While running the ReactJS program. **Thing need to take note while doing it.**

* Add all the required library. If we don’t have compiler for ES6(like babel), then have to add browser.min.js to make it work in the browser
* Create jsx file to create component.
* Add it to the html file. Make note of it while adding we have to add like:

<script src=”js/app.jsx” type=”text/babel”/>

* Also while linking the component to the html file, we have to provide proper id, else will generate error in the browser console.
* We have to always refer class level member using ‘this’ keyword. Else we will get accessing property as not defined even though it is available inside the component class.

**Things need to study:**

Stateful and stateless component

events

Ref

Key

Router

LifeCycle of Component if having time

**Q) what is Ref and its use in the REACT?**

Ans:

Ref property of the html tag inside the component class, make it possible to read the tag value or assign some value to the tag(input, select or text area or span).

To make a reference, place the ref attribute with a function value on any React element or component. Then, inside of the function, the first parameter within the scope of the function will be a reference to the element or component the ref is on.

Ex:

<input type="text" ref={(val)=>{this.inputVal=val}}/><br></br>

myFun:**function**(){

this.inputVal.value='enter name';

console.log(this.inputVal.value);

},

**Note**: if we will not use lambda function, we can’t use ‘this’ keyword within that normal function.

**Note**: in render function we can’t display array or json object to html file directly, we can only display in single information.

**Q) how to add data to state array in react?**

Ans:

**var** newData={user:this.user.value,id:this.id.value}

this.setState({data:this.state.data.concat(newData)});

**Q) How can we remove some property from State?**

Ans: one way to override the same property with new value.

So assign ‘**undefined’** value to the property we want to remove.

Also you can use **delete**,

delete this.state.foo;

this.setState(this.state);

**Q) What is difference between a.push(b) and a.concat(b) in javascript? Where a and b are array of element.**

Ans:

a.push(b); => add new element to the array a and return length of the new array a.

a.concat(b) => add new array or element to the array and return the combined array.

a.reduc()=>

var result = numbers.reduce(function(accumulator, currentValue) {

return accumulator + currentValue;

});

a.map(function(element){})

{this.state.data.map((element,i)=><ChildComp key={i} value={element}/>)}

**Q) How can we work with the html tag in the react function? Like reading and updating value in the tag(input, textArea, select, span etc)?**

Ans:

There are two ways to achieve this:

<input type="text" ref={(el)=>this.user=el} onChange={this.myFuncall.bind(this)}/>

Ref:- **ref={(el)=>this.user=el}** //this.user is indication input tag itself.

using this we have to create a gobal variable (that is component level scope) which content the whole element and then we can access the html tag using the variable as:

Any\_fun(){ //this we have to call as need

Var inputValue=this.user.value; or

This.user.value=”new value”

}

**There is another way**, **onChange={this.myFuncall.bind(this)}**

where we want to access the tag value if any event is happening on the tag then we can call to custom event listener and pass the tag and using that we can monitor the html tag.

call:**function**(event){

event.target.innerText="this is new span value"//event is for span

event.target.value=”rohit kumar” //event is for input tag

console.log(event);

}

**Q) How does component decide to render a component?**

Ans:

A re-render can only be triggered if a component’s state has changed.

The state can change from a props change, or from a direct setState change.

By overriding method: **shouldComponentUpdate**

**shouldComponentUpdate** : return ture means react will render the component and will not for false return.

We can control the rendering of the component. Means even some value of state is changing we ignore that and render the component as per our requirement.

**Important note**

Returning false does not prevent child components from re-rendering when *their* state changes.

**Q) Why the span value didn’t bind with input in react.js? i.e. if we change the value of this.typed anywhere in the component, it will not reflect here.**

<span>{this.typed}</span>

Ans: Because you change a typed variable, which does not trigger component re-rendering.

Use React's state functionality so it will re-render when the value of type changes.

Q) what is default props and stateless component?

Q) Explain the following component of react router. (BrowserRouter, Route, NavLink, withRouter, and switch)