Configure Stripe:

In your React component, configure Stripe by loading the Stripe object with your publishable key.

tsx

Copy code

// src/components/CheckoutForm.tsx

import { loadStripe } from '@stripe/stripe-js';

const stripePromise = loadStripe('your-publishable-key');

const CheckoutForm: React.FC = () => {

// Your checkout form logic here

};

export default CheckoutForm;

Handle Payments (Client-Side):

Implement the checkout form to collect payment details and handle payments on the client side.

tsx

Copy code

// src/components/CheckoutForm.tsx

import React from 'react';

import { useStripe, useElements } from '@stripe/stripe-js';

const CheckoutForm: React.FC = () => {

const stripe = useStripe();

const elements = useElements();

const handleSubmit = async (event: React.FormEvent) => {

event.preventDefault();

// Implement payment handling using stripe.confirmCardPayment

};

return (

<form onSubmit={handleSubmit}>

{/\* Your payment form fields, including CardElement \*/}

<button type="submit">Pay Now</button>

</form>

);

};

export default CheckoutForm;

<!-- Include Stripe.js script on your web page -->

<script src="https://js.stripe.com/v3/"></script>

<!-- Your payment form -->

<form id="payment-form">

<!-- A Stripe Element will be inserted here. -->

<div id="card-element"></div>

<!-- Used to display form errors. -->

<div id="card-errors" role="alert"></div>

<button type="submit">Submit Payment</button>

</form>

<script>

// Create a Stripe client

var stripe = Stripe('your-publishable-key');

// Create an instance of Elements

var elements = stripe.elements();

// Create an instance of the card Element.

var card = elements.create('card');

// Add an instance of the card Element into the `card-element` div.

card.mount('#card-element');

// Handle real-time validation errors from the card Element.

card.addEventListener('change', function(event) {

var displayError = document.getElementById('card-errors');

if (event.error) {

displayError.textContent = event.error.message;

} else {

displayError.textContent = '';

}

});

// Handle form submission.

var form = document.getElementById('payment-form');

form.addEventListener('submit', function(event) {

event.preventDefault();

// Use Stripe.js to create a token and handle the payment.

stripe.createToken(card).then(function(result) {

if (result.error) {

// Inform the user if there was an error.

var errorElement = document.getElementById('card-errors');

errorElement.textContent = result.error.message;

} else {

// Send the token to your server.

stripeTokenHandler(result.token);

}

});

});

// Submit the token to your server

function stripeTokenHandler(token) {

// Send the token to your server for further processing

console.log(token);

// Here, you would typically make an AJAX request to your server

// and include the token in the payload.

}

</script>