//importanto o modulo uuid para BD ficticio

//const { v4: uuid } = require('uuid'); //For generating ID's

//importanto os MODELS

const { where } = require('sequelize');

const { Usuario, Endereco, Estado, Contato } = require('../models');

const { Tipousuario } = require('../models');

const { Login } = require('../models');

const { Router } = require('express');

const roteador = Router();

roteador.get('/', async(req, res) => {

    try {

        log = await Login.findOne({

            include: [

              /\*  { model: Usuario },\*/

                { model: Tipousuario }

            ]

        });

        res.render('login/index', { log });

        //res.status(201).send(login);

    } catch (err) {

        //return res.status(500).send({ err });

        res.status(401).send('Algo deu errado!!! Não foi possível acessar a tela de Login!!!' + err);

    }

})

roteador.post('/', async(req, res) => {

    const usuariologin = req.body.usuariologin;

    const senha = req.body.senha;

    //var usuariologin = "admin";

    //var senha = "12345";

    //recuperar dados via POST    //instalar npm bodyparser

    console.log(req.body.usuariologin);

    console.log(req.body.senha);

    log = await Login.findAll({

        include: [

            { model: Tipousuario },

               { model: Usuario,

            }

        ],

        where: { usuariologin: usuariologin, senha: senha }

    });

    /\* include: [

                    { model: Contato },

                    {

                        model: Endereco,

                        include: [

                            { model: Estado }

                        ],

                    },

                ],\*/

    // 1 - Administrador

    // 2- Supervidor

    // 3 - Atendente

    // 4 - Gerente

    // 5 - Cliente

    if (log.length > 0) {

        if (log[0].statuslogin == 'Ativo' && log[0].tipousuarioid == '1' || log[0].tipousuarioid == '2' || log[0].tipousuarioid == '3' || log[0].tipousuarioid == '4') {

            //session mostrará "Logado com sucesso!"

            console.log(log.length);

            l = log[0];

            req.session.usuariologin = log[0].usuariologin;

            req.session.tipousuarioid = log[0].tipousuarioid;

            // req.session.Tipousuario.tipousuario = log[0].Tipousuario.tipousuario;

            // session == true;

            res.render('login/logado', { usuariologin: log[0].usuariologin, l }); //{ usuariologin: log[0].usuariologin,} são PARÂMETROS para renderizar pagina web

            console.log("ID do perfil logado é: " + req.session.tipousuarioid);

            console.log("Login utilizado é: " + req.session.usuariologin);

            console.log("Tipo de perfil logado é: " + l.Tipousuario.tipousuario);

            console.log("ID do Usuário logado é: " + l.Usuario.id);

            console.log("Nome do Usuário logado é: " + l.Usuario.nome);

        } else if (log[0].statuslogin == 'Ativo' && log[0].tipousuarioid == '5') {

            l = log[0];

            req.session.usuariologin = log[0].usuariologin;

            req.session.tipousuarioid = log[0].tipousuarioid;

            console.log(log);

            res.render('login/loginPerfilNegado', { usuariologin: log[0].usuariologin, tipousuarioid: log[0].tipousuarioid, l });

        } else if (log[0].statuslogin == 'Inativo') {

            /\*Este é para identificar LOGIN DE USUARIO INATIVO  \*/

            console.log("ID do perfil logado é: " + log[0].tipousuarioid);

            console.log("Login utilizado é: " + log[0].usuariologin);

            console.log("Tipo de perfil logado é: " + log[0].Tipousuario.tipousuario);

            console.log("ID do Usuário logado é: " + log[0].Usuario.id);

            console.log("Nome do Usuário logado é: " + log[0].Usuario.nome);

            res.render('login/login\_inativo');

        }

    } else {

        console.log(log);

        res.render('login/login\_erro');

        //(usuariologin != log[0].usuariologin && senha != log[0].senha)

    }

});

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// NEW - renders a form

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

roteador.get('/new', async(req, res) => {

        const tipousuario = await Tipousuario.findAll();

        const estados = await Estado.findAll();

        res.render('login/new', { tipousuario, estados });

    })

    // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    // CREATE - creates a new usuarios

    // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

roteador.post('/new', async(req, res) => {

    try { //dados vindos de Models/Migrations Login (usuarioid)

        //dados vindos de Models/Migrations usuarios

        const nome = req.body.nome;

        const cpf = req.body.cpf;

        const datanasc = req.body.datanasc;

        //dados vindos de Models/Migrations para o Login do Tipo Usuario (tipousuarioid)

        //dados vindos de Models/Migrations Login (usuarioid)

        const usuariologin = req.body.usuariologin;

        const senha = req.body.senha;

        const statuslogin = req.body.statuslogin;

        const tipousuarioid = req.body.tipousuario;

        //dados vindos de Models/Migrations Estados

        const estadosid = req.body.estados;

        //dados vindos de Models/Migration Contatos

        const email = req.body.email;

        const telefone = req.body.telefone;

        //dados vindos de Models/Migrations Enderecos

        const endereco = req.body.endereco;

        const nro = req.body.nro;

        const complemento = req.body.complemento;

        const bairro = req.body.bairro;

        const cidade = req.body.cidade;

        const cep = req.body.cep;

        const novologin = await Login.create({

            usuariologin,

            senha,

            statuslogin,

            tipousuarioid

        });

        const novoendereco = await Endereco.create({

            endereco,

            nro,

            complemento,

            bairro,

            cidade,

            cep,

            estadosid

        });

        const novocontato = await Contato.create({

            email,

            telefone

        });

        const loginid = novologin.id;

        const enderecosid = novoendereco.id;

        const contatosid = novocontato.id;

        await Usuario.create({

            nome,

            cpf,

            datanasc,

            loginid,

            enderecosid,

            contatosid

        });

        res.redirect('/login');

    } catch (err) {

        //return res.status(500).send({ err });

        res.status(401).send('Algo deu errado!!! Não foi possível CRIAR uma NOVO LOGIN!!!' + err);

    }

});

roteador.get('/:id/perfil', async(req, res) => {

    const tipousuario = await Tipousuario.findAll();

    const usuario = await Usuario.findAll();

    res.render('login/perfil', { tipousuario, usuario });

})

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// UPDATE - updates a particular login

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

roteador.patch('/:id', async(req, res) => {

    const { id } = req.params;

    /\*

        const {

            login,

            senha,

            statuslogin,

            e

   //dados vindos de Models/Migrations Perfis

    const tipoperfil = req.body.tipoperfil;

    const statusperfil = req.body.statusperfil;

\*/

    //dados vindos de Models/Migrations Login

    const usuariologin = req.body.usuariologin;

    const senha = req.body.senha;

    const statuslogin = req.body.statuslogin;

    await Login.update({

        usuariologin,

        senha,

        statuslogin

    }, {

        where: {

            id: id

        }

    });

    res.redirect('/home')

})

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// DELETE/DESTROY- removes a single login

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

roteador.delete('/:id', async(req, res) => {

    const { id } = req.params;

    await Login.destroy({

        where: {

            id: id

        }

    });

    res.redirect('/login');

})

module.exports = roteador;