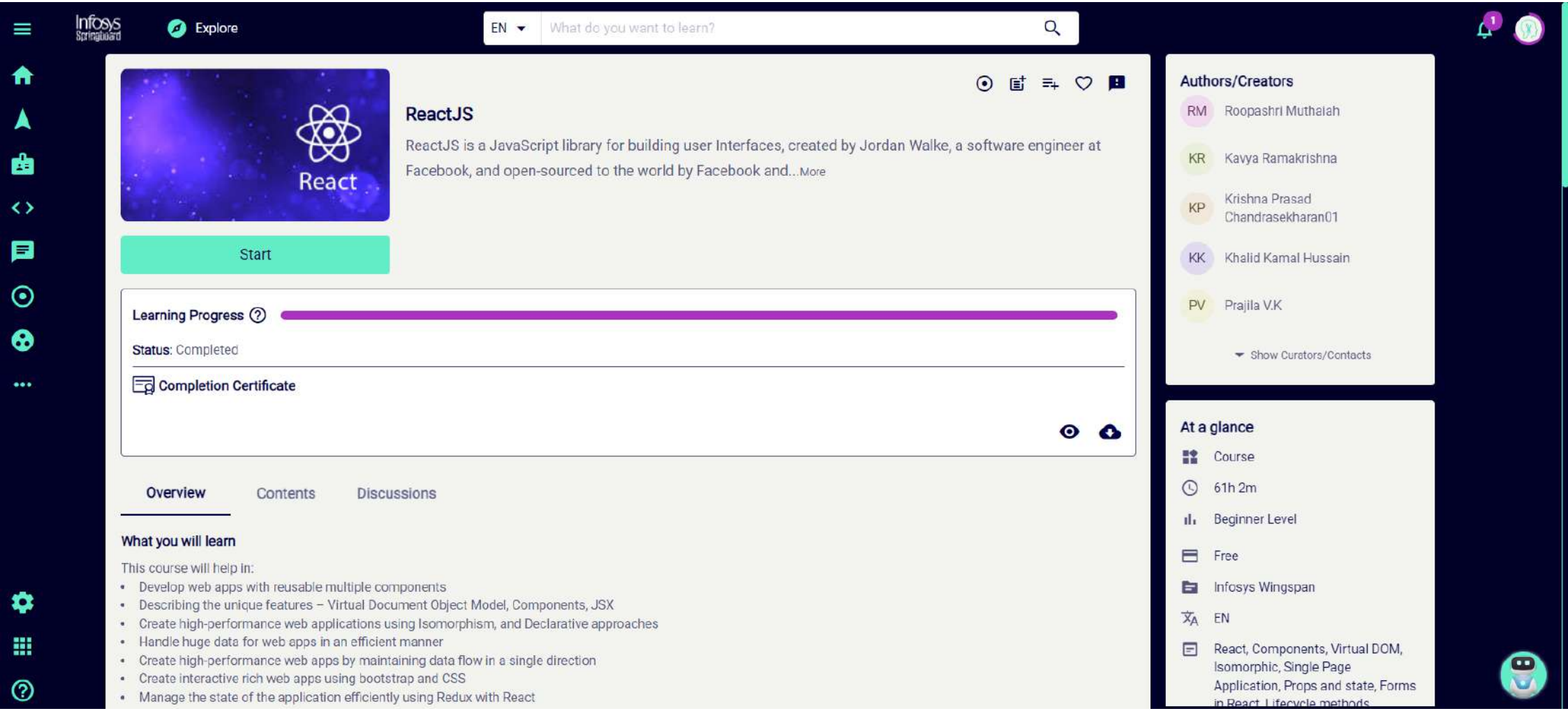

React JS - Assessment

Thank you. Your test submitted.

You have cleared this assessment.

