Hands-on Lab: Build the Notification Component

Estimated time needed: **90** minutes

Introduction

In this lab, you will create a **Notification** component, which must be available on every website page after an appointment is booked and disappears when the appointment is canceled.

Objectives

After completing this lab, you will be able to:

- Display notifications on appointment booking
- Remove notifications on appointment cancellation

Prerequisites

- You should have completed the prerequisite courses, especially the **Developing Front-End Apps with React** course.
- You must have completed the following labs:
    - Design Website Layouts
    - Create a GitHub Repository for your Project
    - Build Static Website Layouts
    - Set up the React Environment
    - Convert Static Pages To Dynamic React Components
    - Build the Appointment Booking Component
    - Integrate Existing Functionality

Exercise 1: Create the Notification component

1. In case you exited and are re-entering the Skills Network lab environment:
    1. Open new terminal.
    2. Clone your React project's GitHub repository.

        Refer to the **Capstone Project Reference: Git Commands** reading for details about Git command syntax and use relevant to the project.
2. Create the **Notification** folder in the **Components** folder under the React project's **src** folder.
3. Create the **Notification.js** component and the **Notification.css** file in the **Notification** folder.

4. Make sure to add the component code logic to the **Notification.js** file. Add code to ensure the notification disappears when a patient cancels a booking. Include logic to set the state variable in the **Notification** component to indicate that the notification should no longer be displayed.

    1. In the **Notification** component, add an event handler or effect that listens for changes in the state variable that indicates the cancellation of an appointment.

    2. When the state variable is updated to reflect a canceled appointment, set the state in the **Notification** component to hide the notification.

    3. Ensure that this state change in the **Notification** component triggers a re-render of the component to remove the notification from the page.

▶ Click here to for **sample solution**.

5. Enhance the existing code to incorporate additional notification details like the user name who booked appointment, appointment time, and appointment date, ensuring that these elements are appropriately displayed within the notification.

6. Implement the necessary elements and styling for the **Notification** component, such as a container `div` with appropriate CSS styles to display the notification at the desired position on the page.

7. Add a **state** variable to determine when the notification should be displayed or when not.

8. **Take a screenshot** of the output and save it as **notification_output.png**.

▶ Click here to for **sample screenshot**.

9. Perform `git add`, `git commit`, and `git push` commands to update changes into your React project's GitHub repository for proper code management.

    Refer to the **Capstone Project Reference: Git Commands** reading for details about Git command syntax and use relevant to the project.

Exercise 2: Integrate the Notification component in your application

1. In case you exited and are re-entering the Skills Network lab environment:

    1. Open new terminal.
    2. Clone your React project's GitHub repository.

2. Import the **Notification** component in the App.js component and place it in a way that the notification appears on every website page once a patient books an appointment.

▶ Click here for a **hint**.
▶ Click here to view the **exemplar code**.

3. **Take a screenshot** of code snippet for the integration and save it as **notification_integration.png**.

4. Perform `git add`, `git commit`, and `git push` commands to update changes into your React project's GitHub repository for proper code management.

    Refer to the **Capstone Project Reference: Git Commands** reading for details about Git command syntax and use relevant to the project.

Exercise 3: Test and refine the Notification component

Make sure your project client is running using `npm start` and also server side is running after navigating to server folder using `node index`.

1. Verify that the **Notification** component is rendered and visible on every website page.

2. Test the cancellation of an appointment and ensure that the notification disappears from all pages.

3. **Take a screenshot** of the output and save it as **notification_test.png**.

Remember to continuously test, refine, and iterate on your components to ensure they work smoothly and meet your requirements.

Screenshot checklist

You should have taken the following screenshots as part of this lab:

- *notification_output.png*
- *notification_integration.png*
- *notification_test.png*

Note about data management and persistence

To ensure the proper management and persistence of your data in a GitHub repository, it is crucial to follow a few essential steps:

- **Regular Updates:** Whenever you make changes or add new components to your project, it is essential to add, commit, and push the updates to your GitHub repository. This ensures that your latest work is safely stored and accessible to collaborators.

- **Session Persistence:** During an active session, your data remains accessible. However, it's important to note that if your session expires or you log out, you will need to clone the repository again to resume work.

- **Ignoring node modules:** When pushing data to GitHub, it's best practice to exclude the node modules folder from both your server and client directories. This folder contains external dependencies and can be quite large, making the repository heavy and slowing down the process. By adding it to the .gitignore file, you prevent it from being pushed to the repository, keeping your commits cleaner and more focused.

By adhering to these guidelines, you can maintain a well-organized and efficient GitHub repository, ensuring that your work is securely stored and easily accessible to you and your collaborators.

Author(s)

Richa Arora