

ASSIGNMENT -2

INSTITUTION NAME :SMART BRIDGE

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.

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.

Component Integration:

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

CARD COMPONENT :

REACT JS CODE

// Card.js

```
import React from 'react';
```

```
import './Card.css'; // Assuming you have a Card.css file
```

```
function Card({ imgSrc, imgAlt, title, description, buttonText, link }) {  
  return (  

```

```
    <div className="card">
```

```
      <img src={imgSrc} alt={imgAlt} />
```

```
      <div className="card-body">
```

```
        <h2>{title}</h2>
```

```
        <p>{description}</p>
```

```
        <a href={link}>{buttonText}</a>
```

```
      </div>
```

```
    </div>
```

```
  );
```

```
}
```

```
export default Card;
```

we can submit css code to make it look better

internal work of card component is done on react js but the look of the card is depend on css code

CSS CODE

```
/* Card.css */
```

```
.card {  
  border: 1px solid #ddd;  
  border-radius: 8px;  
  overflow: hidden;  
  margin: 16px;  
  width: 300px; /* Adjust the width as needed */  
}
```

```
.card img {  
  width: 100%;  
  height: auto;  
}
```

```
.card-body {  
  padding: 16px;  
}
```

```
.card-body h2 {  
  margin-bottom: 8px;  
}
```

```
.card-body p {  
  color: #555;  
}
```

```
.card-body a {  
  display: inline-block;  
  padding: 8px 16px;  
  background-color: #007bff;  
  color: #fff;  
  text-decoration: none;  
  border-radius: 4px;  
  margin-top: 8px;  
}
```

BUTTON COMPONENT:

Creating button component:

```
// Button.js
import React from 'react';

const Button = ({ label, onClick }) => (
  <a href="#" onClick={onClick} className="card-button">
    {label}
  </a>
);

export default Button;
```

Integrating button component into my card component:

```
// Card.js
import React from 'react';
import Button from './Button'; // Make sure to adjust the path based on your
project structure

const Card = ({ title, description, imageUrl, buttonLabel, onClick }) => (
  <div className="card">
    <img src={imageUrl} alt="Card Image" />
    <div className="card-body">
      <h2>{title}</h2>
      <p>{description}</p>
      <Button label={buttonLabel} onClick={onClick} />
    </div>
  </div>
);

export default Card;
```

Styling the button component

`/* Add this to your existing CSS file or create a new one */`

```
.card-button {  
  display: inline-block;  
  padding: 8px 16px;  
  background-color: #007bff;  
  color: #fff;  
  text-decoration: none;  
  border-radius: 4px;  
  margin-top: 8px;  
}  
  
.card-button:hover {  
  background-color: #0056b3;  
}
```

Using button component in our application

```
// App.js or any other component where you want to use the Card  
import React from 'react';  
import Card from './Card'; // Adjust the path based on your project structure  
  
const App = () => {  
  const handleClick = () => {  
    // Define the action you want to perform on button click  
    console.log('Button clicked!');  
  };  
  
  return (  
    <div>  
      <Card  
        title="Example Card"  
        description="This is a sample card description."  
        imageUrl="https://example.com/image.jpg"  
        buttonLabel="Click me"  
        onClick={handleClick}  
      />  
    </div>  
  );  
};  
  
export default App;
```

CONCLUSION:

the provided code establishes a flexible and visually appealing Card component for displaying content. The component incorporates responsive styling, an image, title, description, and a customizable button. The integration of a Button component enhances modularity and allows for seamless interaction within the Card. By following this approach, the code promotes maintainability, reusability, and a clean separation of concerns, contributing to an organized and efficient React application structure.

THE LOOK OF MY CARD COMPONENT:

