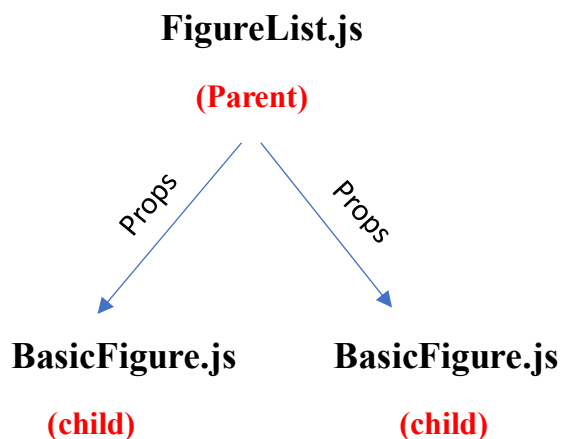


Experiment-5

Develop a React application that demonstrates component composition and the use of props to pass data. Create two components: **FigureList**: A parent component responsible for rendering multiple child components. **BasicFigure**: A child component designed to display an image and its associated caption. Use the **FigureList** component to dynamically render multiple **BasicFigure** components. Pass image URLs and captions as props from the **FigureList** component to each **BasicFigure** component. Style the **BasicFigure** components to display the image and caption in an aesthetically pleasing manner. Arrange the **BasicFigure** components within the **FigureList** in a grid or list format. Allow users to add or remove images dynamically. Add hover effects or animations to the images for an interactive experience.



Probs- image URLs and captions

App.js

```
import React from "react";
import FigureList from "../FigureList";
import "../App.css";

function App() {
  return (
    <div className="App">
      <h1>Figure List</h1>
      <FigureList />
    </div>
  );
}

export default App;
```

FigureList.js

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

// BasicFigure Component (Child)

```
const BasicFigure = ({ image, caption, onRemove }) => {  
  return (  
    <div className="basic-figure">  
      <img src={image} alt={caption} className="figure-image" />  
      <figcaption className="figure-caption">{caption}</figcaption>  
      <button className="remove-button" onClick={onRemove}>  
        Remove  
      </button>  
    </div>  
  );  
};
```

// FigureList Component (Parent)

```
const FigureList = () => {  
  const [figures, setFigures] = useState([  
    { id: 1, image: "https://www.w3schools.com/w3images/lights.jpg", caption: "Beautiful  
    Lights" },  
    { id: 2, image: "https://www.w3schools.com/w3images/nature.jpg", caption: "Nature" },  
  ]);  
  const [newImage, setNewImage] = useState("");  
  const [newCaption, setNewCaption] = useState("");  
  const addFigure = () => {  
    if (newImage && newCaption) {  
      setFigures([  
        ...figures,  
        {  
          id: Date.now(), image: newImage, caption: newCaption },  
      ]
```

```
]);
setNewImage("");
setNewCaption("");
}
};

const removeFigure = (id) => {
  setFigures(figures.filter((figure) => figure.id !== id));
};

return (
  <div className="figure-list">
    <div className="form">
      <input
        type="text"
        placeholder="Image URL"
        value={newImage}
        onChange={(e) => setNewImage(e.target.value)}
      />
      <input
        type="text"
        placeholder="Caption"
        value={newCaption}
        onChange={(e) => setNewCaption(e.target.value)}
      />
      <button onClick={addFigure}>Add Image</button>
    </div>
    <div className="grid">
      {figures.map((figure) => (
        <BasicFigure
          key={figure.id}
```

```
image={figure.image}  
caption={figure.caption}  
onRemove={() => removeFigure(figure.id)}  
/>  
)))  
</div>  
</div>  
);  
};  
export default FigureList;
```

App.css

```
.App {  
font-family: Arial, sans-serif;  
text-align: center;  
padding: 20px;  
}  
.figure-list .form {  
margin-bottom: 20px;  
}  
.figure-list .form input {  
margin: 5px;  
padding: 8px;  
width: 200px;  
border: 1px solid #ccc;  
border-radius: 5px;  
}  
.figure-list .form button {  
padding: 8px 12px;  
background-color: #4caf50;
```

```
color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.figure-list .form button:hover {

background-color: #45a049;

}

.grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));

gap: 20px;

}

.basic-figure {

position: relative;

background: #f9f9f9;

padding: 10px;

border-radius: 10px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

text-align: center;

}

.figure-image {

max-width: 100%;

border-radius: 8px;

transition: transform 0.3s ease;

}

.figure-image:hover {

transform: scale(1.05);

}
```

```
.figure-caption {  
margin-top: 10px;  
font-size: 14px;  
color: #555;  
}  
  
.remove-button {  
position: absolute;  
top: 5px;  
right: 5px;  
padding: 5px;  
background-color: #e74c3c;  
color: white;  
border: none;  
border-radius: 50%;  
cursor: pointer;  
}  
  
.remove-button:hover {  
background-color: #c0392b;  
}
```