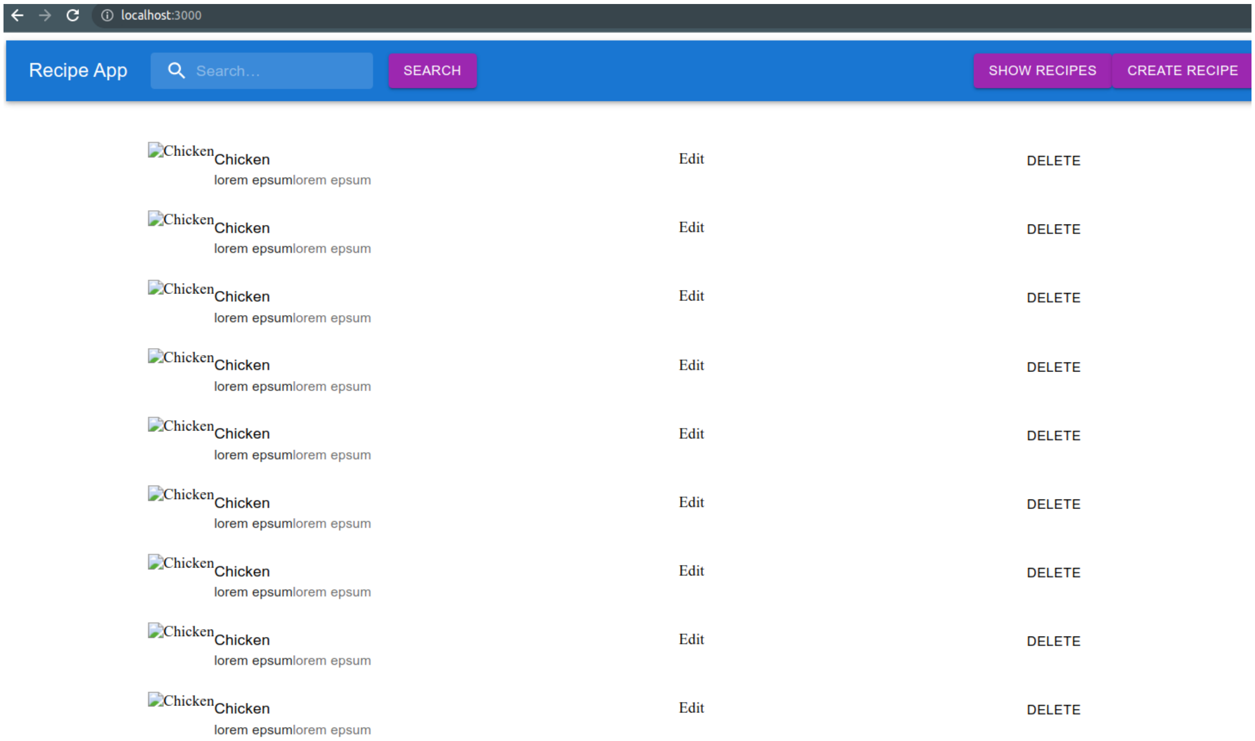
Show Recipes

```
import FormControl from "@mui/material/FormControl";
import InputLabel from "@mui/material/InputLabel";
import { Box, OutlinedInput, Typography } from "@mui/material";
import { Button } from "@mui/material";
import { useState } from "react";

const CreateRecipe = () => {
 const [loading, setLoading] = useState(false);
 const [recipe, setRecipe] = useState({
   title: "",
   description: "",
   ingredients: "",
   instructions: "",
   image: "",
 });
 const handleChange = (event) => {
   setRecipe({
     ...recipe,
     [event.target.name]: event.target.value,
   });
 };

 const onSubmit = async (event) => {
   event.preventDefault();
   setLoading(true);
   const { title, ingredients, instructions, description } = recipe;
   console.log(recipe);
   const response = await fetch("http://localhost:4000/recipes", {
     method: "POST",
     headers: {
       "Content-Type": "application/json",
     },
     body: JSON.stringify({
       title,
       ingredients,
       instructions,
       description,
     }),
   });
   const data = await response.json();
   console.log(data);
   setLoading(false);
 };

 return (
   <>
     {loading ? (
       <Typography variant="h4">Loading...</Typography>
     ) : (
       <Box
         sx={{
           display: "flex",
           flexDirection: "column",
           justifyContent: "center",
           width: "100%",
           marginTop: "2%",
           marginBottom: "2%",
           marginLeft: "10%",
         }}
       >
         <Typography variant="h4">Create Recipe</Typography>
         <form
           onSubmit={onSubmit}
           style={{
             display: "flex",

             flexDirection: "column",
             justifyContent: "center",
             width: "100%",
             marginTop: "2%",
           }}
         >
           <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
             <InputLabel htmlFor="component-outlined">Title</InputLabel>
             <OutlinedInput
```

```
               id="component-outlined"
               value={recipe.title}
               name="title"
               onChange={handleChange}
               label="Title"
