Hands-on Lab: Build the Appointment Booking Component

Estimated time needed: **90** minutes

Introduction

In this lab, you will build the **Appointment Booking** component with a search bar, cards for available doctors, and functionality to book or cancel appointments.

Objectives

After completing this lab, you will be able to create the dynamic React components associated with Appointment Booking including:

- Search bar
- Ability to list available doctors based on the search criteria
- Book appointments
- Cancel appointments

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
    - Convert Static Pages To Dynamic React Components
    - Instant Consultation

Exercise 1: Create the FindDoctorSearch 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 **FindDoctorSearch** folder in the **Components** folder under the React project's **src** folder.

3. Create the **FindDoctorSearch.js** component and the **FindDoctorSearch.css** file in the **FindDoctorSearch** folder.

4. Add the search bar functionality within the **FindDoctorSearch.js** component. An example of search bar functionality is an input field for entering a doctor's specialty.

5. Implement onFocus for displaying a list of doctor specialties and onBlur to hide doctors's list when clicked outside anywhere of the webpage.

    Refer to the **Capstone Project Technical Reference** reading for details about the onFocus and onBlur event handlers and other technical concepts.

    Note: For steps 4 and 5, you can use the **FindDoctorSearchIC.js** file within **InstantConsultation** –> **FindDoctorSearchIC** folder as reference.

6. Optionally, include additional search parameters or filters based on your requirements.

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

7. Implement stylesheet for **FindDoctorSearch.css**. You can leverage the stylesheet **FindDoctorSearch.css** from the **FindDoctorSearchIC.css** in **InstantConsultation** –> **FindDoctorSearchIC** folder.

8. **Take a screenshot** of the code snippet and save it as **docsearch.png**.

9. **Take a screenshot** of the output to search doctors and save it as **docsearch_output.png**.

10. 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: Create the DoctorCard 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. Create the **DoctorCard** folder in the **Components** folder under the React project's **src** folder.

3. Create the **DoctorCard.js** component and the **DoctorCard.css** file in the **DoctorCard** folder.

4. Add different elements for doctor details, such as name, experience, and rating, in the **DoctorCard.js** component.

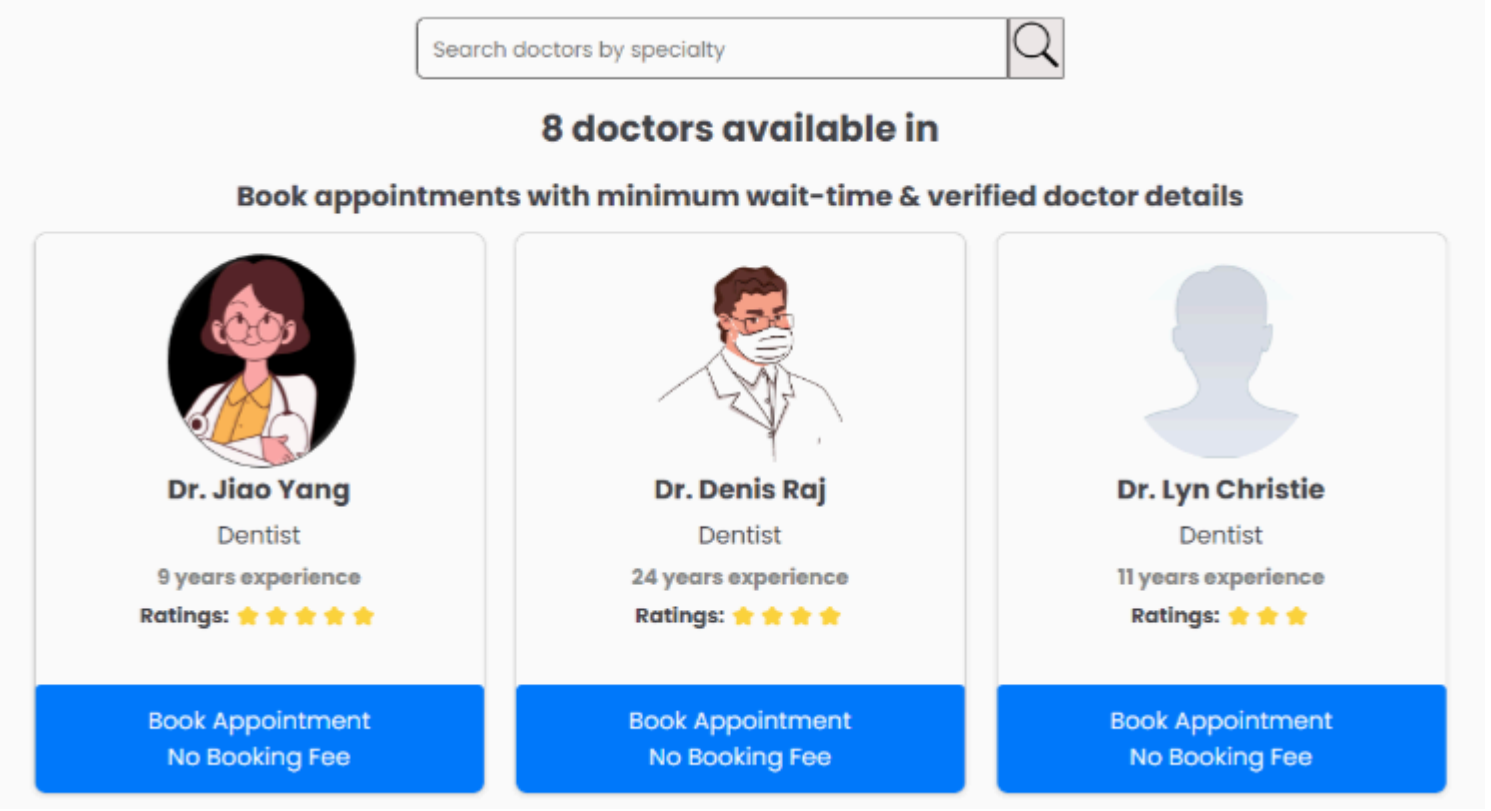
5. Optionally, include additional elements like an image and career profile, in the **DoctorCard.js** component.

