REACT-UNDERSTANDING JSX

IIHT

Time To Complete: 10 to 12 hr

CONTENTS

1 Project Abstract

In this micro-frontend unit, participants will explore the basics of JSX and dynamic rendering in React. The goal is to build a small React SPA that introduces key concepts such as JSX syntax, conditional rendering, variable embedding, and reusable components. This foundational project sets the stage for more advanced component communication and state management.

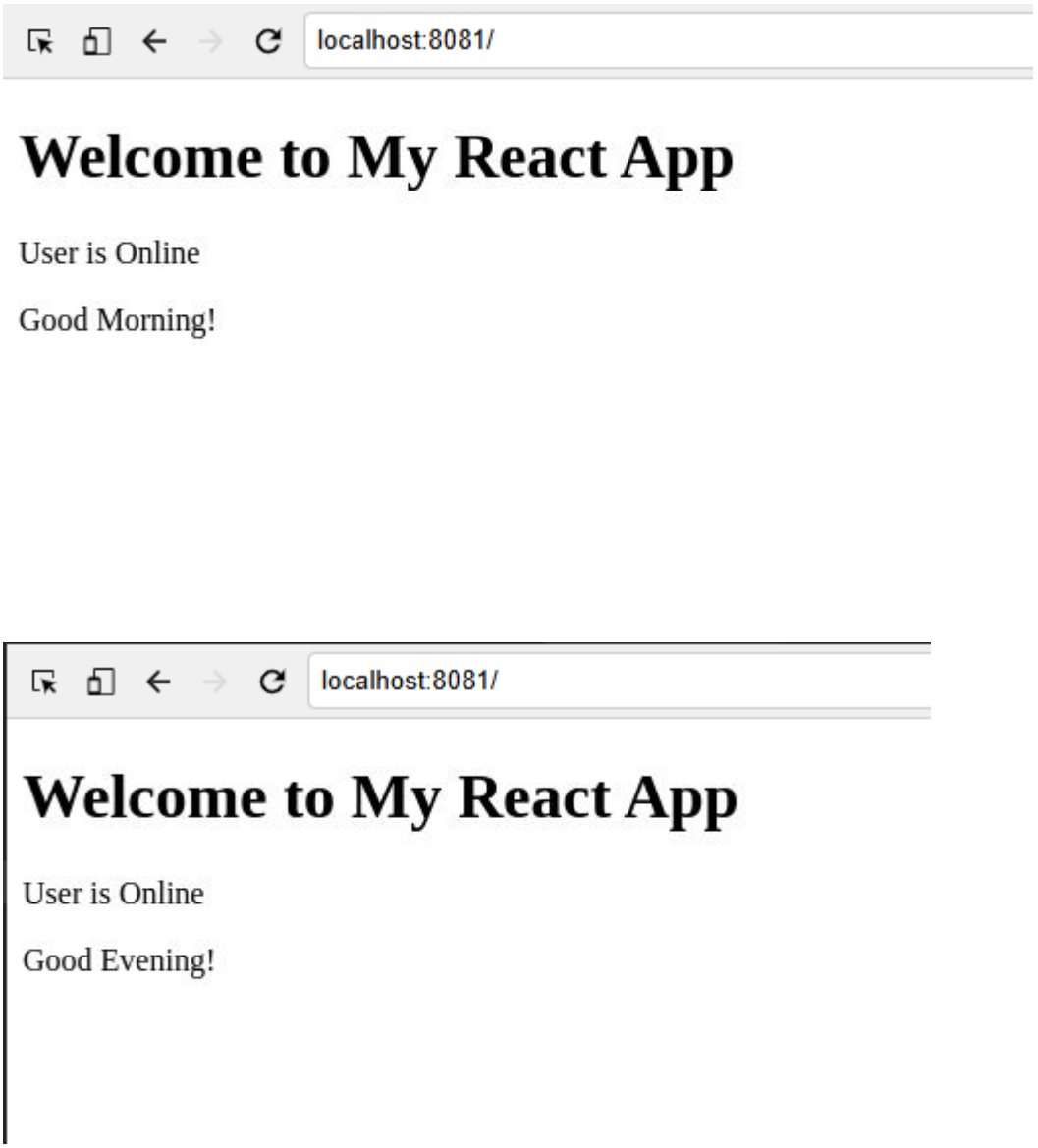
2 Problem Statement

You are tasked with creating a React Single Page Application (SPA) that demonstrates JSX rendering with conditional statements and embedded variables. The application will simulate a user interface that greets users based on the time of day, displays user online/offline status, and showcases core JSX features.

3 Proposed Understanding JSX Application Wireframe

UI needs improvisation and modification as per given use case and to make test cases passed.

3.1 ScreenShots



4 BUSINESS-REQUIREMENT:

As an application developer, develop the React-Understanding JSX (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	As a user I should be able to visit the welcome page as the default page. Acceptance criteria: **AppComponent:** 1. Display a static greeting message at the top of the page. 2. Display a message indicating whether the user is online or offline. 3. Display a different greeting depending on whether it is morning or evening. **State & Functionality Overview:** 1. **No dynamic state** is used in this component — all values are defined as static constants within the function body. **Variables Used:** 1. greetingMessage ● Type: `string` ● Purpose: A static heading message welcoming the user. 2. isOnline ● Type: `boolean` ● Purpose: Simulates the user's online status. 3. timeOfDay ● Type: `string` ● Purpose: Simulates whether it's morning or evening. 4. userStatus ● Value derived using a ternary operator. ● Displays `"User is Online"` if `isOnline` is `true`; otherwise `"User is Offline"`. 5. timeGreeting ● Conditioned using `if` statements based on `timeOfDay`. ● Shows `"Good Morning!"` or `"Good Evening!"`

		Conditional Rendering Logic: 1. Use of **ternary operator**: const userStatus = isOnline ? "User is Online" : "User is Offline"; 2. Use of **`if` statements** to set a message based on the time of day: • If `timeOfDay` is `"morning"` → set `timeGreeting` to `"Good Morning!"` • If `timeOfDay` is `"evening"` → set `timeGreeting` to `"Good Evening!"` **HTML Structure:** 1. Wrap the entire content in a `<div>` tag. 2. Use an `<h1>` to render the **greetingMessage**. 3. Use a `<p>` tag to show the **userStatus**. 4. Use another `<p>` tag to display the **timeGreeting** message. **** Kindly refer to the screenshots for any clarifications. ****

5 CONSTRAINTS

1. You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.

6 MANDATORY ASSESSMENT GUIDELINES

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command**

compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.

5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**

   Note: The application will not run in the local browser
7. **You can follow series of command to setup React environment once you are in your project-name folder:**
   a. **npm install -> Will install all dependencies -> takes 10 to 15 min.**
   b. **npm run start -> To compile and deploy the project in browser. You can press the <Ctrl> key while clicking on localhost:4200 to open the project in the browser -> takes 2 to 3 min.**
   c. **npm run test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min.**
8. **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.**
9. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**