MODULE-5

PART-A

1. How do you create a React app?

npm create-react-app my-app

cd my-app

npm start

2. Explain JSX with a code example. Why can't browsers read it?

JSX stands for JavaScript XML.

JSX allows us to write HTML in React.

JSX makes it easier to write and add HTML in React.

JSX converts HTML tags into react elements.

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = <h1>I Love JSX!</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

Browsers can't read JSX because there is no inherent implementation for the browser engines to read and understand them. JSX is not intended to be implemented by the engines or browsers, it is intended to be used by various transpilers to transform these JSX into valid JavaScript code.

4. Give a code example to demonstrate embedding two or more components into one.

Child1.js

```
import React, { Component } from 'react';
class Child1 extends Component {
render() {
        return (
        <li>
                This is child1 component.
        </li>
        );
}
}
export default Child1;
```

Child2.js

```
import React, { Component } from 'react';
class Child2 extends Component {
render() {
        return (
        <li>
                This is Child2 component.
        </li>
        );
}
}
```

```
export default Child2;
```

App.js

```
import React, { Component } from 'react';
import Child1 from './components/child1';
import Child2 from './components/child2';
class App extends Component {
render() {
        return (
        <div>
                <div>This is a parent component</div>
                <Child1 />
                <Child2 />
        </div>
        );
}
}
export default App;
```

5. Give a code example to modularize code in React.

Filename: index.js

```
import React from "react";
import ReactDOM from "react-dom";
import "./index.css";
// Importing default export
import File from "./DefaultExport";
```

```
// Importing named exports
import { NamedExport } from "./NamedExport";
ReactDOM.render(
<React.StrictMode>
        <File />
        <NamedExport />
</React.StrictMode>,
document.getElementById("root")
);
```

Filename:DefaultExport.js

```
import React from "react";
const DefaultExport = () => {
return (
        <div>
        <h1>This is from default export</h1>
        <h2>Hello Coders</h2>
        </div>
);
};
// Default export
export default DefaultExport;
```

Filename:NamedExport.js

```
import React from "react";
const NamedExport = () => {
```

```
return (
        <div>
        <h1>This is from named export</h1>
        <h2>Nice to see you</h2>
        </div>
);
};
// Named Export
export { NamedExport };
```

6. Write a sample code to update the state of a component in React?

Update the State of Class-Based Components:

- Go inside the App.js file and clear everything.
- At the top of the App.js file import React,{Component} from 'react'.
- Create a Class based component named 'App'. This is the default App component that we have reconstructed.
- Create a state object named text, using this.state syntax. Give it a value.
- Create another method inside the class and update the state of the component using 'this.setState()' method.
- Pass the state object in a JSX element and call the method to update the state on a specific event like button click.

Filename:App.js

```
import React,{Component} from 'react';
class App extends Component {
constructor(){
        super()
        this.state={
        text : 'Welcome to Geeksforgeeks'
        }
}
goPremium(){
        this.setState({
        text:'Subscription successful'
        })
}
render() {
        return (
                <div>
                        <h1>{this.state.text}</h1>
                        <button onClick={() => this.goPremium()}>
                        Go Premium
                        </button>
                </div>
        );
}
}
export default App;
```

7. How do you implement React routing

Step 1 - Install a React Router

A simple way to install the **react-router** is to run the following code snippet in the **command prompt** window.

```
C:\Users\username\Desktop\reactApp>npm install react-router
```

Step 2 - Create Components

In this step, we will create four components. The **App** component will be used as a tab menu. The other three components **(Home), (About)** and **(Contact)** are rendered once the route has changed.

main.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Router, Route, Link, browserHistory, IndexRoute } from 'react-router'

class App extends React.Component {
  render() {
    return (
      <div>
        <ul>
        <li>Home</li>
        <li>About</li>
        <li>Contact</li>
        </ul>
        {this.props.children}
      </div>
    )
  }
}
export default App;

class Home extends React.Component {
  render() {
    return (
      <div>
        <h1>Home...</h1>
      </div>
    )
  }
}
export default Home;

class About extends React.Component {
  render() {
    return (
      <div>
        <h1>About...</h1>
      </div>
    )
  }
}
export default About;
```

```
class Contact extends React.Component {
  render() {
    return (
      <div>
        <h1>Contact...</h1>
      </div>
    )
  }
}
export default Contact;
```

Step 3 - Add a Router

Now, we will add routes to the app. Instead of rendering **App** element like in the previous example, this time the **Router** will be rendered. We will also set components for each route.

main.js

