

# Search functionality project

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

const App = () => {

  const [searchValue, setSearchValue] = useState("")

  const [itemList, setItemList] = useState(['cricket', 'apple','apple juice', 'mango',
'zebra', 'mango juice', 'football', 'rabbit', 'india', 'bangladesh', 'nepal', 'country'])

  const [copyItemList, setCopyItemList] = useState(['cricket', 'apple','apple juice', 'mango',
'mango juice', 'football', 'rabbit', 'india', 'bangladesh', 'nepal', 'country'])

  const handleOnChange = (value) => {

    setSearchValue(value)

    const result = copyItemList.filter((item, index)=> {

      return item.includes(value)

    })

    setItemList(result)

  }

  return (

    <div>

      <input type='text' value={searchValue} onChange={(e)=>handleOnChange(e.target.value)}
placeholder='search...' />

      {itemList.map((item)=> {

        return <p key={item}>{item}</p>

      })}

    </div>

  )

}

export default App
```

# Todo project

```

import { useState } from "react";
import { GrEdit } from "react-icons/gr";
import { AiOutlineDelete } from "react-icons/ai";
import { v4 as uuidv4 } from "uuid";

const App = () => {

  const [todoInput, setTodoInput] = useState("");

  const [todoList, setTodoList] = useState([]);

  const handleTodo = () => {

    const copyOfTodoList = todoList.slice(); //taking exiting todo

    const addObj = {

      id: uuidv4(),

      name: todoInput,

    };

    copyOfTodoList.push(addObj); //adding one more todo

    setTodoList(copyOfTodoList); //updating todo list

    setTodoInput("");

  };

  const handleDelete = (selectedId) => {

    console.log(selectedId, "todoList", todoList);

    const deleteTodo = todoList.filter((item) => {

      console.log(item.id, selectedId, "item", item.id !== selectedId);

      return item.id !== selectedId;

    });

    setTodoList(deleteTodo);

  };

  return (

    <div style={{ margin: "20px" }}>

      <input

        type="text"

        placeholder="Enter..."

        value={todoInput}

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

      />

      <button

        type="button"

        onClick={handleTodo}

```

```

        style={{ backgroundColor: "#000", color: "#fff" }}

      >

      Add

    </button>

    {todoList.map((item) => {

      return (

        <div key={item.id} className="todo-list">

          <p>{item.name}</p>

          <div>

            <AiOutlineDelete

              style={{ marginRight: "10px" }}

              onClick={() => handleDelete(item.id)}

            />

          </div>

        </div>

      );

    })}

  </div>

);

};

export default App;

```

## Contact List

```

// ContactListUi.js

import React, { useState } from "react";

import { Button, Modal, Form } from "react-bootstrap";

import { MdDelete } from "react-icons/md";

import { BiEditAlt } from "react-icons/bi";

import { CgProfile } from "react-icons/cg";

import { ToastContainer, toast } from "react-toastify";

import "react-toastify/dist/ReactToastify.css";

import { v4 as uuidV4 } from "uuid";

const ContactListUi = () => {

  const [contacts, setContacts] = useState([]);

  const [showModal, setShowModal] = useState(false);

```

```
const [selectedContact, setSelectedContact] = useState({});

const [name, setName] = useState("");

const [phone, setPhone] = useState("");

const [search, setSearch] = useState("");
```

```
const handleClose = () => {

  setShowModal(false);

  setSelectedContact({});

  setName("");

  setPhone("");

};
```

```
const handleSave = () => {

  if (!name || !phone) {

    return toast.error("Please Enter Mandatory Fields, error");

  }

}
```

```
if (!/^[6-9]{1}[0-9]{9}$/.test(phone)) {

  return toast.error("Please Enter Valid Phone Number, error");

}
```

```
if (selectedContact.id) {

  // Edit existing contact

  const copyOfContacts = [...contacts];

  copyOfContacts.forEach((contact) => {

    if (contact.id === selectedContact.id) {

      contact.name = name;

      contact.phone = phone;

    }

    return contact;

  });

  setContacts(copyOfContacts);

} else {

  // Add new contact

  const addContact = [...contacts, { id: uuidV4(), name, phone }];
```

```

    setContacts(addContact);
  }

  handleClose();
};

```

```

const handleEdit = (contact) => {
  setSelectedContact(contact);

  setName(contact.name);

  setPhone(contact.phone);

  setShowModal(true);
};

```

```

const handleDelete = (id) => {

  const deleteSelectedContact = contacts.filter((contact) => contact.id !== id);

  setContacts(deleteSelectedContact);

};

```

```

const filteredContacts = contacts.filter((contact) =>

  contact.name.toLowerCase().trim().includes(search.toLowerCase().trim())

);

```

```

return (
  <div>

    <ToastContainer />

    <div className="d-flex justify-content-center my-3">

      <input

        className="mx-1 p-1 border-1 rounded-1"

        type="text"

        value={search}

        placeholder="search..."

