<u>Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110 (An Autonomous Institution, Affiliated to Anna University, Chennai</u>

Internet Programming H.W. By:

Sai Rahul.T CSE B 3122215001090

Date:11/04/2024

Aim: The aim of this experiment is to design a ToDo list component using HTML elements and apply specific styling using CSS.

Learning Outcomes:

1. We understand the structure of HTML elements required to create a ToDo list component.

2. We learn to apply CSS styling using classes and IDs to achieve the desired visual appearance.

3. We gain proficiency in using relative positioning to adjust the layout of elements.

4. We enhance our understanding of object-oriented styling concepts in CSS.

5. We improve our skills in designing user interface components with a focus on aesthetics and usability.

6. We gain practical experience in implementing a simple interactive feature on a web page.

7. We learn to organize code effectively to maintain and update the ToDo list component.

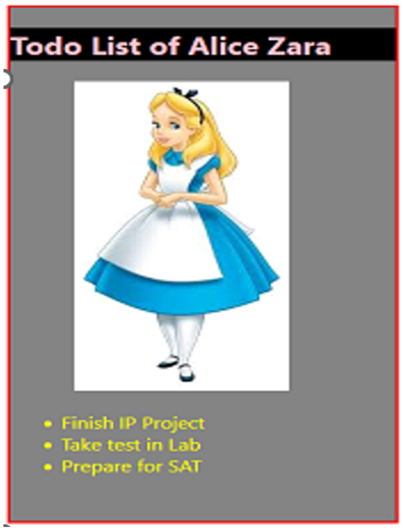
8. We understand the importance of user interface design in improving user experience.

9. We gain insights into the use of colors and borders to enhance visual appeal.

10. We develop problem-solving skills by overcoming challenges in implementing the desired design.

Question)

Design ToDo List as follows:
Create ToDo List Component with h1, img and ul HTML elements
Keep the elements in div
div:
border red color, text color - yellow and keep it in div using object in style
h1:
fname refers to 'Alice', lname to 'Zara' form person object, style it using id selector with background and text color.
Person object has name and theme
Theme is another object having background-color and color;
image
Use relative positioning with 20px



Todo List of Alice Zara

- Finish IP Project
- Take test in Lab
- Prepare for SAT

Answer)

Code:

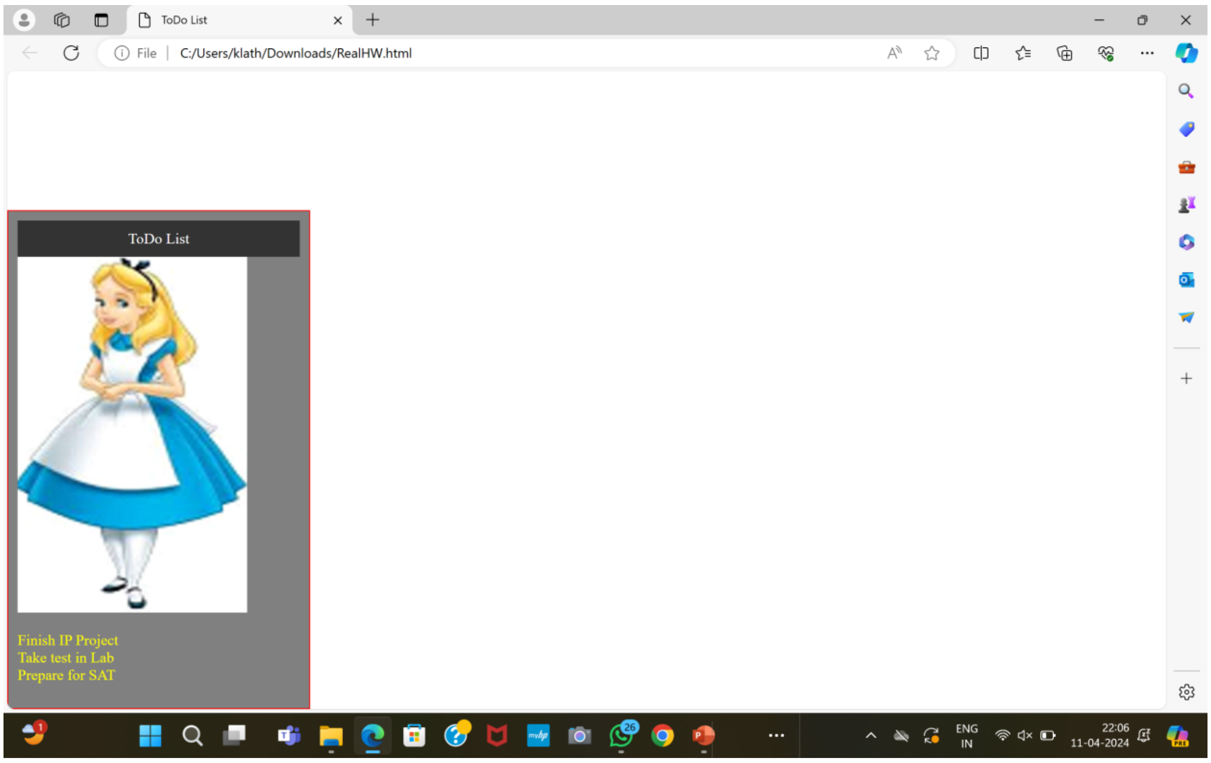
```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>ToDo List</title>
<style>
   .todo-container {
      position: absolute;
      bottom: 0;
      left: 0;
      border: 1px solid red;
      color: yellow;
      padding: 10px;
      width: 300px;
      background-color: grey;
   }

   #header {
      background-color: #333;
      color: white;
      padding: 10px;
      text-align: center;
   }

   #todo-list {
      list-style-type: none;
      padding: 0;
   }



   #todo-list img {
      width: 20px;
      margin-right: 5px;
      position: relative;
```

```
        top: 20px;
    }
</style>
</head>
<body>
    <div class="todo-container">
        <div id="header">ToDo List</div>
        <img src="Alice.png" alt="Checkmark">
        <ul id="todo-list">
            <li>Finish IP Project</li>
            <li>Take test in Lab</li>
            <li>Prepare for SAT</li>
        </ul>
    </div>
</body>
</html>
```

This code is saved as RealHW.html

Output:


ToDo List

Finish IP Project
Take test in Lab
Prepare for SAT

Explanation of the code:

1. The HTML '<!DOCTYPE html>' declaration indicates that the document is an HTML5 document.

2. The '<html lang="en">' tag specifies the language of the document as English.

3. The '<head>' section includes meta tags for character encoding and viewport settings.

4. The '<title>' tag sets the title of the document to "ToDo List."

5. The '<style>' tag contains CSS rules for styling the ToDo list container and its items.

6. The '.todo-container' class styles the container for the ToDo list, positioning it absolutely at the bottom of the page.