Obtained Percentage	Obtained Marks
86.67 %	26 / 30

Best Attempt Score:86.67 % on 11-08-2024

Review Your Attempt

Result: CLEARED

Overall Percentage: 86.67% (26/30)

React JS - Assessment

Section Score: 26/30

Q1 of 30

Consider the below App component which renders the input field on the UI.

```
const App = () => {
  const [data,setData] = useState('Initial render')
        return (< input type = "text" value = {data} / > );
}
export default App
```

Identify the statements which are/is true with respect to the above component. (Select any two)

- [x] 'Initial render' text will be displayed in the input field
- [ ] The component is re-rendered when value is entered in the input field
- [x] The value can be typed in the input field
- [ ] It just displays the blank input field

Q2 of 30

Which of the following statement is/are false about React JS?

a. React JS uses phantom js to render the webpages on server-side

b. React JS is a framework

c. React JS improves the performance of the applications by using the concept of virtual DOM

d. In React JS, the data flow will happen in a single direction

- a, b and d
- Both b and d
- Both a and b
- Only b

Q3 of 30

Which of the following is the correct way to update the state?

a.
```
const auth = function(state = {status: "logged out", value: 'guest'}, action)
{
        switch (action.type)
        {
                case 'LOGIN':
                   return Object.assign({}, state, {
                      status: 'logged in',
                      value: action.value
                })
        }
}
```

b.
```
const auth = function(state = {status: "logged out", value: 'guest'}, action)
{
        switch (action.type)
        {
                case 'LOGIN':
                   let newState=Object.assign({}, state, {
                      status: 'logged in',
                      value: action.value
                })
                return newState;
        }
}
```

c.
```
const auth = function(state = {status: "logged out", value: 'guest'}, action)
{
        switch (action.type)
        {
                case 'LOGIN':
                   let newState= {
                      status: 'logged in',
                      value: action.value
                })
                return newState;
        }
}
```

d.
```
const auth = function(state = {status: "logged out", value: 'guest'}, action)
{
        switch (action.type)
        {
                case 'LOGIN':
                state={
                      status: 'logged in',
                      value: action.value
                }
                return state;
        }
}
```

- Both a and b
- Only a
- Both c and d
- All of the given options

Q4 of 30

Consider the below code snippet

```
const mapStateToProps = (state) => {
 return {

   courses: state.courses
 }
};

const mapDispatchToProps = (dispatch) => {
 return {
   createCourse: course => dispatch(courseActions.createCourse(course))
 }
};

const CourseDetails = () => {
return (
 // JSX elements
)
}
```

Identify the correct option to connect the React component to Redux stores:

- connect(mapStateToProps, mapDispatchToProps)(CourseDetails);
- connect(mapStateToProps(), mapDispatchToProps())(CourseDetails);
- connect(mapStateToProps(state), mapDispatchToProps(dispatch))(CourseDetails);
- connect(mapStateToProps), (mapDispatchToProps)(CourseDetails);

Q5 of 30

Given below are the different ways a JSX expressions might be used to evaluate a certain condition. Choose the option(s) with no error:

a.

```
return < div> {name === "Jade" ? "Hello, Jade!" : "No name found"} < /div>;
```

b.

```
if (name === "Jade") {
  var message = "Hello, Jade!";
} else {
  message = "No name found";
}
return < div>{message}< /div>;
```

- Option a
- Option b
- Both a and b
- None of the options

Q6 of 30

Consider the below given code snippet (assume all the required libraries are imported with index.html page):

```
function Warning(props){
 if(!props.warn){
    return < div>No Warning..!!< /div>;
 }
 return(
    < div>Warning..!!< /div>
 )
}
function DisplayWarning(){
 const [status,setStatus] = useState(true)
    return(
    < Warning warn={status}/ >
    )
}
ReactDOM.render(< DisplayWarning / >, document.getElementById('root'))
```

Choose the correct output from below options when it is rendered:

- ○ Component rendered without any output
- ○ No Warning..!!
- ◉ Warning..!!
- ○ Error: Warning Component not found

Q7 of 30

The counter is passed with the initial value 0 to the reducer and the counter will be incremented by 1 when the increment button is clicked. When clicked on the increment button CallIncrement() function is called.