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

      />

      <Button variant="primary" onClick={() => setShowModal(true)}>

        Add

      </Button>

```

```
</div>
```

```
<Modal show={showModal} onHide={handleClose}>

  <Modal.Header closeButton>

    <Modal.Title>

      {selectedContact.id ? "Edit Contact" : "Add Contact"}

    </Modal.Title>

  </Modal.Header>

  <Modal.Body>

    <Form>

      <Form.Group controlId="formName">

        <Form.Label>

          Name <span style={{ color: "red" }}>*</span>

        </Form.Label>

        <Form.Control

          type="text"

          placeholder="Enter Name"

          value={name}

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

        />

      </Form.Group>

      <br />

      <Form.Group controlId="formphone">

        <Form.Label>

          Phone Number <span style={{ color: "red" }}>*</span>

        </Form.Label>

        <Form.Control

          type="number"

          placeholder="Enter phone Number"

          value={phone}

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

        />

      </Form.Group>

    </Form>

  </Modal.Body>

  <Modal.Footer>
```

```

    <Button variant="danger" onClick={handleClose}>

      Close

    </Button>

    <Button variant="success" onClick={handleSave}>

      Save

    </Button>

  </Modal.Footer>

</Modal>

```

```

<div>

  {filteredContacts.map((contact) => (

    <div key={contact.id} className="contact-list">

      <div className="d-flex align-items-center">

        <CgProfile style={{ fontSize: "30px" }} />

        <div className="mx-2">

          <h4> {contact.name}</h4>

          <p>{contact.phone}</p>

        </div>

      </div>

    </div>

    <div>

      <BiEditAlt onClick={() => handleEdit(contact)} className="mx-1"
style={{ cursor: 'pointer'}} />

      <MdDelete

        style={{ color: "red", cursor: 'pointer'}}

        className="mx-1"

        onClick={() => handleDelete(contact.id)}

      />

    </div>

  </div>

  )))

</div>

</div>

);

};

```

```

export default ContactListUi;

```

# Instagram

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

import { BiMessageRoundedDots } from "react-icons/bi";
import { FaRegHeart } from "react-icons/fa";
import instaLogo from "./instagram-new-logo.png";
import profileImg from "./profile-img.jpg";
import { HiDotsVertical } from "react-icons/hi";
import uploadedImg from "./uploaded-img.jpg";
import { FiMessageCircle } from "react-icons/fi";
import { LuSend } from "react-icons/lu";
import saved from "./saved-icon.png";
import { instaJsonResp } from "./utils";
import { FaHeart } from "react-icons/fa";
import savedProfile from "./savedimg.jpeg";
import { IoMdSend } from "react-icons/io";

const Instagram = () => {

  const [instaAcctDetails, setInstaAcctDetails] = useState(instaJsonResp);

  const [enableMsg, setEnableMsg] = useState(false);

  const [selectedId, setSelectedId] = useState("");

  const [message, setMessage] = useState("");
```

```
const handleLikeImg = (selectedId) => {

  let copyOfInstaAcctDetails = [...instaAcctDetails];

  copyOfInstaAcctDetails.forEach((item) => {

    if (item.id == selectedId) {

      item.like = item.like + 1;

    }

    return item;

  });
```



```
    setInstaAcctDetails(copyOfInstaAcctDetails);  
  };
```

```
const handleMessage = (id) => {  
  setSelectedId(id);  
  setEnableMsg(true);  
};
```

```
const updateMessage = (id) => {  
  let copyOfInstaAcctDetails = [...instaAcctDetails];  
  copyOfInstaAcctDetails.forEach((item) => {  
    if (item.id == selectedId) {  
      let copyOfMessage = item.message;  
      copyOfMessage.push(message);  
      item.message = copyOfMessage;  
    }  
    return item;  
  });  
  setMessage("");  
  setEnableMsg(false);  
  setInstaAcctDetails(copyOfInstaAcctDetails);  
};
```

```
const handleSave = (selectedId) => {  
  let copyOfInstaAcctDetails = [...instaAcctDetails];  
  copyOfInstaAcctDetails.forEach((item) => {  
    if (item.id == selectedId) {  
      item.saved = !item.saved;  
    }  
    return item;  
  });  
  setInstaAcctDetails(copyOfInstaAcctDetails);  
};
```

```
console.log(instaAcctDetails, "instaAcctDetails");
```

```

return (
  <div className="instagram-block">
    <div className="instagram-block-header d-flex justify-content-between align-items-center">
      <img src={instaLogo} width={"200px"} />
      <div>
        <FaRegHeart className="insta-icon" />
        <BiMessageRoundedDots className="insta-icon" />
      </div>
    </div>
  </div>
)

```

```

{instaAcctDetails.map((item, index) => {
  return (
    <div key={item.id} className="instagram-block-body">
      <div className="d-flex justify-content-between align-items-center mx-2 my-4">
        <div className="d-flex align-items-center">
          <img
            src={profileImg}
            alt=""
            width={"45px"}
            height={"45px"}
            className="profile-img"
          />
          <h2 className="mx-3">{item.name}</h2>
        </div>
        <HiDotsVertical />
      </div>
      <div>
        <img
          src={uploadedImg}
          alt=""
          width={"100%"}
          onClick={() => handleLikeImg(item.id)}
        />
        <div className="d-flex align-items-center justify-content-between">
          <div>

