```
const express = require("express");
const path = require("path");
const mongoose = require("mongoose");

const PORT = process.env.PORT || 3001;
const app = express();
const apiRoutes = require("./routes/apiRoutes");

// Define middleware here
app.use(express.urlencoded({ extended: true }));
app.use(express.json());

// Serve up static assets (usually on heroku)
if (process.env.NODE_ENV === "production") {
  app.use(express.static("client/build"));
}

// Connect to the Mongo DB
mongoose.connect(
  process.env.MONGODB_URI || "mongodb://localhost/reactrecipes",
  { useUnifiedTopology: true, useNewUrlParser: true, useCreateIndex: true }
);

// Use apiRoutes
app.use("/api", apiRoutes);

// Send every request to the React app
// Define any API routes before this runs
app.get("*", function(req, res) {
  res.sendFile(path.join(__dirname, "./client/build/index.html"));
});
app.listen(PORT, function() {
  console.log(`□ ==> API server now on port ${PORT}!`);
});
```

Recipes pt1

```
import axios from "axios";
// The getRecipes method retrieves recipes from the server
// It accepts a "query" or term to search the recipe api for
export default {
  getRecipes: function(query) {
    return axios.get("/api/recipes", { params: { q: query } });
  }
};
```

```
const router = require("express").Router();
const db = require("../models");

router.get("/recipes", (req, res) => {
  // Use a regular expression to search titles for req.query.q
  // using case insensitive match.
            https://docs.mongodb.com/manual/reference/operator/query/regex/index.html
  db.Recipe.find({
    title: { $regex: new RegExp(req.query.q, 'i')}
  })
    .then(recipes => res.json(recipes))
    .catch(err => res.status(422).end());
});
module.exports = router;
```

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;

const recipeSchema = new mongoose.Schema({
  title: {
    type: String,
    required: true
  },
  // url string for thumbnail image
  thumbnail: {
    type: String,
    default: ""
  },
  // url for recipe web page - unique index
  href: {
    type: String,
    default: "",
    unique: true
  },
  // Not all ingredients, just the recommended ingredients from scraped web pages
  // from which seed data was sourced
  ingredients: [String]
});

const Recipe = mongoose.model("Recipe", recipeSchema);
module.exports = Recipe;
```

Recipes pt2

```
event => {
  // When the form is submitted, prevent its default behavior, get recipes update the recipes state
  event.preventDefault();
  API.getRecipes(recipeSearch)
    .then(res => setRecipes(res.data))
    .catch(err => console.log(err));
};

return (
  <div>
    <Nav />
    <Jumbotron />
    <Container>
      <Row>
        <Col size="md-12">
          <form>
            <Container>
              <Row>
                <Col size="xs-9 sm-10">
                  <Input
                    name="RecipeSearch"
                    value={recipeSearch}
                    onChange={handleInputChange}
                    placeholder="Search For a Recipe"
                  />
                </Col>
                <Col size="xs-3 sm-2">
                  <Button
                    onClick={handleFormSubmit}
                    type="success"
                    className="input-lg"
                  >
                    Search
                  </Button>
                </Col>
              </Row>
            </Container>
          </form>
        </Col>
      </Row>
      <Row>
        <Col size="xs-12">
          {!recipes.length ? (
            <h1 className="text-center">No Recipes to Display</h1>
          ) : (
            <RecipeList>
              {recipes.map(recipe => {
                return (
                  <RecipeListItem
                    key={recipe.title}
                    title={recipe.title}
                    href={recipe.href}
                    ingredients={recipe.ingredients}
                    thumbnail={recipe.thumbnail}
                  />
                );
              })}
            </RecipeList>
          )}
        </Col>
      </Row>
    </Container>
  </div>
);
}
export default App;
```

```
import React from "react";
import "./style.css";
// The Thumbnail component renders a div that uses
        some CSS to render a background image
// It will always keep square proportions at any size
        without the image warping
// The "role" and "aria label" are there to identify the
        element's purpose as an image for accessibility purposes
function Thumbnail({ src }) {
  return (
    <div
      className="thumbnail"
      role="img"
      aria-label="Recipe Image"
      style={{
        backgroundImage: `url(${src})`
      }}
    />
  );
}
export default Thumbnail;
```

