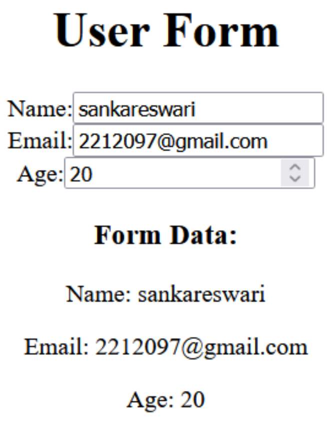
| Ex.No:6 | Programs on Functional Components |
|---|---|
| | |

**User Form**

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

const initialState = {
  name: '',
  email: '',
  age: 0,
};

function reducer(state, action) {
  switch (action.type) {
    case 'SET_NAME':
    return { ...state, name: action.payload };
    case 'SET_EMAIL':
    return { ...state, email: action.payload };
    case 'SET_AGE':
    return { ...state, age: action.payload };
    default:
    return state;
  }
}

const UserForm = () => {
  const [state, dispatch] = useReducer(reducer, initialState);

  const handleChange = (e) => {
    dispatch({
    type: `SET_${e.target.name.toUpperCase()}`,
    payload: e.target.value,
    });
  };

  return (
    <div>
    <h1>User Form</h1>
    <form>
        <div>
        <label>Name:</label>
        <input
            type="text"
            name="name"
            value={state.name}
            onChange={handleChange}
        />
        </div>
        <div>
        <label>Email:</label>
         <input
            type="email"
            name="email"
            value={state.email}
            onChange={handleChange}
        />
        </div>
        <div>
        <label>Age:</label>
        <input
            type="number"
         name="age"
            value={state.age}
            onChange={handleChange}
        />
        </div>
    </form>
    <div>
        <h3>Form Data:</h3>
        <p>Name: {state.name}</p>
        <p>Email: {state.email}</p>
        <p>Age: {state.age}</p>
    </div>
    </div>
  );
};

export default UserForm;
```

**OUTPUT:**



User Form

Name: sankareswari

Email: 2212097@gmail.com

Age: 20

Form Data:

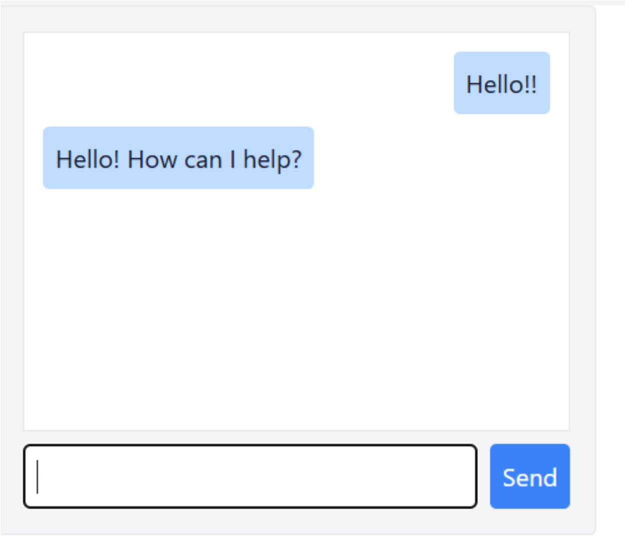
Name: sankareswari

Email: 2212097@gmail.com

Age: 20

## Simple Chat Application

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

const ChatApp = () => {
  const [messages, setMessages] = useState([]);
  const [input, setInput] = useState("");

  const sendMessage = () => {
    if (input.trim() !== "") {
      setMessages([...messages, { text: input, sender: "You" }]);
      setInput("");
      setTimeout(() => {
        setMessages((prev) => [
          ...prev,
          { text: "Hello! How can I help?", sender: "Bot" }
        ]);
      }, 1000);
    }
  };

  return (
    <div className="p-4 w-96 bg-gray-100 border rounded">
      <div className="h-64 overflow-y-auto border p-2 bg-white">
        {messages.map((msg, index) => (
          <div key={index} className={`p-1 ${msg.sender === "You" ? "text-right" : "text-left"}`}>
            <span className="inline-block p-2 bg-blue-200 rounded">{msg.text}</span>
          </div>
        ))}
      </div>
      <div className="mt-2 flex">
        <input
          type="text"
          className="flex-1 p-2 border rounded"
          value={input}
          onChange={(e) => setInput(e.target.value)}
          onKeyDown={(e) => e.key === "Enter" && sendMessage()}
        />
        <button className="ml-2 p-2 bg-blue-500 text-white rounded" onClick={sendMessage}>
          Send
        </button>
      </div>
    </div>
  );
};

export default ChatApp;
```


Hello!!

