

Hands-on Lab: Build the Reviews Component



Estimated time needed: **90** minutes

Introduction

In this lab, you will create a **Reviews** component, which will enable users to provide a feedback and rating for an instant or scheduled consultation.

Objectives

After completing this lab, you will be able to create review components to allow users to:

- Write and submit feedback
- Review feedback after submission

Prerequisites

- You should have completed the prerequisite courses, specially 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](#)
 - [Build the Appointment Booking Component](#), in particular the **Appointment Booking** component must be fully functional
 - [Integrate Existing Functionality](#)
 - [Build the Notification Component](#)

Exercise 1: Create the ReviewForm 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.
2. After that, create the **ReviewForm** folder within the **components** folder under the **src** folder. Inside create **ReviewForm.js** component and **ReviewForm.css** file.
src->components->ReviewForm->ReviewForm.js,ReviewForm.css
3. Define the **ReviewForm** component using the React Function Component syntax.

4. Display necessary information for the consultation to allow the patient to provide feedback and create a button to access the feedback form.

► [Click here to view the sample **Reviews** page.](#)

5. 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: Implement the input element and rating selector

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. Create a form that will appear after clicking on **Click Here** under the **Provide feedback** column. Add input field to allow users to enter their name and text area or input field to review text.

3. **Take a screenshot** of the Feedback form that users will submit and save it as **review_form.png**.

4. Include a **Submit** button within the **ReviewForm** component to allow users to submit their review.

► [Click here to view the sample **Feedback** form.](#)

► [Click here to view the **sample solution**.](#)

5. The example serves as a guide demonstrating the process of capturing a review message and posting it upon clicking the Submit button.

1. Your task involves displaying this message within the designated section of the ReviewForm component, denoted by the highlighted red border, as depicted in the screenshot below.

Reviews

Serial Number	Doctor Name	Doctor Speciality	Provide feedback	Review Given
1	Dr. John Doe	Cardiology	<button>Click Here</button>	
2	Dr. Jane Smith	Dermatology	<button>Click Here</button>	

2. You need to implement logic to disable button from which user can give feedback to submit feedback for once.

3. **Take a screenshot** and save it as **disable_review-button.png**.

6. You need to implement a rating selector element, which can include either stars or numbers ranging from 1-5.

► [Click here to view the sample **Feedback** page with star rating.](#)

7. **Take a screenshot** of the form with the rating feature and save it as **rating_selector.png**.

8. 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: Style the ReviewForm 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.
2. Apply appropriate CSS styles to the ReviewForm component to make it visually appealing and ensure it fits well within the layout of your application.
3. Consider using CSS frameworks or custom styles to enhance the user experience.
4. **Take a screenshot** of the integration and save it as **review_form_style.png**.
5. Remember to perform `git add`, `git commit`, and `git push` all changes in your forked GitHub repository.

Refer to the [Capstone Project Reference: Git Commands](#) reading for details about Git command syntax and use relevant to the project.

Exercise 4: Integrate the ReviewForm 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.
2. Import the **ReviewForm** component into the relevant page or component where you want to display the review form.

► **Click here for a hint.**

3. Ensure you have access to the necessary data, such as the doctor's information and the appointment details, to display context-specific information in the review form.
4. Remember to perform `git add`, `git commit`, and `git push` all changes in your forked GitHub repository.

Refer to the [Capstone Project Reference: Git Commands](#) reading for details about Git command syntax and use relevant to the project.

Exercise 5: Test and refine the ReviewForm 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.
2. Test the **ReviewForm** component by entering a review and rating, submitting the form, and verifying that the data is handled correctly.
3. Validate the behavior of the review component in different scenarios, such as when the user has multiple appointments or when reviewing other doctors.
4. **Take a screenshot** of the out and save it as **test_rev.png**.
5. Remember to perform `git add`, `git commit`, and `git push` all changes in your forked GitHub repository.

Refer to the [Capstone Project Reference: Git Commands](#) reading for details about Git command syntax and use relevant to the project.

Screenshot checklist

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

- *review_form.png*
- *disable_review-button.png*
- *rating_selector.png*
- *review_form_style.png*
- *test_rev.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

© IBM Corporation. All rights reserved.