The action file is as shown below

```
/* action file */
function onIncrement(step){
  return {
    type:"INC",
    step
  }
}

export function CallIncrement(){
  return (dispatch,getState)=>{
    setTimeout(()=>{
                console.log(getState())
        dispatch(onIncrement());
    },3000)
  }
}
```

What value would be logged to the console when the increment button is clicked for the third time?

- Error because getState() method can be called only using store object
- 3
- 2
- Error because getState() method cannot be called within middleware function

Q8 of 30

Consider the below code written for rendering a submit button of a component.

< Button type="submit"> Submit < /Button>

The requirement is to apply the below react-bootstrap styles to the button:

- size of the button - 'small'
- button style - 'primary'

Choose the correct option to achieve the above requirement.

(Assume all required React-bootstrap imports are done)

- < Button type="submit" style="primary" size="sm"> Submit < /Button>
- < Button type="submit" variant="primary" size="sm"> Submit < /Button>
- < Button type="submit" class="primary" class="sm"> Submit < /Button>
- < Button type="submit" className="primary" className="sm"> Submit < /Button>

Q9 of 30

Tom (React application developer) has created the below component to display examination details for the students.

```
const ExamDetails = () => {
    return(Code to display examination details);
}
```

And he wants to display location as Mysore by default, if the preferred location is not provided by student at the form submission time.

Select an appropriate method to be added to the component from the below options.

- Set default value using defaultProps
- Set default value using propTypes
- Use forceUpdate() method and update state as Mysore
- Set default value using useState hook

Q10 of 30

Choose the correct statements about the below code:

```
const [name,setName]= useState('')

const [age,setAge] = useState(0)


useEffect(()=>{

console.log("useEffect method called")

},[age])
```

- useEffect will be called whenever name and age state changes
- (●) useEffect is invoked once after initial render and then everytime when age state changes
- useEffect is invoked once once after initial render
- useEffect is not invoked

Q11 of 30

Listed below are the two components Home and Product. When you click the "Buy Now" button in the Product, that component should be added to the cart. Select the appropriate code from the options given to be written inside the handleClick() method in the Product component so that productId will be passed to the Home component. All the required import and export statements are included.

function Home() { const handleAdd = (productId) => { //Implement logic to purchase the specified product. }; return < Product onPurchase={handleAdd} / >; } export default Home; function Product(props) { const [productId, setProductId] = useState("p-023"); const handleClick = () => { //Select the appropriate line of code. }; return (< div> < div>MobilePhone< /div> < div>Price:1234< /div> < button onClick={handleClick}>Buy Now< /button> < /div>); }

- Home.handleAdd(productId);
- props.onPurchase(productId);
- handleAdd(productId);
- useNavigate(productId)

Q12 of 30

The useEffect() method will run whenever any state of the component changes, when there is no second argument passed to it.

Please validate the above statement.

- (●) True
- ( ) False

Q13 of 30

Consider the below Reducers.

```
const Reducer1 = (state={
    currentDepts:{
        a1:"a101",b1:"a102",c1:"a103",d1:"a104"
    }
},action) =>{
          return Object.assign({},state,{ currentDepts:{a1:"a201",b1:"a202",c1:"a203",d1:"a204"}});
}
export default Reducer1;


const Reducer2 = (state={
    currentDepts:{
        a2:"a101",b2:"a102",c2:"a103",d2:"a104"
    }
},action) =>{
          return Object.assign({},state,{ currentDepts:{a2:"a201",b2:"a202",c2:"a203",d2:"a204"}});
}
export default Reducer2;
```

Which of the below statements are true?
i. The value of a1 can be modified within Reducer2
ii. The state is shared between Reducer1 and Reducer2

