**Checkout/Checkout Enterprise Component Documentation**

**Introduction**

The Checkout component is a React functional component designed to handle the subscription checkout process for the pricing plans in the application. It integrates with the payment gateway to initiate subscription purchases and handle payment verifications.

**Usage**

To use the Checkout component, import it into your React application and place it within the pricing plan cards where users can initiate the checkout process.

import Checkout from './Checkout';

<Checkout/>

**Features**

1. **Subscription Purchase**: The Checkout component allows users to purchase a subscription plan by clicking the "Buy" button.
2. **Login Check**: Before initiating the checkout process, the component checks whether the user is logged in. If not logged in, it redirects the user to the login page.
3. **Payment Integration**: The component integrates with the payment gateway to handle subscription payments securely. It sends a request to create a subscription and opens the payment gateway popup for users to complete the payment.
4. **Subscription Validation**: After the payment is completed, the component verifies the payment and redirects the user to the appropriate page based on the payment status (success or failure).

**Example**

import React from 'react';

import Checkout from './Checkout';

function PricingPage({ user }) {

return (

<div className="pricing-page">

<div className="pricing-plan">

<h2>Basic Plan</h2>

<p>Includes basic features</p>

<Checkout user={user} />

</div>

{/\* Other pricing plans \*/}

</div>

);

}

export default PricingPage;

**Dependencies**

The Checkout component relies on the following external libraries:

* axios: For making HTTP requests to the backend server to create subscriptions and verify payments.
* react-toastify: For displaying toast notifications to provide feedback to users during the checkout process.

Ensure that axios and react-toastify are installed in your project before using the Checkout component.

**Important Notes**

* Update the URLs used for making HTTP requests to match your backend API endpoints for subscription creation and payment verification.
* Customize the payment options and configurations based on your payment gateway integration requirements.
* Ensure that the payment gateway configuration is properly set up and tested in your application environment before deploying it to production.
* Resource: watch the relevant videos in series **https://www.youtube.com/watch?v=buHUUupb-2Q&ab\_channel=6PackProgrammer**