/admin/category/updateCategory

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

import AdminLayout from "../AdminLayout";

import { useRouter } from "next/router";

import { getDocument } from "@/functions/firebase/getData";

import { useAuth } from "@/functions/context";

import CategoryForm from "./categoryForm";

import { toast } from "react-toastify";

import { updateDoc, doc } from "firebase/firestore";

import { db } from "@/functions/firebase";

// update category page will be same states will send to categoryForm

const UpdateCategoryMain = () => {

const { query, replace } = useRouter();

const { id } = query;

// same in

const [title, setTitle] = useState("");

const [image, setImage] = useState({ url: "", name: "" });

const isupdate = true;

const { setPageLoading, pageLoading } = useAuth();

useEffect(() => {

const getData = async () => {

// col , id will change every time data change cats ,id

const singledoc = await getDocument("cats", id);

console.log("single-->", singledoc);

setTitle(singledoc?.title);

setImage(singledoc?.image);

};

if (id) getData();

}, [id]);

//

const handleClick = async (e) => {

console.log("Update");

e.preventDefault();

setPageLoading(true);

const data = {

title: title,

image: image,

};

try {

await updateDoc(doc(db, "cats", id), data);

toast.success("Hero updated successfully");

} catch (error) {

setPageLoading(false);

toast.error(error?.message);

}

setPageLoading(false);

};

return (

<AdminLayout>

<CategoryForm

{...{ title, setTitle, image, setImage, handleClick, isupdate }}

/>

</AdminLayout>

);

};

export default UpdateCategoryMain;

CategoryForm.js

import React from "react";

import { useState } from "react";

import { uploadFile } from "@/functions/firebase/addImage";

import { toast } from "react-toastify";

import { handleDeleteImage } from "@/functions/firebase/addImage";

const CategoryForm = ({

title,

setTitle,

image,

setImage,

handleClick,

isupdate = false,

}) => {

const [file, setFile] = useState("");

const uploadImage = async (e) => {

console.log("file-->", file);

e.preventDefault();

if (!file) {

// return with nothing will stop function and all logic will came after it

toast.error("file is Empty you have to add some files before upload");

return;

}

// if you select same image again dont delete or add any thing

if (file.name === image.name) {

toast.error('sorry you select same image again')

return

}

const filePath = `cats/${file?.name}`;

// from updateCategory page this image is not empty

// if image came from updatecategory page

// delete it then add new image

//&& image?.name !== file.name

if (isupdate && image?.name) {

// (from ,img) from is folder of image , image is object have url and name to delete it

await handleDeleteImage("cats", image);

toast.success("old image deleted success");

}

try {

// upload file data and folder path to uploadFile Function

// if this function success will return image url {link}

const url = await uploadFile(file, filePath);

setImage({ name: file.name, url: url });

toast.success("Image Uploaded Successfully");

setFile({});

console.log("url in form", url);

} catch (error) {

toast.error("something wrong");

}

};

//

return (

<div>

<div className="w-full p-4">

<div className="w-[70%] md:!w-[40%] ">

<div className=" ">

<div className=" text-center mb-4 text-orange-400 font-semibold text-2xl ">

{isupdate ? "Update Category Form" : "Add Category Form"}

</div>

{file?.name}

<p className=" mb-2 text-center font-semibold">Category title</p>

<input

onChange={(e) => setTitle(e.target.value)}

className="w-full border-2 text-black font-medium rounded-md border-teal-400 py-3 px-6"

type="text"

placeholder="title"

value={title}

/>

</div>

<div className=" my-4">

<div className="w-full flex">

<input

type="file"

id="file"

name=""

className="text-black font-medium rounded-md border-teal-400 py-3 px-6 border-2 border-r-0 rounded-r-none"

onChange={(e) => setFile(e.target.files[0])}

/>

<button

onClick={uploadImage}

type="button"

className="rounded-l-none inline-block shrink-0 rounded-md border border-teal-600 bg-teal-600 px-12 py-3 text-sm font-medium text-white transition hover:text-teal-700 focus:outline-none focus:ring active:text-teal-500 "

// onClick={uploadImage}

>

Upload Image

</button>

</div>

<div className="text-xl font-cutiveMono text-center py-2">or</div>

<div className="w-full">

<input

className="w-full border-2 text-black font-medium rounded-md border-teal-400 py-3 px-6"

type="text"

placeholder="Image url"

onChange={(e) => setImage({ ...image, url: e.target.value })}

value={image?.url}

/>

</div>

</div>

{/\* --------show image in updateCategory page ---- \*/}

{isupdate && (

<div>

<img

className=" rounded-sm object-cover object-center my-4 w-[200px] h-[200px] mx-auto"

src={image?.url}

alt=""

/>

</div>

)}

<div className="w-full md:w-full text-center">

<button

onClick={handleClick}

className=" block w-[40%] m-auto shrink-0 rounded-md border border-teal-600 bg-teal-600 px-12 py-3 text-sm font-medium text-white transition focus:outline-none focus:ring active:text-teal-500 "

>

Publish

</button>

</div>

</div>

</div>

</div>

);

};

export default CategoryForm;

/functions/addImage

import { FaBuromobelexperte } from "react-icons/fa6";

import { storage } from "./index";

import {

getDownloadURL,

ref,

uploadBytes,

deleteObject,

} from "firebase/storage";

export const uploadFile = (file, filepath) => {

return new Promise(async (resolve, reject) => {

const storageRef = ref(storage, filepath);

try {

await uploadBytes(storageRef, file);

const url = await getDownloadURL(storageRef);

console.log("image Url in function --->", url);

resolve(url);

} catch (error) {

reject(error);

}

});

};

export const handleDeleteImage = async (from, img) => {

try {

console.log("data", from, img);

const deleteRef = ref(storage, `${from}/${img.name}`);

await deleteObject(deleteObject);

} catch (error) {

console.log("error", error?.message);

}

};