             />
           </FormControl>
           <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
             <InputLabel
htmlFor="component-outlined">Description</InputLabel>
             <OutlinedInput
               id="component-outlined"
               value={recipe.description}
               name="description"
               onChange={handleChange}
               label="Description"
             />
           </FormControl>
           <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
             <InputLabel
htmlFor="component-outlined">Ingredients</InputLabel>
             <OutlinedInput
               id="component-outlined"
               value={recipe.ingredients}
               name="ingredients"
               onChange={handleChange}
               label="Ingredients"
             />
           </FormControl>
           <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
             <InputLabel
htmlFor="component-outlined">Instructions</InputLabel>
             <OutlinedInput
               id="component-outlined"
               value={recipe.instructions}
               name="instructions"
               onChange={handleChange}
               label="Instructions"
             />
           </FormControl>
           <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
             <Button variant="contained" component="label">
               Upload File
               <input type="file" hidden />
             </Button>
           </FormControl>
           <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
             <button
               type="submit"
               style={{
                 backgroundColor: "#4CAF50",
                 color: "white",
                 padding: "14px 20px",
                 margin: "8px 0",
                 border: "none",
                 cursor: "pointer",
                 width: "100%",
                 borderRadius: "4px",
                 fontSize: "16px",
                 fontWeight: "bold",
               }}
             >
               Submit
             </button>
           </FormControl>
         </form>
       </Box>
     )}
   </>
 );
};
export default CreateRecipe;
```