- Both i and ii
- Neither i and ii
- Only i
- Only ii

Q14 of 30

Observe the below given JavaScript code and predict the JSX equivalent syntax from the given options.

```
const App = () => {
return  React.createElement("div", null, React.createElement ("h1", {}, "Hello World"), React.createElement("p", null));
}
export default App;
```

- < div> < h1> Hello World < /h1> < p>null< /p> < /div>
- < div> < h1> Hello World < /h1> < p> < /p> < /div>
- < div>null < h1> Hello World < /h1> < p > null< /p> < /div>
- < div null> < h1> Hello World < /h1> < p null > < /p> < /div>

Q15 of 30

Sophie is using React Router in her SPA and wants to create a link that can help her to navigate to "/products" URL. Select the appropriate line of code from the below options.

- < Link href=”/products” > Products< /Link >
- < a href=”/products” > Products< /a >
- < Link to=”/products”>Products< /Link >
- < Link path=”/products”>Products< /Link>

Q16 of 30

Match the following key points of ReactJS (Set A) with their appropriate options (Set B).

Set A:	Set B:
a. PascalCasing naming convention	a. Plain JavaScript
b. className	b. Component name
c. Props	c. class becomes className in JSX
d. React.createElement	d. Read - Only

- a - b b - c c - d d - a
- a - c b - b c - d d - a
- a - c b - a c - d d - b
- a - d b - c c - b d - a

Q17 of 30

Order the below steps to match the data flow in Redux

a. The root reducer combines the new state returned from multiple reducers into a single state

b. Component dispatches an action

c. The store gets the initial state from root reducer passed to it

d. The action reaches the root reducer

e. Store saves the entire state of the application

- (●) c,b,d,a,e
- ( ) c,b,a,d,e
- ( ) a,c,b,d,e
- ( ) a,c,d,b,e

Q18 of 30

James a front-end developer had a requirement to create a Single page application. He decides to create the application using React. Which of the below listed advantages can James get by using React?

a. Modularity can achieved by using React JS

b. Optimize DOM manipulation

c. Follows MVC architecture

d. Components can also be created using pure JavaScript

- (●) Only a and b are true
- ( ) Only a, c and d are true
- ( ) Only a, b and d are true
- ( ) All the given options are true

Q19 of 30

Jemmy, a React developer has a requirement of handling state in a functional component and she has written the below code

```
const App = () => {
const [count, setCount] = useState(0);
}
```

Which of the below statement should she use to increment the count variable of the above component?

- (●) setCount(count + 1);
- ( ) this.setState({ this.state.count + 1});
- ( ) this.setCount(count + 1);
- ( ) setCount(this.count + 1);

Q20 of 30

Match the following React methods (Set A) to its category (Set B).

Set A:	Set B:
a. ReactDOM.unmountComponentAtNode();	a. Handling props
b. useState	b. Return JSX elements
c. render()	c. Data handling
d. defaultProps	d. React DOM method

- a - c b - a c - d d - b
- a - c b - b c - d d - a
- a - d b - c c - b d - a
- a - a b - b c - d d - c

Q21 of 30

Consider the given two components Home and Contact.

The requirement is to render the Contact component within Home when the Home component is rendered

```
const Home = () => {
 return (
  < div>
   < p>
    Below, you can find the contact of the R&D Department.
    < Contact />
   < /p>
  < /div>
 );
};

const Contact = () => {
 return (
  < div>
   < p>123 Southern Avenue, Brooksville, Ph: 9999987878< /p>
  < /div>
 );
};

export default Home;
```

Select the appropriate way to achieve the requirement

(Assume all the required environment setup, import, and export is done.)(Select any two)

- [x] ReactDOM.render(< Home> < /Home>, document.getElementById('root'))
- [ ] ReactDOM.render(< Contact> , document.getElementById('root'))
- [x] ReactDOM.render(< Home />, document.getElementById('root'))
- [ ] ReactDOM.render({Contact}, document.getElementById('root'))

