ASSIGNMENT - 2

React.js Component Creation Assignment

Overview:

In this assignment, you will create reusable components like card components and buttons using React.js. You will apply your knowledge of JSX syntax, component structure, and state management to build these components.

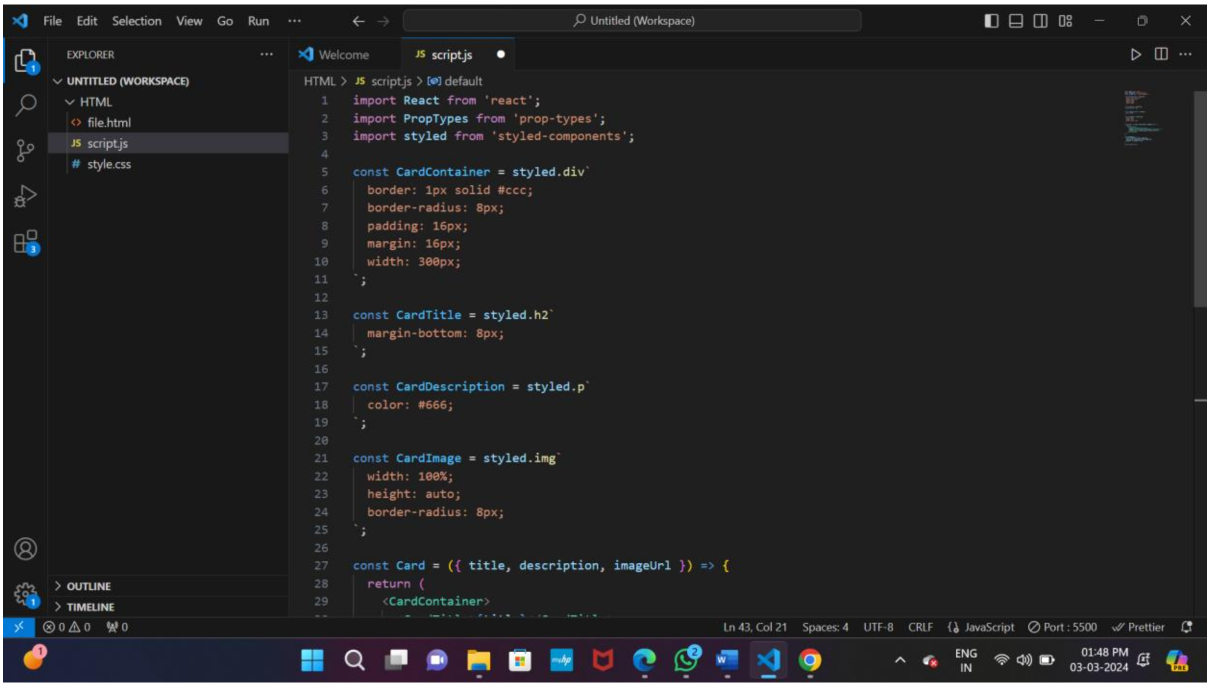
Objective:

The objective of this assignment is to familiarize yourself with React.js and practice creating reusable components. By completing this assignment, you will gain hands-on experience in building user interface elements using React.js.

Assignment Details:

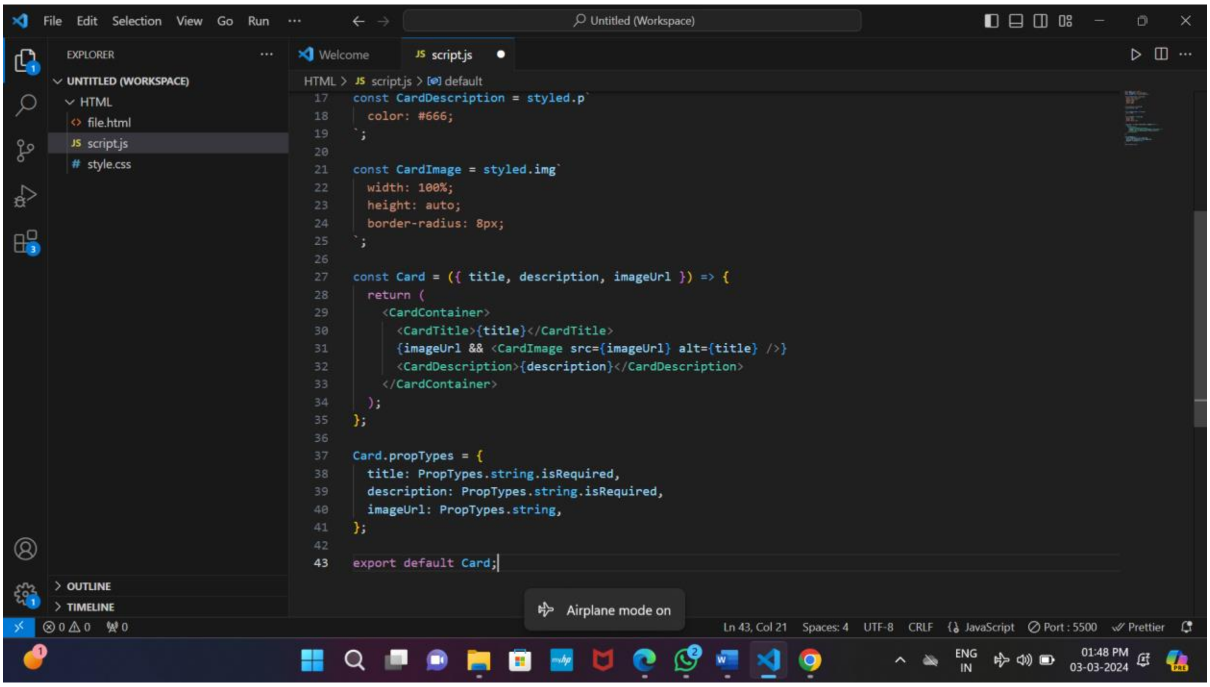
Card Component:

Create a reusable card component that displays information such as a title, description, and an optional image. The card should have a consistent layout and styling.



```
import React from 'react';
import PropTypes from 'prop-types';
import styled from 'styled-components';

const CardContainer = styled.div`
  border: 1px solid #ccc;
  border-radius: 8px;
  padding: 16px;
  margin: 16px;
  width: 300px;
`;

const CardTitle = styled.h2`
  margin-bottom: 8px;
`;

const CardDescription = styled.p`
  color: #666;
`;

const CardImage = styled.img`
  width: 100%;
  height: auto;
  border-radius: 8px;
`;

const Card = ({ title, description, imageUrl }) => {
  return (
    <CardContainer>
```

