




# Full Stack App with React and a REST API

## Checklist

	 Needs Work	 Meets Expectations	 Exceeds Expectations
React Project	<ul style="list-style-type: none"><li>• Project not set up using create-react-app.</li><li>• The React project's folder isn't named <code>client</code>.</li><li>• Warnings or errors in console about unused/missing assets.</li><li>• Running <code>npm start</code> doesn't launch the app.</li></ul>	<ul style="list-style-type: none"><li>• Project set up using create-react-app.</li><li>• The React project's folder is named <code>client</code>.</li><li>• No warnings or errors in console about unused/missing assets.</li><li>• Running <code>npm start</code> launches the app.</li></ul>	<ul style="list-style-type: none"><li>• N/A</li></ul>
REST API	<ul style="list-style-type: none"><li>• The REST API project's folder isn't named <code>api</code>.</li><li>• The REST API hasn't been updated with support for Cross-Origin Resource Sharing (CORS).</li></ul>	<ul style="list-style-type: none"><li>• The REST API project's folder is named <code>api</code>.</li><li>• The REST API has been updated with support for Cross-Origin Resource Sharing (CORS).</li></ul>	<ul style="list-style-type: none"><li>• N/A</li></ul>

## Stateful Class Components

- The app doesn't contain one or more of the required stateful class components.
  - One or more of the stateful class components doesn't behave in the expected way.
- The app contains the following stateful class components:
    - `Courses` - The component retrieves the list of courses from the REST API, renders a list of courses, links each course to its respective "Course Detail" screen, and renders a link to the "Create Course" screen.
    - `CourseDetail` - The component retrieves the detail for a course from the REST API, renders the course details, an "Update Course" button for navigating to the "Update Course" screen, and a "Delete Course" button that when clicked sends a DELETE request to the REST API to delete a course.
    - `UserSignIn` - The component renders a form allowing the user to sign using their existing account information, a "Sign In" button that when clicked signs in the user, and a "Cancel" button that returns the user to the default route (i.e. the list of courses).
  - The `CourseDetail` component redirects users to the `/notfound` path if the requested course isn't returned from the REST API.
  - The `UserSignIn` component redirects users back to the previous screen after successfully signing in.
  - The `UpdateCourse` component:
    - Redirects users to the `/notfound` path if the requested course isn't returned from the REST API.
    - Redirects users to the `/forbidden` path if the requested course isn't owned by the authenticated user.
  - Components redirect users to the `/error` path when requests to the REST API return a "500 Internal Server Error" HTTP status code.

- `UserSignUp` - The component renders a form allowing a user to sign up by creating a new account, a "Sign Up" button that when clicked sends a POST request to the REST API's `/api/users` route and signs in the user, and a "Cancel" button that returns the user to the default route (i.e. the list of courses).
- `CreateCourse` - The component renders a form allowing a user to create a new course, a "Create Course" button that when clicked sends a POST request to the REST API's `/api/courses` route, and a "Cancel" button that returns the user to the default route (i.e. the list of courses).
- `UpdateCourse` - The component renders a form allowing a user to update one of their existing courses, an "Update Course" button that when clicked sends a PUT request to the REST API's `/api/courses/:id` route, and a "Cancel" button that returns the user to the "Course Detail" screen.

### Stateless Functional Components

- The app doesn't contain one or more of the required stateless functional components.
  - One or more of the stateless functional components doesn't behave in the expected way.
- The app contains the following stateless functional components:
    - `Header` - The component renders the top menu bar and buttons for signing in and signing up (if there's not an authenticated user) or the user's first and last name and a button for signing out (if there's an authenticated user).
    - `UserSignOut` - The component signs out the authenticated user and redirects the user to the default route (i.e. the list of courses).
- The app contains the following stateless functional components:
    - `NotFound` - The component renders a message letting the user know that the requested page can't be found.
    - `Forbidden` - The component renders a message letting the user know that they can't access the requested page.
    - `UnhandledError` - The component renders a message letting the user know that an unexpected error has occurred.

