REACT-WORKING WITH COMPONENTS

IIHT

Time To Complete: 10 to 12 hr

CONTENTS

1 Project Abstract

Component-based architecture is at the heart of React. In this assignment, participants will build a simple application that demonstrates how to break down a UI into reusable, isolated components. The goal is to understand how to structure a React application using multiple components like Header, Card, and Footer, and how to organize them inside the main `App` component.

2 Problem Statement

You are tasked with creating a React application that renders a structured layout using three core components:

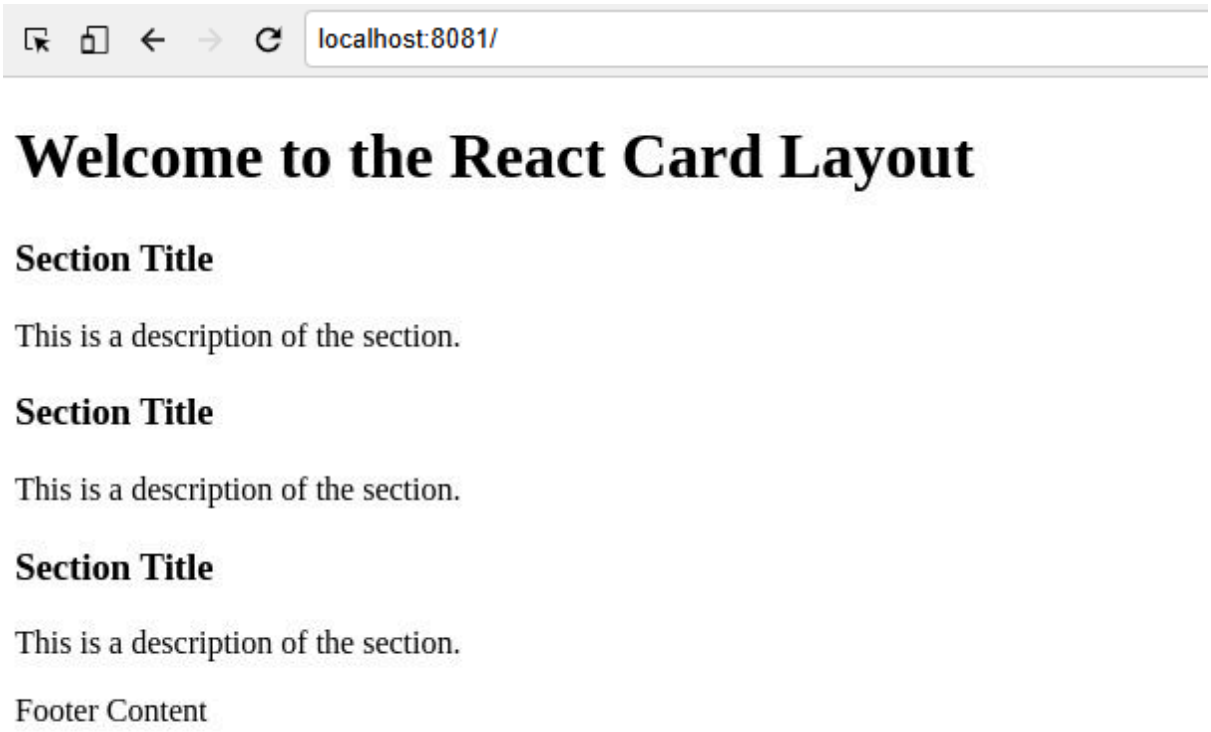
- `Header` – Displays a welcome message.
- `Card` – Displays repeated content sections.
- `Footer` – Displays closing content.

The main application (App.js) should import and use these components to build a reusable and clean layout.

3 Proposed Working with Components 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 Working with Components (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: 1. The page should render a header at the top with a welcome message. 2. Display **three identical cards**, each with a title and a short description. 3. Display a footer at the bottom with a simple footer message.

4. All UI should be composed using **separate functional components**.

Component Overview:

AppComponent:

1. Acts as the root container.
2. Renders:
    - `<Header />` at the top
    - A container with three `<Card />` components
    - `<Footer />` at the bottom

Header Component:

1. Displays a large heading at the top of the page.
2. Static text: **"Welcome to the React Card Layout"**

Card Component:

1. Represents a content section.
2. Includes:
    - A `<h3>` tag with static title: `"Section Title"`
    - A `<p>` tag with static description: `"This is a description of the section."`

Footer Component:

1. Displays a simple message in a `<footer>` tag.
2. Static text: **"Footer Content"**

HTML Structure:

1. **App Component** layout:
    - Use a top-level `<div>` to wrap everything.
    - Inside:
        - ➔ Render the `Header` component.
        - ➔ Render a `<div>` containing **three Card components**.
        - ➔ Render the `Footer` component.
2. **Header:**
    - Contains a `<header>` element with a main heading (`<h1>`).
3. **Card:**
    - Contains a `<div>` with a:
        - ➔ Heading (`<h3>`)
        - ➔ Paragraph (`<p>`)

		4. Footer: • Contains a `<footer>` element with a `<p>` tag. **** 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 jest -> to run all test cases and see the summary.**

d. **npm run test -> to run all test cases and register results on the server. 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.**