**Namaste React**

**Chapter 9: Jo Dikhta hai wo Bikta hai!**

How to style React Component?

1. Native CSS Styling

We are maintaining the index.css file and add all the css in this index.css file.

1. Using SCSS and SASS

Sass is a CSS pre-processor. Sass files are executed on the server and sends CSS to the browser. projects.

Install Sass by running this command in your terminal:

npm i sass

Create a Sass file the same way as you create CSS files, but Sass files have the file extension .scss. In Sass files you can use variables and other Sass functions:

Example

**my-sass.scss:**

Create a variable to define the color of the text:

$myColor: red;

h1 {

color: $myColor;

}

Import the Sass file the same way as you imported a CSS file:

import './my-sass.scss';

1. **Using Inline styling**

To style an element with the inline style attribute, the value must be a JavaScript object:

<div>

<h1 style={{color: "red"}}>Hello Style!</h1>

<p>Add a little style!</p>

</div>

**camelCased Property Names**

Since the inline CSS is written in a JavaScript object, properties with two names, like background-color, must be written with camel case syntax

**JavaScript Object**

You can also create an object with styling information, and refer to it in the style attribute

const mystyle = {

color: "white",

backgroundColor: "DodgerBlue",

padding: "10px",

fontFamily: "Arial"

};

return (

<div>

<h1 style={mystyle}>Hello Style!</h1>

<p>Add a little style!</p>

</div>

);

1. Using libraries like Material UI

What are the pro’s and cons of using libraries for CSS?

Pro’s

-easy to use

-reusable

-consistency in UI

Etc

**Tailwind CSS**

**What is Tailwind CSS?**

Tailwind CSS is a utility-first CSS framework designed to enable users to create applications faster and easier. You can use utility classes to control the layout, color, spacing, typography, shadows, and more to create a completely custom component design — without leaving your HTML or writing a single line of custom CSS.

Advantages of Tailwind CSS

* CSS on the go(in the same file)

We can write css on the go in the same file we can write html and css

* Reusability

Tailwind css comes with lot of prebuild classes that we can use multiple time

* Less bundle size

It is not very big framework, as it is minimal css that it offers to us it is less bundle size

* Flexible UI

We can customise it however we want to

**What is PostCSS**

PostCSS is a JavaScript tool that transforms your CSS code into an abstract syntax tree (AST) and then provides an API (application programming interface) for analyzing and modifying it using JavaScript plugins.

Despite its name, it is neither a post-processor nor a pre-processor, it is just a transpiler that turns a special PostCSS plugin syntax into a Vanilla CSS. You can think of it as the Babel tool for CSS.

In other words

Postcss config files in our project (postcss.config.js or .postcssrc) takes the configuration we give in it and tells our bundler(parcel) that in out project we will be seeing lot of tailwind classes so we need to configure it that when we are bundling things up compile the tailwind css into the normal css

When we are using the tailwind css we will not be writing any css inside of the index.css instead we will be adding below 3 lines only.

@tailwind base;

@tailwind components;

@tailwind utilities;