7. The 'border: 1px solid red;' property adds a red border around the container.

8. The 'color: yellow;' property sets the text color inside the container to yellow.

9. The 'padding: 10px;' property adds padding inside the container.

10. The 'width: 300px;' property sets the width of the container to 300 pixels.

11. The 'background-color: grey;' property sets the background color of the container to grey.

12. The '#header' ID styles the header of the ToDo list with a dark background, white text, and centered alignment.

13. The '#todo-list' ID styles the unordered list that contains the ToDo items, removing default list styles.

14. The 'list-style-type: none;' property removes the bullet points from the list items.

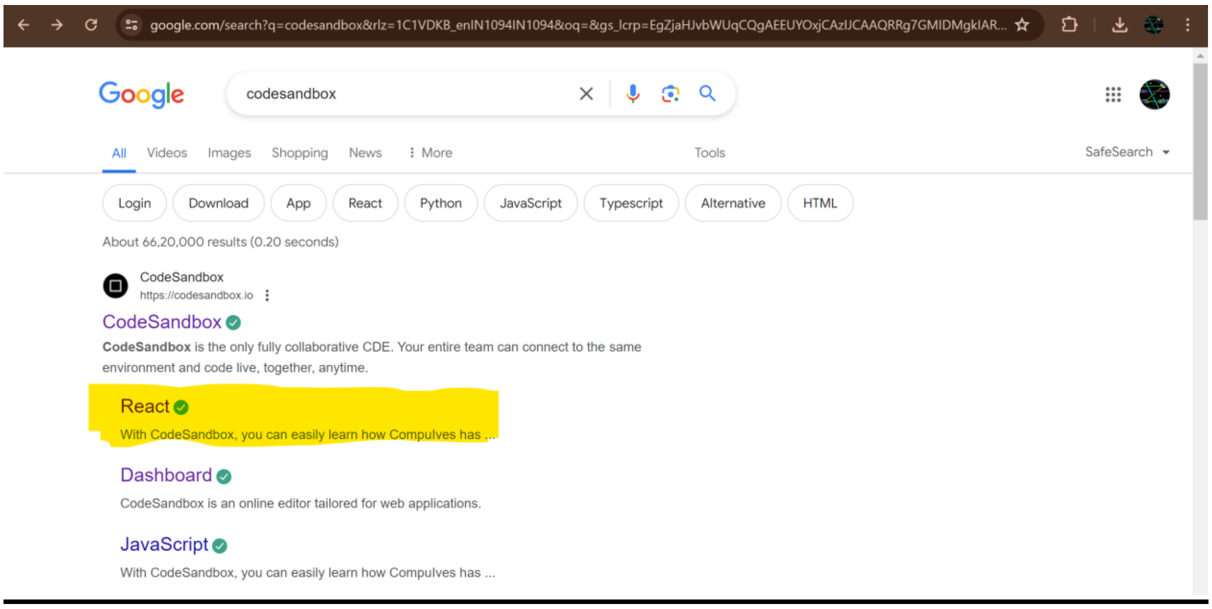
15. Inside the container, an '<img>' tag is used to display an image (Alice.png) before each list item.

16. The 'position: relative;' property in the '#todo-list img' selector sets the image's positioning relative to its normal position in the document flow.

17. The 'top: 20px;' property moves the image 20 pixels down from its normal position within the list item.

18. The document structure and styling combine to create a visually appealing and functional ToDo list interface.

We can do Using CodeSandbox(React/ReactJS) too:



About 66,20,000 results (0.20 seconds)

CodeSandbox
https://codesandbox.io

CodeSandbox

CodeSandbox is the only fully collaborative CDE. Your entire team can connect to the same environment and code live, together, anytime.

React

With CodeSandbox, you can easily learn how Compulves has ...

Dashboard

CodeSandbox is an online editor tailored for web applications.

JavaScript

With CodeSandbox, you can easily learn how Compulves has ...

The codes are:

Index.html:

```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
  <meta name="theme-color" content="#000000">
  <!--
      manifest.json provides metadata used when your web app is added
to the
      homescreen on Android. See
https://developers.google.com/web/fundamentals/engage-and-retain/web-
app-manifest/
    -->
  <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
  <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
  <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during
the build.
      Only files inside the `public` folder can be referenced from the
HTML.

      Unlike "/favicon.ico" or "favicon.ico",
"%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root
public URL.
      Learn how to configure a non-root public URL by running `npm run
build`.
    -->
  <title>React App</title>
</head>

<body>
  <noscript>
```

```
    You need to enable JavaScript to run this app.
  </noscript>
  <div id="root"></div>
  <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty
page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body>
tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn
build`.
    -->
</body>

</html>
```

App.js:

```
import "./styles.css";

export default function App() {
    return (
        <div className="todo-container">
            <div id="header">ToDo List</div>
            <img src="https://i.imgur.com/p9nvUi1.png"alt="hwimage" />
            <ul id="todo-list">
                <li>Finish IP Project</li>
                <li>Take test in Lab</li>
                <li>Prepare for SAT</li>
            </ul>
        </div>
    );
}
```

Index.js:

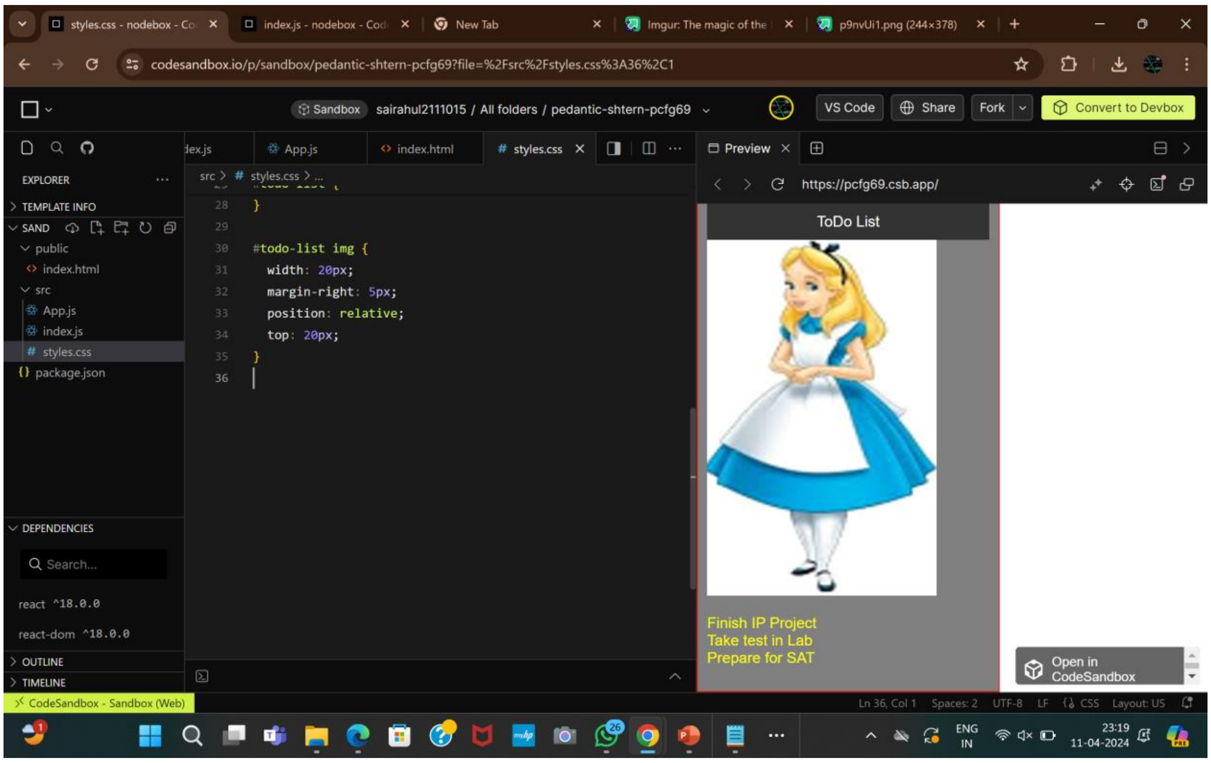
```
import { StrictMode } from "react";
import { createRoot } from "react-dom";

import App from "./App";

const rootElement = document.getElementById("root");
const root = createRoot(rootElement);

root.render(
    <StrictMode>
        <App />
    </StrictMode>
);
```

Styles.css:

```
body {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
}

.todo-container {
  position: absolute;
  bottom: 0;
  left: 0;
  border: 1px solid red;
  color: yellow;
  padding: 10px;
  width: 300px;
```

```css
  background-color: grey;
}

#header {
  background-color: #333;
  color: white;
  padding: 10px;
  text-align: center;
}

#todo-list {
  list-style-type: none;
  padding: 0;
}

#todo-list img {
  width: 20px;
  margin-right: 5px;
  position: relative;
  top: 20px;
}
```

Output:



```
}

#todo-list img {
  width: 20px;
  margin-right: 5px;
  position: relative;
  top: 20px;
}
```

ToDo List

Finish IP Project
Take test in Lab
Prepare for SAT

Result:

By the end of this experiment, we successfully create a visually appealing ToDo list component using HTML and CSS, demonstrating proficiency in designing user interface elements and applying styling techniques.