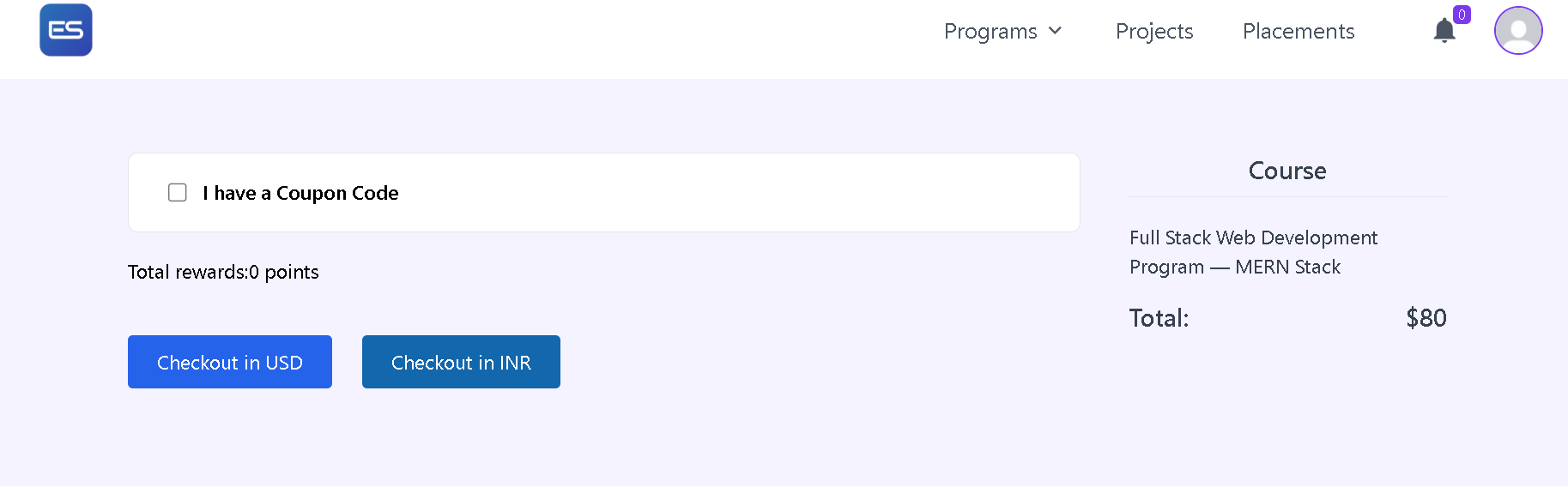
**Checkout** [**https://citria.org/course/checkout/649714c70b1be79305efd21b**](https://citria.org/course/checkout/649714c70b1be79305efd21b)

****

The provided code is a React component named Checkout. It seems to be part of a larger web application, most likely an e-commerce platform for purchasing online courses. Let me break down the code and explain its main functionalities:

Imports:

The component utilizes several third-party libraries and React-related modules, such as Axios for making HTTP requests, React Router for navigation, and some custom components like Header, PageTitle, Spinner, ApplyCoupon, and Summary.

Razorpay Integration:

The component includes functions for loading the Razorpay payment script dynamically and handling the payment verification process using Razorpay API.

State Management:

The component uses React useState hooks to manage various states, such as isRedeemed, rewardData, rewardDiscount, updatedPrice, isDiscount, isChecked, billingAddress, btnLoading, and razorpayRes.

Effect Hooks:

The useEffect hook is utilized to perform side effects in the component.

One useEffect fetches billing address data when the component mounts.

Another useEffect listens for changes in razorpayRes and verifies the payment using a backend API.

Billing Address:

The component fetches billing address data from the server and displays an error message if the address is not available.

Form Submission:

The handleSubmit function is triggered when the user submits the checkout form.

Depending on the selected payment method (usd or inr), the component either redirects the user to a Stripe checkout URL or initiates a Razorpay payment.

Coupon Code:

The component allows users to enter a coupon code and fetches discount information from the server using the ApplyCoupon component.

Reward Points:

Users can redeem reward points for discounts on the course price.

The handeleReedemRewardPoint function makes a request to the server to redeem reward points and updates the component state accordingly.

UI Rendering:

The component renders UI elements, including buttons for selecting the payment currency, information about reward points, and a summary of the selected course.

Loading Spinner:

A loading spinner is displayed when certain asynchronous operations, such as fetching data or processing payments, are in progress.

Conditional Rendering:

Certain UI elements are conditionally rendered based on the state of the component, such as whether a coupon code is checked, reward points are redeemed, or the page is in a loading state.

Navigation:

The useNavigate hook from React Router is used for programmatic navigation.

In summary, this component represents a checkout page for purchasing a course, providing features like coupon code application, reward point redemption, and integration with payment gateways like Razorpay and Stripe.

The provided code defines a React component named **StripeCheckoutMsg.** This component is designed to display a message based on the URL query parameters, particularly focusing on the success or cancellation of a payment.

Component Structure:

PageTitle Component:

Utilizes the PageTitle component to set the title of the page dynamically based on the URL query parameters (success or canceled).

Conditional Rendering:

Uses conditional rendering to display different messages and UI elements based on whether the payment was successful (queryObj?.success) or canceled (queryObj?.canceled).

Icons:

Utilizes icons from the react-icons library (BsCheck2Circle and SlClose) to visually represent success and cancellation.

Navigation:

Uses the useNavigate hook from react-router-dom for programmatic navigation when the user clicks the "OK" button.

Styling:

Applies styles using Tailwind CSS classes for responsiveness and aesthetics.

Message Display Logic:

If the payment is canceled (queryObj?.canceled is truthy), it displays a message indicating the payment cancellation along with an error icon.

If the payment is successful (queryObj?.success is truthy), it displays a message indicating the payment success along with a success icon.

User Interaction:

Provides an "OK" button that, when clicked, navigates the user to the home page ('/') if the payment is canceled or to the user's course profile page ('/profile/course') if the payment is successful.

This component is likely used in conjunction with a Stripe checkout process, and it provides a clear and visually appealing way to communicate the payment outcome to the user.

The **ApplyCoupon** component is a React functional component designed to handle the application of coupon codes within a React application. Below is an explanation of the code:

Component Structure:

State Hooks:

Utilizes the useState hook to manage three state variables:

state: Stores the user-inputted coupon code.

error: Stores error messages, if any, during the coupon application process.

loading: Indicates whether a coupon application request is in progress.

Navigation Hook:

Uses the useNavigate hook from react-router-dom to enable programmatic navigation.

handleApplyCoupon Function:

An asynchronous function triggered when the user clicks the "Apply" button.

Initiates the loading state to provide feedback to the user.

Makes an asynchronous POST request to a coupon application endpoint using Axios.

Updates the discount if the coupon application is successful and handles errors gracefully.

Input Field:

Renders an input field for the user to enter the coupon code.

Updates the state variable on user input.

Error Display:

Conditionally displays an error message using the BsExclamationCircleFill icon if an error occurs during coupon application.

Apply Button:

Triggers the handleApplyCoupon function on click.

Styled using Tailwind CSS classes for appearance and interactivity.

Loading Spinner:

Renders a spinner component (Spinner) while the coupon application is in progress.

Usage:

Import this component into the parent component where coupon functionality is required.

Pass the necessary props (id, price, setIsDiscount) to enable coupon application.

Dependencies:

Axios for making HTTP requests.

React Icons for visual elements.

React Router DOM for navigation.

Tailwind CSS for styling.

This component provides a user-friendly interface for applying coupon codes with real-time feedback on errors and loading states.