ChickenChicken
lorem epsumlorem epsum Edit DELETE

ChickenChicken
lorem epsumlorem epsum Edit DELETE

ChickenChicken
lorem epsumlorem epsum Edit DELETE

ChickenChicken
lorem epsumlorem epsum Edit DELETE

ChickenChicken
lorem epsumlorem epsum Edit DELETE

ChickenChicken
lorem epsumlorem epsum Edit DELETE

ChickenChicken
lorem epsumlorem epsum Edit DELETE

ChickenChicken
lorem epsumlorem epsum Edit DELETE

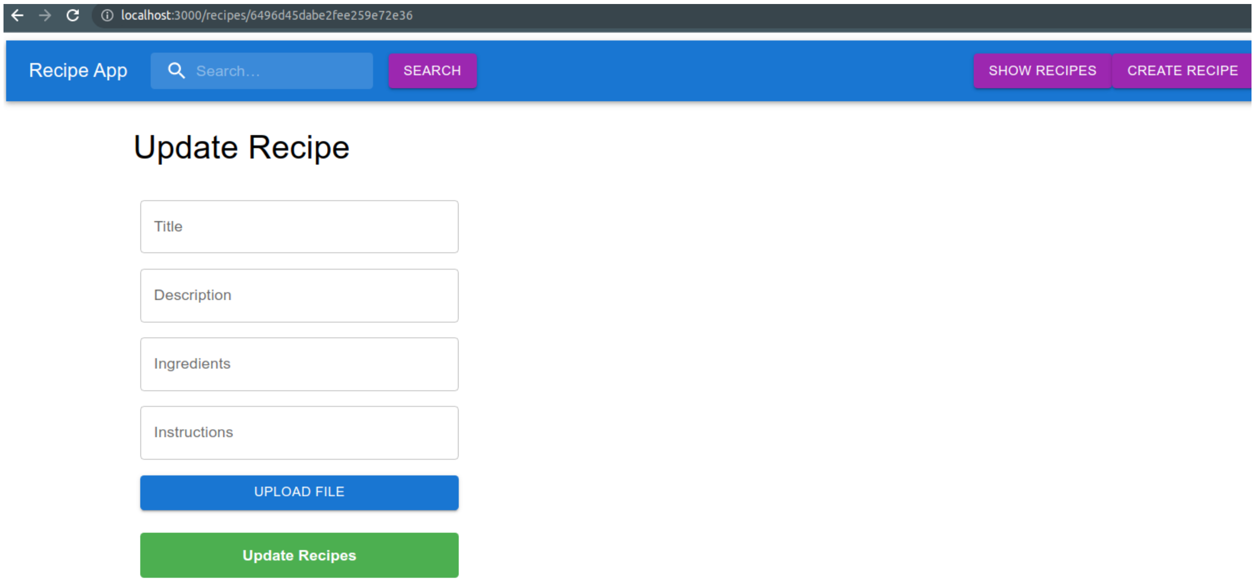
ChickenChicken
lorem epsumlorem epsum Edit DELETE

Edit Recipe

```
import FormControl from "@mui/material/FormControl";
import InputLabel from "@mui/material/InputLabel";
import { Box, OutlinedInput, Typography } from "@mui/material";
import { Button } from "@mui/material";
import { useEffect, useState } from "react";
import { Routes, Route, useParams } from "react-router-dom";
import Recipes from "./Recipes";

const EditRecipe = () => {
 let { id } = useParams();
 const [loading, setLoading] = useState(true);

 const [recipe, setRecipe] = useState({
   title: "",
   description: "",
   ingredients: "",
   instructions: "",
   image: "",
 });
 useEffect(() => {
   const fetchRecipe = async () => {
     const response = await fetch(`http://localhost:4000/recipes/${id}`);
     const data = await response.json();
     setRecipe(data);
   };
   fetchRecipe();
   setLoading(false);
 }, [id]);

 const handleChange = (event) => {
   setRecipe({
     ...recipe,
     [event.target.name]: event.target.value,
   });
 };

 const onSubmit = async (event) => {
   event.preventDefault();
   const { title, ingredients, instructions, description } = recipe;
   console.log(recipe);
   setLoading(true);
   const response = await fetch(`http://localhost:4000/recipes/${id}`, {
     method: "PUT",
     headers: {
       "Content-Type": "application/json",
     },
     body: JSON.stringify({
       title,
       ingredients,
       instructions,
       description,
     }),
   });
   const data = await response.json();
   setLoading(false);
   console.log(data);
```

```
};

return (
  <>
    {loading ? (
      <Typography variant="h4">Loading...</Typography>
    ) : (
      <Box
        sx={{
          display: "flex",
          flexDirection: "column",
          justifyContent: "center",
          width: "100%",
          marginTop: "2%",
          marginBottom: "2%",
          marginLeft: "10%",
        }}
      >
        <Typography variant="h4">Update Recipe</Typography>
        <form
          onSubmit={onSubmit}
          style={{
            display: "flex",

            flexDirection: "column",
            justifyContent: "center",
            width: "100%",
            marginTop: "2%",
          }}
        >
          <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
            <InputLabel htmlFor="component-outlined">Title</InputLabel>
            <OutlinedInput
              id="component-outlined"
              value={recipe.title}
              name="title"
              onChange={handleChange}
              label="Title"
            />
          </FormControl>
          <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
            <InputLabel
htmlFor="component-outlined">Description</InputLabel>
            <OutlinedInput
              id="component-outlined"
              value={recipe.description}
              name="description"
              onChange={handleChange}
              label="Description"
            />
          </FormControl>
          <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
            <InputLabel
htmlFor="component-outlined">Ingredients</InputLabel>
            <OutlinedInput
              id="component-outlined"
              value={recipe.ingredients}
              name="ingredients"
              onChange={handleChange}
              label="Ingredients"
            />
          </FormControl>
          <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
            <InputLabel
htmlFor="component-outlined">Instructions</InputLabel>
            <OutlinedInput
              id="component-outlined"
              value={recipe.instructions}
              name="instructions"
              onChange={handleChange}
              label="Instructions"
            />
          </FormControl>
          <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
            <Button variant="contained" component="label">
              Upload File
              <input type="file" hidden />
            </Button>
          </FormControl>
          <FormControl sx={{ m: 1, width: "25%" }} variant="outlined">
```

