Profe Utilice este "c.nombre AS categoría FROM productos p INNER JOIN categorias c ON p.idcategoria = c.idcategoria" para mostrar el nombre de la categoría, de lo contrario no lo muestra, solo el resto que está en la consulta igualmente en la función para mostrar el formulario de edición del producto la utilice por la misma razón

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

// función para listar los productos

exports.indexProd = async (req, res) => {

try {

/\* Utilize este "c.nombre AS categoria FROM productos p INNER JOIN categorias c ON p.idcategoria = c.idcategoria"

para mostrar el nombre de la categoria, de lo contrario no lo muestra, solo el resto que esta en la consulta

igualmente en la funcion para mostrar el formulario de edición del producto la utlize por la misma razon\*/

const consulta = `SELECT p.\*, c.nombre AS categoria

FROM productos p INNER JOIN categorias c ON p.idcategoria = c.idcategoria;`;

const resultado = await conexP.query(consulta);

resultado.rows.forEach(producto => {

producto.valor = producto.precio \* producto.stock;

});

res.render('post/productos', { productos: resultado.rows });

} catch (error) {

console.error('Error al obtener todos los productos:', error);

res.status(500).json({

message: 'Hubo un error al obtener los productos',

error: error.message

});

}

};

// Funcion para mostrar el formulario de crear un nuevo producto

exports.create = async (req, res) => {

res.render('post/create');

};

// función para registrar un nuevo producto

exports.store = async (req, res) => {

const { nombre, stock, precio, estado, idcategoria } = req.body;

const valorTotal = stock \* precio;

try {

const consulta = `SELECT insertar\_producto($1, $2, $3, $4, $5, $6);`;

const values = [nombre, stock, precio, estado, idcategoria, valorTotal];

const resultado = await conexP.query(consulta, values);

res.redirect('/post');

} catch (error) {

console.error('Error al registrar el producto:', error);

res.status(500).json({

message: 'Hubo un error al registrar el producto',

error: error.message

});

}

};

// Función para mostrar el formulario de edición del producto

exports.edit = async (req, res) => {

const { id } = req.params;

try {

const consulta = `SELECT p.\*, c.nombre AS categoria

FROM productos p INNER JOIN categorias c ON p.idcategoria = c.idcategoria

WHERE p."idProd" = $1;`;

const resultado = await conexP.query(consulta, [id]);

if (resultado.rows.length === 0) {

return res.status(404).json({ message: 'Producto no encontrado' });

}

res.render('post/edit', { producto: resultado.rows[0] });

} catch (error) {

console.error('Error al obtener el producto:', error);

res.status(500).json({

message: 'Hubo un error al obtener el producto.',

error: error.message

});

}

};

// Función para actualizar un producto

exports.update = async (req, res) => {

const { id } = req.params;

const { nombre, stock, precio, estado, idcategoria } = req.body;

const valorTotal = stock \* precio;

try {

const consulta = `SELECT editar\_producto($1, $2, $3, $4, $5, $6, $7);`;

const values = [id, nombre, stock, precio, estado, idcategoria, valorTotal];

const resultado = await conexP.query(consulta, values);

if (resultado.rows.length === 0) {

return res.status(404).json({ message: 'Producto no encontrado' });

}

// Redirigir a la lista de productos

res.redirect('/post');

} catch (error) {

console.error('Error al actualizar el producto:', error);

res.status(500).json({

message: 'Hubo un error al actualizar el producto.',

error: error.message

});

}

};

// Eliminar un producto

exports.delete = async (req, res) => {

const { id } = req.params;

try {

const consulta = `SELECT eliminar\_producto($1);`;

const values = [id];

const resultado = await conexP.query(consulta, values);

if (resultado.rows.length === 0) {

return res.status(404).json({ message: 'Producto no encontrado' });

}

res.redirect('/post');

} catch (error) {

console.error('Error al eliminar el producto:', error);

res.status(500).json({

message: 'Hubo un error al eliminar el producto.',

error: error.message

});

}

};