Q22 of 30

Mily wants to fetch the list of product data from a backend API. Received data will be set as the state of a component i.e. in 'result'. Which of the following code snippets will help her in achieving this requirement? (Assume all required set up is given)

```
const[result,setResult] = useState("")
```

- axios.get(“api_url”).subscribe(res => setResult(result))
- axios.fetch(“api_url”).then(res => setResult({result : res}))
- axios.get(“api_url”).then(res => setResult(res.data))
- axios.fetch(“api_url”).then(res => setResult(res.data))

Q23 of 30

Which of the following h1 element will be rendered with styles?

- < h1 style={{color: "Blue"; font-family:"Arial"}}> I'm styled < /h1>
- < h1 style={{color: "Blue font-family:Arial"}}> I'm styled < /h1>
- < h1 style={{color: "Blue",fontFamily:"Arial"}}> I'm styled < /h1>
- < h1 style={{color: "Blue",FontFamily:"Arial"}}> I'm styled < /h1>

Q24 of 30

John is developing a functionality, where on clicking view details button of a particular product from the list of prodcts, the details of that product should be displayed in a different view. For that, he is passing the productId through the URL. Choose the appropriate code snippet to achieve the requirement

- let params = useParams(); {params.productId}
- {useParams().productId}
- both a and b
- neither a nor b

Q25 of 30

Which of the below methods helps in unmounting a component from the DOM

- unmountComponentAtNode
- unmountComponent
- useEffect
- componentWillUnmount

Q26 of 30

Consider the below code snippet

```
function Dialog() {
 return (
   < >
    < Child>// JSX content to render a dialog box with some content< /Child>
   < />
 );
}
```

Which of the following option should be used within Child component to render the dialog box

- child
- IndexRoute
- children
- React.children

Q27 of 30

Which of the following statement is true with respect to forms in React?

a. ref is used to give the reference name to the form element

b. useRef hook returns a ref object

c. onChange prop must be added to make form interactive

- ○ Only a
- ○ Only b
- ○ Only a & b
- ◉ a, b, c

Q28 of 30

The below code snippet produces compilation error. Identify the options that helps in removing the compilation error (Assume all the required environment setup has been done with the HTML file ready):

```
const Button = () => {
  const [cntr, setCntr] = useState(100);

  return (
    < div>
      < button onClick={handleClick}>update< /button>
      < Resultant new={cntr} / >
    < /div>
  );
};

const Resultant = (props) => {
  const handleClick = () => {
    setCntr(cntr + 1);
  };

  return (
    < div>
      < h2>{props.new}< /h2>
    < /div>
  );
};
export default Button;

ReactDOM.render(< Button />, document.getElementById("root"));
```

render(){ return (< div>< button onClick={this.props.handleClick}>100< /button> <Resultant new={this.state.Cntr} / > < /div>); } } class Resultant extends React.Component{ handleClick(){ this.setState({Cntr:this.state.Cntr+1}); } render(){ return < div>< h2> {this.props.new}< /h2> < /div> } } ReactDOM.render(< Button / >, document.getElementById('app'));

- Passing handleClick method from Resultant to Button component
- Defining the handleClick method within the Button component
- Binding the handleClick method in Resultant component
- Defining the cntr state within Resultant component

Q29 of 30

Predict the output of the below given code snippet when it is rendered:

```
const Calculator = () => {
  var a = 7;
  var b = 6;
  var c = 5;
  return < h1>{"(a+b)-c"}< /h1>;
};
```

- a+b-c
- 8
- (a+b)-c
- 7

Q30 of 30

Consider the below Route mappings and identify which component will be rendered on encountering the route /

```
< Routes>
   < Route path="/" element={<Header />}>
    < Route index element={<Home />}/>
    < Route path="home" element={<Home />}/>
   < /Route>
 < /Routes>
```

- ○ Header
- ○ Home
- ◉ Header and Home
- ○ None of the components will be rendered