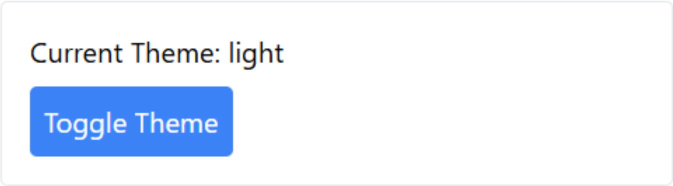
Hello! How can I help?

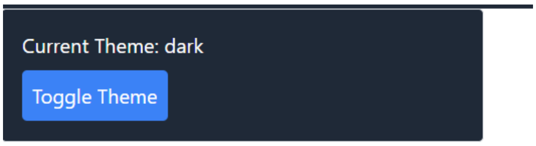
### Responsive Theme Switcher

```
import React, { useState, useEffect, createContext, useContext } from "react";

const ThemeContext = createContext();

const ThemeProvider = ({ children }) => {
  const [theme, setTheme] = useState(() => {
    return localStorage.getItem("theme") || "light";
  });

  useEffect(() => {
    localStorage.setItem("theme", theme);
    document.documentElement.className = theme;
  }, [theme]);

  const toggleTheme = () => {
    setTheme((prevTheme) => (prevTheme === "light" ? "dark" : "light"));
  };

  return (
    <ThemeContext.Provider value={{ theme, toggleTheme }}>
      {children}
    </ThemeContext.Provider>
  );
};

const useTheme = () => useContext(ThemeContext);

const ThemeSwitcher = () => {
  const { theme, toggleTheme } = useTheme();

  return (
    <div className={`p-4 w-96 border rounded ${theme === "dark" ? "bg-gray-800 text-white" : "bg-
white text-black"}`}>
      <p>Current Theme: {theme}</p>
```

```
   <button
    className="mt-2 p-2 bg-blue-500 text-white rounded"
    onClick={toggleTheme}
   >
    Toggle Theme
   </button>
  </div>
 );
};

const App = () => {
 return (
  <ThemeProvider>
   <ThemeSwitcher />
  </ThemeProvider>
 );
};

export default App;
```

**Output:**

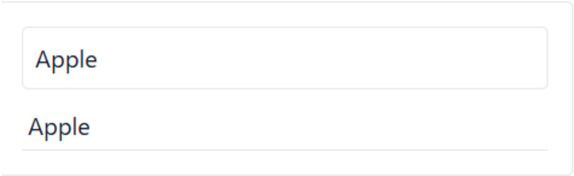

Current Theme: light


Current Theme: dark

### Real-time Search Filter

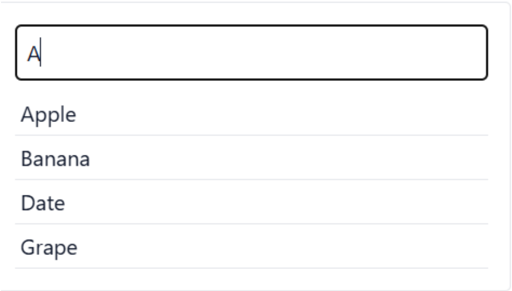
```
import React, { useState } from "react";

const ProductSearch = () => {
 const [searchQuery, setSearchQuery] = useState("");
 const products = ["Apple", "Banana", "Cherry", "Date", "Elderberry", "Fig", "Grape", "Honeydew"];

 const filteredProducts = products.filter(product =>
  product.toLowerCase().includes(searchQuery.toLowerCase())
 );

 return (
  <div className="p-4 w-96 bg-white border rounded">
   <input
    type="text"
    placeholder="Search products..."
    className="p-2 border rounded w-full"
    value={searchQuery}
    onChange={(e) => setSearchQuery(e.target.value)}
   />
   <ul className="mt-2">
    {filteredProducts.map((product, index) => (
     <li key={index} className="p-1 border-b">{product}</li>
    ))}
   </ul>
  </div>
 );
};

const App = () => {
 return <ProductSearch />;
};
export default app
```

