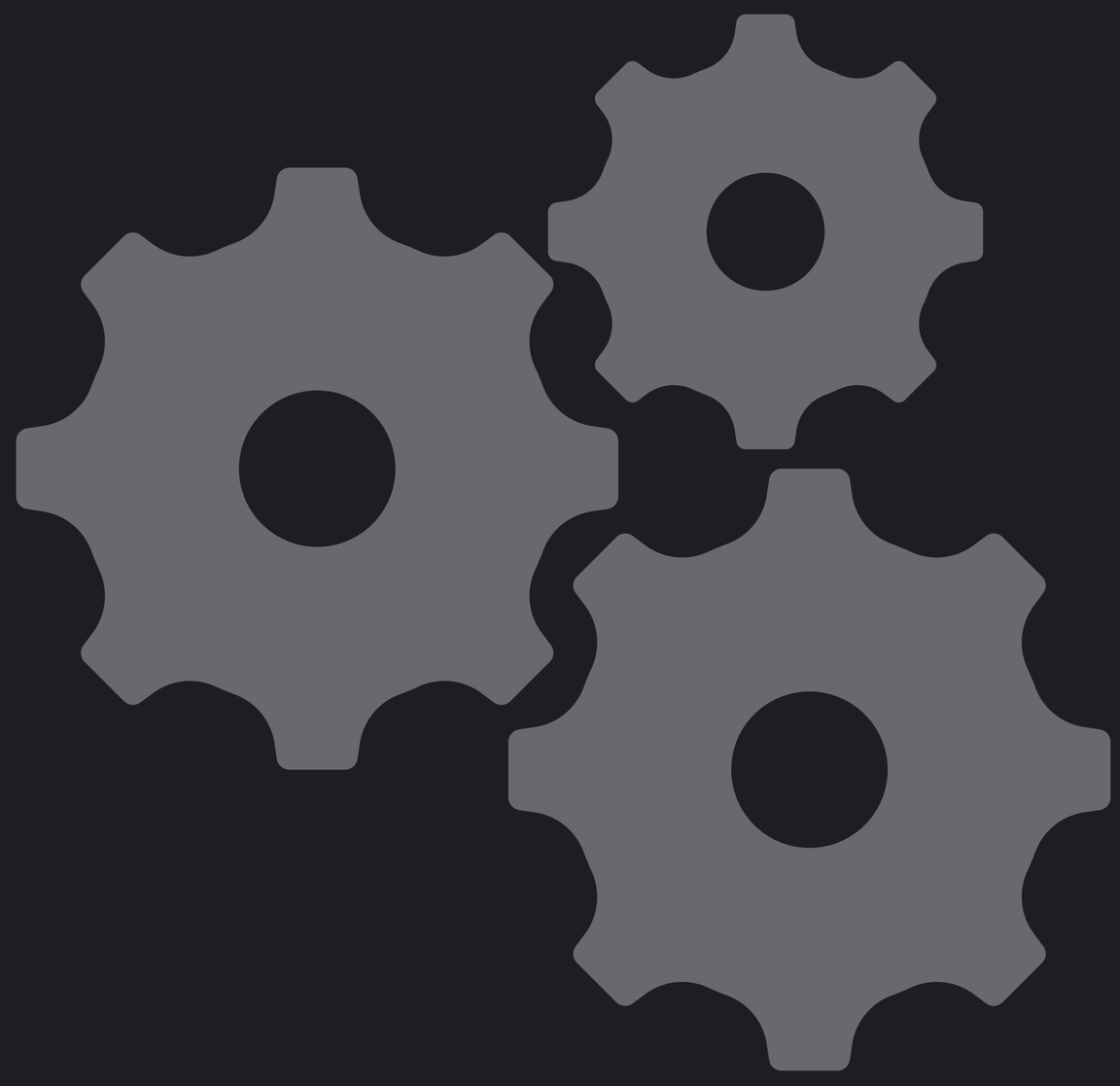
CodeWithAnjum
@codewithanjum



HOW REACT WORKS

React

Swipe →

React.js
is a popular JavaScript library used for building user interfaces. It was developed by Facebook and has gained widespread adoption in the web development community.

Here's how React works:

Component-based Architecture:
React follows a component-based architecture, where the UI is divided into reusable, self-contained components. Each component represents a part of the user interface and can contain its own logic, styling, and data.

Virtual DOM (Document Object Model):
One of the key concepts in React is the Virtual DOM. Instead of directly manipulating the actual DOM, React creates a lightweight virtual representation of the DOM. This virtual representation is kept in memory and is synchronized with the actual DOM in an efficient manner.

Render Method:
Each React component has a **render()** method that returns a description of what the UI should look like based on its current state and props (properties).

Component Lifecycle:
React components have a lifecycle with various methods that are called at different stages, such as componentDidMount(),componentDidUpdate(), and componentWillUnmount(). These lifecycle methods allow developers to control what happens when a component is created, updated, or removed from the DOM.

State Management:

React components can have state, which is a JavaScript object that holds data specific to that component. When the state of a component changes, React automatically re-renders the component, updating the Virtual DOM.

One-way Data Binding:

React follows a one-way data binding approach. Data flows from the parent components to the child components. When the data changes in the parent component, it automatically updates the child component's data and triggers a re-rendering.

Event Handling:

React allows developers to define event handlers for various user interactions, such as button clicks or form submissions. These event handlers can update the component's state, leading to a re-render and, in turn, updating the UI.

Conditional Rendering:

Developers can use conditional statements within the render() method to conditionally show or hide components based on certain conditions or state values.

Component Reusability:

React promotes reusability by allowing developers to compose complex UIs from smaller, reusable components. This makes the codebase more organized and easier to maintain.

React Hooks:

Introduced in React 16.8, hooks provide a way to use state and other React features without writing class components. Hooks enable developers to use state and other React features in functional components, making the code more concise and easier to understand.

Overall, React.js provides a powerful and efficient way to build interactive, scalable, and maintainable user interfaces for web applications. Its focus on components, virtual DOM, and efficient rendering makes it a preferred choice for many developers when building modern web applications.

Did you Find this Post a **bit Useful?**

Tap on like button, Share with your friends & save for Later.