```
import React from "react";
import Thumbnail from "../Thumbnail";
import { Container, Row, Col } from "../Grid";

// Exporting both RecipeList and RecipeListItem from this file
// RecipeList renders a bootstrap list item
export function RecipeList({ children }) {
  return <ul className="list-group">{children}</ul>;
}
// RecipeListItem renders a bootstrap list item containing data
        from the recipe api call
export function RecipeListItem({
  thumbnail,
  title,
  ingredients,
  href
}) {
  return (
    <li className="list-group-item">
      <Container>
        <Row>
          <Col size="xs-4 sm-2">
            <Thumbnail src={thumbnail || "https://placehold.it/300x300"} />
          </Col>
          <Col size="xs-8 sm-9">
            <h3>{title}</h3>
            <p>Ingredients: {ingredients}</p>
            <a rel="noreferrer noopener" target="_blank" href={href}>
              Go to recipe!
            </a>
          </Col>
        </Row>
      </Container>
    </li>
  );
}
```

```
import React from "react";
// This component lets us use a bootstrap input element without worrying about class names
// or manually wrapping the input with a form-group div
// All of the props passed to this component are spread onto the input element
function Input(props) {
  return (
    <div className="input-group input-group-lg">
      <input className="form-control" type="text" {...props} />
    </div>
  );
}
export default Input;
```

AJAX Books

```
import React, { useEffect, useState } from "react";
import Jumbotron from "../components/Jumbotron";
import DeleteBtn from "../components/DeleteBtn";
import API from "../utils/API";
import { Col, Row, Container } from "../components/Grid";
import { List, ListItem } from "../components/List";
import { Input, TextArea, FormBtn } from "../components/Form";

function Books() {
  // Setting our component's initial state
  const [books, setBooks] = useState([])
  const [formObject, setFormObject] = useState({})

  // Load all books and store them with setBooks
  useEffect(() => {
    loadBooks()
  }, [])

  // Loads all books and sets them to books
  function loadBooks() {
    API.getBooks()
      .then(res =>
        setBooks(res.data)
      )
      .catch(err => console.log(err));
  };
    return (
      <Container fluid>
        <Row>
          <Col size="md-6">
            <Jumbotron>
              <h1>What Books Should I Read?</h1>
            </Jumbotron>
            <form>
              <Input
                onChange={() => {}}
                name="title"
                placeholder="Title (required)"
              />
              <Input
                onChange={() => {}}
                name="author"
                placeholder="Author (required)"
              />
              <TextArea
                onChange={() => {}}
                name="synopsis"
                placeholder="Synopsis (Optional)"
              />
              <FormBtn
                disabled={!(formObject.author && formObject.title)}
                onClick={() => {}}
              >
                Submit Book
              </FormBtn>
            </form>
          </Col>
          <Col size="md-6 sm-12">
            <Jumbotron>
              <h1>Books On My List</h1>
            </Jumbotron>
            {books.length ? (
              <List>
                {books.map(book => {
                  return (
                    <ListItem key={book._id}>
                      <a href={"/books/" + book._id}>
                        <strong>
                          {book.title} by {book.author}
                        </strong>
                      </a>
                      <DeleteBtn onClick={() =>{}} />
                    </ListItem>
                  );
                })}
              </List>
            ) : (
              <h3>No Results to Display</h3>
            )}
          </Col>
        </Row>
      </Container>
    );
  }

export default Books;
```

```
import React from "react";
import "./style.css";

// This file exports both the List and ListItem components
export function List({ children }) {
  return (
    <div className="list-overflow-container">
      <ul className="list-group">{children}</ul>
    </div>
  );
}
export function ListItem({ children }) {
  return <li className="list-group-item">{children}</li>;
}
```

```
import React from "react";
import "./style.css";

// The ...props means, spread all of the passed props
        onto this element
// That way we don't have to
        define them all individually
function DeleteBtn(props) {
  return (
    <span className="delete-btn" {...props}
        role="button" tabIndex="0">
      □
    </span>
  );
}
export default DeleteBtn;
```

