Exercise 2

Design and implement the user interface for the Application using Tailwind CSS

Aim:

Using **Tailwind CSS** in a **MERN (MongoDB, Express, React, Node.js)** stack project to speed up development with utility-first CSS classes.

Steps to Set Up Tailwind CSS in a MERN Stack Project

---

1. Install Tailwind CSS in the React Project

Now, let's set up Tailwind CSS in your React project (inside the `client` folder).

1. **Navigate to the `client` directory** and install Tailwind CSS dependencies.

   ```
   cd client
   npm install -D tailwindcss postcss autoprefixer
   ```

2. **Generate the `tailwind.config.js` and `postcss.config.js` files**. Tailwind provides a CLI to generate configuration files for PostCSS.

   ```
   npx tailwindcss init
   ```

   This will generate a `tailwind.config.js` file. You'll also need a `postcss.config.cjs` file:

   ```
   touch postcss.config.cjs
   ```

3. **Configure `tailwind.config.js`**: In the `tailwind.config.js`, make sure to set up the content path to include all your JSX/JS files. Here's an example of what the config might look like:

   ```
   // tailwind.config.js
   module.exports = {
     content: [
       "./src/**/*.{js,jsx,ts,tsx}",
     ],
     theme: {
       extend: {},
     },
     plugins: [],
   };
   ```

4. **Configure `postcss.config.js`**: Add the following content to the `postcss.config.cjs` file:

```
// postcss.config.js
module.exports = {
  plugins: {
    tailwindcss: {},
    autoprefixer: {},
  },
};
```

---

3. Add Tailwind Directives to Your CSS File

Now, you'll need to add the Tailwind CSS directives to your main stylesheet.

1. Create a new CSS file, for example, `src/index.css`, or open the existing one if it's already there.
2. Add the following lines to the `index.css` (or create a new one if it doesn't exist):

```
/* src/index.css */
@tailwind base;
@tailwind components;
@tailwind utilities;
```

---

4. Import the CSS File

Make sure your CSS file is being imported into your React app.

1. Open `src/index.js` (or `src/App.js` if necessary) and import the CSS file:

```
import './index.css';

// Or the file you used to import Tailwind's styles
```

2. Run the React development server to make sure everything is working:

```
npm run dev
```

At this point, Tailwind CSS should be fully integrated into your React frontend. You can now use Tailwind's utility classes to style your components.

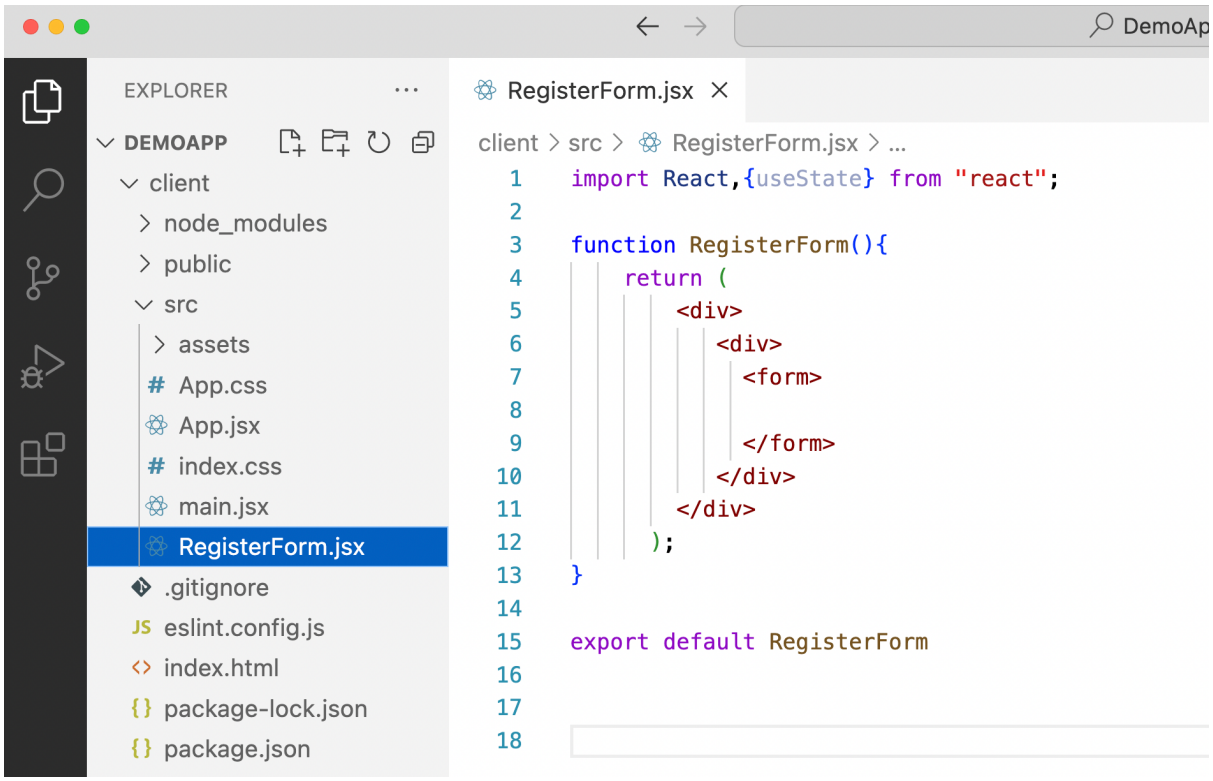
Question

Design a Registration Form for the allotted application themes[Refer GCR] using Tailwind CSS utility classes. [Refer Notes given below]

