

## React Projects

### Day 2 Class Exercise Project1

#### HelloWorld.jsx

```
import React from 'react';

const HelloWorld = () => {

  return(

    <p>Hello World</p>

  )

}

export default HelloWorld;
```

### Day 2 Class Exercise Project2

#### InlineStyling.jsx

```
import React from 'react';

const InlineStyling = () => {

  return (

    <>

    <h1 style={{color:'Green'}}>Inline Style in JSX Example.</h1>

    <div data-testid="app" style={{backgroundColor:'lightblue', color:'darkblue', padding:'10px', border:'1px solid blue', borderRadius:'5px'}}>

      <p style={{color:'darkblue', fontSize:'16px'}}>This is a paragraph with inline styles applied</p>

    </div>

    </>

  )

}

export default InlineStyling;
```

### Day 2 Practice at Home Project1

#### Form.jsx

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

const Form =() => {

  const [formData,setFormData]=useState({

    Name:",
```

```

    Email:",
    Message:' '
  }
)
const handleChangeEvent = (event) => {
  const{name,value} = event.target;
  setFormData({
    ...formData,
    [name]: value
  })
}
const handleSubmit = (event)=>{
  event.preventDefault();
  console.log(formData);
}
return(
  <>
  <h1>Basic Form</h1>
  <form method='POST' onSubmit={handleSubmit}>
    <label htmlFor="Name">Name:</label>
    <input type="text" name="Name" id="Name" onChange={handleChangeEvent}/>
    <br></br>
    <label htmlFor="Email">Email:</label>
    <input type="text" name="Email" id="Email" onChange={handleChangeEvent} />
    <br></br>
    <label htmlFor="Message">Message:</label>
    <input type="text" name="Message" id="Message" onChange={handleChangeEvent} />
    <br></br>
    <button type="Submit" onSubmit={handleSubmit}>Submit</button>
  </form>

```

```

    </>
  )
}
export default Form;

```

### Day 3 Class Exercise Project1

#### DynamicRendering.jsx

```

import React,{useState} from 'react';
const DynamicRendering = () => {
  const [btnTxt,setBtnTxt]=useState('Show Component');
  const [showData,setShowData]=useState(false);
  const handleEvent = () => {
    if(btnTxt==='Show Component'){
      setBtnTxt('Hide Component');
      setShowData(true);
    }
    else{
      setBtnTxt('Show Component');
      setShowData(false);
    }
  }
  return (
    <>
      { showData ? <p>Hi! How are You!!!</p> : <p></p>}
      <button onClick={handleEvent}>{btnTxt}</button>
    </>
  )
}
export default DynamicRendering;

```

### Day 3 Practice at Home Project1

#### App.jsx

```

import React, { useState } from 'react';

const App = () => {
  const [transformation, setTransformation] = useState('Kaioken');
  const toggleTransformation = () => {
    setTransformation((prevTransformation) =>
      prevTransformation === 'Kaioken' ? 'SuperSaiyan' : 'Kaioken'
    );
  };
  return (
    <div>
      <h1 className={transformation}>Hello, World!</h1>
      <button onClick={toggleTransformation}>Toggle Class</button>
    </div>
  );
};

export default App;

```

## Day 4 Class Exercise Project1

### FruitSurvey.jsx

```

import React, { useState } from 'react';
import { TextField, Autocomplete, Button } from '@mui/material';
import styled from 'styled-components';

const Fruits = ['Apple', 'Banana', 'Cherry', 'Durian', 'Elderberry'];

const FruitSurvey = () => {
  const [name, setName] = useState('');
  const [selectedValue, setSelectedValue] = useState(null);
  const handleSubmit = (event) => {
    event.preventDefault();
  };
  return (
    <Form onSubmit={handleSubmit}>

```

```

<h1>Enter your name favourite fruit :</h1>

<TextField

  style={{ marginBottom: '10px' }}

  variant="standard"

  value={name}

  onChange={(e) => setName(e.target.value)}

  label="Name"

  data-testid="name"