```
import axios from "axios";
export default {
  // Gets all books
  getBooks: function() {
    return axios.get("/api/books");
  },
  // Gets the book with the given id
  getBook: function(id) {
    return axios.get("/api/books/" + id);
  },
  // Deletes the book with the given id
  deleteBook: function(id) {
    return axios.delete("/api/books/" + id);
  },
  // Saves a book to the database
  saveBook: function(bookData) {
    return axios.post("/api/books", bookData);
  }
};
```

AJAX Form

```
import React, { useEffect, useState } from "react";
import Jumbotron from "../../components/Jumbotron";
import DeleteBtn from "../../components/DeleteBtn";
import API from "../../utils/API";
import { Col, Row, Container } from "../../components/Grid";
import { List, ListItem } from "../../components/List";
import { Input, TextArea, FormBtn } from "../../components/Form";

function Books() {
  // Setting our component's initial state
  const [books, setBooks] = useState([])
  const [formObject, setFormObject] = useState({
    title: "",
    author: "",
    synopsis: ""
  })
  // Load all books and store them with setBooks
  useEffect(() => {
    loadBooks()
  }, [])
  // Loads all books and sets them to books
  function loadBooks() {
    API.getBooks()
      .then(res =>
        setBooks(res.data)
      )
      .catch(err => console.log(err));
  };
  // Deletes a book from the database with a given id,
                then reloads books from the db
  function deleteBook(id) {
    API.deleteBook(id)
      .then(res => loadBooks())
      .catch(err => console.log(err));
  }
  // Handles updating component state when the user types into the input field
  function handleInputChange(event) {
    const { name, value } = event.target;
    setFormObject({...formObject, [name]: value})
  };
  // When the form is submitted,
                use the API.saveBook method to save the book data
  // Then reload books from the database
  function handleFormSubmit(event) {
    event.preventDefault();
    if (formObject.title && formObject.author) {
      API.saveBook({
        title: formObject.title,
        author: formObject.author,
        synopsis: formObject.synopsis
      })
        .then(() => setFormObject({
          title: "",
          author: "",
          synopsis: ""
        }))
        .then(() => loadBooks())
        .catch(err => console.log(err));
    }
  };

  return (
    <Container fluid>
      <Row>
        <Col size="md-6">
          <Jumbotron>
            <h1>What Books Should I Read?</h1>
          </Jumbotron>
          <form>
            <Input
              onChange={handleInputChange}
              name="title"
              placeholder="Title (required)"
              value={formObject.title}
            />
            <Input
              onChange={handleInputChange}
              name="author"
              placeholder="Author (required)"
              value={formObject.author}
            />
            <TextArea
              onChange={handleInputChange}
              name="synopsis"
              placeholder="Synopsis (Optional)"
              value={formObject.synopsis}
            />
            <FormBtn
              disabled={!(formObject.author && formObject.title)}
              onClick={handleFormSubmit}
            >
              Submit Book
            </FormBtn>
          </form>
        </Col>
        <Col size="md-6 sm-12">
          <Jumbotron>
            <h1>Books On My List</h1>
          </Jumbotron>
          {books.length ? (
            <List>
              {books.map(book => {
                return (
                  <ListItem key={book._id}>
                    <a href={"/books/" + book._id}>
                      <strong>
                        {book.title} by {book.author}
                      </strong>
                    </a>
                    <DeleteBtn onClick={() => deleteBook(book._id)} />
                  </ListItem>
                );
              })}
            </List>
          ) : (
            <h3>No Results to Display</h3>
          )}
        </Col>
      </Row>
    </Container>
  );
}

export default Books;
```

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;

const bookSchema = new Schema({
  title: { type: String, required: true },
  author: { type: String, required: true },
  synopsis: String,
  date: { type: Date, default: Date.now }
});

const Book = mongoose.model("Book", bookSchema);

module.exports = Book;
```

