const express = require("express");

const app = express();

const {Pet, Owner, PetType} = require('./models');

app.set("view engine", "pug");

app.use(express.urlencoded( { extended: true }));

app.get("/pets", async (req, res) =>{

  const data = {};

  const pets = await Pet.findAll({

    include: [PetType, Owner],

    order: ['name']

  });

  const petTypes = await PetType.findAll({

    order: ['type']

  });

  const owners = await Owner.findAll({

    order: ['lastName', 'firstName']

  });

  data.pets = pets;

  data.petTypes = petTypes;

  data.owners = owners;

  res.render('pets', data);

});

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

  const data = {};

  const owners = await Owner.findAll({

    order: ['lastName', 'firstName']

  });

  data.owners = owners;

  res.render('owners', data);

});

app.post("/pets", async (req, res)=>{

  const {body: data} = req;

  const pet = await Pet.create({

    name: data.name,

    age: data.age,

    petTypeId: data.typeId

  });

  const ownerIds = data.ownerIds.split(',');

  const owners = await Owner.findAll({

    where: {id: ownerIds} });

  for (let owner of owners) {

    await pet.addOwner(owner);

  }

  res.redirect('/pets');

});

app.post("/owners", async (req, res) =>{

  const { body: data} = req;

  await Owner.create({

    firstName: data.firstName,

    lastName: data.lastName

  });

  res.redirect('/owners');

});

app.get("/", (\_, res) => res.redirect("/pets"));

app.use(express.static('./assets'));

const port = 8081;

app.listen(port, ()=>{

  console.log(`Server accepting connections on ${port}`);

});

PUG files Layout.pug

<!DOCTYPE html>

html(lang="en")

  head

    meta(charset="UTF-8")

    meta(name="viewport", content="width=device-width, initial-scale=1.0")

    link(rel='stylesheet', href="https://unpkg.com/purecss@2.0.3/build/pure-min.css")

    title Pet Track

  body

    header

      ul

        li

          a(href='/pets') Pets

        li

          a(href="/owners") Owners

    main

      block mainContent

    footer

//owners.pug

extends layout

mixin ownerRow(firstName, lastName)

  tr

    td=firstName

    td=lastName

    td

block mainContent

  table.pure-table.pure-table-striped

    thead

      tr

        th First Name

        th Last Name

        th Pets

    tbody

      each owner in owners

        +ownerRow(owner.firstName, owner.lastName)

      else

        tr

          td(colspan=4) No Owners

  form.pure-form(method="post", action="/owners")

    fieldset

      legend Create a owner

      input(type="text", name="firstName", placeholder="First Name", required)

      input(type="text", name="lastName", placeholder="Last Name", required)

      button(type="submit") Create this owner

//pets.pug

extends layout.pug

block mainContent

  table.pure-table.pure-table-striped

    thead

      tr

        th Name

        th Type

        th Age

        th Owners

    tbody

      each pet in pets

        tr

          td= pet.name

          td= pet.PetType.type

          td= pet.age

          td

            ul

              each owner in pet.Owners

                li #{owner.lastName}, #{owner.firstName}

      else

        tr

          td(colspan=4) No pets

  form.pure-form(method="post", action="/pets")

    fieldset

      legend Create a pet

      input(type="text", name="name", placeholder="name", required)

      input(type="number", name="age", placeholder="age", required)

      select(name="typeId")

        each petType in petTypes

          option(value=petType.id)=petType.type

      select(name='ownerIds' multiple)

        each owner in owners

          option(value=owner.id) #{owner.lastName}, #{owner.firstName}

      button(type="submit") Create this pet