**Output:**



Apple

Apple



A

Apple

Banana

Date

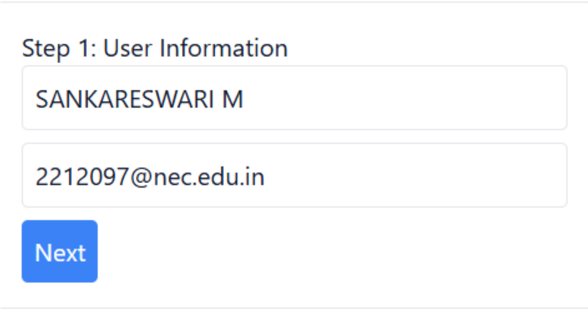
Grape

**Form with Multiple Steps**

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

const MultiStepForm = () => {
 const [step, setStep] = useState(1);
 const [formData, setFormData] = useState({
  name: "",
  email: "",
  street: "",
  city: "",
  zip: "",
  creditCard: ""
 });

 const handleChange = (e) => {
  setFormData({ ...formData, [e.target.name]: e.target.value });
 };

 const nextStep = () => setStep((prev) => prev + 1);
 const prevStep = () => setStep((prev) => prev - 1);

 return (
  <div className="p-4 w-96 bg-white border rounded">
   {step === 1 && (
    <div>
```

```
     <h2>Step 1: User Information</h2>
     <input name="name" placeholder="Name" value={formData.name} onChange={handleChange}
className="p-2 border rounded w-full mb-2" />
     <input name="email" placeholder="Email" value={formData.email} onChange={handleChange}
className="p-2 border rounded w-full mb-2" />
     <button onClick={nextStep} className="p-2 bg-blue-500 text-white rounded">Next</button>
    </div>
   )}
   {step === 2 && (
    <div>
     <h2>Step 2: Address Information</h2>
     <input name="street" placeholder="Street" value={formData.street}
onChange={handleChange} className="p-2 border rounded w-full mb-2" />
     <input name="city" placeholder="City" value={formData.city} onChange={handleChange}
className="p-2 border rounded w-full mb-2" />
     <input name="zip" placeholder="Zip Code" value={formData.zip} onChange={handleChange}
className="p-2 border rounded w-full mb-2" />
     <button onClick={prevStep} className="p-2 bg-gray-500 text-white rounded mr-
2">Previous</button>
     <button onClick={nextStep} className="p-2 bg-blue-500 text-white rounded">Next</button>
    </div>
   )}
   {step === 3 && (
    <div>
     <h2>Step 3: Payment Information</h2>
     <input name="creditCard" placeholder="Credit Card" value={formData.creditCard}
onChange={handleChange} className="p-2 border rounded w-full mb-2" />
     <button onClick={prevStep} className="p-2 bg-gray-500 text-white rounded mr-
2">Previous</button>
     <button className="p-2 bg-green-500 text-white rounded">Submit</button>
    </div>
   )}
  </div>
 );
};

const App = () => {
 return <MultiStepForm />;
};

export default App;
```

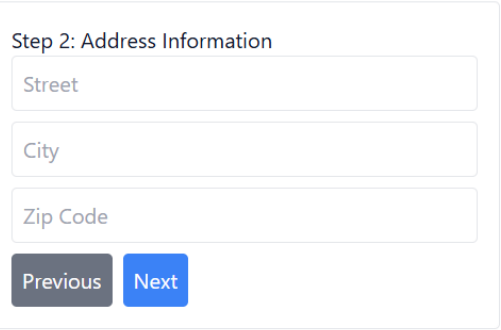
**OUTPUT**



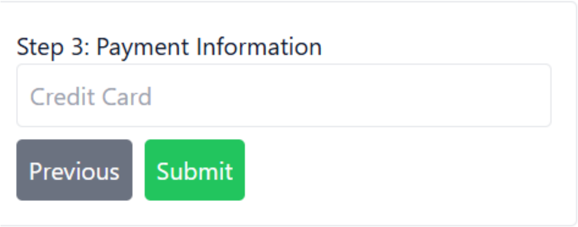
Step 1: User Information

SANKARESWARI M

2212097@nec.edu.in



Step 2: Address Information



Step 3: Payment Information

| **Problem Identification**<br>**(5)** | **Execution**<br>**(5)** | **Time management**<br>**(5)** | **Viva**<br>**(5)** | **Total**<br>**(20)** |
|---|---|---|---|---|
| | | | | |

**Result:**

Thus the above functional component programs run successfully and verified.