```
ReactDOM.render((
  <Router history = {browserHistory}>
    <Route path = "/" component = {App}>
      <IndexRoute component = {Home} />
      <Route path = "home" component = {Home} />
      <Route path = "about" component = {About} />
      <Route path = "contact" component = {Contact} />
    </Route>
  </Router>
), document.getElementById('app'))
```

8. Web Application without Redux

Yes, you can create web application without redux. Redux has nothing to do with React. They can work together, but it's not a rule of thumb. Although Redux is really helpful when managing states on large React applications, you can choose to use or not.

9. Web Application with Redux (Using Store and reducers)

Redux is a state management library that can be used to **manage the state** of any application, not just React. When using Redux, you will see that we may need to write more code to get the same things done. That is by design but your application state is more manageable, and testing becomes easy.

Three Pillars of Redux

Store

Action

Reducer

10. List some of the cases when you should use Refs.

- Managing focus, text selection, or media playback.
- Triggering imperative animations.
- Integrating with third-party DOM libraries.

PART-B

1. What is an event in React?

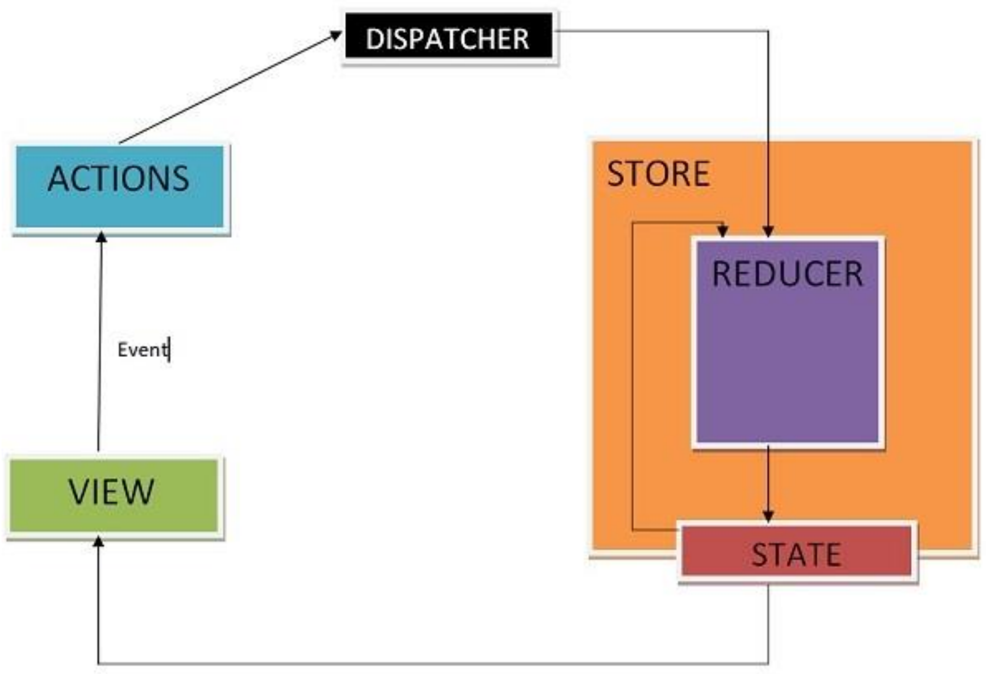
An event is an action that could be triggered as a result of the user action or system generated event. For example, a mouse click, loading of a web page, pressing a key, window resizes, and other interactions are called events.

React has its own event handling system which is very similar to handling events on DOM elements. The react event handling system is known as Synthetic Events. The synthetic event is a cross-browser wrapper of the browser's native event.

Handling events with react have some syntactic differences from handling events on DOM. These are:

1. React events are named as **camelCase** instead of **lowercase**.
2. With JSX, a function is passed as the **event handler** instead of a **string**.

2. Draw a diagram showing how data flows through Redux



3. How will you distinguish Redux from Flux?

Redux: Redux is a predictable state container for JavaScript apps. Redux itself library that can be used with any UI layer or framework, including React, Angular, Ember, and vanilla JS. Redux can be used with React, both are independent of each other. Redux is a state managing library used in JavaScript apps. It simply manages the state of your application or in other words, it is used to manage the data of the application. It is used with a library like React.

Flux: Flux is the application architecture or we can say JavaScript architecture that uses for building client-side web applications or UI for client applications. you can start using flux without a lot of new code. flux overcome the drawbacks of MVC such as instability and complexity.

4. What are the advantages of using Redux?

1. Centralized state management system i.e. Store

2. Performance Optimizations

3. Pure reducer functions

4. Storing long-term data

5. Time-travel Debugging

6. Great supportive community

5.Where would you put AJAX calls in your React code?

it is totally recommended that API calls should be made in **componentDidMount()** life cycle method.

6. You must've heard that "In React, everything is a component." What do you understand from the statement?