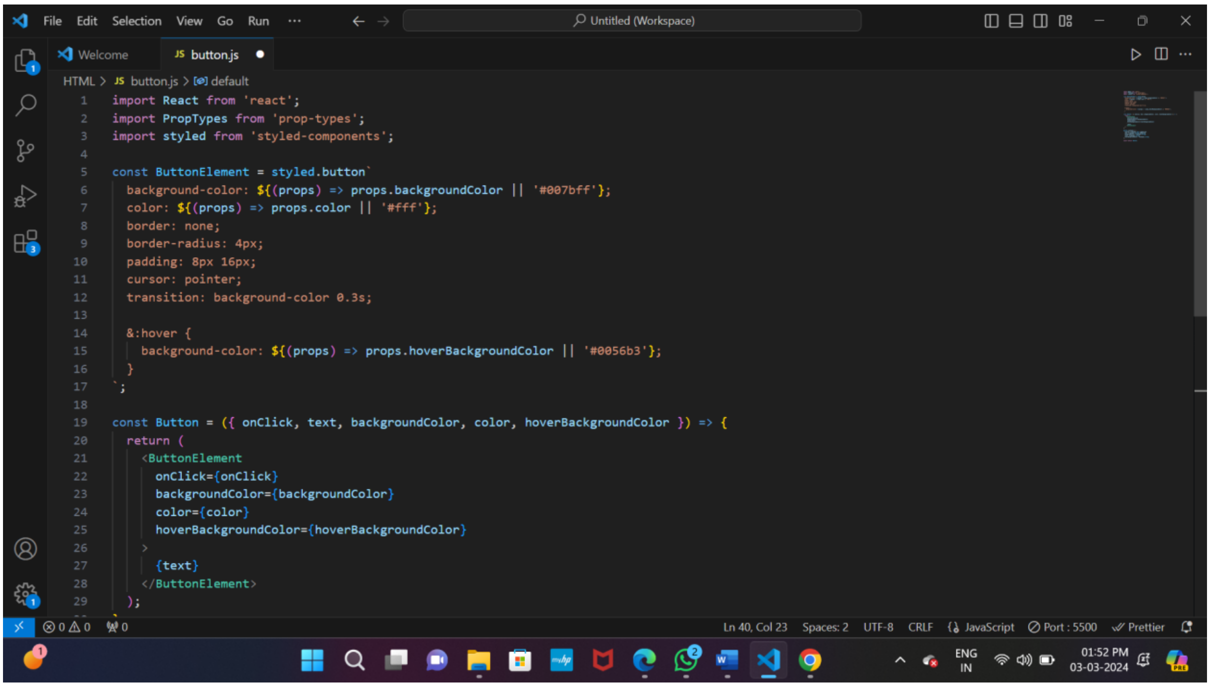
ASSIGNMENT - 2



```
const CardDescription = styled.p`
  color: #666;
`;

const CardImage = styled.img`
  width: 100%;
  height: auto;
  border-radius: 8px;
`;

const Card = ({ title, description, imageUrl }) => {
  return (
    <CardContainer>
      <CardTitle>{title}</CardTitle>
      {imageUrl && <CardImage src={imageUrl} alt={title} />}
      <CardDescription>{description}</CardDescription>
    </CardContainer>
  );
};

Card.propTypes = {
  title: PropTypes.string.isRequired,
  description: PropTypes.string.isRequired,
  imageUrl: PropTypes.string,
};

export default Card;
```

Button Component:

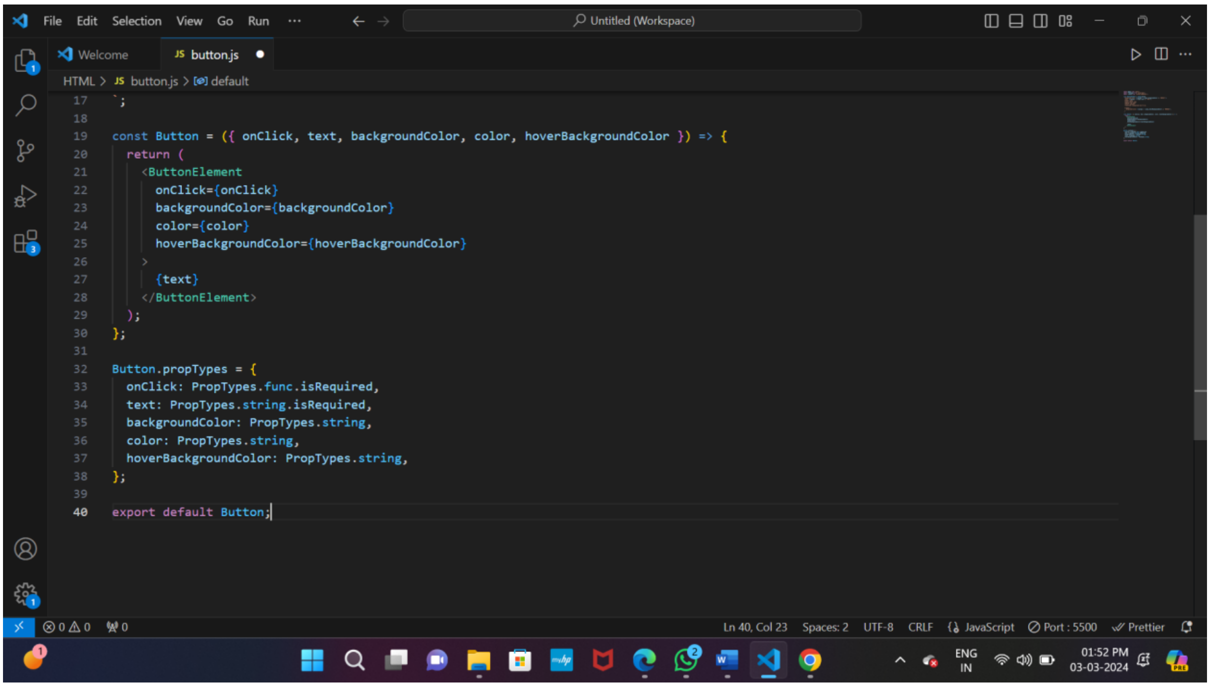
Create a reusable button component that can be customized with different styles and text. The button should be clickable and visually indicate when it is being hovered over or clicked.



```
import React from 'react';
import PropTypes from 'prop-types';
import styled from 'styled-components';

const ButtonElement = styled.button`
  background-color: ${(props) => props.backgroundColor || '#007bff'};
  color: ${(props) => props.color || '#fff'};
  border: none;
  border-radius: 4px;
  padding: 8px 16px;
  cursor: pointer;
  transition: background-color 0.3s;

  &:hover {
    background-color: ${(props) => props.hoverBackgroundColor || '#0056b3'};
  }
`;

const Button = ({ onClick, text, backgroundColor, color, hoverBackgroundColor }) => {
  return (
    <ButtonElement
      onClick={onClick}
      backgroundColor={backgroundColor}
      color={color}
      hoverBackgroundColor={hoverBackgroundColor}
    >
      {text}
    </ButtonElement>
  );
```