Procedure to add new JSX components to React client app

Step 1: Install Tailwind CSS as per the procedure given above.

Step 2: Create a new jsx file called RegisterForm.jsx in client/src folder with the following code




```
import React,{useState} from "react";

function RegisterForm(){
    return (
        <div>
          <div>
            <form>


            </form>
          </div>
        </div>
      );
}

export default RegisterForm
```

Step 3: Include the necessary <label> and form elements as per your application themes.

Step 4: Apply Tailwind utility classes to design an attractive User Registration Form.

Step 5: Include the Registration Form component in APP UI.

Open App.jsx file and include the registration form component using the component name <RegisterForm/>

```
import { useState,useEffect } from 'react'
import './App.css'
import axios from 'axios';
```

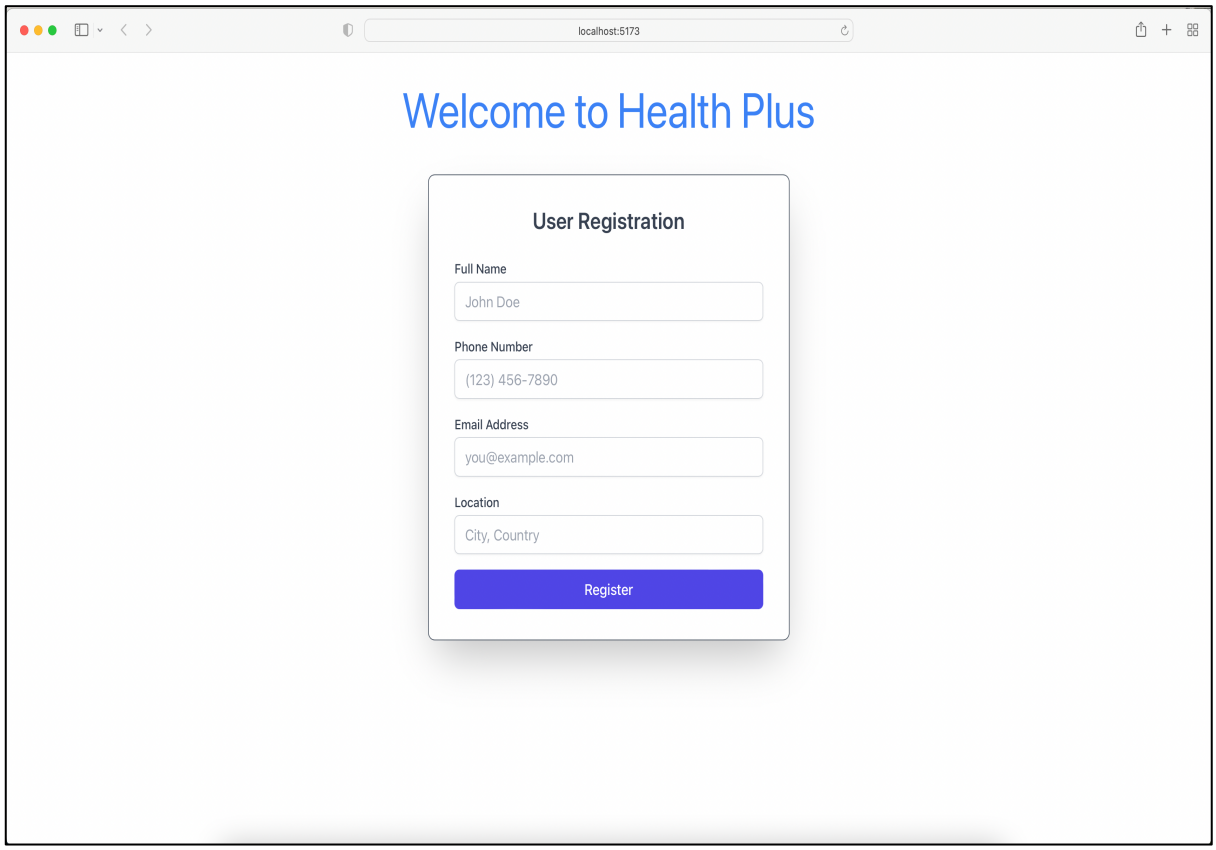
```
import './index.css';
import RegisterForm from './RegisterForm';

function App() {
  const [message,setMessage] = useState('')
  useEffect(()=>{
    //Fetch ApI
    axios.get("http://localhost:9000/")
         .then(response => {
            setMessage(response.data)
         })
         .catch(error=>{
           setMessage(error.message)
         })
  },[])

  return (
    <>
      <h1 className='text-blue-500'>Welcome to Health Plus</h1>
      <RegisterForm/>
    </>
  )
}

export default App
```

Sample Design



Basic Tailwind's Utility Classes Notes

1. container

The container class sets the max-width of an element to match the min-width of the current breakpoint. This is useful if you'd prefer to design for a fixed set of screen sizes instead of trying to accommodate a fully fluid viewport.