The building blocks of a React application's UI are called components. Any app UI created using React is divisible into a number of small independent and reusable pieces, known as components. React renders each of the components independent of each other. Hence, there is no effect of rendering a component on the rest of the app UI.

7. Can browsers read JSX?

Browsers can't read JSX because there is no inherent implementation for the browser engines to read and understand them. JSX is not intended to be implemented by the engines or browsers, it is intended to be used by various transpilers to transform these JSX into valid JavaScript code.

8. Define HOC in React?What are the benefits of HOC?

A higher-order component (HOC) is an advanced technique in React for reusing component logic. HOCs are not part of the React API, per se. They are a pattern that emerges from React's compositional nature. Concretely, **a higher-order component is a function that takes a component and returns a new component.**

9. What is a store in Redux?

A store is an immutable object tree in Redux. A store is a state container which holds the application's state. Redux can have only a single store in your application. Whenever a store is created in Redux, you need to specify the reducer.

The only way to change the state inside it is to dispatch an action on it.

A store is not a class. It's just an object with a few methods on it. To create it, pass your root reducing function to `createStore`.

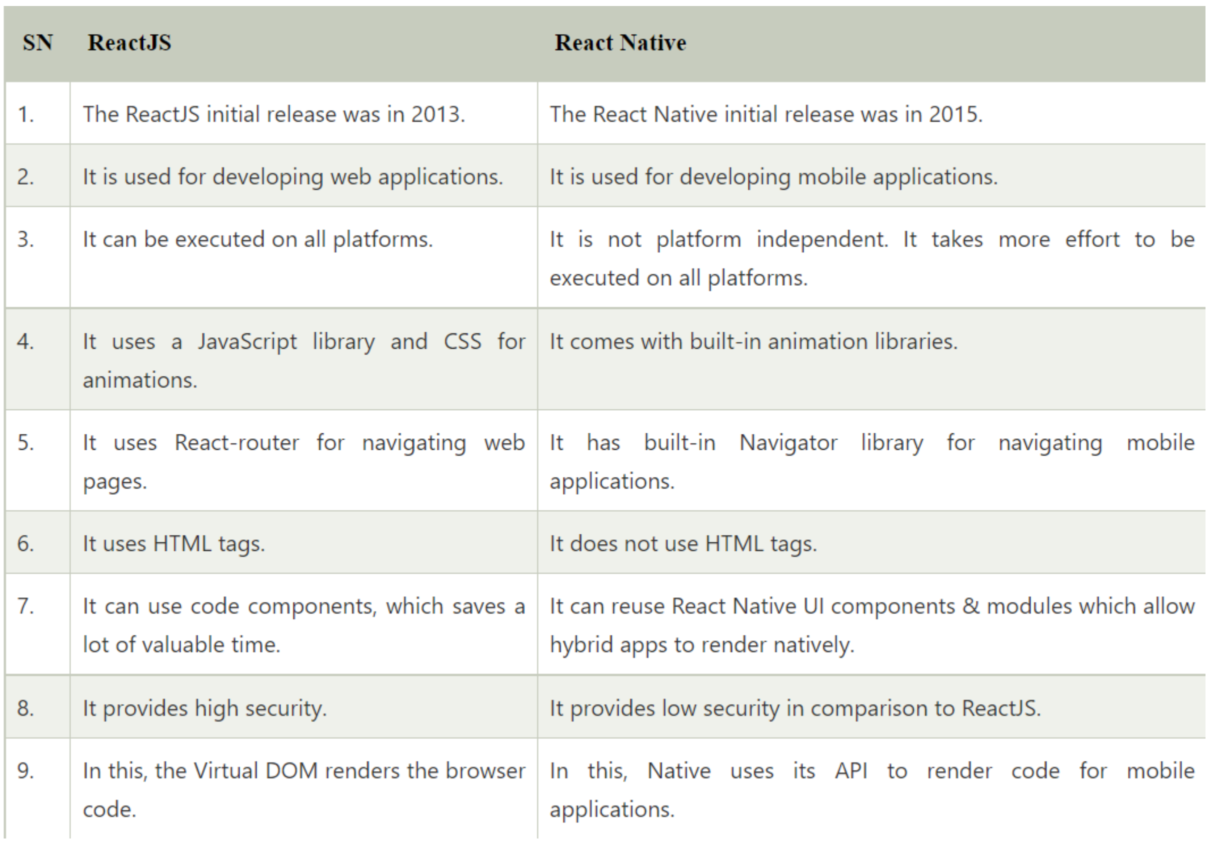
10. What is an arrow function and how is it used in React?

With arrow functions there is no binding of `this`.

In regular functions the `this` keyword represented the object that called the function, which could be the window, the document, a button or whatever.

With arrow functions, the `this` keyword *always* represents the object that defined the arrow function.

11. How is React different from React Native?



SN	ReactJS	React Native
1.	The ReactJS initial release was in 2013.	The React Native initial release was in 2015.
2.	It is used for developing web applications.	It is used for developing mobile applications.
3.	It can be executed on all platforms.	It is not platform independent. It takes more effort to be executed on all platforms.
4.	It uses a JavaScript library and CSS for animations.	It comes with built-in animation libraries.
5.	It uses React-router for navigating web pages.	It has built-in Navigator library for navigating mobile applications.
6.	It uses HTML tags.	It does not use HTML tags.
7.	It can use code components, which saves a lot of valuable time.	It can reuse React Native UI components & modules which allow hybrid apps to render natively.
8.	It provides high security.	It provides low security in comparison to ReactJS.
9.	In this, the Virtual DOM renders the browser code.	In this, Native uses its API to render code for mobile applications.

12. How is React different from Angular?

React is a declarative, efficient, and flexible JavaScript library for building user interfaces.ReactJS is an open-source, component-based front-end library responsible only for the view layer of the application. It is maintained by Facebook. React uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible. It designs simple views for each state in your application, and React will efficiently update and render just the right component when your data changes. The declarative view makes your code more predictable and easier to debug.

