map it to image with initial state provided to map object.

# Coding Problem

Implement a React component "AirportChooser" which is a button used to select one of many airports. The button when clicked should show a dropdown with list of all available airports. Each airport should display its name, city/country, and airport code. Display a loading state in case list of airports is not ready initially. Ensure the component can effectively support a large number of airports (40,000+). Include CSS styles for the component as well as a usage example. Do not use any 3rd party components for this project, only React itself. The actual list of airports should not be hardcoded in the component, but uploaded from a file. This list of airports may be useful for testing: <https://gist.github.com/tdreyno/4278655>