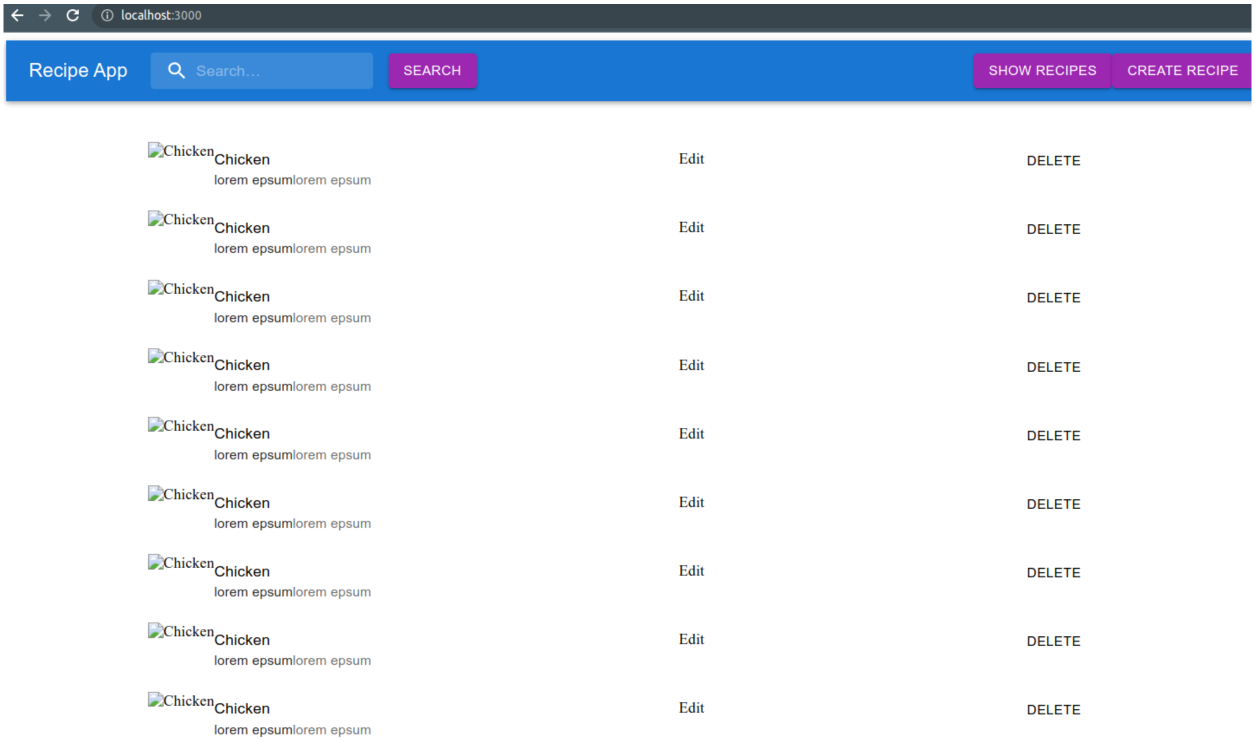
```
             <button
               type="submit"
               style={{
                 backgroundColor: "#4CAF50",
                 color: "white",
                 padding: "14px 20px",
                 margin: "8px 0",
                 border: "none",
                 cursor: "pointer",
                 width: "100%",
                 borderRadius: "4px",
                 fontSize: "16px",
                 fontWeight: "bold",
               }}
             >
               Update Recipes
             </button>
           </FormControl>
         </form>
       </Box>
     )}
   </>
 );
};
export default EditRecipe;
```



