Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2025-26

Class:	T.E.	**Semester:**	V
Course Code:	CSC502	**Course Name:**	WEB COMPUTING

Name of Student:	SHRUTI GAUCHANDRA
Roll No. :	18
Assignment No.:	03
Title of Assignment:	Apply concepts of React fundamentals to develop front end applications.
Date of Submission:	28/08/25
Date of Correction:	01/09/25

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Completeness	5	5
Demonstrated Knowledge	3	3
Legibility	2	2
Total	10	10

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Completeness	5	3-4	1-2
Demonstrated Knowledge Legibility	3	2	1
Legibility	2	1	0

Checked by

Name of Faculty : Ms. KSHITIJA GHARAT

Signature : Ghatar

Date : 1/9/25

CSC502.3 Apply concepts of React fundamental to develop front end application.

Q1. Implement a React component that manages a contact list. Users should be able to add contacts with fields for name, phone number, and email. Include a search bar that filters the list of contacts based on the input. Provide a "Remove" button next to each contact for deletion.

→

```
import {useState} from "react";
export default function ContactList(){
 const [contacts, setContacts] = useState([]);
 const [form, setForm] = useState({name:"", phone:"",
                                   email:""});
 const [search, setSearch] = useState("");

const addContact=()=>{
 if(form.name && form.phone && form.email){
   setContacts([....contacts, form]);
   setForm({name:"", phone:"", email:""});
  }
};
const removeContact = (i)=>{
  setContacts(contacts.filter((_,index)=> index!==i));
};
const filtered = contacts.filter((c)=>
 c.name.toLowerCase().includes(search.toLowerCase())
);
```

Sundaram FOR EDUCATIONAL USE

```
return (
 <div>
 <h2> Contact Manager </h2>
 <input
  value={search}
  onChange={(e) => setSearch(e.target.value)}
  placeholder = "search by name"}
  />
 <br /> <br />
 <input
   value ={form.name}
   onChange={ (e) => setForm ({... form, name : e.target.
                    value})}
   placeholder = "Name"
 />
 <input
   value = {form.phone}
   onChange = {(e) => setForm({... form, phone : e.target.value}
                )}
   placeholder = "Phone".
 />
 <input
   value = {form.email}
   onChange = {(e) -> setForm({ .... form, email : e.target.value}
              )}
   placeholder = "Email".
 />
<button onClick = {addContact}> Add Contact </button>
<ul>
{filtered.map ((c, i) => (
```

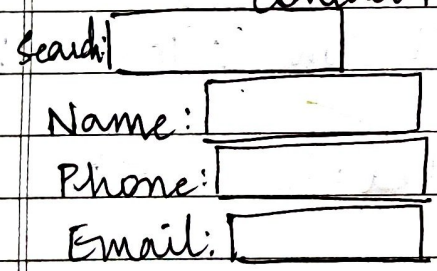
FOR EDUCATIONAL USE

Sundaram®

```
<li key={i}>
  {c.name} - {c.phone} - {c.email}
  <button onClick={() => removeContact(i)}>Remove
  </button>
 </li>
))}
</ul>
</div>
);
}
```

Q2. Write a code:

Output:



When we add a contact, it looks in a list like:

John Doe - 1235467890 - john@example.com [Remove]

Sundaram
FOR EDUCATIONAL USE

Q2. Write a code making use of React Hooks that displays four buttons namely, "Red", "Blue", "Green", "Yellow". On clicking any of these buttons, the code displays the message that you have selected that particular color.

→
```
import {useState} from "react";
export default function ColorSelector(){
 const[color, setColor] = useState("");
return(
  <div>
  <h2> ColorSelector </h2>
  <button onClick={()=> setColor("Red")}>Red</button>
  <button onClick={()=> setColor("Blue")}>Blue</button>
  <button onClick={()=> setColor("Green")}>Green</button>
  <button onClick={()=> setColor("Yellow")}>Yellow</button>

  {color && <p>You have selected: {color}</p>}
  </div>
);
}
```

Output:

Color Selector

[Red] [Blue] [Green] [Yellow]

When you click a button, it shows

You have selected: Red.

Sundaram
FOR EDUCATIONAL USE

Q3. Write the code making use of Hooks usestate function that displays the number of times button named "CLICK" is clicked.

→
```
import {useState} from "react";
export default function ClickCounter(){
const [count, setCount] = useState(0);

return(
  <div>
  <h2>Click Counter</h2>
  <button onClick={() => setCount(count+1)}>CLICK
  </button>
  <p>Button clicked {count} times</p>
 </div>
);
}
```

Output:

Click Counter

[CLICK]

A text showing below, how many times the button was clicked.

Button clicked 0 times

After clicking:

Button clicked 1 times

Button clicked 2 times