## Routes

- The `react-router-dom` npm package isn't installed or listed as a dependency in the `package.json` file.
- Clicking a link doesn't navigate the user to the correct route or fails to display the appropriate info.
- The current route isn't always reflected in the URL.
- One or more of the required routes aren't configured.
- The `react-router-dom` npm package is installed and listed as a dependency in the `package.json` file.
- Clicking a link navigates the user to the correct route and displays the appropriate info.
- The current route is always reflected in the URL.
- The following routes are configured (listed in the format `path - component`):
  - `/ - Courses`
  - `/courses/create - CreateCourse`
  - `/courses/:id/update - UpdateCourse`
  - `/courses/:id - CourseDetail`
  - `/signin - UserSignIn`
  - `/signup - UserSignUp`
  - `/signout - UserSignOut`
- The following routes are configured (listed in the format `path - component`):
  - `/notfound - NotFound`
  - `/forbidden - Forbidden`
  - `/error - UnhandledError`
- React Router is configured so that if a route isn't matched the `NotFound` component is rendered.

## User Authentication

- The app's global state isn't kept in the `App` component or managed using the React Context API.
- A `signIn()` method isn't globally available or doesn't implement all of the required functionality.
- A `signOut()` method isn't globally available or doesn't implement all of the required functionality.
- The app's global state is kept in the `App` component or managed using the React Context API.
- A `signIn()` method is globally available that:
  - Authenticates a user using their email address and password.
  - Persists the authenticated user's information (including their password) in the global state.
- A `signOut()` method is globally available that removes the authenticated user's information (including their password) from the global state.
- The app persists user credentials using an HTTP cookie or local storage so that the user's authenticated state is maintained even if the application is reloaded or loaded into a new browser tab.

## Protected Routes

- The app doesn't contain a higher-order component (HOC) named `PrivateRoute`.
- One or more of the following routes aren't configured to use the `PrivateRoute` component:
  - `/courses/create`
  - `/courses/:id/update`
- The app contains a higher-order component (HOC) named `PrivateRoute` that is used to configure protected routes (i.e. routes that require authentication).
- The following routes are configured using the `PrivateRoute` component:
  - `/courses/create`
  - `/courses/:id/update`
- N/A

User Authorization	<ul style="list-style-type: none"> <li>The <code>CourseDetail</code> component renders the "Update Course" and "Delete Course" buttons regardless if: <ul style="list-style-type: none"> <li>There's an authenticated user.</li> <li>The authenticated user's ID matches that of the user who owns the course.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>The <code>CourseDetail</code> component only renders the "Update Course" and "Delete Course" buttons if: <ul style="list-style-type: none"> <li>There's an authenticated user.</li> <li>The authenticated user's ID matches that of the user who owns the course.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>N/A</li> </ul>
Validation Errors	<ul style="list-style-type: none"> <li>One or more of the "Sign Up", "Create Course", and "Update Course" screens don't display validation errors returned from the REST API.</li> </ul>	<ul style="list-style-type: none"> <li>The "Sign Up", "Create Course", and "Update Course" screens display validation errors returned from the REST API.</li> </ul>	<ul style="list-style-type: none"> <li>N/A</li> </ul>
Validation Errors	<ul style="list-style-type: none"> <li>One or more of the "Sign Up", "Create Course", and "Update Course" screens don't display validation errors returned from the REST API.</li> </ul>	<ul style="list-style-type: none"> <li>The "Sign Up", "Create Course", and "Update Course" screens display validation errors returned from the REST API.</li> </ul>	<ul style="list-style-type: none"> <li>N/A</li> </ul>
Markdown Support	<ul style="list-style-type: none"> <li>The "Course Detail" screen doesn't render the <code>courseDescription</code> or <code>materialsNeeded</code> properties as markdown formatted text.</li> </ul>	<ul style="list-style-type: none"> <li>The "Course Detail" screen renders the <code>courseDescription</code> and <code>materialsNeeded</code> properties as markdown formatted text.</li> </ul>	<ul style="list-style-type: none"> <li>N/A</li> </ul>
HTML and CSS	<ul style="list-style-type: none"> <li>Provided HTML or CSS is not used and/or significant layout issues are keeping the important aspects of the app from generally resembling the mockups.</li> </ul>	<ul style="list-style-type: none"> <li>Provided HTML and CSS is used and the important aspects of the app generally resemble the mockups.</li> </ul>	<ul style="list-style-type: none"> <li>N/A</li> </ul>
Valid JavaScript	<ul style="list-style-type: none"> <li>Code produces syntax errors.</li> </ul>	<ul style="list-style-type: none"> <li>Free of syntax errors.</li> </ul>	<ul style="list-style-type: none"> <li>N/A</li> </ul>

#### Code Comments

- No comments describing how your code and functions work.
- Detailed code comments explaining how your functions work.
- N/A