    Note: For steps 4 and 5 you can refer **DoctorCardIC.js** file within **InstantConsultation** –> **DoctorCardIC** folder. You need to ignore the code written in between `<div className="doctor-card-options-container"> </div>`, which is part of **AppointmentForm** that you will perform in next exercise, instead you need to perform step 6.

6. Add a button element to allow users to book an appointment for the doctor, in the **DoctorCard.js** component.

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

7. Perform steps 4 to 6 for multiple doctors.



8. Implement stylesheet for **DoctorCard.css**. You can leverage the stylesheet **DoctorCard.css** from the **DoctorCardIC.css** in **InstantConsultation** –> **DoctorCardIC** folder.

9. **Take a screenshot** of the doctor card, which has been created by you and named as **doctor_card.png**

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

    For details on Git command syntax and use relevant to the project, refer to the **Capstone Project Reference: Git Commands** reading.

Exercise 3: Create the AppointmentForm 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 a folder named **AppointmentForm** in the **Components** folder under the **src** folder.

3. Next, create the **AppointmentForm.js** component file in the **AppointmentForm** folder.

4. Create form in **AppointmentForm.js** file to be displayed after clicking the **Book Appointment** button.

5. Implement the form elements within the **AppointmentForm.js** component. Examples of elements that you can include in the form include patient's name, appointment date, and appointment time.

6. Link the **AppointmentForm.js** component within **DoctorCard.js** component for booking an appointment.

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

7. You need to add two more elements, one for to book time slot and other to book for specified date for the appointment to book for coming dates.

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

9. Optionally, include validation logic to ensure the form is filled out correctly before submission.

10. **Take a screenshot** of the code snippet and save it as **apptform.png**.

11. 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 4: Implement book/cancel appointment functionality

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. Implement the booking and cancellation functionality within the **DoctorCard** component. This can be done by adding buttons or elements for booking and cancelling appointments, with appropriate event handlers.

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

6. 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 5: Integrate the components into 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. Create one **BookingConsultation.js** file inside components folder.

3. Import DoctorCard.js and FindDoctorSearch.js inside **BookingConsultation.js** and integrate their functionality within this component.

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

4. Import the necessary **BookingConsultation.js** component into **App.js** to book and cancel the appointment.

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 6: Test and refine the components

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.

    3. Perform `npm install` in the React project's **root** folder and the **server** folder.
2. From the React project's server folder, perform `node index`.
3. From the React project's root folder, perform `npm start`.
4. Launch the client side of the React project.
5. Validate the **FindDoctorSearch** functionality by entering different specialties and verifying that the search results are displayed correctly.
6. Test the **DoctorCard** component to ensure it accurately displays the doctor's information.
7. Verify that the **AppointmentForm** component captures the entered data correctly and allows form submission.
8. Test the overall flow of the **BookAppointment** functionality, from searching for doctors to booking appointments, ensuring that data is passed correctly between components and the desired actions are triggered to handle booking and cancellation events appropriately.
9. **Take a screenshot** of the output and save it as **appttest.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:

- *docsearch.png*
- *docsearch_output.png*
- *doctor_card.png*
- *apptform.png*
- *appttest.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.