React-Node-Shopping-Cart

* **npm init**
* **npm I –g nodemon**
* yarn add bcrypt@2.0.1 body-parser@1.18.3 cloudinary@1.11.0 concurrently@3.6.0 cookie-parser@1.4.3 dotenv@6.0.0 express@4.16.3 express-formidable@1.0.0 jsonwebtoken@8.3.0 moment@2.22.2 mongoose@5.1.6 [multer@1.3.0](mailto:multer@1.3.0)
* [bcrypt@2.0.1](mailto:bcrypt@2.0.1) =🡺 password,hash password
* [body-parser@1.18.3](mailto:body-parser@1.18.3) ===🡺 expresss,middleware
* [cloudinary@1.11.0](mailto:cloudinary@1.11.0) =🡺 store images
* [cookie-parser@1.4.3](mailto:cookie-parser@1.4.3) =🡺 server cookies
* [dotenv@6.0.0](mailto:dotenv@6.0.0) 🡺 create env variable
* [jsonwebtoken@8.3.0](mailto:jsonwebtoken@8.3.0) 🡺 create web tokens
* [multer@1.3.0](mailto:multer@1.3.0)==> upload images

Create Client and Server folders

Change In packge.json

"test": "echo \"Error: no test specified\" && exit 1",

"start":"node server/server.js",

"server":"nodemon server/server.js"

Yarn run server

[nodemon] restarting due to changes...

[nodemon] starting `node server/server.js`

Server Running 5000

config

module.exports = {

mongoURI:

"mongodb://shopping:Sandhani12345@ds115154.mlab.com:15154/shopping\_cart",

secretOrKey: "secret"

};

Server.js

const express = require('express');

const bodyParser = require('body-parser');

const cookieParser = require('cookie-parser');

const app = express();

const mongoose = require('mongoose');

// BOdy Parser middleware

app.use(bodyParser.urlencoded({ extended: false }));

app.use(bodyParser.json());

app.use(cookieParser());

// Mango config

const db = require("./../config/key").mongoURI;

// MangoDB connet

mongoose

.connect(

db

)

.then(() => console.log("mongodb conneneted "))

.catch(err => console.log(err));

app.get("/", (req, res) => res.send("hello Run Server")

const port = process.env.PORT || 5000;

app.listen(port,()=>{

console.log(`Server Running ${port}`);

})

Output

[nodemon] restarting due to changes...

[nodemon] starting `node server/server.js`

Server Running 5000

mongodb conneneted

MODELS

User.js

const mongoose =require('mongoose');

const Schema = mongoose.Schema;

const bcrypt = require('bcrypt');

const jwt = require("jsonwebtoken");

require('dotenv').config();

//Create Schema Form User

const UserSchema = new Schema({

email:{

type:String,

require:true,

trim:true,

unique:1

},

password:{

type:String,

require:true,

minlength:5

},

name:{

type:String,

require:true,

maxlength:100

},

lastname:{

type:String,

require:true,

maxlength:100

},

cart:{

type:Array,

default:[]

},

history:{

type:Array,

default:[]

},

role:{

type:Number,

default:0

},

token:{

type:String

}

});

//bcrypt the password

UserSchema.pre('save', function(next){

var user = this;

//NEW Feature

if(user.isModified('password')){

bcrypt.genSalt(10)

.then(salt=>bcrypt.hash(user.password,salt)

.then(hash=>{

user.password = hash;

next()

})

.catch(err=>next(err)))

.catch(err=>next(err));

//OLD Feature

// bcrypt.genSalt(10,(err,salt)=>{

// if(err) return next(err);

// bcrypt.hash(user.email,salt,(err,hash)=>{

// if(err) return next(err);

// user.password = hash;

// next();

// })

// })

}else{

next()

};

});

// Method to compare password for login

UserSchema.methods.comparePassword = function (candidatePassword, cb) {

bcrypt.compare(candidatePassword, this.password, (err, isMatch) => {

if (err) { return cb(err); }

cb(null, isMatch);

});

};

//JWT TOKEN

UserSchema.methods.generateToken = function(cb){

var user = this;

var token = jwt.sign(user.\_id.toHexString(),process.env.SECRET);

user.token = token;

user

.save()

.then(user=>cb(null,user))

.catch(err=>cb(err))

}

//Auth

UserSchema.statics.findByToken = function(token,cb){

var user = this;

jwt.verify(token,process.env.SECRET,(err,decode)=>{

user.findOne({'\_id':decode,'token':token})

.then(user=>cb(null,user))

.catch(err=>cb(err));

})

}

module.exports = User = mongoose.model('user',UserSchema);

Router/api/user.js

const express = require("express");

const router = express.Router();

const bcrypt = require('bcrypt')

//Model

const User = require('./../../models/User');

//Middleware

const auth = require('./../../middleware/auth');

// @route GET api/users/test

// @desc Test User router

// @access Public

router.get("/test", (req, res) => res.json({ msg: "Users Work" }));

// @route POST api/users/register

// @desc Test User router

// @access Public

router.post('/register',(req,res)=>{

User.findOne({email:req.body.email}).then(user=>{

if(user){

return res.status(400).json({email:user.email + "email alredy exits" });

}

});

const newUser = new User(req.body);

//bcrypt password

// bcrypt.genSalt(10,(err, salt)=>{

// bcrypt.hash(newUser.password,salt,(err,hash)=>{

// if(err) return err;

// newUser.password = hash;

// newUser

// .save()

// .then(user=>res.json({success:true,user}))

// .catch(err => console.log({success:false,err}));

// });

// })

//NEW style

newUser

.save()

.then(user=>res.status(200).json({success:true,userdata:user}))

.catch(err=>res.json({success:false,err}))

//OLD style

// newUser.save((err,doc)=>{

// if(err) return res.json({success:false,err})

// res.status(200).json({

// success:true,

// userdata:doc

// })

// })

})

// @route POST api/users/login

// @desc Test User router

// @access Public

router.post('/login',(req,res)=>{

// find email

User.findOne({'email':req.body.email},(err,user)=>{

if(!user) return res.json({loginSuccess:false,message:"auth fail,email not found"});

user.comparePassword(req.body.password,(err,isMatch)=>{

if(!isMatch) return res.status(400).json({login:'password not matched'})

user.generateToken((err,user)=>{

if(err) return res.status(400).send(err);

res.cookie('w\_auth',user.token).status(200).json({

loginSuccess:true,

userdata:user

})

})

})

})

})

// @route GET api/users/auth

// @desc Test User router

// @access Pravite

router.get(

'/auth',

auth,

(req,res) => {

res.status(200).json({

isAdmin:req.user.role === 0 ? false : true,

isAuth:true,

email:req.user.email,

name:req.user.name,

lastname:req.user.lastname,

role:req.user.role,

cart:req.user.cart,

history:req.user.history

})

});

// @route GET api/users/logout

// @desc Test User router

// @access Pravite

router.get(

'/logout',

auth,

(req,res) => {

User.findOneAndUpdate(

{\_id:req.user.\_id},

{token:''},

(err,doc)=>{

if(err) return err;

return res.status(200).json({

success:true

})

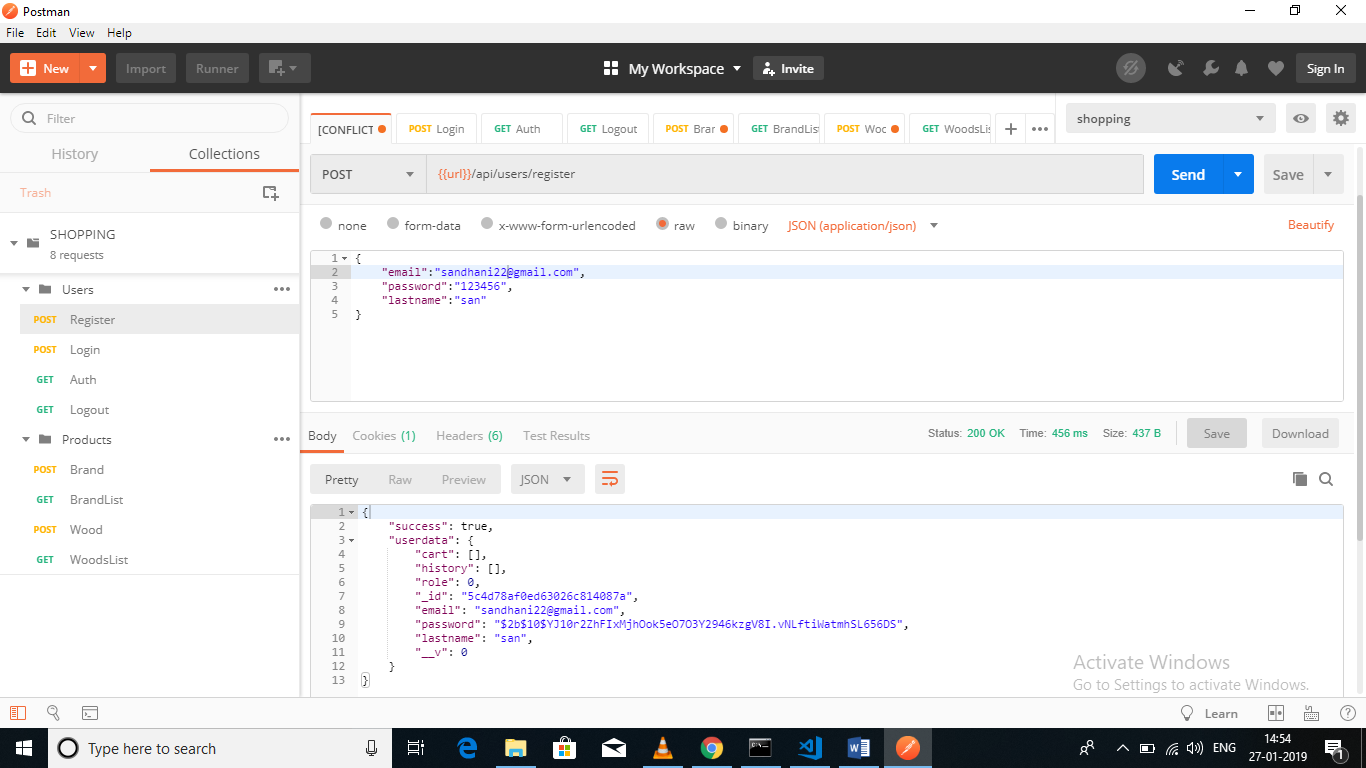
}

)

