**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/backend/util.js” file, update the following code:**

import jwt from 'jsonwebtoken';

import config from './config';

// const jwt = require('jsonwebtoken');

// const config = require('./config');

const getToken = (user) => {

    return jwt.sign({

        \_id: user.\_id,

        name: user.name,

        email: user.email,

        isAdmin: user.isAdmin,

    }, config.JWT\_SECRET, {

        expiresIn: '48h'

    })

}

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

    const token = req.headers.authorization;

    if(token) {

        const onlyToken = token.slice(7, token.length);

        jwt.verify(onlyToken, config.JWT\_SECRET, (err, decode) => {

            if(err) {

                return res.status(401).send({ message: 'Invalid Token' });

            }

            req.user = token;

            next();

            return

        });

    }

    return res.status(401).send({ message: "Token is not supplied." });

}

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

    if(req.user && req.user.isAdmin) {

        return next();

    }

    return res.status(401).send({ message: 'Admin Token is not valid.' });

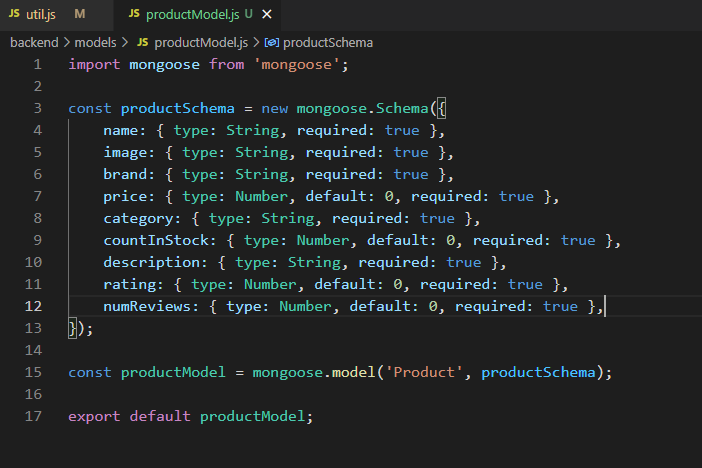
}

export {

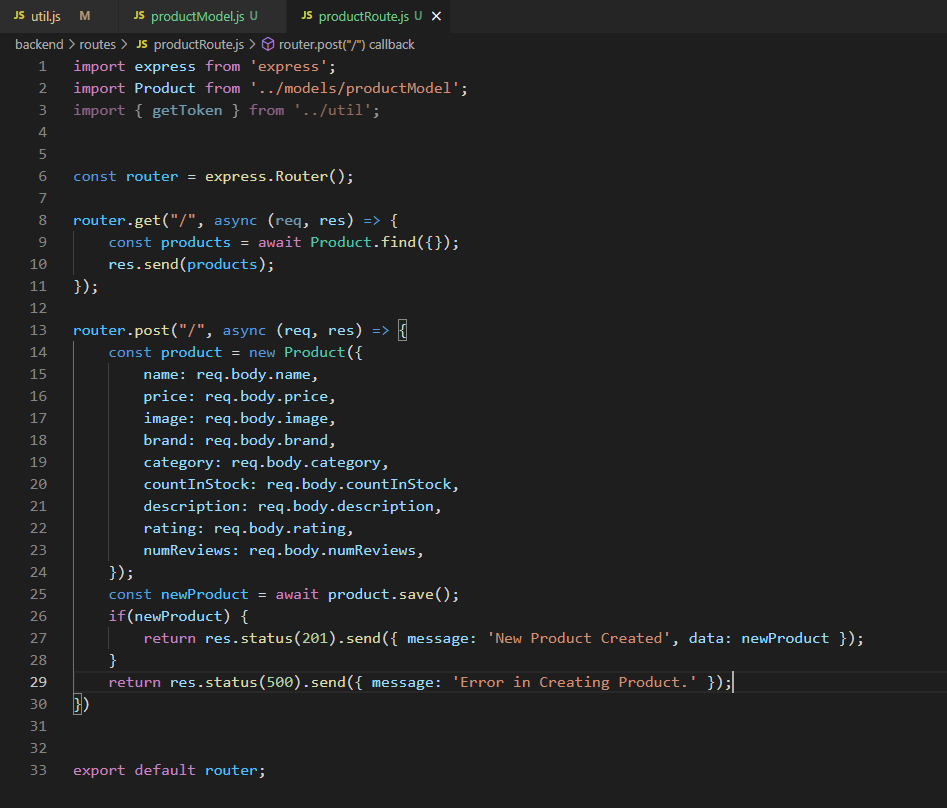
    getToken, isAuth, isAdmin

}

**- Create “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/backend/models/productModel.js” file, add the following code:**



**- Create “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/backend/routes/productRoute.js” file, add the following code:**



**- Create “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/screens/ProductsScreen.js” file, add the following code:**

import React, { useEffect, useState } from 'react';

import { Link } from 'react-router-dom';

import { useSelector, useDispatch } from 'react-redux';

import { signin } from '../actions/userActions';

function ProductsScreen (props) {

    const [name, setName] = useState('');

    const [price, setPrice] = useState('');

    const [image, setImage] = useState('');

    const [brand, setBrand] = useState('');

    const [category, setCategory] = useState('');

    const [countInStock, setCountInStock] = useState('');

    const [description, setDescription] = useState('');

    const [rating, setRating] = useState('');

    const [numReview, setNumReview] = useState('');

    const productSave = useSelector(state => state.userSignin);

    const { loading: loadingSave, success: successSave, error: errorSave } = productSave;

    const dispatch = useDispatch();

    useEffect(() => {

        return () => {

            //

        };