Angular is a popular open-source JavaScript framework created by Google for developing web applications. Front-end developers use frameworks like Angular or React for presenting and manipulating data efficiently. Updated Angular is much more efficient compared to the older version of Angular, especially, the core functionality was moved to different modules. That's why it becomes so much fast and smooth compare to the older one. Newly added angular CLI. With that package, you can create a scaffolding for your Angular project.

13. What are the components of Redux?

STORE: A Store is a place where the entire state of your application lists. It manages the status of the application and has a dispatch(action) function. It is like a brain responsible for all moving parts in Redux.

ACTION: Action is sent or dispatched from the view which are payloads that can be read by Reducers. It is a pure object created to store the information of the user's event. It includes information such as type of action, time of occurrence, location of occurrence, its coordinates, and which state it aims to change.

REDUCER: Reducer read the payloads from the actions and then updates the store via the state accordingly. It is a pure function to return a new state from the initial state.

14. What are pure components?

Pure Components restricts the re-rendering when there is no use of re-rendering of the component. Pure Components are Class Components which extends **React.PureComponent**.

15. How would you create a form in React?

16. What are the advantages of using Redux?

1. Centralized state management system i.e. Store

2. Performance Optimizations

3. Pure reducer functions

4. Storing long-term data

5. Time-travel Debugging

6. Great supportive community

17. What is Redux?

Redux is a predictable state container for JavaScript apps. As the application grows, it becomes difficult to keep it organized and maintain data flow. Redux solves this problem by managing application's state with a single global object called Store. Redux fundamental principles help in maintaining consistency throughout your application, which makes debugging and testing easier.

18. Web Application without Redux

Yes, you can create web application without redux. Redux has nothing to do with React. They can work together, but it's not a rule of thumb. Although Redux is really helpful when managing states on large React applications, you can choose to use or not.

19. What are the three principles that Redux follows?

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.

20. What do you understand by "Single source of truth"?

Single source of truth

The global state of your application is stored in an object tree within a single store.

This makes it easy to create universal apps, as the state from your server can be serialized and hydrated into the client with no extra coding effort. A single state tree also makes it easier to debug or inspect an application; it also enables you to persist your app's state in development, for a faster development cycle. Some functionality which has been traditionally difficult to implement - Undo/Redo, for example - can suddenly become trivial to implement, if all of your state is stored in a single tree.