Hands-on Lab: Integrate Existing Functionality

Estimated time needed: **30** minutes

Introduction

In this lab, you will integrate the **Instant Consultation** component into your project. The **Instant Consultation** component provides real-time functionality for your application. You will clone the **Instant Consultation** component from a GitHub repository and integrate it with other components of your project. By the end of this lab, you will have a fully functioning project with integrated instant functionality.

Objectives

After completing this lab, you will be able to:

- Integrate an existing React component for Instant Consultation
- Test the integration

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

Exercise 1: Integrate the Instant Booking Component

1. Download the folder from given link, which includes **Instant Consultation** folder.

   InstantConsultationBooking

2. Add the **Instant Consultation** folder in the **Components** folder under the React project's **src** folder. Import the **Instant Consultation.js** component into the **App.js** component.

3. You will need to install reactjs-popup using `npm i reactjs-popup` in the terminal of React project's **root** folder.

   Refer to the **Capstone Project Technical Reference** reading for details about the reactjs-popup library and other technical concepts.

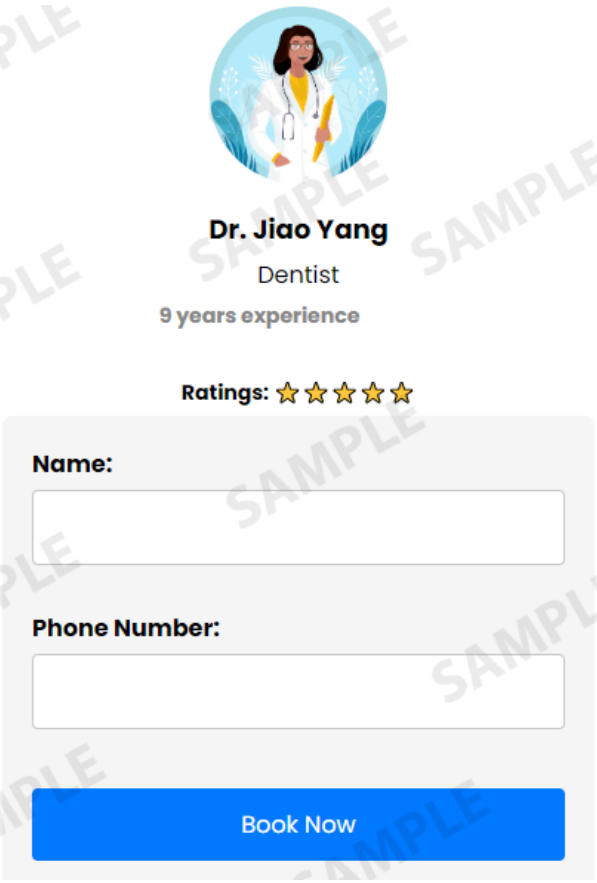
4. Review the documentation or README file provided with the **Instant Consultation** component. Understand the component structure, dependencies, and configuration requirements.

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

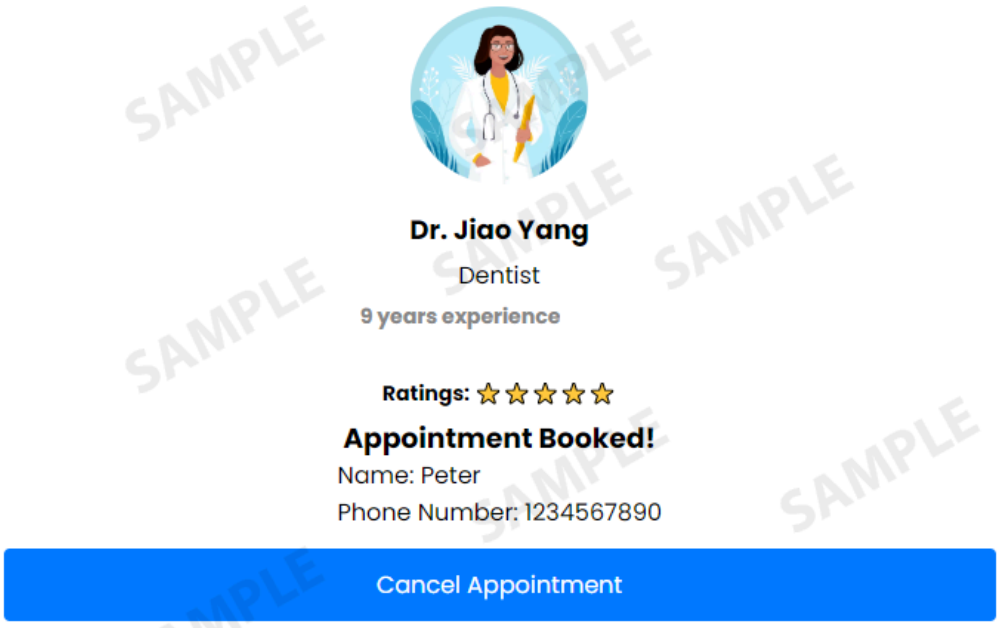
    Refer to the **Capstone Project Reference: Git Commands** reading for details about Git command syntax and use relevant to the project.

6. Launch the React project's client side.

7. You should see the Instant Consultation form similar to the image below.



8. When a user confirms an instant booking, the card will look similar to the image below.



9. You can change the design layout according to your application theme.

Exercise 2: Integrate the Instant Consultation component

1. Identify the integration points in your existing React project where the **Instant Consultation** component must be included.

2. Integrate the **Instant Consultation** component in your existing React project inside **App.js** as:
   `<Route path="/instant-consultation" element={<InstantConsultation />} />`

3. **Take a screenshot** for the working of this component **instant_consultation.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.

Screenshot checklist

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

- *instant_consultation.png*

Note about data management and persistence

Make sure that your project client side is running using `npm start` and server side is running using `node index`. 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