Example:

mx-auto – used to align the items in centre of the container
Refer: https://v1.tailwindcss.com/docs/container

2. flex

The CSS **flexbox** is a vital feature to develop the frontend, there is four flex available in tailwind CSS all the properties are covered as in class form. It is the alternative of CSS flex Property for fast development of front-end. It is used to set the length of flexible items. The flex class is much responsive and mobile-friendly. It is easy to positioning child elements and the main container. The margin doesn't collapse with the content margins. The order of any element can be easily changed without editing the HTML section.

Example:

```
<div class="flex m-2 text-white">
     <div class="bg-green-900 flex-1">
               Section 1
     </div>
     <div class="bg-green-800 flex-1">
               Section 2
     </div>
 </div>
```

Refer: https://v1.tailwindcss.com/docs/flex

3. Alignment

This class is used to describe the alignment of the flexible box container. It contains the space between and around content items along the main axis of a flex container. It is basically used for controlling how flex and grid items are positioned along a container's main axis.

Justify Content classes:

- justify-start
- justify-end
- justify-center

- justify-between
- justify-around
- justify-evenly

Refer: https://v1.tailwindcss.com/docs/justify-content

4. Tailwind CSS Padding

This class is used to create space around the element, inside any defined border. We can set different paddings for individual sides (top, right, bottom, left). It is important to add border properties to implement padding properties.

p-0 This class is used to define the padding on all sides.
py-0 This class defines padding on the y-axis (padding-top and padding-bottom).
px-0 This class defines padding on the x-axis (padding-left and padding-right).
pt-0 This class is specifically used to add padding on top.
pr-0 This class is specifically used to add padding on the right.
pb-0 This class is specifically used to add padding on the bottom.
pl-0 This class is specifically used to add padding on the left.

Here, 0 – represents size in 'rem' units, it can be vary from 0 to 20

Refer: https://v1.tailwindcss.com/docs/padding

5. *Tailwind CSS Margin*

This class is used to create space around the element, outside any defined border. We can set different margins for individual sides(top, right, bottom, left). It is important to add border properties to implement margin classes.

Example Classes:

m-0: This class is used to define the margin on all sides.
my-0: This class is used to define margin on the y-axis i.e margin-top and margin-bottom.
mx-0: This class is used to define margin on the x-axis i.e margin-left and margin-right.
mt-0: This class is specially used to add a margin on top.

Here, 0 – represents size in 'rem' units, it can be vary from 0 to 20

Refer: https://v1.tailwindcss.com/docs/margin

6. width and height

Refer:
https://v1.tailwindcss.com/docs/width
https://v1.tailwindcss.com/docs/height

7. Font Properties

Refer:
https://v1.tailwindcss.com/docs/font-family
https://v1.tailwindcss.com/docs/font-size
https://v1.tailwindcss.com/docs/font-style
https://v1.tailwindcss.com/docs/font-weight

8. Text Properties

Refer:
https://v1.tailwindcss.com/docs/text-align
https://v1.tailwindcss.com/docs/text-color
https://v1.tailwindcss.com/docs/text-opacity
https://v1.tailwindcss.com/docs/text-decoration
https://v1.tailwindcss.com/docs/text-transform

9. Background Properties

Refer:
https://v1.tailwindcss.com/docs/background-color
https://v1.tailwindcss.com/docs/background-image
https://v1.tailwindcss.com/docs/background-opacity
https://v1.tailwindcss.com/docs/background-position
https://v1.tailwindcss.com/docs/background-repeat
https://v1.tailwindcss.com/docs/background-size

10. Border Properties

Refer:
https://v1.tailwindcss.com/docs/border-color
https://v1.tailwindcss.com/docs/border-opacity
https://v1.tailwindcss.com/docs/border-style
https://v1.tailwindcss.com/docs/border-width
https://v1.tailwindcss.com/docs/border-radius
https://v1.tailwindcss.com/docs/divide-width

Rounded corners:
Use utilities like rounded-sm, rounded, or rounded-lg to apply different border radius sizes to an element.
11. Box Shadow

Refer: https://v1.tailwindcss.com/docs/box-shadow

12. Button

Refer: https://www.material-tailwind.com/docs/html/button

13. Form Input Tailwind CSS properties

Refer: https://www.material-tailwind.com/docs/html/input