React Router

```
import React from "react";
import { BrowserRouter as Router, Route, Switch } from "react-router-dom";
import Books from "./pages/Books";
import Detail from "./pages/Detail";
import NoMatch from "./pages/NoMatch";
import Nav from "./components/Nav";

function App() {
  return (
    <Router>
      <div>
        <Nav />
        <Switch>
          <Route exact path={["/", "/books"]}>
            <Books />
          </Route>
          <Route exact path="/books/:id">
            <Detail />
          </Route>
          <Route>
            <NoMatch />
          </Route>
        </Switch>
      </div>
    </Router>
  );
}
export default App;
```

```
import React, { useEffect, useState } from "react";
import { Link, useParams } from "react-router-dom";
import { Col, Row, Container } from "../components/Grid";
import Jumbotron from "../components/Jumbotron";
import API from "../utils/API";

function Detail(props) {
  const [book, setBook] = useState({})
  // When this component mounts, grab the book
                with the _id of props.match.params.id
  // e.g. localhost:3000/books/599dcb67f0f16317844583fc
  const {id} = useParams()

  useEffect(() => {
    API.getBook(id)
      .then(res => setBook(res.data))
      .catch(err => console.log(err));
  }, [])

  return (
    <Container fluid>
      <Row>
        <Col size="md-12">
          <Jumbotron>
            <h1>
              {book.title} by {book.author}
            </h1>
          </Jumbotron>
        </Col>
      </Row>
      <Row>
        <Col size="md-10 md-offset-1">
          <article>
            <h1>Synopsis</h1>
            <p>
              {book.synopsis}
            </p>
          </article>
        </Col>
      </Row>
      <Row>
        <Col size="md-2">
          <Link to="/">← Back to Authors</Link>
        </Col>
      </Row>
    </Container>
  );
}

export default Detail;
```

```
import React, { useState, useEffect } from "react";
import DeleteBtn from "../components/DeleteBtn";
import Jumbotron from "../components/Jumbotron";
import API from "../utils/API";
import { Link } from "react-router-dom";
import { Col, Row, Container } from "../components/Grid";
import { List, ListItem } from "../components/List";
import { Input, TextArea, FormBtn } from "../components/Form";

function Books() {
  // Setting our component's initial state
  const [books, setBooks] = useState([])
  const [formObject, setFormObject] = useState({})
  // Load all books and store them with setBooks
  useEffect(() => {
    loadBooks()
  }, [])
  // Loads all books and sets them to books
  function loadBooks() {
    API.getBooks()
      .then(res =>
        setBooks(res.data)
      )
      .catch(err => console.log(err));
  };
  // Deletes a book from the database with a given id,
              then reloads books from the db
  function deleteBook(id) {
    API.deleteBook(id)
      .then(res => loadBooks())
      .catch(err => console.log(err));
  }
  // Handles updating component state
              when the user types into the input field
  function handleInputChange(event) {
    const { name, value } = event.target;
    setFormObject({...formObject, [name]: value})
  };
  // When the form is submitted,
              use the API.saveBook method to save the book data
  // Then reload books from the database
  function handleFormSubmit(event) {
    event.preventDefault();
    if (formObject.title && formObject.author) {
      API.saveBook({
        title: formObject.title,
        author: formObject.author,
        synopsis: formObject.synopsis
      })
        .then(res => loadBooks())
        .catch(err => console.log(err));
    }
  };
  return (
    <Container fluid>
      <Row>
        <Col size="md-6">
          <Jumbotron>
            <h1>What Books Should I Read?</h1>
          </Jumbotron>
          <form>
            <Input
              onChange={handleInputChange}
              name="title"
              placeholder="Title (required)"
            />
            <Input
              onChange={handleInputChange}
              name="author"
              placeholder="Author (required)"
            />
            <TextArea
              onChange={handleInputChange}
              name="synopsis"
              placeholder="Synopsis (Optional)"
            />
            <FormBtn
              disabled={!(formObject.author && formObject.title)}
              onClick={handleFormSubmit}
            >
              Submit Book
            </FormBtn>
          </form>
        </Col>
        <Col size="md-6 sm-12">
          <Jumbotron>
            <h1>Books On My List</h1>
          </Jumbotron>
          {books.length ? (
            <List>
              {books.map(book => (
                <ListItem key={book._id}>
                  <Link to={"/books/" + book._id}>
                    <strong>
                      {book.title} by {book.author}
                    </strong>
                  </Link>
                  <DeleteBtn onClick={() => deleteBook(book._id)} />
                </ListItem>
              ))}
            </List>
          ) : (
            <h3>No Results to Display</h3>
          )}
        </Col>
      </Row>
    </Container>
  );
}

export default Books;
```