**Chapter 10: Jo Diktha Hai, Vo Bikta Hai**

***Theory:***

Q) Explore all the ways of writing CSS

1. **Inline CSS:** Inline CSS is written directly in the HTML document, within the style attribute of an HTML element

Eg: <p style="color: blue; font-size: 16px;">This is a paragraph with inline CSS.</p>

1. **Internal CSS:** Internal CSS is placed within the <style> element in the <head> section of an HTML document

Eg:

<head>

<style>

p {

color: red;

font-size: 18px;

}

</style>

</head>

<body>

<p>This is a paragraph with internal CSS.</p>

</body>

1. **External CSS:** External CSS is defined in a separate .css file and linked to the HTML document using the <link> element in the <head> section.

Eg:

/\* style.css \*/

p {

color: green;

font-size: 20px;

}

Then, link it in the html page using <link> Element

<head>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<p>This is a paragraph with external CSS.</p>

</body>

1. **CSS Frameworks:** We can also write CSS using CSS Libraries / CSS Framework such as Bootstrap, Chakra, Material UI, Ant Design, Tailwind etc.. Using these css libraries offers

* Faster development
* Consistency
* Easy maintenance
* Responsive design
* Cross browser compatibility

Q) How do we configure Tailwind?

A: Here are the steps to configure Tailwind in the APP for the “Parcel” Bundler.

1. First of all, install the tailwind css in the app using the command,

**npm install -D tailwindcss postcss**

1. Then run the command **npx tailwindcss init**. This will initialize the tailwind css by creating the configuration file ‘tailwind.config.js’ in the projects root directory
2. After that, create a .postcssrc file in the project root directory, and enable the tailwindcss plugin.
3. Add the Tailwind directives to your CSS by creating ./src/index.css file and add the @tailwind directives for each of Tailwind’s layers.

@tailwind base;

@tailwind components;

@tailwind utilities;

These are directives used in the project's CSS file to control which parts of the Tailwind CSS framework get included in the final CSS output

1. Finally start the Build once again to start using the tailwinf in the project

Q) In tailwind.config.css, what does all the keys mean (content, theme, plugins, extend)?

1. **Content:** The content section of tailwind.config.js file is where we configure the paths to all the files that contain Tailwind class names.
2. **Theme:** The theme section of tailwind.config.js file is where we define our project’s color palette, type scale, fonts, breakpoints, border radius values, and more.

We can provide default theme with a very generous set of values to get you started, but we can change it as and when we want the changes in the theme

1. **Plugins:** Plugins allows us to register new styles for Tailwind to inject into the user’s stylesheet using JavaScript instead of CSS.
2. **Extend:** the extends property in the tailwind.config.js file is used to extend or merge an external configuration file with the default Tailwind CSS configuration. It allows us to add or override utility classes, define custom styles, and tailor Tailwind's default configuration to suit our specific project requirements.

Q) Why do we have .postcssrc file?

A: The .postcssrc file is a configuration file for PostCSS, and it is used to specify the plugins that we want to process our CSS.

**NOTE:** PostCSS is a popular tool used in web development to process and transform CSS. It's a post-processor that takes standard CSS code and applies various transformations to it.

In our case, The bundler(Parcel in this app) uses postcss to understand the tailwind css.

***Coding:***

Q) Configure Tailwind and try to build your whole APP using Tailwind.

A: Done. It was really great experience using tailwind.

https://github.com/nandinihulsurkar/me-learning-react.git