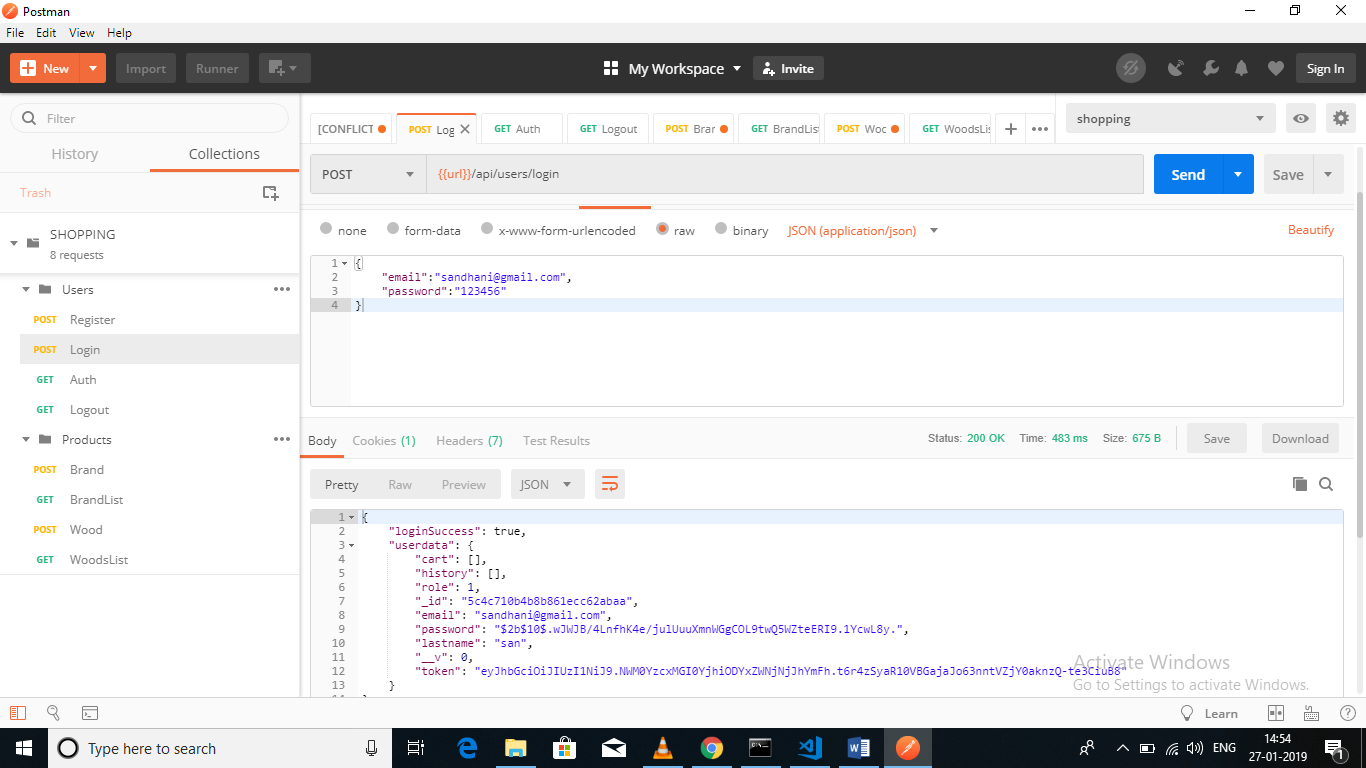
});

module.exports = router;

