Amagon

**Material UI**

npm install @material-ui/icons

npm install @material-ui/core

**>props**

{Array(rating)

.fill(value,startindex,endindex)

.map((\_,i) =>

<p>⭐️</p>

)}

>**CurrencyFormat**

npm I react-currency-format

<https://www.npmjs.com/package/react-currency-format>

**>react context api/hook (***try and learn more***)**

import React,{createContext,useContext,useReducer} from 'react';

create

export const StateContext = createContext();

reducer(val,action)

export const StateProvider = ({reducer,initialState,children}) => (

<StateContext.Provider value={useReducer(reducer,initialState)}>

{children}

</StateContext.Provider>

)

use

export const useStateValue = () => useContext (StateContext);

export const initialState = {

basket: [],

};

const reducer = (state, action) => {

console.log(action);

switch (action.type) {

case 'ADD\_TO\_BASKET':

return {

...state,

basket: [...state.basket, action.item],

};

case 'EMPTY\_BASKET':

return{

...state,

basket:[]

}

default:

return state;

}

}

export default reducer;

>***either useHistory or link to button***

***>*** ***splice(index, howmany)***

***> <input type='text' value={email} onChange={e => setEmail(e.target.value)} />***

>**Firebase**

**Sudo npm Install -g firebase-tools**

**Npm I firebase**

import firebase from "firebase";

// For Firebase JS SDK v7.20.0 and later, measurementId is optional

const firebaseConfig = {

apiKey: "AIzaSyDtxVylA\_1aQorKrnu15RUk-FP3VwDupU8",

authDomain: "clone-9d4da.firebaseapp.com",

projectId: "clone-9d4da",

storageBucket: "clone-9d4da.appspot.com",

messagingSenderId: "1065534154113",

appId: "1:1065534154113:web:ca83a0c80b158b0266b480",

measurementId: "G-NN68NFRE3D"

};

const firebaseApp = firebase.initializeApp(firebaseConfig);

const db = firebaseApp.firestore();

const auth = firebase.auth();

export {db,auth};

const register = (e) => {

e.preventDefault();

auth

.createUserWithEmailAndPassword(email, password)

.then((auth) => {

// it successfully created a new user with email and password

if (auth) {

history.push("/");

}

})

.catch((error) => alert(error.message));

};

const signIn = (e) => {

e.preventDefault();

auth

.signInWithEmailAndPassword(email, password)

.then((auth) => {

// it successfully created a new user with email and password

history.push("/");

})

// not successfull

.catch((error) => alert(error.message));

};

>{!user && "/login"}

If there is no user then login page

>npm I @stripe/stripe-js

npm I @stripe/react-stripe-js

set up stirp accocunt

get publish key

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

import {Elements} from "@stripe/react-stripe-js";

app.js

const promise = loadStripe("pk\_test\_51IBywyEAjIG7X6ReTdObYhUvN28ikHSLbD0qvfFw2ST3pR47DjsEmXD69Yq3GveW4RKfzC88fvzlN0GaUSe3Li2Z00yDPEBngb");

<Route path="/payment">

<Header />

<Elements stripe={promise}>

<Payment/>

</Elements>

</Route>

Payment.js

import { CardElement, useStripe, useElements } from "@stripe/react-stripe-js";

import { getBasketTotal } from "./reducer";

import CurrencyFormat from "react-currency-format";

import axios from "./axios";

const stripe = useStripe();

const elements = useElements();

const [error, setError] = useState(null);

const [disabled, setDisabled] = useState(true);

const [disabled, setDisabled] = useState(true);

const [succeeded, setSucceeded] = useState(false);

const [processing, setProcessing] = useState("");

const history = useHistory();

const handleSubmit = async (event) => {

// do all other stuffs

event.preventDefault();

setProcessing(true);

const payload = await stripe

.confirmCardPayment(clientSecret, {

payment\_method: {

card: elements.getElement(CardElement),

},

})

.then(({ paymentIntent }) => {

db.collection("users")

.doc(user?.uid)

.collection("orders")

.doc(paymentIntent.id)

.set({

basket: basket,

amount: paymentIntent.amount,

created: paymentIntent.created,

});

// payment confirmation

setSucceeded(true);

setError(null);

setProcessing(false);

dispatch({

type: "EMPTY\_BASKET",

});

history.replace("/orders");

});

};

const handleChange = (event) => {

// listen to changes in the card element

// also display any error as the customer type thier card details

setDisabled(event.empty);

setError(event.error ? event.error.message : "");

};

<form onSubmit={handleSubmit}>

<CardElement onChange={handleChange} />

<div className="payment\_\_priceContainer">

<p>

<span>For Demo : </span>Type <span>Card Number: </span> 5454

5454 5454 5454, <span>MM/YY:</span> 05 / 45, <span>CVC:</span>{" "}

454

</p>

<CurrencyFormat

renderText={(value) => <h3>Order Total : {value}</h3>}

decimalScale={2}

value={getBasketTotal(basket)}

displayType={"text"}

thousandSeparator={true}

prefix={"₹"}

/>

<button disabled={processing || disabled || succeeded}>

<span>{processing ? <p>Processing</p> : "Buy Now"}</span>

</button>

</div>

{/\* error \*/}

{error && <div>{error}</div>}

</form>

>ClientSecrete in stripe

--create axios.js

Npm I axios

import axios from "axios";

const instance = axios.create({

baseURL:'http://localhost:5001/clone-9d4da/us-central1/api'

// the api cloud function url

});

export default instance;

payment.js

const [clientSecret, setClientSecret] = useState(true);

useEffect(() => {

// generate teh special stripe secrete which allow us to charge a customer

const getClientSecret = async () => {

const response = await axios({

method: "post",

// stripe expects the total in currency submits

url: `/payments/create?total=${getBasketTotal(basket) \* 100}`,

});

setClientSecret(response.data.clientSecret);

};

getClientSecret();

}, [basket]);

console.log("THE SECRET IS >>", clientSecret);

firebase Strip->

firebase init

select functions

javascript

y

y

---open function backend folder

Npm I express

Npm I cors

Npm I stripe

--open index.js in function

const functions = require('firebase-functions');

const express = require("express");

const cors = require("cors");

const stripe = require("stripe")('sk\_test\_51IBywyEAjIG7X6ReN7Xk9I0tznJCmgOtQ1cqV0X744gtCFaYhUYI4FBUPdi8PmJ7fRN9bAaQnBA89yDzkFd7KuHj00ucejTMdS')

// api

// app config

const app = express();

// middlewarts

app.use(cors({origin:true}));

app.use(express.json());

// api routes

app.get("/",(request,response) => response.status(200).send('hello chitvan'));

used in payment.js

app.post("/payments/create", async (request,response) =>{

const total = request.query.total;

console.log('Payment Request Recieved BOOM for this amount >>>',total);

const paymentIntent = await stripe.paymentIntents.create({

amount: total,

// subunits of the currency

currency:"INR",

});

response.status(201).send({

clientSecret:paymentIntent.client\_secret,

});

});

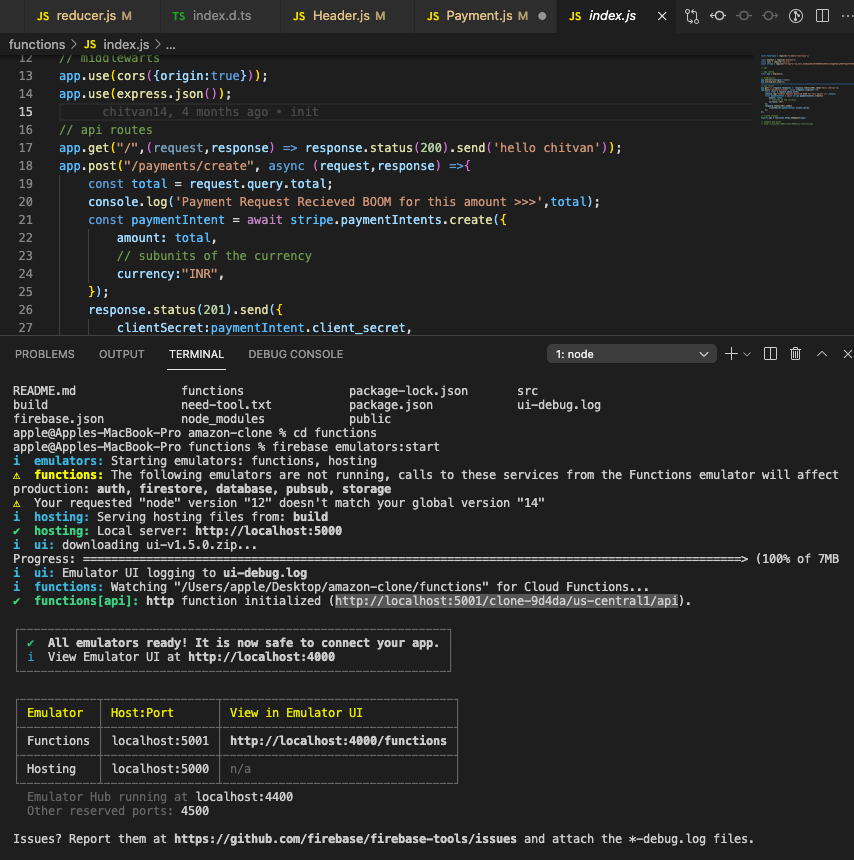
// listen command

exports.api = functions.https.onRequest(app);

// example end point

// http://localhost:5001/clone-9d4da/us-central1/api

To start the functions -> firebase emulators:start



>send data to firestore

import { db } from "./firebase";

db.collection("users")

.doc(user?.uid)

.collection("orders")

.doc(paymentIntent.id)

.set({

basket: basket,

amount: paymentIntent.amount,

created: paymentIntent.created,

});

> <p>{moment.unix(order.data.created).format("MMMM Do YYYY, h:mma")}</p>