

Topic: ReactTS

Submission Date: 11 August 2025

Time: 09:00

Please submit by pushing to your github repository and submitting the [submission form](#)

Title: Task 3 - ReactTS Job Application Tracker

Objective

The objective of this task is to assess your understanding of React navigation, routing, URL queries and URL parameters, while testing your research skills when it comes to finding out ways to use new technology.

This is the third official task for React based on Lesson 3.

Scenario: You are tasked with building a simple (MVP) of a web application that allows job applicants to track the number of jobs they've applied for, which will help them assess how many applications are successful, pending and rejected. (You can extend the app in your free time to add more functionality such as an actual database, as this is only a mock application).

REQUIREMENTS

Interface

1. Create a user-friendly interface that is intuitive and easy to use
2. The interface should be responsive to different screens
3. Interface should make use of aesthetically pleasing color combinations
4. The interface should have a good layout that is easy to navigate
5. Pages:
 - a. Six Pages: Login, Registration, Home, landing, more details page (job page), 404 page
 - b. Login Page: Users can log in with their credentials.
 - c. Registration Page: New users can register with the following details:
 - i. Username
 - ii. Password
 - d. Home Page: Display Jobs you applied for

- e. Landing Page: Displays details about the purpose of the web application. You can be creative in terms of the other details to include
- f. Job Page: This page can be used to display more details about the job and the company, such as address, contact details, duties, requirements, and any data applicants could find relevant to know about the company for interview purposes. You can use your creativity to see what details you want users to optionally add, bear in mind to include all necessary inputs.
- g. 404 page: Catch all non-existent paths

Functionality:

1. Add Function: Users can add new jobs to the Job Application Tracker with the following details:
 - a. Company name
 - b. Role
 - c. Status(Applied, Interviewed, Rejected)
 - d. Date applied
 - e. Extra details about the company and the job such as job duties, requirements, etc
2. Delete Function: Users can delete existing Jobs from the job-application-tracker.
3. Update Function: Users can edit existing jobs on the job-application tracker.
4. Search Function: Users can search for items by company or role. The searched item should reflect on the URL bar
5. Status colors: use colours to represent status (e.g., Red for Rejected, Yellow for Applied, Green for Interviewed).
6. Filter function: Users can filter the jobs according to job status. The filter should reflect on the URL bar
7. Sort function: Users can sort by date. Descending and ascending. This should reflect on the URL bar

General Requirements:

- Implement CRUD (Create, Read, Update, Delete) operations for jobs you applied for.

- Use [JSON Server](#) to store user information and jobs you applied for.
- Ensure the application is responsive and user-friendly.
- Use proper validation for input fields to prevent errors.
- Implement user authentication and authorization to protect user data.
- Make sure to use queries, parameters in your application. Allowing users to interact with the application from the URL bar
- Make your components dynamic
- Use protected routing routing

Persistence

1. Make use of localStorage to store the link details

Concepts the task covers

1. Arrays and array methods
2. Objects and object methods
3. LocalStorage
4. React components
5. React state
6. React props
7. React hooks
8. JSON object and its methods
9. Web page responsiveness

Instructions:

1. Come up with your own design for the problem statement
2. Make use of reusable components wherever applicable
3. In terms of CSS the portal should be as responsive as possible, meaning it should be fully accessible on all platforms, whether desktop, laptop or mobile devices. All this without distorting the inputs you will have on the screen.
4. The main point of data storage will be [JSON server](#), which will allow you to add, update and delete from it.
5. Third party libraries for styling and responsiveness are allowed.

Evaluation Criteria:

1. User-friendliness of the design
 - Is the design intuitive
 - Does the design have a proper and understandable layout
 - Does the app flow from one function to another in an understandable and easy to follow manner
 - Does the app show notifications for processes occurring in the background
2. Overall design of the application
 - Does the design have colors that blend well together
 - Does the design maintain a consistent typography
3. Page interactivity
 - Does the mouse cursor change when hovering over clickable elements
 - Do clickable elements change color when hovered over to indicate interactivity
 - It is easy to navigate through the application's different pages (if more than one utilised)
4. Utilisation of React.js features
 - Did trainee create their own components
 - Were the components reused where appropriate
 - Was React state utilised properly
 - Were props sent and handled properly
 - Were reused components customised for similar design elements instead of creating new elements
 - Were queries utilised
 - Were parameters utilised
 - Were protected routes utilised correctly
 - Were all page 404s handled correctly
5. Responsiveness of the page
 - Is the page responsive to different web view sizes at different breakpoints, below is an example of common breakpoints:

- 320px
- 480px
- 768px
- 1024px
- 1200px