**Pricing Component Documentation**

**Introduction**

The Pricing component is a React functional component designed to display different pricing plans for the application. It allows users to choose from various subscription options and provides information about each plan's features and pricing details.

**Usage**

To use the Pricing component, import it into your React application and place it at the desired location within your application layout.

import Pricing from './Pricing';

<Pricing/>

**Features**

1. **Plan Selection**: The Pricing component displays multiple pricing cards, each representing a different subscription plan. Users can choose the plan that best suits their needs.
2. **Plan Details**: Each pricing card includes details about the plan, such as the plan name, pricing, billing frequency, and a list of features included in the plan.
3. **Payment Integration**: The component integrates with the payment system to allow users to subscribe to their chosen plan seamlessly. It includes checkout components for handling the payment process.
4. **Dynamic Pricing**: Pricing details are fetched dynamically from the server, ensuring that users see the most up-to-date pricing information.

**Example**

import React from 'react';

import Pricing from './Pricing';

function App() {

return (

<div className="App">

{/\* Your main content \*/}

<Pricing />

</div>

);

}

export default App;

**Dependencies**

The Pricing component relies on the following external libraries:

* axios: For making HTTP requests to fetch pricing details and handle payment interactions.
* react-toastify: For displaying toast notifications to provide feedback to users during the payment process.

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

**Important Notes**

* Update the URLs used for fetching pricing details to match your backend API endpoints.
* Customize the pricing card details, including plan names, pricing, and features, to reflect your application's subscription options.
* Ensure that the payment integration is properly configured and tested in your application environment before deploying it to production.