Recipes

```
import * as React from "react";
import List from "@mui/material/List";
import ListItem from "@mui/material/ListItem";
import Divider from "@mui/material/Divider";
import ListItemText from "@mui/material/ListItemText";
import { ImageListItemBar, ListItemButton } from "@mui/material";
import { ImageListItem } from "@mui/material";
import { Link } from "@mui/material";
import Button from "@mui/material/Button";

import Typography from "@mui/material/Typography";
import { Box } from "@mui/material";

export default function Recipes({ recipesList }) {
 const [recipes, setRecipes] = React.useState(recipesList || []);
 const [loading, setLoading] = React.useState(true);
 const [error, setError] = React.useState(null);
 React.useEffect(() => {
   if (recipesList) {
     setLoading(false);

     return;
   }
   fetch("http://localhost:4000/recipes/")
     .then((response) => response.json())
     .then((data) => {
       setRecipes(data);
       setLoading(false);
     })
     .catch((error) => {
       console.error("Error fetching data: ", error);
       setError(error);
       setLoading(false);
     });
 }, []);
 if (loading) return "Loading...";
 if (error) return "Error!";
 const handleDelete = async (id) => {
   const response = await fetch(`http://localhost:4000/recipes/${id}`, {
     method: "DELETE",
   });
   const data = await response.json();
   console.log(data);
   setRecipes(recipes.filter((recipe) => recipe.id !== id));
 };

 return (
   <Box
     sx={{
       display: "flex",
       flexDirection: "column",
       justifyContent: "center",
       width: "100%",
       marginTop: "2%",
       marginBottom: "2%",
       marginLeft: "10%",
     }}
   >
     <List
```

```
       sx={{
         width: "100%",
         maxWidth: "100%",
         marginRight: "10%",
         bgcolor: "background.paper",
       }}
     >
       {recipes.length === 0 ? (
         <Typography variant="h4">No recipes found!</Typography>
       ) : (
         recipes.map((recipe) => (
           <div key={recipe.id}>
             <ListItem alignItems="flex-start" width="100%">
               <ImageListItem key={recipe?.imageName}>
                 <img

src={`${"/public/logo192.png"}?w=248&fit=crop&auto=format`}

srcSet={`${recipe?.img}?w=248&fit=crop&auto=format&dpr=2 2x`}
                   alt={recipe.title}
                   loading="lazy"
                 />
                 <ImageListItemBar position="below" title={recipe?.author}
/>
               </ImageListItem>
               <ListItemText
                 primary={recipe.title}
                 secondary={
                   <React.Fragment>
                     <Typography
                       sx={{ display: "inline" }}
                       component="span"
                       variant="body2"
                       color="text.primary"
                     >
                       {recipe.description}
                     </Typography>
                     {recipe.ingredients}
                   </React.Fragment>
                 }
               />
               {/* add Edit Button */}
               <ListItemButton
                 component={Link}
                 to={`/recipes/${recipe["_id"]}`}
               >
                 Edit
               </ListItemButton>
               <ListItemButton
                 component={Button}
                 onClick={async () => await handleDelete(recipe["_id"])}
                 to={`/recipes/${recipe["_id"]}`}
               >
                 Delete
               </ListItemButton>
             </ListItem>
           </div>
         ))
       )}
     </List>
   </Box>
 );
}
```



```
ChickenChicken                    Edit          DELETE
lorem epsumlorem epsum
ChickenChicken                    Edit          DELETE
lorem epsumlorem epsum
ChickenChicken                    Edit          DELETE
lorem epsumlorem epsum
ChickenChicken                    Edit          DELETE
lorem epsumlorem epsum
ChickenChicken                    Edit          DELETE
lorem epsumlorem epsum
ChickenChicken                    Edit          DELETE
lorem epsumlorem epsum
ChickenChicken                    Edit          DELETE
lorem epsumlorem epsum
ChickenChicken                    Edit          DELETE
lorem epsumlorem epsum
ChickenChicken                    Edit          DELETE
lorem epsumlorem epsum
```

