Contents

# 1 Introduction

## 1 basic command

/\*Use\*/ **“use strict”;** /\*in Script to type in strict mode in order to avoid bugs in js\*/

npx /\*borrow react app for project only \*/ /\*npm ka chota bhai \*/ use **npm start** /\*to start react app\*/ class === className tab == tabindex react js use single tag <h1> poppy </h1> will give error put full html code in <> code </> **npm run build**  /\*to deploy project in public domain and never deploy public private file in public project\*/

React has two types of Components

Function-based Component: In short they are the Javascript functions. Nowadays, Function-based components are primarily used.

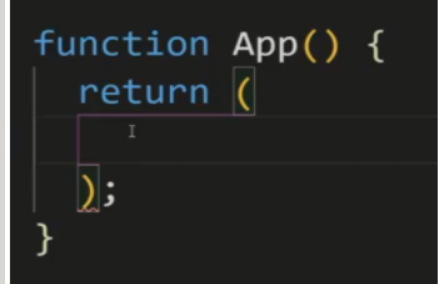
Class-based Component: They are quite complex than the function-based component as the class components can work with each other.

JSX: It stands for Javascript XML. It is the syntax extension of Javascript which allows HTML to co-exist with Javascript in React.

Some Points to remember while using JSX

You have to write ‘ClassName’ instead of ‘Class’ as class is the reserved keyword in Javascript

In place of ‘For’ you have to use ‘htmlFor’(not htmlfor).

Firstly, Remove the provided content of return() of App.js

## 2 FIRST HTML/XML AND CSS

function App(){ return( <>

<nav> <li>Home</li> <li>About</li> <li>Contact us</li> </nav>

<div className= “container”> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p> </div> </> );}

## 3 Props

/\*create components folder in src\*/ /\*create js props in components folder with first latter capital\*/ **‘rfc’** /\*it will generate code\*/

import React from 'react' export default function Navbar(props) { return ( <div> </div> ) } <Navbar title= “poppy” gf= “pogli”></ Navbar >/\*use such components to shorten the code\*+{props.title}+{props.gf}/ /\*if above is save as Navbar.js\*/ /\*in main app.js navbar component could be used to store all component of navbar\*/ /\*props is prop innitilization in Navbar\*/ /\*props.title will be replaced by title given by coder\*/ eg in above case poppy as title and pogli as gf\*/

**impt /\***it will give command for proptypes**\*/** import PropTypes from 'prop-types'

Navbar.propTypes = { title : PropTypes.string} /\*to define nature of props type\*/

Navbar.defaultProps = { Title : "Set title here" }; /\*when no props were used they will be shown in case of props \*/

Navbar.propTypes = { title : PropTypes.string.isRequired} /\*if prop is not present it will show an error\*/

<div my-4="true"> /\* to create space of 4px to make project look spacious\*/

LET name = “poppy”; {name} could be as variable woth object poppy stored in it. Import ‘ .\app.css’; /\*to import CSS in react above fn App()\*/

## 4 States and handling events

const [mode, setMode] = useState("light"); // Whether dark mode is enabled or not

  const toggleMode = () => { if (mode === "light") { setMode("dark");

document.body.style.backgroundColor = "#042743"; }

else { setMode("light"); document.body.style.backgroundColor = "white"; } };

/\*above fn to enable darkmode\*/

## 

## 5 function given to textbox

import React, {useState} from 'react' /\*to initialize state in js\*/

export default function TextForm(props) {

const[text, setText]=useState("Enter text here"); /\*State to get text input from user and rewrite text with uptext output and text as input\*/

const handleUpcase = ()=>{ let uptext=text.toUpperCase(); setText(uptext) }

const handleOnchange = (event)=>{ setText(event.target.value) } return (

/\* event.target.value syntax to get value form uptext and display in parent \*/

<textarea name="myBox" value= {text}  onChange={handleOnchange} rows="10"></textarea>   /\*onChange is the action which must be taken when text changes in textarea\*/

<button onClick={handleUpcase}>Convert to upper case</button> /\*onClick is the action which must be taken in case of clicking button\*/

className= mx-2; /\*It will give margin of 1 char between 2 button\*/

text.split(" ").length} /\*to find no of words in text output\*/

{text.length} /\*to find no of character in text output

## 6 dark mode

const toggleMode = () => { if (mode === "light") { setMode("dark"); /\*changing mode uding toggle button\*/ document.body.style.backgroundColor = "#042743"; showAlert("Dark mode has been enabled", "success"); /\*to enable alert\*/ document.title = 'TextUtils - Dark Mode'; /\* set document titles \*/}

else { setMode("light"); document.body.style.backgroundColor = "white"; showAlert("Light mode has been enabled", "success"); document.title = 'TextUtils - Light Mode'; } }; mode={mode} /\*to enable dark mode\*/

<div className={`form-check form-switch text-${props.mode==='light'?'dark':'light'}`}>

<input className="form-check-input" onClick={props.toggleMode} type="checkbox" id="flexSwitchCheckDefault"/> <label className="form-check-label" htmlFor="flexSwitchCheckDefault">Enable DarkMode</label> </div> /\*above to create toggle switch to enable darkmode

style={{color: props.mode==='dark'?'white':'#042743'}}