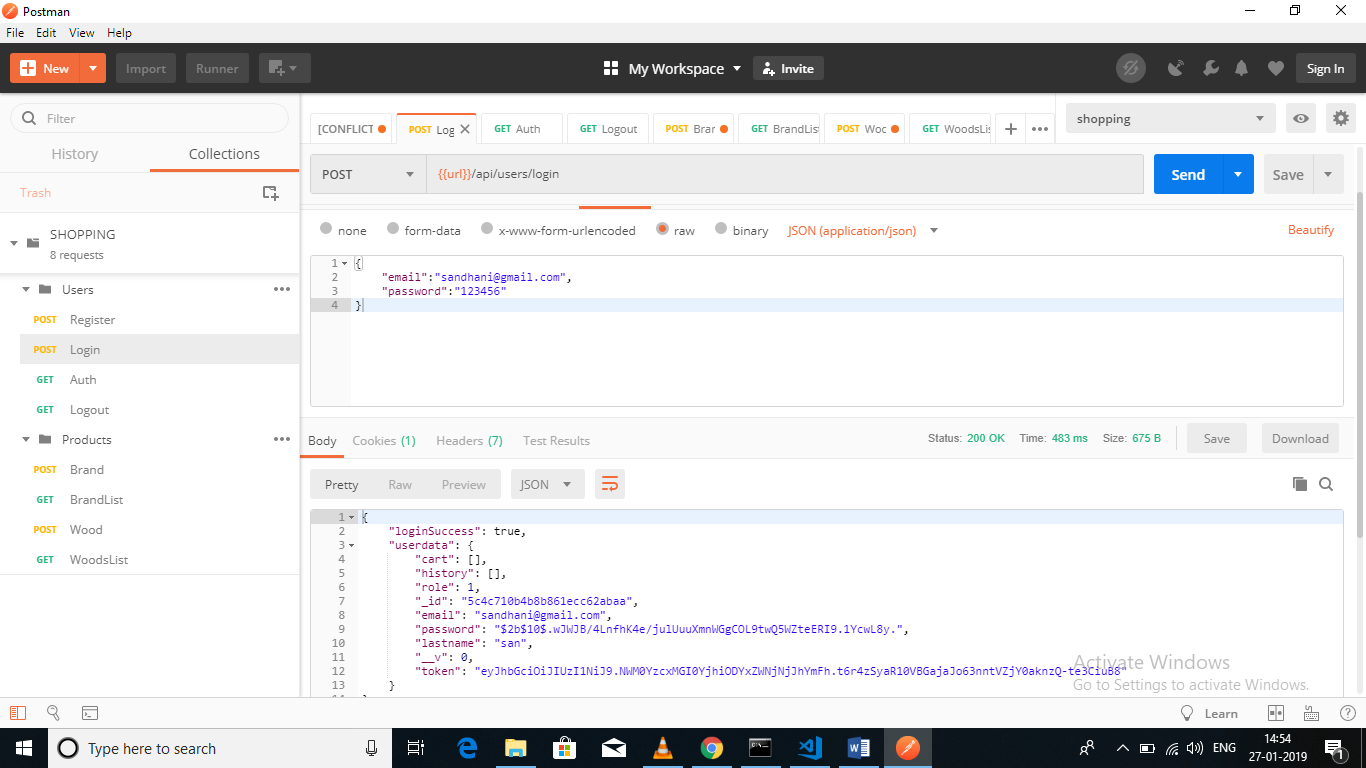
**USER API OUTPUT**



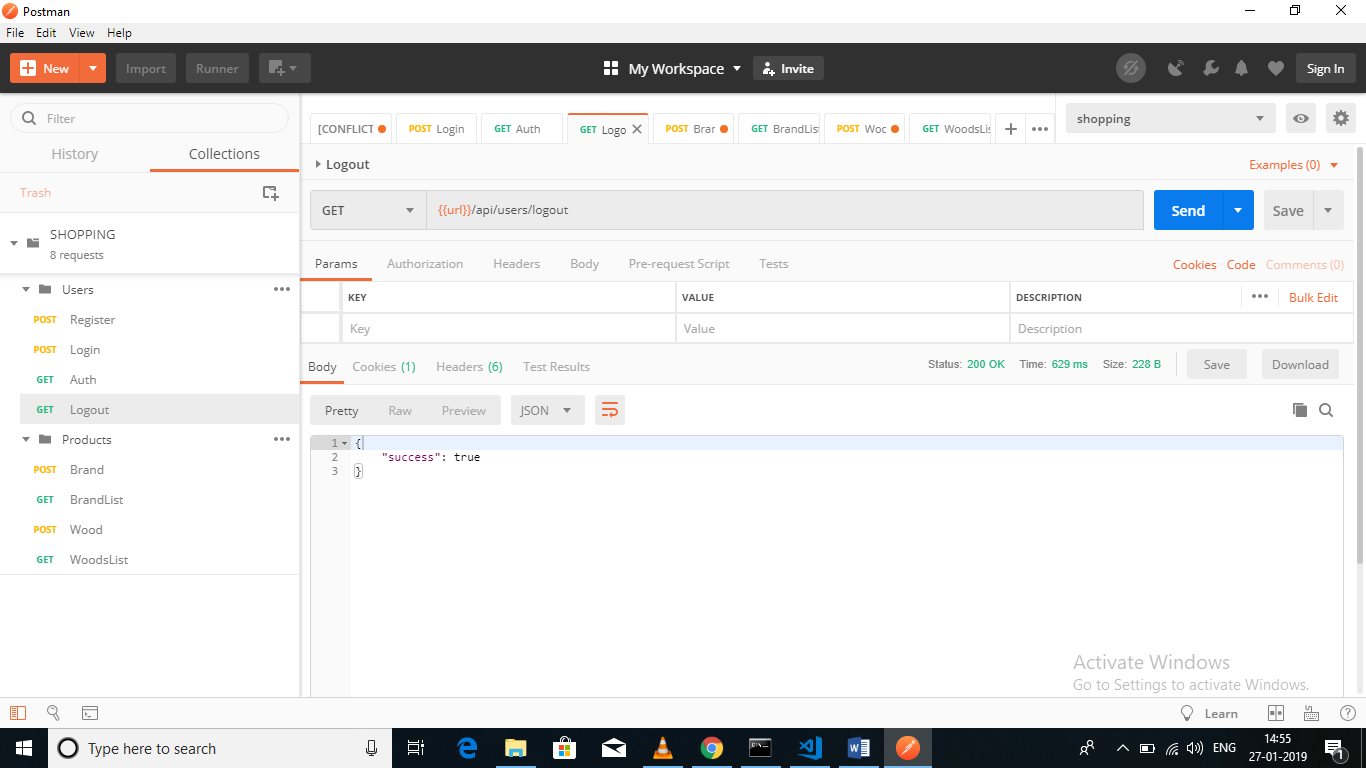
**Login**



**Auth**



**Logout**



Models/Brand.js

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const BrandSchema = new Schema({

name:{

type:String,

require:true,

unique:1,

maxlength:100

}

})

module.exports = Brand = mongoose.model('brands',BrandSchema);

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const WoodSchema = new Schema({

name:{

type:String,

require:true,

unique:1,

maxlength:100

}

})

module.exports = Wood = mongoose.model('woods',WoodSchema);

Products.js

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const ProductSchema = new Schema({

name:{

type:String,

require:true,

unique:1,

maxlength:100

},

description:{

type:String,

require:true,

maxlength:100000

},

price:{

type:Number,

require:true,

maxlength:25

},

brand:{

type:Schema.Types.ObjectId,

ref:'brands',

require:true

},

wood:{

type:Schema.Types.ObjectId,

ref:'woods',

require:true

},

shipping:{

require:true,

type:Boolean

},

available:{

type:Boolean,

require:true

},

frets:{

require:true,

type:Number

},

sold:{

type:Boolean,

require:true,

default:0

},

publish:{

require:true,

type:Boolean

},

images:{

type:Array,

default:[]

}

},{timestamps:true});

module.exports = Product = mongoose.model('Products',ProductSchema);

create admin for auth validation

middleware/admin.js

let admin = (req,res,next)=>{

if(req.user.role === 0 ){

return res.json({message:"your not allowd get out of here"});

}

next();

}

module.exports = admin;

Authntication

Middleware/auth

const User = require('../models/User');

const auth = (req,res,next)=> {

let token =req.cookies.w\_auth;

User.findByToken(token,(err,user)=>{

if(err) return err;

if(!user) return res.json({

isAuth:false,

error:true

});

req.token = token;

req.user = user;

next()

})

}

module.exports = auth;

Router/product.js

const express = require("express");

const router = express.Router();

//Model

const Brand = require('./../../models/Brand');

const Wood = require('./../../models/Wood');

const Product = require('./../../models/Product');

//Middleware

const auth = require('./../../middleware/auth');

const admin = require('./../../middleware/admin');

// @route POST api/products/brand

// @desc Test Brand router

// @access Pravite

router.post('/brand',auth,admin,(req,res)=>{

const brand = new Brand(req.body);

brand.save()

.then(brand => res.status(200).json({

success:true,brand:brand

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route get api/products/brand

// @desc Test Brand router

// @access Public

router.get('/brand',(req,res)=>{

Brand.find()

.then(brand=>res.json({

brand:brand

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route POST api/products/wood

// @desc Test Brand router

// @access Pravite

router.post('/wood',auth,admin,(req,res)=>{

const wood = new Wood(req.body);

wood.save()

.then(brand => res.status(200).json({

success:true,wood:wood

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route get api/products/wood

// @desc Test Brand router

// @access Public

router.get('/wood',(req,res)=>{

Wood.find()

.then(wood=>res.json({

wood:wood

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route get api/products/wood

// @desc Test Brand router

// @access Public

router.get('/wood',(req,res)=>{

Wood.find()

.then(wood=>res.json({

wood:wood

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route POST api/products/product

// @desc Test Brand router

// @access Pravite

router.post('/product',auth,admin,(req,res)=>{

const product = new Product(req.body);

product.save()

.then(product => res.status(200).json({

success:true,product:product

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route get api/products/product

// @desc Test Brand router

// @access Public

router.get('/product',(req,res)=>{

Brand.find()

.then(product=>res.json({

product:product

}))

.catch(err=> res.json({

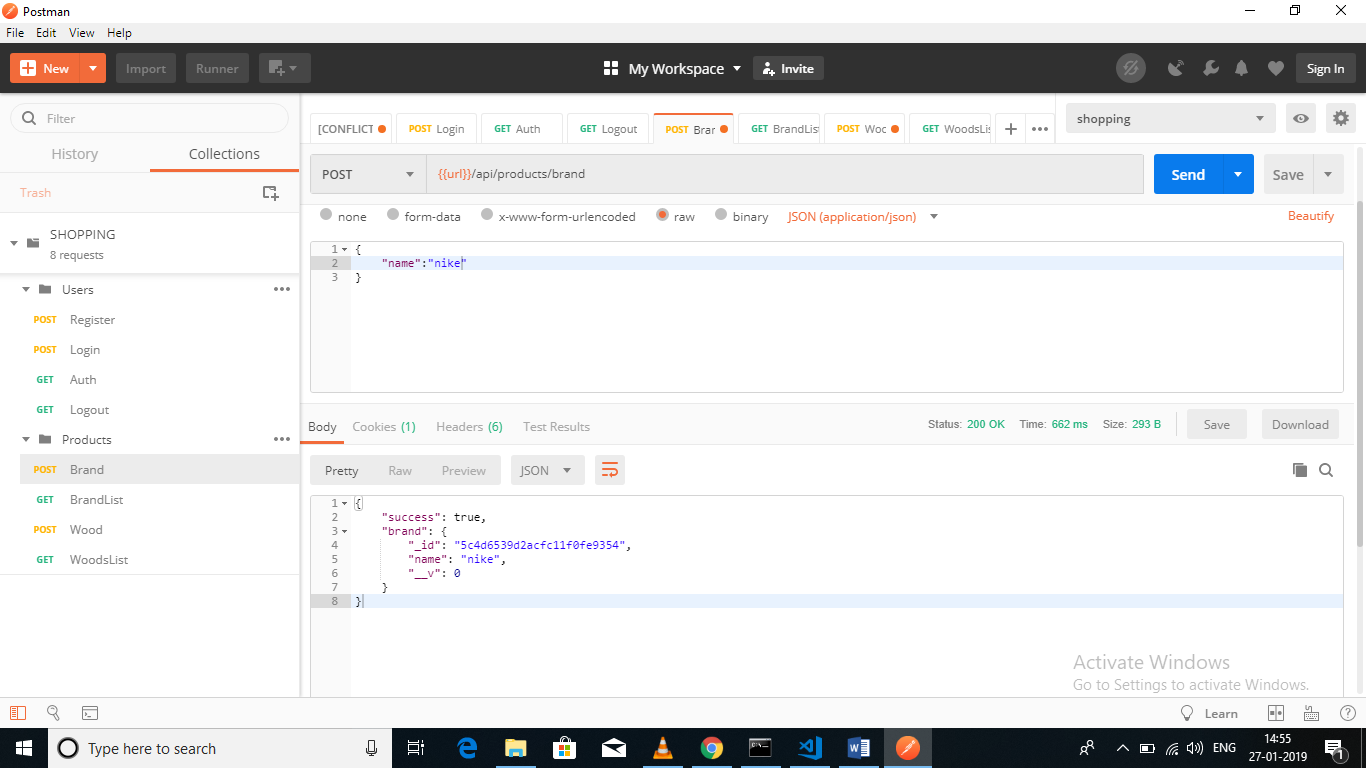
success:false,

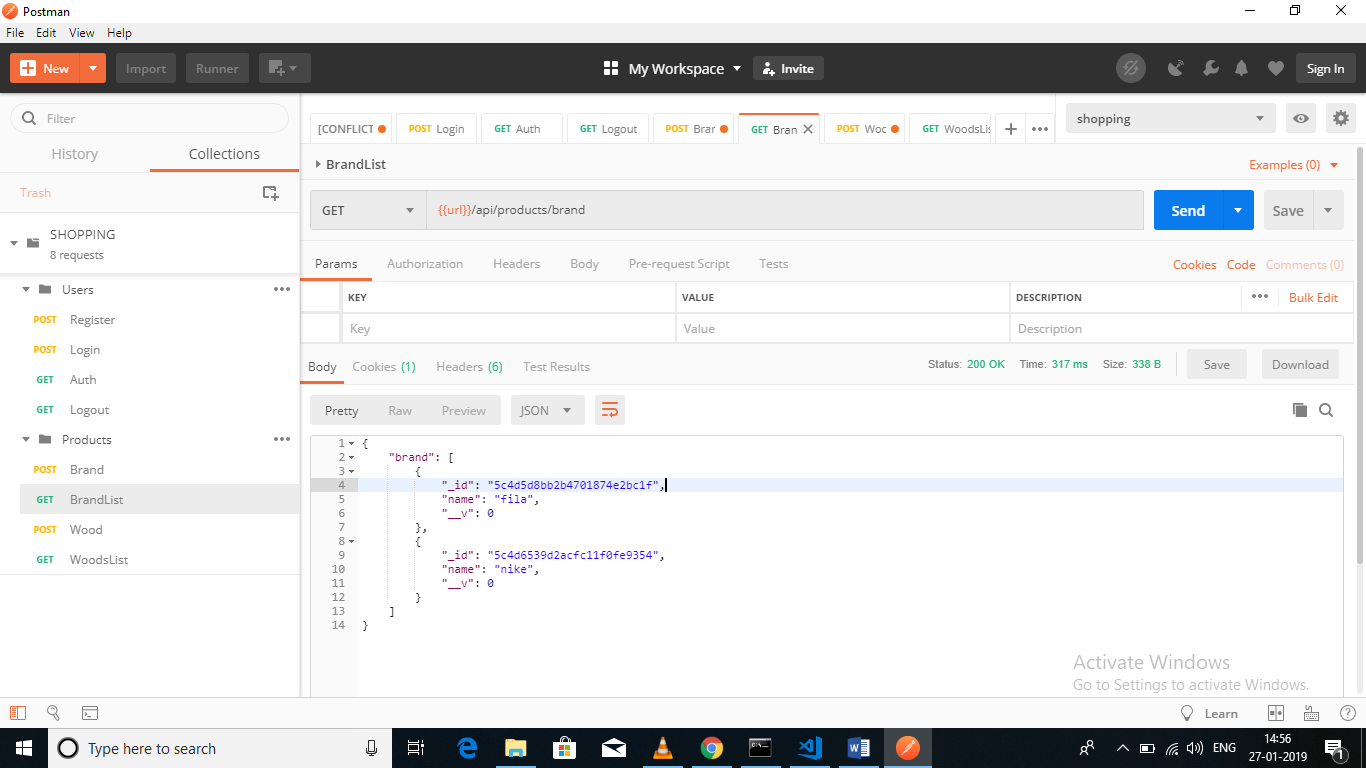
err:err

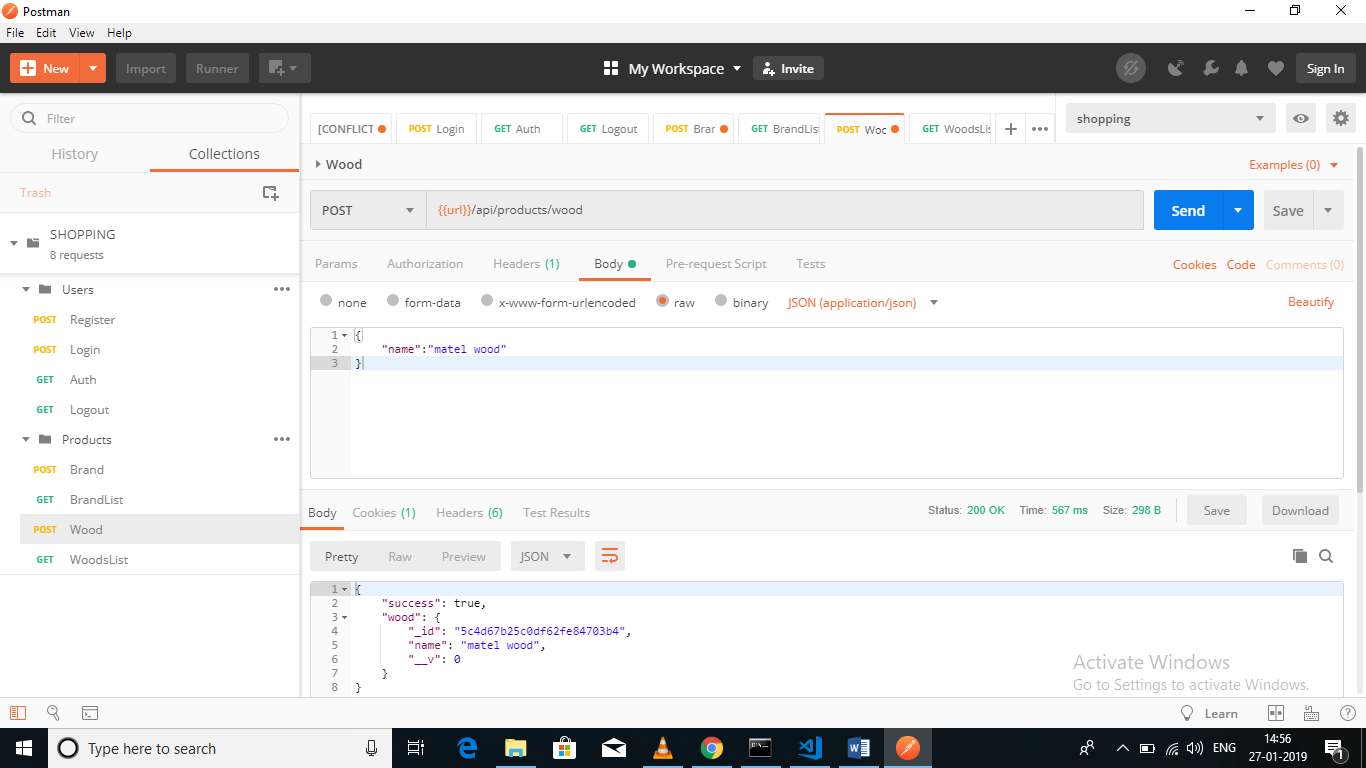
}));

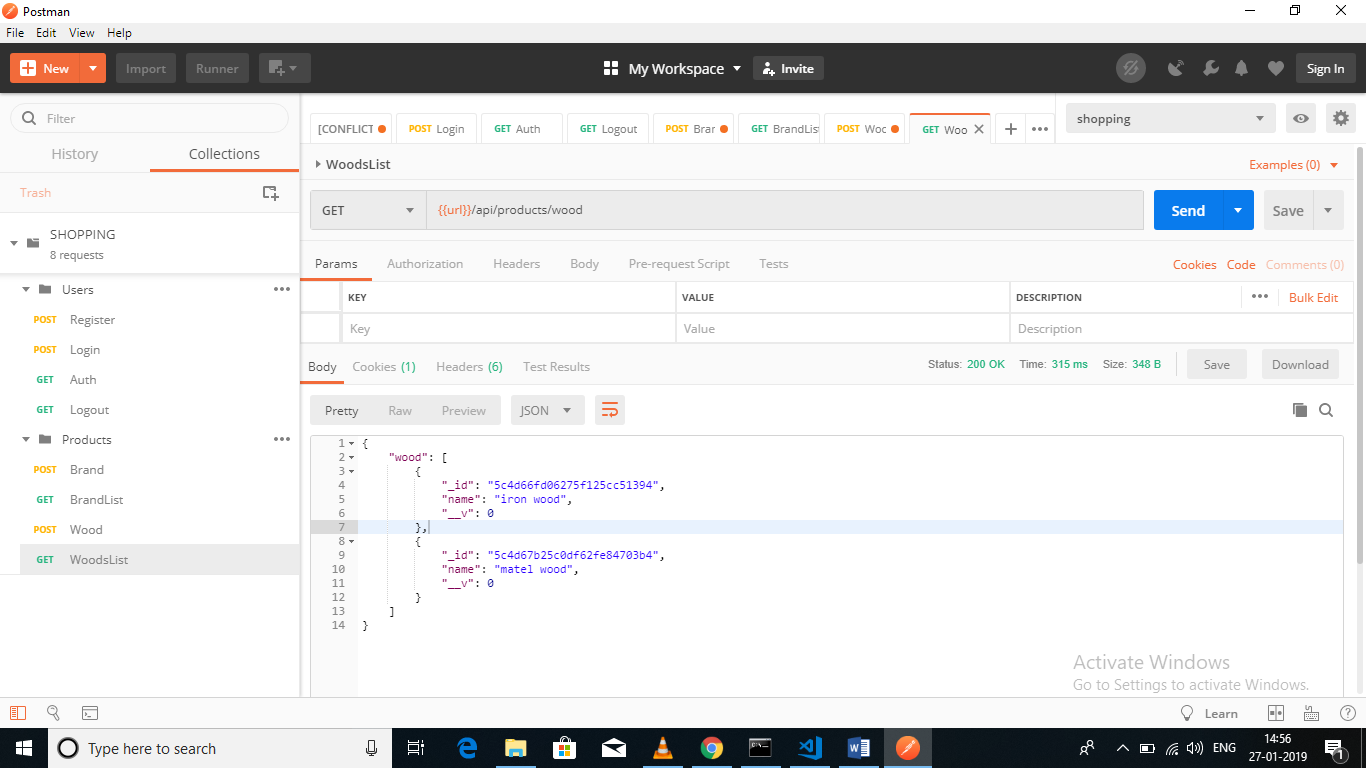
})

module.exports = router;









server.js

const express = require('express');

const bodyParser = require('body-parser');

const cookieParser = require('cookie-parser');

const app = express();

const mongoose = require('mongoose');

// BOdy Parser middleware

app.use(bodyParser.urlencoded({ extended: false }));

app.use(bodyParser.json());

app.use(cookieParser());

// Mango config

const db = require("./../config/key").mongoURI;

// MangoDB connet

mongoose

.connect(

db

)

.then(() => console.log("mongodb conneneted "))

.catch(err => console.log(err));

//app.get("/", (req, res) => res.send("hello Run Server"));

// Routes Refer

const users = require("./routes/api/users");

const products = require("./routes/api/products");

// const profile = require("./routes/api/profile");

// User Routes

app.use("/api/users", users);

app.use("/api/products", products);

// app.use("/api/posts", posts);

const port = process.env.PORT || 5000;

app.listen(port,()=>{

console.log(`Server Running ${port}`);

})

Product.js

const express = require("express");

const router = express.Router();

const mongoose =require('mongoose');

//Model

const Brand = require('./../../models/Brand');

const Wood = require('./../../models/Wood');

const Product = require('./../../models/Product');

//Middleware

const auth = require('./../../middleware/auth');

const admin = require('./../../middleware/admin');

// @route POST api/products/brand

// @desc Test Brand router

// @access Pravite

router.post('/brand',auth,admin,(req,res)=>{

const brand = new Brand(req.body);

brand.save()

.then(brand => res.status(200).json({

success:true,brand:brand

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route get api/products/brand

// @desc Test Brand router

// @access Public

router.get('/brand',(req,res)=>{

Brand.find()

.then(brand=>res.json({

brand:brand

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route POST api/products/wood

// @desc Test Brand router

// @access Pravite

router.post('/wood',auth,admin,(req,res)=>{

const wood = new Wood(req.body);

wood.save()

.then(brand => res.status(200).json({

success:true,wood:wood

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route get api/products/wood

// @desc Test Brand router

// @access Public

router.get('/wood',(req,res)=>{

Wood.find()

.then(wood=>res.json({

wood:wood

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route POST api/products/product

// @desc Test Brand router

// @access Pravite

router.post('/article',auth,admin,(req,res)=>{

const product = new Product(req.body);

product.save()

.then(product => res.status(200).json({

success:true,

product:product

}))

.catch(err=> res.json({

success:false,

err:err

}));

})

// @route get api/products/article

// @desc Test Brand router

// @access Public

router.get('/article',(req,res)=>{

Product.find()

.then(product=>res.json({

product:product

}))

.catch(err=> res.json({

success:false,

err:err

}));

});

// @route get {{url}}/api/products/article\_by\_id?id=5c4d7d4d73f7181f1052a3a9,5c4f418970ba4d2bcc3698c3,5c4f419170ba4d2bcc3698c4&type=array

// @desc Test Brand router

// @access Public

//using query string

router.get('/article\_by\_id',(req,res)=>{

let type = req.query.type;

let items = req.query.id;

if(type==='array'){

let ids = req.query.id.split(',');

// console.log(ids)

// items = [];

items = ids.map(item=>{

return mongoose.Types.ObjectId(item);

})

Product.find({'\_id':{$in:items}})

.populate('brand')

.populate('wood')

.exec((err,doc)=>{

res.status(200).send(doc)

})

}

})

// @route get {{url}}/api/products/article?sortBy=sold&order=desc&limit=100&skip=5

// @desc Test Brand router

// @access Public

// Using Query String

router.get('/articles',(req,res)=>{

let sortBy = req.query.sortBy ? req.query.sortBy : '\_id';

let order = req.query.order ? req.query.order : 'asc';

let limit = req.query.limit ? parseInt(req.query.limit) : 100;

Product

.find({})

.populate('wood')

.populate('brand')

.sort([[sortBy,order]])

.limit(limit)

.exec((err,doc)=>{

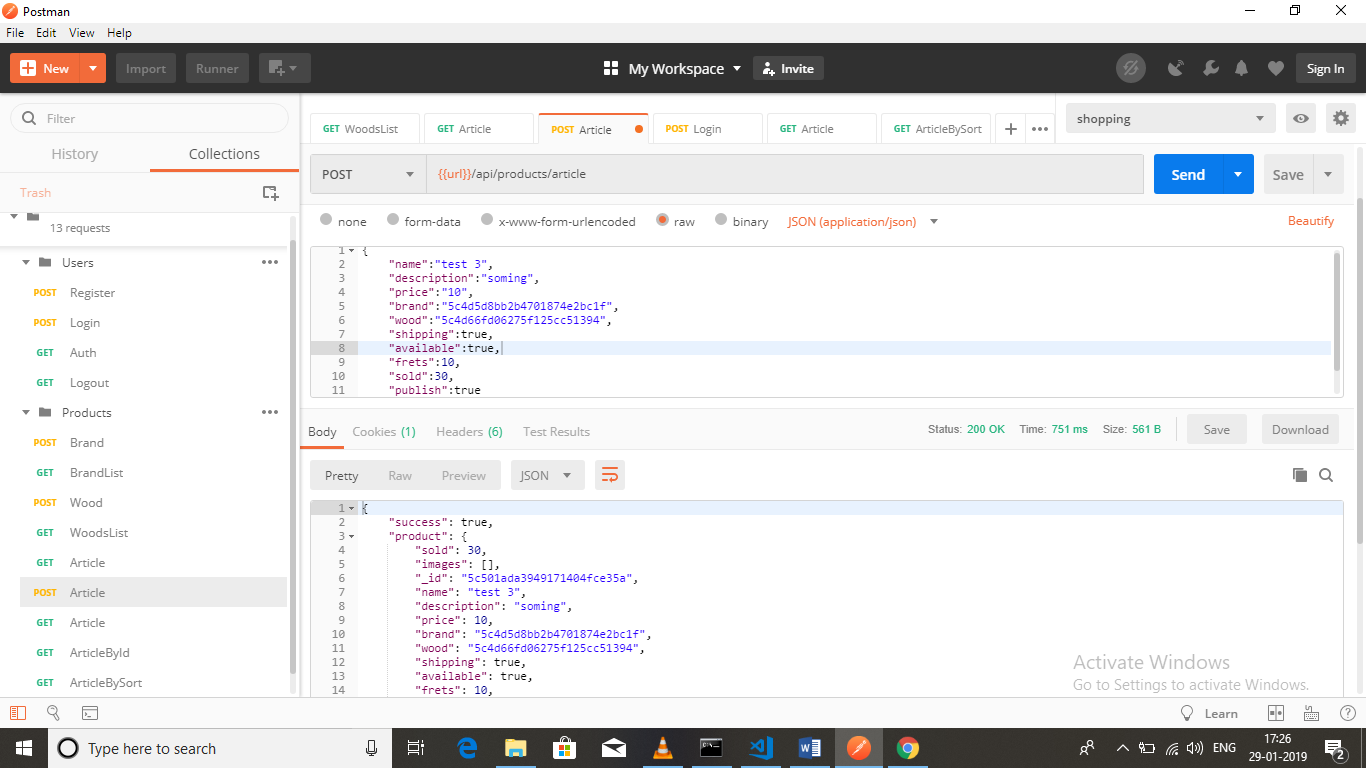
if(err) return res.status(400).send(err);

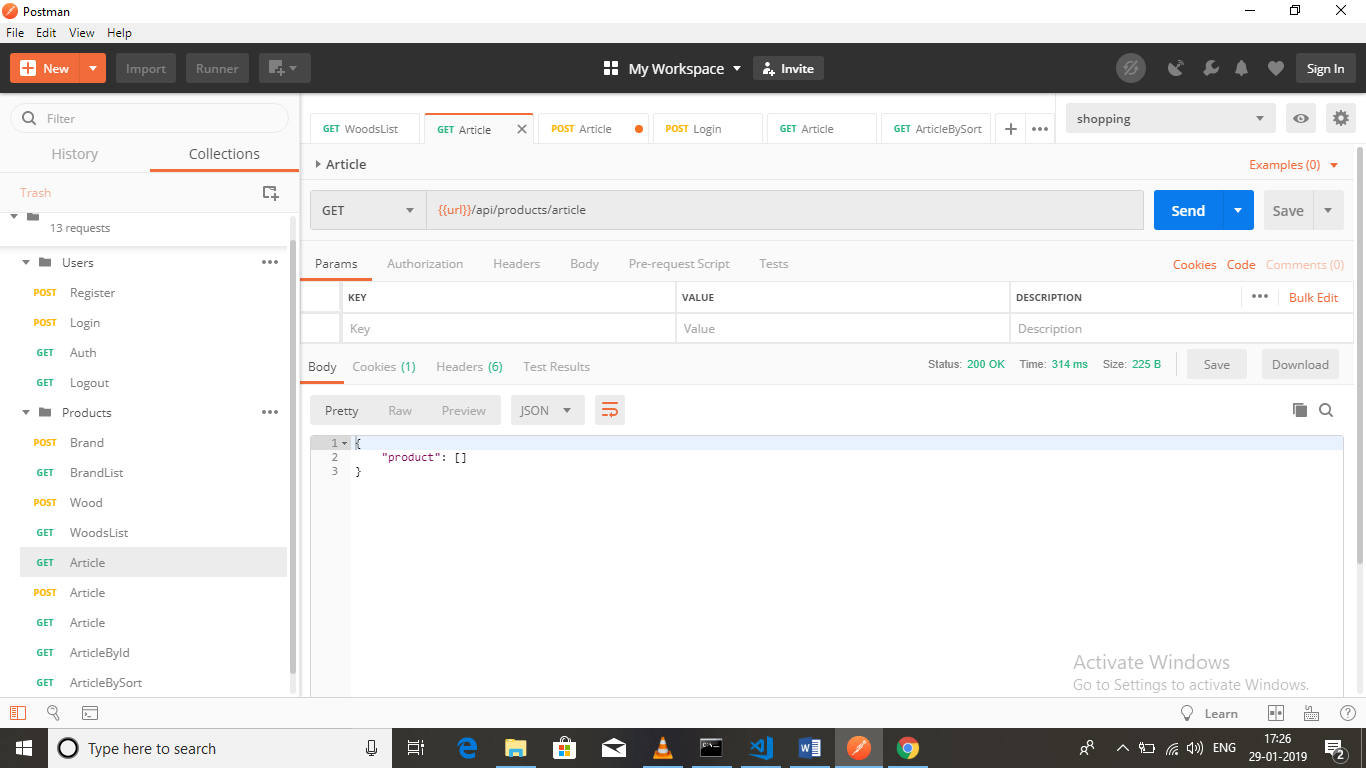
res.status(200).send(doc);

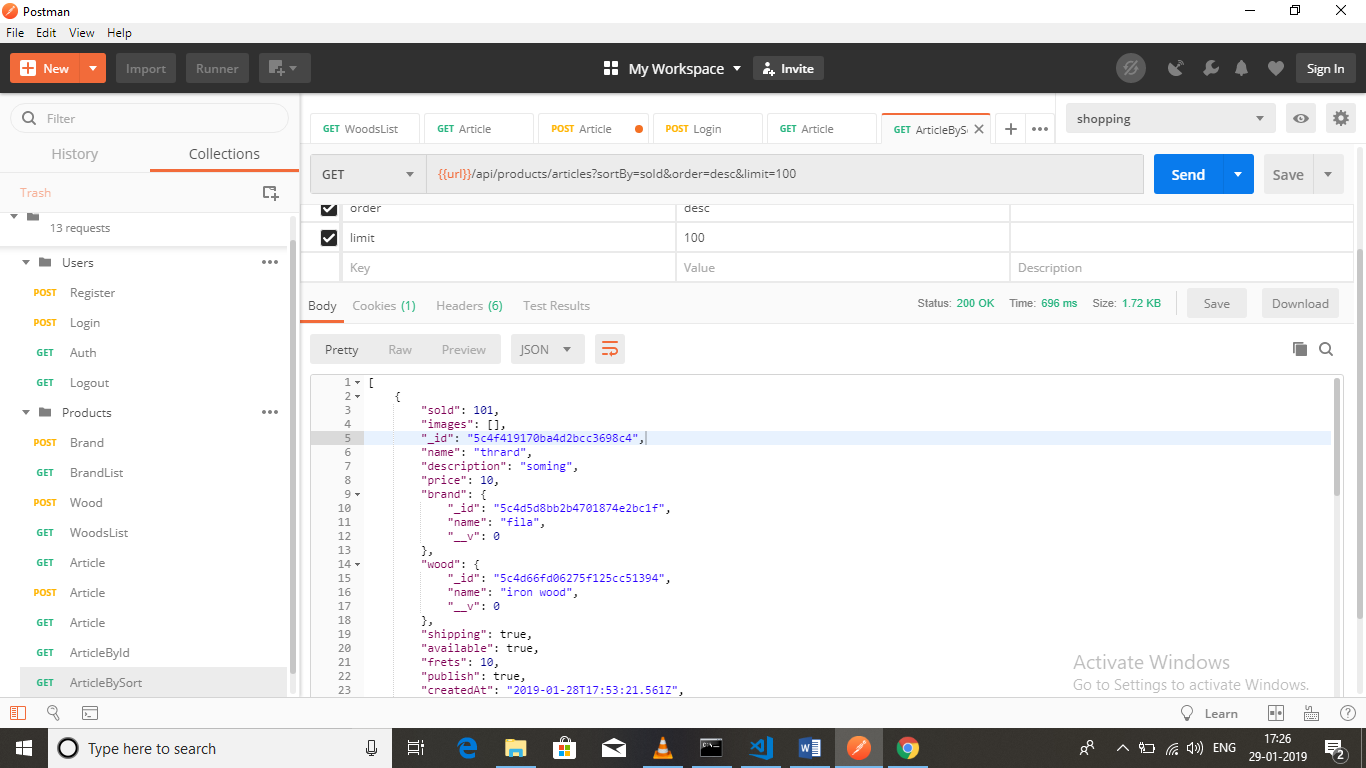
})

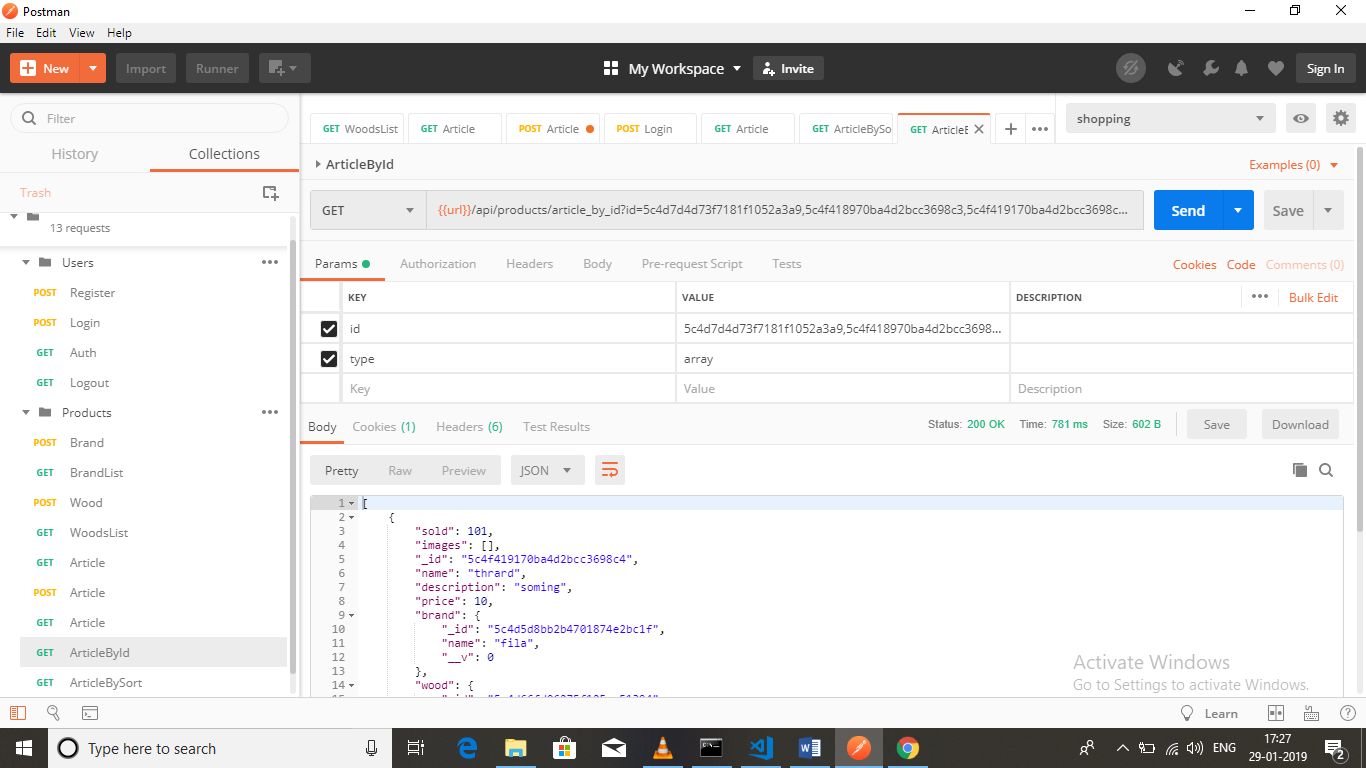
})

module.exports = router;









Client Side

Npx create-react-app client

Cd client

Client=🡺

PS D:\Practice\2019\react-node-shopping-cart\client> yarn add @fortawesome/fontawesome@1.1.8 @fortawesome/fontawesome-free-solid@5.0.13 @fortawesome/react-fontawesome@0.0.20 @material-ui/core@1.2.2 axios@0.18.0 react-images@0.5.17 react-redux@5.0.7 react-router-dom@4.3.1 react-slick@0.23.1 redux@4.0.0 redux-promise@0.6.0 redux-thunk@2.3.0 react-dropzone@4.2.12 react-moment@0.7.7 [react-paypal-express-checkout@1.0.4](mailto:react-paypal-express-checkout@1.0.4)

Run server and clinet at a time ‘using cuncurently

In main main package.json

Node package.json

"scripts": {

"client-install": "npm install --prefix client",

"start": "node server/server.js",

"server": "nodemon server/server.js",

"client": "npm start --prefix client",

"dev": "concurrently \"npm run server\" \"npm run client\""

},

Index.js

import React from 'react';

import ReactDOM from 'react-dom';

import {BrowserRouter, Router} from 'react-router-dom'

import Routes from './routes';

//css

import './Resources/css/styles.css';

ReactDOM.render(

<BrowserRouter>

<Routes />

</BrowserRouter>

,document.getElementById('root'));

Routes.js

import React from 'react'

import { Switch,Route } from 'react-router-dom';

//components

import Home from './components/Home';

import Layout from './hoc/Layout';

import Products from './components/Products'

const Routes = () => {

return (

<div>

<Layout>

<Switch>

<Route path='/' exact component={Home}/>

<Route path='/product' component={Products} />

</Switch>

</Layout>

</div>

)

}

export default Routes;

hoc/layout.js

import React, { Component } from 'react'

// components

import Header from './../components/Header';

import Footer from './../components/Footer';

class Layout extends Component {

render() {

return (

<div>

<Header />

<div className="page\_container">

{this.props.children}

</div>

<Footer />

</div>

)

}

}

export default Layout;

component/header.js

import React from 'react'

const Header= () => {

return (

<header className="bck\_b\_light">

<div className="container">

<div className="left">

<div className="logo">

waves

</div>

</div>

<div className="right">

<div className="top">

Link

</div>

<div className="bottom">

Link

</div>

</div>

</div>

</header>

)

}

export default Header;

Footer.js

import React from 'react'

import FontAwesomeIcon from '@fortawesome/react-fontawesome';

import faCompass from '@fortawesome/fontawesome-free-solid/faCompass';

import faPhone from '@fortawesome/fontawesome-free-solid/faPhone';

import faClock from '@fortawesome/fontawesome-free-solid/faClock';

import faEnvelope from '@fortawesome/fontawesome-free-solid/faEnvelope'

const Footer= () => {

return (

<footer className="bck\_b\_dark">

<div className="container">

<div className="logo">

waves

</div>

<div className="flex">

<div className="left">

<h2>Contant inforamtion</h2>

<div className="business\_nfo">

<div className="tag">

<FontAwesomeIcon

icon={faCompass}

className="icon"

/>

<div className="nfo">

<div>Address</div>

<div>Bangalore Kundanahalli gate</div>

</div>

</div>

<div className="tag">

<FontAwesomeIcon

icon={faPhone}

className="icon"

/>

<div className="nfo">

<div>Phone</div>

<div>+91-9030701091</div>

</div>

</div>

<div className="tag">

<FontAwesomeIcon

icon={faClock}

className="icon"

/>

<div className="nfo">

<div>Working hr</div>

<div> 24/7</div>

</div>

</div>

<div className="tag">

<FontAwesomeIcon

icon={faEnvelope}

className="icon"

/>

<div className="nfo">

<div>Email Address</div>

<div>sandhanireactdeveloper@gmail.com</div>

</div>

</div>

</div>

</div>

<div className="left">

<h2>be the first to know </h2>

<div>

<div> sets up your development environment so that you can use the latest JavaScript features, provides a nice developer experience, and optimizes your app for production. You’ll need to have Node >= 6 and npm >= 5.2 on your machine. To create a project, run:</div>

</div>

</div>

</div>

</div>

</footer>

)

}

export default Footer;

Home.js

import React from 'react'

const Home = () => {

return (

<div>

home

</div>

)

}

export default Home;

Resource/style.js

body {

margin:0;

padding:0;

font-family: 'Oswald', sans-serif;

color: #404040;

}

.container {

width:1100px;

margin: 0 auto;

}

.flex{

display: flex

}

.page\_container {

min-height: 75vh;

padding-top: 85px;

}

.main\_loader {

text-align: center;

margin-top: 100px;

}

.page\_wrapper {

padding-top: 85px;

}

a {

text-decoration: none;

color:inherit;

}

.bck\_b\_light {

background: #999592;

}

.bck\_b\_dark {

background: #272723;

}

.dialog\_alert {

padding: 51px;

font-size: 30px;

}

.load\_more\_container {

clear: both;

margin-bottom: 50px;

border-top: 5px solid #f5f5f5;

padding-top: 20px;

}

.load\_more\_container span {

cursor: pointer;

background: #2196F3;

color: #ffffff;

font-size: 27px;

padding: 10px 60px;

display: table;

margin: 0 auto;

}

.no\_result {

font-size: 40px;

color: #999592;

text-align: center;

padding-top: 40px;

}

.not\_found\_container {

font-size: 70px;

text-align: center;

padding-top: 70px;

color: #dddddd;

}

.clear:after {

content: '';

display: block;

clear: both;

}

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #dddddd;

}

/\*============================

FORM

==============================\*/

input {

font-size: 15px;

-webkit-box-sizing: border-box;

box-sizing: border-box;

padding: 10px 10px;

width: 100%;

margin: 10px 0px 0px 0px;

}

button {

cursor: pointer;

text-transform: uppercase;

padding: 10px;

background: #e1ddc3;

margin-top: 10px;

font-family: 'Oswald', sans-serif;

font-size: 16px;

}

.error\_label {

color: #F44336;

display:block;

font-size: 15px;

}

textarea {

width: 100%;

height: 51px;

font-size: 15px;

padding: 10px;

box-sizing: border-box;

margin-top: 5px;

border: 1px solid #d1d1d1;

}

.form\_devider {

border-bottom: 3px solid #999592;

margin: 40px 0px;

}

.label\_inputs {

display: inline-block;

background: #999592;

color: #fff;

padding: 5px 5px;

}

select {

font-size: 16px;

height: 38px;

border: 1px solid #e9e9e9;

border-radius: 3px;

width: 100%;

margin: 10px 0px 0px 0px;

}

.formBlock {

margin-bottom: 10px;

}

.form\_block\_two {

display:flex;

}

.form\_block\_two .block{

flex-grow: 1;

box-sizing: border-box;

}

.form\_block\_two .block:nth-child(1) {

padding-right:10px;

}

.form\_block\_two .block:nth-child(2) {

padding-left:10px;

}

.form\_success {

background: #4CAF50;

color: #ffffffff;

padding: 10px;

display: block;

}

/\*============================

BUTTONS

==============================\*/

.link\_default {

text-transform: uppercase;

padding: 10px;

display: table;

background: #e1ddc3;

margin-top: 10px;

}

.card\_link,

.bag\_link,

.add\_to\_cart\_link {

width: 100%;

border: 1px solid #dcdcdc;

padding: 5px 10px;

display: block;

box-sizing: border-box;

text-align: center;

font-size: 14px;

text-transform: uppercase;

color: #c3c0b1;

background: #f8f8f8;

transition: all 200ms ease-in;

cursor: pointer;

}

.card\_link:hover,

.bag\_link:hover,

.add\_to\_cart\_link:hover {

background: #aaaaaa;

color: #ffff;

}

.add\_to\_cart\_link {

color: #6b6b6b;

background: #d9d9d9;

}

.add\_to\_cart\_link svg {

margin-right: 10px;

}

.action\_button {

text-transform: uppercase;

padding: 10px;

display: table;

background: #e1ddc3;

margin-top: 10px;

}

/\*============================

PAGE TOP

==============================\*/

.page\_top {

border-bottom: 1px solid #999592;

padding: 20px 0px;

text-transform: uppercase;

}

/\*============================

HEADER

==============================\*/

header {

position: fixed;

z-index: 999;

width: 100%;

border-bottom: 1px solid white;

}

header .logo {

font-family: 'Monoton', cursive;

color: #ffffff;

font-size: 40px;

padding: 11px 0px;

}

header a {

margin-left: 20px;

}

header .container {

display: flex;

}

header .container .left{

width: 250px;

}

header .container .right{

flex-grow: 1;

border-left: 1px solid #eeeeee;

border-right: 1px solid #eeeeee;

}

header .container .right .top ,

header .container .right .bottom {

text-transform: uppercase;

font-weight: 700;

text-align: right;

color:#ffffff;

font-size: 15px;

padding: 10px 20px;

}

header .container .right .top {

color:#c9c9c9;

border-bottom: 1px solid #ffffff;

}

header .cart\_link {

display: inline-block;

position: relative;

}

header .cart\_link span {

position: absolute;

background: red;

top: 2px;

color: #ffff;

left: -8px;

border-radius: 50%;

padding: 0px 7px 2px 7px;

font-size: 12px;

}

header .log\_out\_link {

cursor: pointer;

display: inline-block;

margin-left: 20px;

}

/\*============================

FOOTER

==============================\*/

footer .logo {

text-align: center;

font-family: 'Monoton', cursive;

color: #ffffff;

font-size: 40px;

padding: 11px 0px;

border-bottom: 1px solid #4d4d4d;

}

footer .wrapper {

display: flex;

color: #828282;

}

footer .left,

footer .right {

flex: 1;

}

footer .business\_nfo {

display: flex;

flex-wrap: wrap;

}

footer .business\_nfo .tag {

flex-grow: 1;

position: relative;

width: 50%;

margin-bottom: 30px;

}

footer .business\_nfo .icon {

position: absolute;

font-size: 26px;

top: 10px;

}

footer .business\_nfo .nfo {

padding-left: 40px;

}

/\*============================

FEATURED IMAGE

==============================\*/

.featured\_container .featured\_image {

position:relative;

background-size: cover !important;

background-position: center left !important;

}

.featured\_container .featured\_action {

position: absolute;

top: 45%;

left: 8%;

}

.featured\_container .featured\_action .tag {

background: #82828287;

color: #ffffff;

padding: 0px 20px;

text-transform: uppercase;

display: table;

}

.featured\_container .featured\_action .tag.title {

font-size: 70px;

font-weight: 700;

}

.featured\_container .featured\_action .tag.low\_title {

font-size: 50px;

font-weight: 300;

margin-top: 10px;

}

/\*============================

FEATURED PROMOTION

==============================\*/

.home\_promotion .home\_promotion\_img {

height: 400px;

background-size: cover !important;

background-position: top center !important;

padding: 73px;

box-sizing: border-box;

}

.home\_promotion .tag {

background: #82828287;

color: #ffffff;

padding: 0px 20px;

text-transform: uppercase;

display: table;

}

.home\_promotion .tag.title {

font-size: 70px;

font-weight: 700;

margin: 10px auto;

}

.home\_promotion .tag.low\_title {

font-size: 50px;

font-weight: 300;

margin-top: 10px;

margin: 0 auto;

}

.home\_promotion .my\_link a {

margin: 10px auto;

}

/\*============================

CARD BLOCK

==============================\*/

.card\_block {

margin: 110px 0px;

}

.card\_block .title {

text-transform: uppercase;

text-align: center;

font-size: 25px;

margin-bottom: 50px;

}

.card\_block\_shop .card\_item\_wrapper {

float: left;

width: 33.3%;

margin-bottom: 20px;

}

/\*============================

CARD ITEM

==============================\*/

.card\_item\_wrapper {

flex-grow: 1;

width: 25%;

padding: 0 20px;

box-sizing: border-box;

position: relative;

overflow: hidden;

}

.card\_item\_wrapper .image {

width: 100%;

height: 280px;

background-position: center !important;

background-size: cover !important;

border: 1px solid #f0f0f0;

}

.card\_item\_wrapper .tags {

position: relative;

border-bottom: 1px solid #f0f0f0;

}

.card\_item\_wrapper .tags .brand,

.card\_item\_wrapper .tags .name,

.card\_item\_wrapper .tags .price {

text-align: center;

margin: 5px;

}

.card\_item\_wrapper .tags .brand {

font-size: 17px;

font-weight: 700;

color: #2196F3;

}

.card\_item\_wrapper .tags .name {

font-size: 21px;

font-weight: 700;

}

.card\_item\_wrapper .tags .price {

font-weight: 300;

font-size: 21px;

}

.card\_item\_wrapper .actions {

display: flex;

padding: 15px 0 10px 0;

}

.card\_item\_wrapper .actions .button\_wrapp:nth-child(1) {

flex-grow: 1;

padding-right: 10px;

}

.card\_item\_wrapper.grid\_bars {

float: none;

width: 100%;

display: flex;

}

.card\_item\_wrapper.grid\_bars .image {

width: 420px;

height: 211px;

border: none;

}

.card\_item\_wrapper.grid\_bars .action\_container {

flex-grow: 1;

border: 1px solid #e6e6e6;

}

.card\_item\_wrapper.grid\_bars .description{

clear:both;

padding: 0px 25px;

border-top: 1px solid #eeeeee;

}

.card\_item\_wrapper.grid\_bars .tags{

border-bottom: 0px;

padding: 5px 20px;

}

.card\_item\_wrapper.grid\_bars .tags .brand {

font-size: 21px;

}

.card\_item\_wrapper.grid\_bars .tags .brand,

.card\_item\_wrapper.grid\_bars .tags .name {

float:left

}

.card\_item\_wrapper.grid\_bars .tags .price {

float:right;

}

.card\_item\_wrapper.grid\_bars .actions {

clear:both;

float:right;

padding: 10px 20px 0px 15px;

}

/\*============================

SHOP

==============================\*/

.shop\_wrapper {

display:flex;

}

.shop\_wrapper .left {

flex-grow: 1;

}

.shop\_wrapper .right {

width: 800px;

}

.collapse\_title span {

text-transform: uppercase;

font-weight: 700;

}

.grid\_btn {

float: right;

padding: 10px 10px;

font-size: 20px;

color: #c6c6c6;

}

.grid\_btn.active {

color: #727272;

}

.shop\_options {

border-bottom: 1px solid #e8e8e8;

margin: 0px 20px 20px 20px;

}

/\*============================

PRODUCT DETAIL

==============================\*/

.product\_detail\_wrapper {

display: flex;

margin-bottom: 140px;

}

.product\_detail\_wrapper .right{

flex-grow: 1;

min-height: 800px;

}

.product\_detail\_wrapper p {

font-size: 17px;

font-family: Arial, Helvetica;

font-weight: lighter;

}

.product\_detail\_wrapper .product\_tags {

padding-bottom: 10px;

border-bottom: 1px solid #d3d3d3;

}

.product\_detail\_wrapper .product\_tags .tag {

display: flex;

margin-bottom: 20px;

}

.product\_detail\_wrapper .product\_tags .tag svg {

font-size: 43px;

color: #e4e4e4;

width: 50px;

}

.product\_detail\_wrapper .product\_tags .tag .tag\_text {

font-size: 14px;

text-transform: uppercase;

color: #999592;

padding-left: 20px;

}

.product\_detail\_wrapper .product\_actions .price{

font-size: 40px;

font-weight: 300;

}

.product\_detail\_wrapper .product\_actions .cart {

width: 149px;

}

.user\_cart\_sum {

font-size: 20px;

text-transform: uppercase;

border: 1px solid #e7e7e7;

background: #f7f7f7;

padding: 20px 30px;

}

.cart\_no\_items svg ,

.cart\_success svg {

font-size: 130px;

color: #717171;

}

.cart\_no\_items,

.cart\_success {

text-align: center;

font-size: 28px;

color: #999592;

}

.product\_detail\_wrapper .product\_actions {

padding: 20px 0px 30px 0px;

border-bottom: 1px solid #d3d3d3;

}

.product\_detail\_wrapper .product\_specifications .item{

font-size: 15px;

text-transform: uppercase;

color: #838383;

}

.product\_detail\_wrapper .product\_image\_container {

margin-top: 30px;

padding-right: 70px;

}

.product\_detail\_wrapper .main\_pic {

box-sizing: border-box;

cursor: pointer;

}

.product\_detail\_wrapper .main\_pic div {

width: 100%;

height: 446px;

border: 1px solid #d9d9d9;

background-size: contain !important;

background-position: center center !important;

}

.product\_detail\_wrapper .main\_thumbs {

margin-top: 10px;

}

.product\_detail\_wrapper .main\_thumbs .thumb {

cursor: pointer;

height: 200px;

width: 50%;

background-size: cover !important;

background-position: bottom center !important;

border: 1px solid #e8e8e8;

box-sizing: border-box;

float: left;

}

.paypal\_button\_container{

margin: 10px 0px;

}

/\*=============================

REGISTER LOGIN

===============================\*/

.register\_login\_container {

display: flex;

border: 1px solid #999592;

border-top: 5px solid #999592;

-webkit-box-sizing: border-box;

box-sizing: border-box;

margin: 80px 0px 110px 0px;

padding: 30px;

min-height: 430px;

}

.register\_login\_container .left,

.register\_login\_container .right {

flex:1

}

.register\_login\_container h2 {

margin: 20px 0px;

text-transform: uppercase;

color: #686868;

}

.register\_login\_container p{

font-weight: 300;

font-size: 16px;

width: 90%;

}

/\*=============================

USER

===============================\*/

.user\_container {

display:flex;

margin-bottom: 50px;

min-height: 700px;

}

.user\_container .user\_left\_nav {

border-right: 1px solid #f8f8f8;

}

.user\_container .user\_left\_nav .links a {

display: block;

border-bottom: 1px solid #dddddd;

font-size: 20px;

font-weight: 300;

padding: 10px 0px;

width: 280px;

}

.user\_container .user\_right {

flex-grow: 1;

padding: 0 0 0 29px;

}

.user\_container .user\_nfo\_panel {

background: #f8f8f8;

padding: 20px;

margin-top: 20px;

}

.user\_container .user\_nfo\_panel h1 {

margin: 0 0 20px 0;

line-height: 32px;

text-transform: uppercase;

font-size: 25px;

}

.user\_container .user\_nfo\_panel span{

display: block;

font-size: 18px;

font-weight: 300;

}

.user\_product\_block {

display: flex;

flex-wrap: wrap;

background: white;

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ebebeb;

}

.user\_product\_block .item {

flex-grow: 1;

}

.user\_product\_block .item .image{

height: 125px;

background-size: contain !important;

background-repeat: no-repeat !important;

background-position: center center !important;

margin-right: 10px;

}

.user\_product\_block .item.btn {

width: 0px;

}

.user\_product\_block .cart\_remove\_btn {

cursor: pointer;

background: #F44336;

display: inline-block;

padding: 5px 10px;

color: #ffffff;

margin-top: 50px;

}

/\*=============================

ADMIN

===============================\*/

.admin\_category\_wrapper {

padding-bottom: 50px;

border-bottom: 4px solid #e9e9e9;

}

.admin\_two\_column {

display: flex;

}

.admin\_two\_column .right {

flex-grow: 1;

}

.brands\_container {

height: 200px;

overflow-y: scroll;

margin-right: 20px;

}

.brands\_container .category\_item {

width: 200px;

font-size: 20px;

font-weight: 300;

border: 1px solid #f1f1f1;

padding: 10px;

position: relative;

background: #999592;

color: #fff;

}

.dropzone {

border: 1px solid #e5e5e5;

margin-bottom: 40px;

}

.dropzone\_box {

float: left;

width: 25%;

height: 160px;

padding: 10px;

box-sizing: border-box;

}

.dropzone\_box .wrap {

background: #eaeaea;

height: 100%;

text-align: center;

padding-top: 31px;

box-sizing: border-box;

font-size: 49px;

background-size: contain !important;

background-position: center center !important;

border: 1px solid #eaeaea;

}

**Redux Intigration**

Create =🡺 Action folder and reducers