/>

<Autocomplete

  style={{ width: '250px', marginBottom: '10px' }}

  value={selectedValue}

  onChange={(_, newValue) => setSelectedValue(newValue)}

  options={Fruits}

  renderInput={({params) => (
    <TextField {...params} data-testid={`option-${selectedValue}`} label="Fruits"
variant="outlined" />
  )}

  data-testid="autocomplete"

/>

  <Button type='submit' variant='contained' style={{ width: '150px' }} data-
testid="button">Submit</Button>

</Form>

);

};

const Form = styled.form`

  width: 500px;

  display: flex;

  flex-direction: column;

  justify-content: center;

  align-items: left;

  margin: 10px 0 0 10px;

```

```
`;  
  
export default FruitSurvey;
```

## Day 4 Practice at Home Project1

### LoginButton.jsx

```
import React, { useState } from 'react';  
  
import {  
  Button,  
  Dialog,  
  DialogTitle,  
  DialogContent,  
  DialogActions,  
  TextField,  
} from '@mui/material';  
  
const LoginButton = () => {  
  const [open, setOpen] = useState(false);  
  const [username, setUsername] = useState('');  
  const [password, setPassword] = useState('');  
  const handleLogin = () => {  
    console.log('Username:', username);  
    console.log('Password:', password);  
    handleClose();  
  };  
  const handleOpen = () => {  
    setOpen(true);  
  };  
  const handleClose = () => {  
    setOpen(false);  
  };  
  return (  
    <div>
```

```
<Button data-testid="login" onClick={handleOpen} variant="outlined">
  Login
</Button>

<Dialog open={open} onClose={handleClose}>
  <DialogTitle>Login</DialogTitle>
  <DialogContent>
    <TextField
      data-testid="username"
      label="Username"
      variant="outlined"
      fullWidth
      value={username}
      onChange={(e) => setUsername(e.target.value)}
    />
    <TextField
      data-testid="password"
      label="Password"
      variant="outlined"
      fullWidth
      type="password"
      value={password}
      onChange={(e) => setPassword(e.target.value)}
    />
  </DialogContent>
  <DialogActions>
    <Button data-testid="cancel" onClick={handleClose} color="primary">
      Cancel
    </Button>
    <Button data-testid="dia-login" onClick={handleLogin} color="primary">
      Login
    </Button>
  </DialogActions>
</Dialog>
```

```

        </DialogActions>
      </Dialog>
    </div>
  );
};
export default LoginButton;

```

## Day 5 Class Exercise Project1

### ErrorBoundary.jsx

```

import React, { Component } from 'react';
import PropTypes from 'prop-types';
class ErrorBoundary extends Component {
  constructor(props) {
    super(props);
    this.state = { hasError: false, error: null };
  }
  static getDerivedStateFromError(error) {
    return { hasError: true, error };
  }
  componentDidCatch(error, errorInfo) {
    console.error(error, errorInfo);
  }
  render() {
    if (this.state.hasError) {
      return <p>Error: {this.state.error.message}</p>;
    }
    return this.props.children;
  }
}
ErrorBoundary.propTypes = {
  children: PropTypes.node.isRequired,

```



```

};

export default ErrorBoundary;

List.jsx

import React, { Component } from 'react';
import ErrorBoundary from './ErrorBoundary';

class List extends Component {
  constructor(props) {
    super(props);
    this.state = { data: [1, 2, 3, 4, 5] };
  }

  componentDidUpdate() {
    throw new Error('An error occurred in List component');
  }

  render() {
    return (
      <ErrorBoundary>
        <ul>
          {this.state.data.map(item => (
            <li key={item}>{item}</li>
          ))}
        </ul>
      </ErrorBoundary>
    );
  }
}

export default List;

```

## Day 5 Practice at Home Project1

### LifeCycle.jsx

```

import React, { Component } from 'react';

class LifeCycle extends Component {

```

```
constructor(props) {  
  super(props);  
  this.state = {  
    msg: 'IamNEO',  
  };  
}  
componentDidMount() {  
  console.log('Component Did MOUNT!');  
}  
componentDidUpdate(prevProps, prevState) {  
  console.log('Component Did UPDATE!');  
}  
componentWillUnmount() {  
  console.log('Component Will UNMOUNT!');  
}  
handleClick = () => {  
  console.log('Button Clicked!');  
  this.setState({  
    msg: 'Welcome to React',  
  });  
};  
render() {  
  return (  
    <div>  
      <h1>{this.state.msg}</h1>  
      <button onClick={this.handleClick}>Change Message</button>  
    </div>  
  );  
}  
}  
export default LifeCycle;
```

## Day 6 Class Exercise Project1

### App.jsx

```
import React from 'react';

import { WithProductList } from './WithProductList';

import ProductListComponent from './ProductListComponent';

const Products = [

  { id: 1, name: 'Product 1', price: 19.99, imageUrl: 'product1.jpg' },

  { id: 2, name: 'Product 2', price: 29.99, imageUrl: 'product2.jpg' },

];

const EnhancedProductList = WithProductList(ProductListComponent, Products);

const App = () => {

  return (

    <div>

      <h1>Your App Title</h1>

      {/* Render the enhanced ProductListComponent */}

      <EnhancedProductList />

    </div>

  );

};

export default App;
```

### ProductListComponent.jsx

```
import React from 'react';

const ProductListComponent = ({ productList }) => {

  return (

    <ul>

      {productList.map((product) => (

        <li key={product.id}>

          <div>{product.name}</div>

          <div>${product.price.toFixed(2)}</div>

        </li>

      ))}

    </ul>

  );

};
```

```

    </ul>

  );
};

export default ProductListComponent;

WithProductList.jsx

import React from 'react';

export const WithProductList = (WrappedComponent, productList) => {
  return () => <WrappedComponent productList={productList} />;
};

```

## Day 6 Practice at Home Project1

### FilterOption.jsx

```

import React, { useContext } from 'react';
import { ProductContext } from '../ProductContext';

const FilterOptions = () => {
  const { state, dispatch } = useContext(ProductContext);
  const handlePriceChange = (e, type) => {
    dispatch({ type: 'SET_PRICE_RANGE', payload: { ...state.priceRange, [type]: e.target.value } });
  };

  return (
    <div>
      <h2>Filter Options</h2>
      <label>
        Min Price:
      </label>
      <input
        type="number"
        value={state.priceRange.min}
        onChange={(e) => handlePriceChange(e, 'min')}
      />
      <label>

```

```

        Max Price:
      </label>

      <input
        type="number"
        value={state.priceRange.max}
        onChange={(e) => handlePriceChange(e, 'max')}
      />
    </div>

  );
};

export default FilterOptions;

```

### **ProductContext.jsx**

```

import React, { createContext, useReducer, useEffect, useMemo } from 'react';
import PropTypes from 'prop-types';
import axios from 'axios';

const initialState = {
  categories: [],
  products: [],
  filteredProducts: [],
  priceRange: { min: 0, max: 1000 },
};

const productReducer = (state, action) => {
  let min, max, filtered;
  switch (action.type) {
    case 'SET_CATEGORIES':
      return { ...state, categories: action.payload };
    case 'SET_PRODUCTS':
      return { ...state, products: action.payload, filteredProducts: action.payload };
    case 'SET_PRICE_RANGE':
      return { ...state, priceRange: action.payload };
    case 'FILTER_PRODUCTS':

```

```

    min = state.priceRange.min;

    max = state.priceRange.max;

    filtered = state.products.filter(product => product.price >= min && product.price <= max);

    return { ...state, filteredProducts: filtered };

  default:

    return state;

  }
};

const ProductContext = createContext();
const ProductProvider = ({ children }) => {

  const [state, dispatch] = useReducer(productReducer, initialState);

  const contextValue = useMemo(() => ({ state, dispatch }), [state, dispatch]);

  const fetchCategories = async () => {

    try {

      const response = await axios.get('API_ENDPOINT/categories');

      dispatch({ type: 'SET_CATEGORIES', payload: response.data });

    } catch (error) {

      console.error('Error fetching categories:', error);

    }

  };

  const fetchProducts = async () => {

    try {

      const response = await axios.get('API_ENDPOINT/products');

      dispatch({ type: 'SET_PRODUCTS', payload: response.data });

    } catch (error) {

      console.error('Error fetching products:', error);

    }

  };

  useEffect(() => {

    fetchCategories();

    fetchProducts();

```

```

    }, []);
    return (
      <ProductContext.Provider value={contextValue}>
        {children}
      </ProductContext.Provider>
    );
  };
};

ProductProvider.propTypes = {
  children: PropTypes.node.isRequired,
};

export { ProductProvider, ProductContext };

ProductList.jsx

import React, { useContext, useEffect } from 'react';
import { ProductContext } from '../ProductContext';

const ProductList = () => {
  const { state, dispatch } = useContext(ProductContext);
  useEffect(() => {
    dispatch({ type: 'FILTER_PRODUCTS' });
  }, [state.priceRange, dispatch]);
  return (
    <div>
      <h2>Product List</h2>
      {state.filteredProducts.map(product => (
        <div key={product.id}>
          <p>{product.name}</p>
          <p>{product.price}</p>
        </div>
      ))}
    </div>
  );
};

```

```
export default ProductList;
```

## Day 7 Class Exercise Project1

### BookList.jsx

```
import React, { useState, useEffect } from 'react';

const BookList = () => {

  const [books, setBooks] = useState([]);

  useEffect(() => {

    const fetchData = async () => {

      try {

        const response = await fetch('https://example.com/books');

        const data = await response.json();

        const transformedData = data.data.map(book => ({

          id: book.id,

          title: book.title,

          author: book.author

        }));

        setBooks(transformedData);

      } catch (error) {

        console.error('Error fetching data:', error);

      }

    };

    fetchData();

  }, []);

  return (

    <div>

      <h1>Book List</h1>

      <ul>

        {books.map(book => (

          <li key={book.id}>

            <strong>{book.title}</strong> by {book.author}

          </li>

        ))}

      </ul>

    </div>

  );

};
```



```

        </li>
      )}
    </ul>
  </div>
);
};
export default BookList;

```

## Day 7 Practice at Home Project1

### ProductCard.jsx

```

import React from 'react';

const ProductCard = ({ name, description, price }) => {
  return (
    <div style={{ border: '1px solid #ccc', padding: '16px', borderRadius: '8px' }}>
      <h2>{name}</h2>
      <p>{description}</p>
      <p>Price: ${price.toFixed(2)}</p>
    </div>
  )
}

export default ProductCard;

```

### ProductList.jsx

```

import React, { useState, useEffect } from 'react';
import ProductCard from './ProductCard';

const ProductList = () => {
  const [products, setProducts] = useState([]);

  useEffect(() => {
    const apiUrl = 'http://localhost:3001/products';

    fetch(apiUrl)
      .then(response => response.json())
      .then(data => setProducts(data))
  });

```

```

        .catch(error => console.error('Error fetching data:', error));
    }, []);
    return (
        <div>
            <h1>Product List</h1>
            <div style={{ display: 'grid', gridTemplateColumns: 'repeat(3, 1fr)', gap: '16px' }}>
                {products.map(product => (
                    <ProductCard
                        key={product.id}
                        name={product.name}
                        description={product.description}
                        price={product.price}
                    />
                ))}
            </div>
        </div>
    );
};
export default ProductList;

```

### Product.json

```

{
  "products": [
    {
      "id": 1,
      "name": "ProductA",
      "description": "This is Product A",
      "price": 10.99
    }
  ]
}

```

### Day 8 Class Exercise Project1

### ReactApp.jsx

```
import React, { useState, useEffect } from 'react';

const ReactApp = () => {

  const [message, setMessage] = useState("Hello dear one, learning React...?");

  useEffect(() => {

    console.log('useEffect hook called');

    const timeoutId = setTimeout(() => {

      setMessage("Great.... This is time to learn about HOOKS");

    }, 2000);

    return () => {

      clearTimeout(timeoutId);

    };

  }, []);

  return (

    <div>

      <p>{message}</p>

    </div>

  );

};

export default ReactApp;
```