File Edit Selection View Go Run
Untitled (Workspace)
Welcome
button.js
HTML > button.js > default
Ln 40, Col 23 Spaces: 2 UTF-8 CRLF JavaScript Port : 5500 Prettier
ENG IN
01:52 PM 03-03-2024

ASSIGNMENT - 2

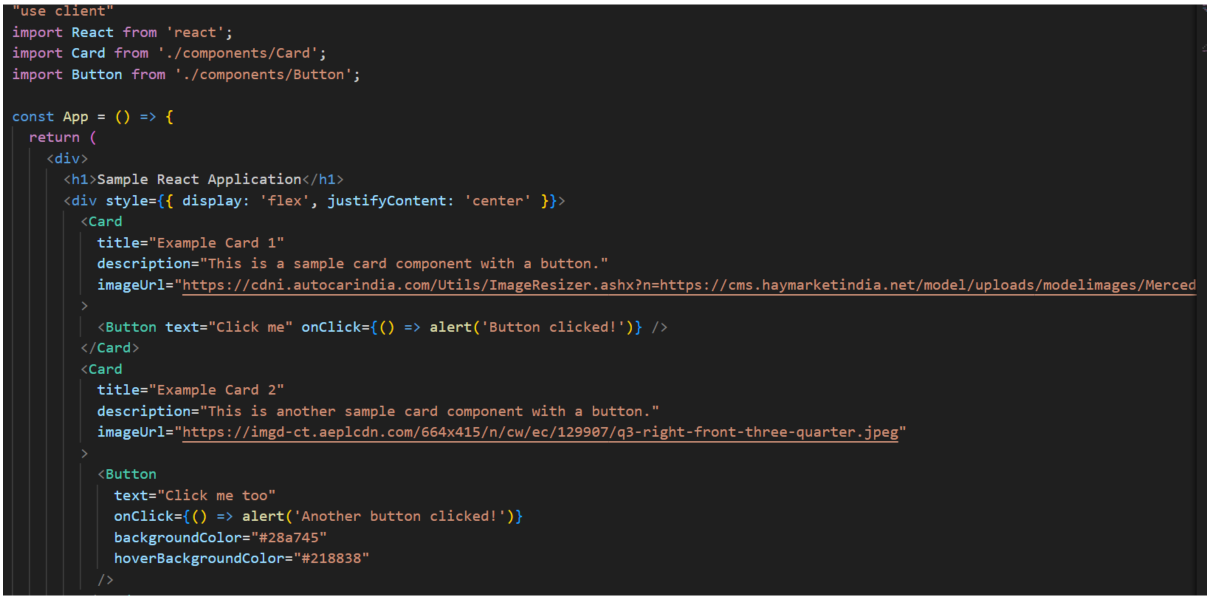


File Edit Selection View Go Run ···
Untitled (Workspace)
Welcome
button.js
HTML > button.js > default
Ln 40, Col 23 Spaces: 2 UTF-8 CRLF JavaScript Port : 5500 Prettier
ENG IN
01:52 PM
03-03-2024

```
`;

const Button = ({ onClick, text, backgroundColor, color, hoverBackgroundColor }) => {
  return (
    <ButtonElement
      onClick={onClick}
      backgroundColor={backgroundColor}
      color={color}
      hoverBackgroundColor={hoverBackgroundColor}
    >
      {text}
    </ButtonElement>
  );
};

Button.propTypes = {
  onClick: PropTypes.func.isRequired,
  text: PropTypes.string.isRequired,
  backgroundColor: PropTypes.string,
  color: PropTypes.string,
  hoverBackgroundColor: PropTypes.string,
};

export default Button;
```

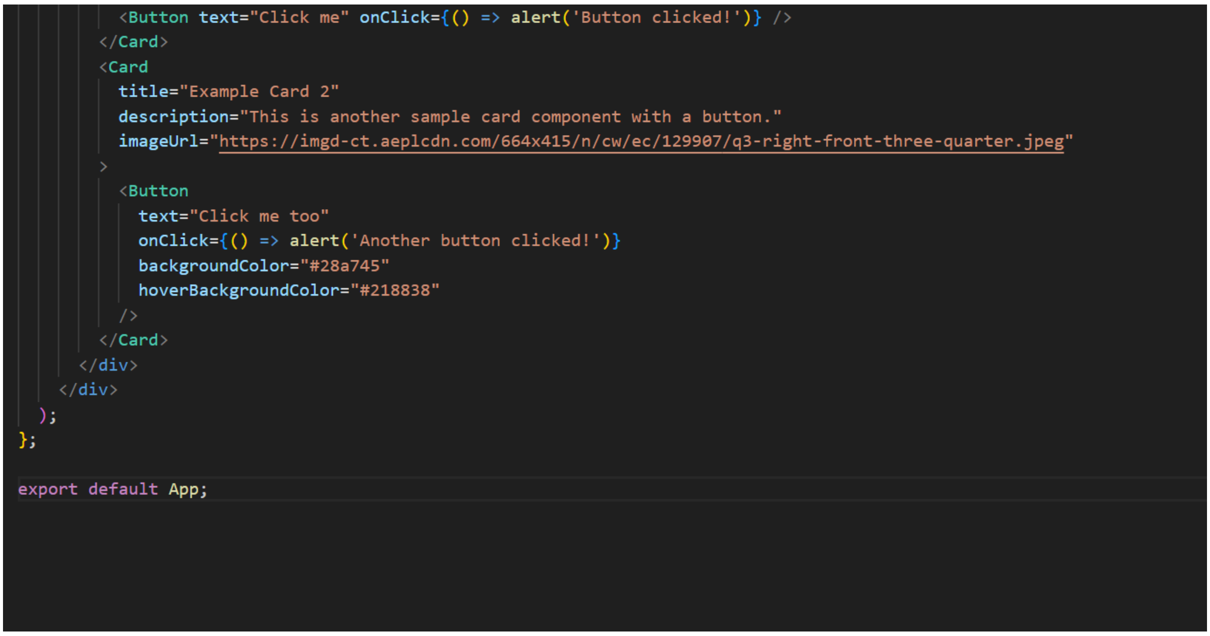
Component Integration:

Integrate the card and button components into a sample React application. Use these components to display a list of cards with buttons.