Index

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
import { createBrowserRouter, RouterProvider } from "react-router-dom";
import Layout from "./components/Layout";
import Recipes from "./components/Recipes";
import CreateRecipe from "./components/CreateRecipe";
import EditRecipe from "./components/EditRecipe";
import SearchRecipes from "./components/SearchRecipes";

const router = createBrowserRouter([
 {
   path: "/",
   element: <Layout children={<Recipes />} />,

   errorElement: <div>Not Found</div>,
 },
 {
   path: "/recipes/create",
   element: <Layout children={<CreateRecipe />} />,
 },
 {
   path: "/recipes/:id",
   element: <Layout children={<EditRecipe />} />,
 },
 {
   path: "/search/:search",
   element: <Layout children={<SearchRecipes />} />,
 },
 {
   path: "/recipes/:id/edit",
   element: <h1>Edit Recipe</h1>,
 },
 {
   path: "/recipes/new",
   element: <h1>New Recipe</h1>,
 },
]);

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
 <React.StrictMode>
   <RouterProvider router={router} />
 </React.StrictMode>
);
```

Server Js

```
const express = require("express");
const app = express();
const cors = require("cors");
const mongoose = require("mongoose");
const Recipe = require("./models/recipe");
const multer = require("multer");
const fs = require("fs");
const port = 4000;
const { MONGO_URI } = require("./config.js");
const upload = multer({ dest: "images/" });

mongoose
 .connect(MONGO_URI)
 .then(() => console.log("MongoDB connected successfully"))
 .catch((err) => console.log(err));

app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use(
 cors({
   origin: "http://localhost:3000",
   credentials: true,
 })
);

app.get("/", (req, res) =>
 res.json({ message: "server is running please got /recipes!" })
);

app.get("/recipes", async (req, res) => {
 try {
   const recipes = (await Recipe.find()).reverse();
   res.json(recipes);
 } catch (err) {
   console.error(err);
 }
});

app.post("/recipes", upload.single("image"), async (req, res) => {
 try {
   const { title, ingredients, instructions, description, imageName } =
     req.body;
   const image = req.file;
   // validate req.body
   // validate req.file
   // save file to disk
   if (!fs.existsSync("images")) {
     fs.mkdirSync("images");
   }
   if (
     title === undefined ||
     ingredients === undefined ||
     instructions === undefined ||
     description === undefined
   ) {
     res.status(400).json({ message: "Please fill out all fields" });
   }

   if (image !== undefined) {
     fs.writeFileSync(imageName + image.mimetype, content);
     //file written successfully
   }

   const recipe = new Recipe({
     title,
     ingredients,
     instructions,
     description,
     image,
   });
   await recipe.save();
   res.json(recipe);
 } catch (err) {
   console.error(err);
 }
});
app.delete("/recipes/:id", async (req, res) => {
 try {
   const { id } = req.params;
   console.log("delete ", id);
   const recipe = await Recipe.findByIdAndDelete(id);
   // TODO: also remove file from disk
   res.json(recipe);
 } catch (err) {
   console.error(err);
 }
});
app.put("/recipes/:id", async (req, res) => {
 try {
   const { id } = req.params;
   console.log(req.body);
   const recipe = await Recipe.findByIdAndUpdate(id, {
     $set: {
       ...req.body,
     },
   });
   res.json(recipe);
 } catch (err) {
   console.error(err);
 }
});

app.get("/search/:search", async (req, res) => {
 try {
   const { search } = req.params;
   console.log(search);
   const recipes = await Recipe.find({
     $or: [{ title: { $regex: search, $options: "i" } }],
   }).limit(5);
   res.json(recipes);
 } catch (err) {
   console.error(err);
   res.json({ message: "error" });
 }
});

app.listen(port, () => console.log(`Example app listening on port
${port}!`));
// Run the backend server with node backend/index.js
```