```

```

      {item.like ? (
        <FaHeart
          className="insta-icon"
          style={{ color: "red" }}
          onClick={() => handleLikeImg(item.id)}
        />
      ) : (
        <FaRegHeart
          className="insta-icon"
          onClick={() => handleLikeImg(item.id)}
        />
      )}
    <FiMessageCircle
      className="insta-icon"
      onClick={() => handleMessage(item.id)}
    />
    <LuSend className="insta-icon" />
    <p style={{ margin: "0 10px" }}>{item.like} Likes</p>
  </div>

```

```

<div>
  {item.saved ? (
    <img
      src={savedProfile}
      alt=""
      width={"30px"}
      height={"25px"}
      className="insta-icon"
      onClick={() => handleSave(item.id)}
    />
  ) : (
    <img
      src={saved}
      alt=""
      width={"15px"}
      className="insta-icon"

```

```

        onClick={() => handleSave(item.id)}

      />

    )}

  </div>

</div>

</div>

{item.message.map((val, index) => {

  return (

    <div key={index}>

      <div className=" m-2 d-flex align-items-center ">

        <img

          src={profileImg}

          alt=""

          width={"20px"}

          height={"20px"}

          className="profile-img"

        />

        <div style={{margin: '0 5px'}}>

          <h6 style={{ margin: "0", fontSize: "10px", color: '#566565' }}>

            {item.name}

          </h6>

          <p style={{margin: "0", fontSize: '13px'}}>>{val}</p>

        </div>

      </div>

    </div>

  );

}})

{enableMsg && item.id == selectedId && (

  <div>

    <input

      type="text"

      className="insta-input-message"

      placeholder="Enter Message..."

      value={message}

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

    />

```

```

        <IoMdSend
          className="insta-icon"
          onClick={() => updateMessage(item.id)}
        />
      </div>
    )}
  </div>
);
}}}

```

```

    { /* <FiSend /> */ }
  </div>
);
};

```

```
export default Instagram;
```

## Index css

```

.dflex {
  display: flex;
  justify-content: space-between;
}

.todo-list {
  display: flex;
  justify-content: space-between;
  align-items: center;
  border: 1px solid rgb(0, 0, 0);
  padding: 10px;
  margin: 10px 0;
  box-shadow: rgba(0, 0, 0, 0.05) 0px 6px 24px 0px, rgba(0, 0, 0, 0.08) 0px 0px 0px 1px;
}

```

```
background-color: rgb(214, 200, 116);  
}
```

```
.todo-list p {  
  margin: 0;
```

```
}
```

```
.card {  
  border: 1px solid #000;  
  margin: 10px;  
  text-align: center;  
  width: 200px;  
  padding: 10px;  
  cursor: pointer;  
}
```

```
.card img {  
  width: 100%;
```

```
}
```

```
.light-dark-mode {  
  width: 100px;  
  height: 100px;  
  border: 1px solid #000;  
  margin: 5px,  
}
```

```
.best-seller-tag {  
  position: absolute;  
  top: 0;  
  left: 10px;  
  margin: 0;  
  background-color: #000;
```

```
color: #fff;

padding: 5px;

border-radius: 2px;

z-index: 99;

}
```

```
.bestseller {

  position: absolute;

  top: 10px;

  left: 10px;

  background-color: rgb(0, 0, 0);

  color: rgb(255, 255, 255);

  z-index: 9;

  padding: 5px;

  border-radius: 3px;

}
```

```
.cartvalue {

  padding: 3px 8px;

  background-color: #000;

  color: #fff;

  border-radius: 50%;

  margin: 5px 0;

}
```

```
.instagram-block {

  width: 60%;

  margin: auto;

  border: 1px solid #ddd;

  padding: 5px;

}
```

```
.instagram-block-header h1 {

  font-family: cursive

}
```

```
.profile-img {  
  border-radius: 50%;  
  border: 1.5px solid #000;  
}
```

```
.instagram-block-body {  
  margin: 10px auto;  
  border: .5px solid #ddd;
```

```
}
```

```
.insta-icon {  
  margin: 10px;  
  cursor: pointer;  
}
```

```
@media only screen and (max-width: 900px) {  
  .instagram-block {  
    width: 80%;  
  }  
}
```

```
@media only screen and (max-width: 500px) {  
  .instagram-block {  
    width: 95%;  
  }  
}
```

```
.insta-input-message {  
  width: 92%;  
  margin-left: 10px;  
  padding: 0 5px;  
  margin: 10px 0 10px 10px;  
}
```



```
.contact-list {  
  box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;  
  border: 1px solid #000;  
}
```

```
.contact-list {  
  box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;  
  border: 1px solid #ddd;  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  width: 37%;  
  margin: auto;  
  padding: 10px;  
  text-transform: capitalize;  
}
```

```
.contact-list h4 {  
  margin: 0;  
  font-size: 14px;  
}
```

```
.contact-list p {  
  margin: 0;  
  font-size: 14px;  
}
```