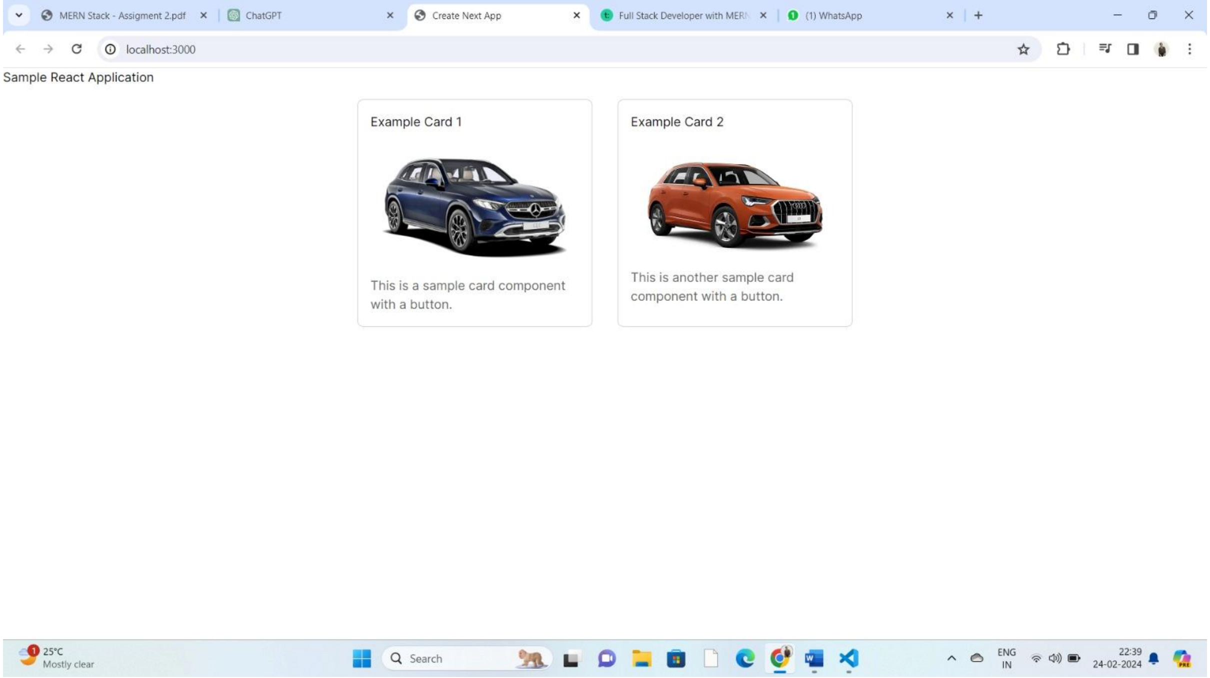
```
"use client"
import React from 'react';
import Card from './components/Card';
import Button from './components/Button';

const App = () => {
  return (
    <div>
      <h1>Sample React Application</h1>
      <div style={{ display: 'flex', justifyContent: 'center' }}>
        <Card
          title="Example Card 1"
          description="This is a sample card component with a button."
          imageUrl="https://cdni.autocarindia.com/Utils/ImageResizer.ashx?n=https://cms.haymarketindia.net/model/uploads/modelimages/Merced
        >
          <Button text="Click me" onClick={() => alert('Button clicked!')} />
        </Card>
        <Card
          title="Example Card 2"
          description="This is another sample card component with a button."
          imageUrl="https://imgd-ct.aeplcdn.com/664x415/n/cw/ec/129907/q3-right-front-three-quarter.jpeg"
        >
          <Button
            text="Click me too"
            onClick={() => alert('Another button clicked!')}
            backgroundColor="#28a745"
            hoverBackgroundColor="#218838"
          />
```

ASSIGNMENT - 2



```
          <Button text="Click me" onClick={() => alert('Button clicked!')} />
        </Card>
        <Card
          title="Example Card 2"
          description="This is another sample card component with a button."
          imageUrl="https://imgd-ct.aeplcdn.com/664x415/n/cw/ec/129907/q3-right-front-three-quarter.jpeg"
        >
          <Button
            text="Click me too"
            onClick={() => alert('Another button clicked!')}
            backgroundColor="#28a745"
            hoverBackgroundColor="#218838"
          />
        </Card>
      </div>
    </div>
  );
};

export default App;
```

OUTCOME ON LIVE SERVER:



Sample React Application

Example Card 1

This is a sample card component with a button.

Example Card 2

This is another sample card component with a button.

Guidelines: Use functional components to create the card and button components. Use props to pass data and event handlers to the components. Use CSS or a CSS-in-JS library like styled-components to style the components. Ensure that the components are reusable and can be easily customized with different data and styles.