

Experiment-7

Develop a React Application featuring a ProfileCard component to display a user's profile information, including their name, profile picture, and bio. The component should demonstrate flexibility by utilizing both external CSS and inline styling for its design. Display the following information: Profile picture, User's name, A short bio or description Use an external CSS file for overall structure and primary styles, such as layout, colors, and typography. Apply inline styles for dynamic or specific styling elements, such as background colors or alignment. Design the ProfileCard to be visually appealing and responsive. Ensure the profile picture is displayed as a circle, and the name and bio are appropriately styled. Add hover effects or animations to enhance interactivity. Allow the background color of the card to change dynamically based on a prop or state.

App.js

```
import React from 'react';
import ProfileCard from './ProfileCard';

function App() {
  return (
    <div className="App">
      <h1 style={{ textAlign: 'center', marginTop: '20px' }}>User Profile</h1>
      <ProfileCard
        name="John Doe"
        bio="A passionate developer with a love for learning and coding."
        profilePicture="https://upload.wikimedia.org/wikipedia/commons/4/45/John_H
        ealey_Official_Cabinet_Portrait%2C_July_2024_%28cropped%29_2.jpg"
        backgroundColor="pink"
      />
    </div>
  );
}
```

```
}  
  
export default App;
```

App.css

```
.app {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
  background-color: #f0f0f0;  
}
```

ProfileCard.js

```
import React, { useState } from 'react';  
import './ProfileCard.css'; // Import external CSS file for overall styles  
const ProfileCard = ({ name, bio, profilePicture, backgroundColor }) => {  
  // Inline styles for dynamic elements  
  const inlineStyles = {  
    backgroundColor: backgroundColor || '#f0f0f0', // Default background if no  
    prop passed  
    textAlign: 'center',  
    padding: '20px',  
    borderRadius: '8px',  
    boxShadow: '0 4px 6px rgba(0, 0, 0, 0.1)',  
    transition: 'background-color 0.3s ease',  
  }; // Hover effect using inline styles
```

```
const [isHovered, setIsHovered] = useState(false);

const hoverStyles = {
  backgroundColor: isHovered ? '#e0e0e0' : inlineStyles.backgroundColor, //
  Change on hover
};

const handleMouseEnter = () => setIsHovered(true);
const handleMouseLeave = () => setIsHovered(false);

return (
  <div
    className="profile-card"
    style={{ ...inlineStyles, ...hoverStyles }}
    onMouseEnter={handleMouseEnter}
    onMouseLeave={handleMouseLeave}
  >
    <img
      src={profilePicture}
      alt="Profile"
      className="profile-image"
    />
    <h2>{name}</h2>
    <p>{bio}</p>
  </div>
);

};

export default ProfileCard;
```

ProfileCard.css

```
ProfileCard.css.profile-image {  
  width: 150px;  
  height: 150px;  
  border-radius: 50%; /* Makes the image circular */  
  object-fit: cover;  
  margin-bottom: 15px;  
  transition: transform 0.3s ease;  
}  
  
.profile-image:hover {  
  transform: scale(1.1); /* Slight zoom effect when hovered */  
}  
  
/* General styling for profile card */  
.profile-card {  
  max-width: 300px;  
  margin: 0 auto;  
  padding: 20px;  
  text-align: center;  
  background-color: #225f83;  
  border-radius: 8px;  
  box-shadow: 0 4px 6px rgba(143, 26, 26, 0.1);  
  transition: transform 0.3s ease;  
}  
  
.profile-card:hover {  
  transform: translateY(-5px); /* Slight lift effect on hover */  
}
```

```
h2 {  
  font-size: 1.5em;  
  margin: 10px 0;  
  font-family: 'Arial', sans-serif;  
}  
  
p {  
  font-size: 1em;  
  color: #555;  
  font-family: 'Arial', sans-serif;  
  margin-top: 10px;  
}
```