    }, []);

    const submitHandler = (e) => {

        e.preventDefault();

        dispatch(saveProduct({ name, price, image, brand, category, countInStock, description, rating, numReview }));

    }

    return <div className="form">

        <form onSubmit={submitHandler}>

            <ul className="form-container">

                <li>

                    <h2>Create Product</h2>

                </li>

                <li>

                    {loading && <div>Loading...</div>}

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

                </li>

                <li>

                    <label htmlFor="name">Name</label>

                    <input type="text" name="name" id="name" onChange={(e) => setName(e.target.value)}></input>

                </li>

                <li>

                    <label htmlFor="name">Price</label>

                    <input type="text" name="name" id="name" onChange={(e) => setName(e.target.value)}></input>

                </li>

                <li>

                    <label htmlFor="name">Image</label>

                    <input type="text" name="name" id="name" onChange={(e) => setName(e.target.value)}></input>

                </li>

                <li>

                    <label htmlFor="brand">Brand</label>

                    <input type="text" name="brand" id="brand" onChange={(e) => setBrand(e.target.value)}></input>

                </li>

                <li>

                    <label htmlFor="category">Category</label>

                    <input type="text" name="category" id="category" onChange={(e) => setCategory(e.target.value)}></input>

                </li>

                <li>

                    <label htmlFor="rating">Rating</label>

                    <input type="text" name="rating" id="rating" onChange={(e) => setRating(e.target.value)}></input>

                </li>

                <li>

                    <label htmlFor="numReview">Num Reviews</label>

                    <input type="text" name="numReview" id="numReview" onChange={(e) => setNumReview(e.target.value)}></input>

                </li>

                <li>

                    <label htmlFor="description">Description</label>

                    <textarea name="description" id="description" onChange={(e) => setDescription(e.target.value)}></textarea>

                </li>

                <li>

                    <button type="submit" className="button primary">Create</button>

                </li>

            </ul>

        </form>

    </div>

}

export default ProductsScreen;

**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/actions/productActions.js” file, update the following code:**

import {  PRODUCT\_DETAILS\_FAIL, PRODUCT\_DETAILS\_REQUEST, PRODUCT\_DETAILS\_SUCCESS, PRODUCT\_LIST\_FAIL, PRODUCT\_LIST\_REQUEST, PRODUCT\_LIST\_SUCCESS, PRODUCT\_SAVE\_FAIL, PRODUCT\_SAVE\_REQUEST, PRODUCT\_SAVE\_SUCCESS } from "../constants/productConstants"

import axios, { Axios } from 'axios';

const listProducts = () => async (dispatch) => {

    try {

        dispatch({ type: PRODUCT\_LIST\_REQUEST });

        const { data } = await axios.get("/api/products");

        dispatch({ type: PRODUCT\_LIST\_SUCCESS, payload: data });

    }

    catch (error) {

        dispatch({ type: PRODUCT\_LIST\_FAIL, payload: error.message });

    }

}

const saveProduct = (product) = async(dispatch, getState) => {

    try {

        dispatch({ type: PRODUCT\_SAVE\_REQUEST, payload: product });

        const { userSignin:{ userInfo } } = getState();

        const { data } = await Axios.post('/api/products', product, {

            headers: {

                'Authorization': 'Bearer' + userInfo.token

            }

        });

        dispatch({ type: PRODUCT\_SAVE\_SUCCESS, payload: data });

    } catch (error) {

        dispatch({ type: PRODUCT\_SAVE\_FAIL, payload: error.message });

    }

}

const detailsProduct = (productId) => async (dispatch) => {

    try {

        dispatch({ type: PRODUCT\_DETAILS\_REQUEST, payload: productId });

        const { data } = await axios.get("/api/products/" + productId);

        dispatch({ type: PRODUCT\_DETAILS\_SUCCESS, payload: data });

    } catch (error) {

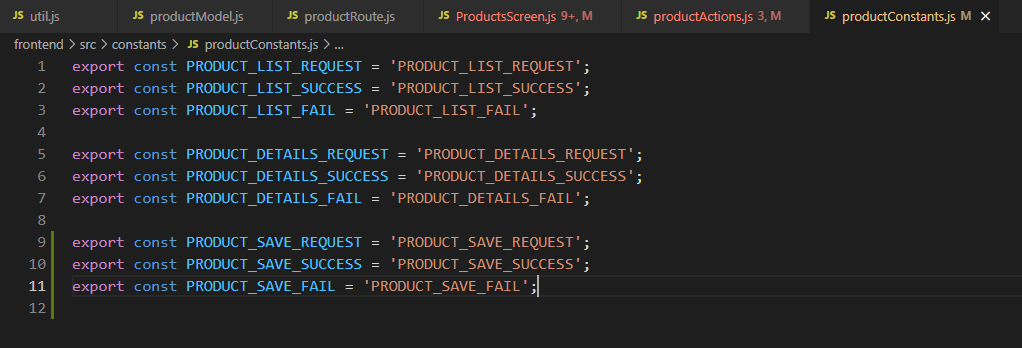
        dispatch({ type: PRODUCT\_DETAILS\_FAIL, payload: error.message });

    }

}

export { listProducts, detailsProduct, saveProduct }

**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/constants/productConstants.js” file, update the following code:**



**- In “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/reducers/productReducers.js” file, update the following code:**

**- Create “SNEAKER\_ECOMMERCE\_WEBSITE\_PROJECT/frontend/src/actions/userActions.js” file, add the following code:**

**(Error 3 ... 3:55:26) Nếu làm như trong video mà tại lúc test đăng ký, khi nhập hết các field và nhấn nút đăng ký mà nó bắn ra lỗi 500.**

**(Solution (Error 3)):**

**- Result: to checking ... 3:55:26**

2