const express = require('express');

const app = express();

const path = require('path');

const mongoose = require('mongoose');

const Campground = require('../models/campground');

const Review = require('../models/review')

const methodOverride = require('method-override');

const ejsMate = require('ejs-mate');

const { nextTick } = require('process');

const { runInNewContext } = require('vm');

const ExpressError = require('./utils/ExpressError');

const { STATUS\_CODES } = require('http');

const campgrounds = require('./routes/campgrounds');

const reviews = require('./routes/reviews');

app.use('/campgrounds', campgrounds);

app.use('/campgrounds/:id/reviews', reviews);

app.set('view engine', 'ejs');

app.set('views', path.join(\_\_dirname, 'views'));

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

app.use(methodOverride('\_method'));

app.engine('ejs', ejsMate);

mongoose.connect('mongodb://localhost:27017/yelp-camp', (() => {

    console.log("connect to DB")

}), (e => {

    console.error(e.message);

}));

app.listen(3000, () => {

    console.log("Listen to port 3000")

})

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

    res.render('home')

})

*// app.get('/makeCampground', (req, res) => {*

*//     const newCampground = new Campground({*

*//         title: "My back yield",*

*//         description: "Cheap & available",*

*//     });*

*//     res.send(newCampground);*

*// })*

app.all('\*', (req, res, next) => {

    next(new ExpressError(404, 'Page not found'))

})

app.use((err, req, res, next) => {

*// const { statusCode = 500, message = 'Something went wrong' } = err;*

*// res.status(statusCode).send(message);*

    const { statusCode = 500 } = err;

*if* (!err.message) { err.message = 'Unspecific Errors' }

    res.status(statusCode).render('error', { err })

})

const express = require('express');

const router = express.Router();

const mongoose = require('mongoose');

const methodOverride = require('method-override');

const catchAsync = require('../utils/catchAsync');

const ExpressError = require('../utils/ExpressError');

const Campground = require('../models/campground');

const { campgroundSchema } = require('../schema.js');

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

    const { error } = campgroundSchema.validate(req.body);

*if* (error) {

        const msg = error.details.map(el => el.message).join(',')

*throw* new ExpressError(400, msg)

    }

*else* {

        next();

    }

}

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

    const campgrounds = *await* Campground.find();

    res.render('campgrounds/index', { campgrounds })

})

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

    res.render('campgrounds/new');

})

router.get('/:id/edit', catchAsync(async (req, res) => {

    const { id } = req.params;

    const campground = *await* (Campground.findById(id))

    res.render('campgrounds/edit', { campground });

}))

router.put('/:id', validateCampground, catchAsync(async (req, res) => {

    const { id } = req.params;

    console.log(id);

    const cmpgrd = *await* Campground.findByIdAndUpdate(id, req.body.campground, { runValidators: true })

    res.redirect(`/campgrounds/${cmpgrd.\_id}`);

}))

router.get('/:id', catchAsync(async (req, res, next) => {

    const { id } = req.params;

    const campground = *await* Campground.findById(id).populate('reviews');

    res.render('campgrounds/show', { campground });

}))

router.post('/', validateCampground, catchAsync(async (req, res, next) => {

    const { campground } = req.body;

    console.log(campground);

    const campgrounds = new Campground(campground);

*await* campgrounds.save();

    res.redirect('/campgrounds');

}))

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

    const { id } = req.params;

*await* Campground.findByIdAndDelete(id);

    res.redirect('/campgrounds');

}))

module.exports = router;

const express = require('express');

const router = express.Router({ mergeParams: true });

const Campground = require('../models/campground');

const Review = require('../models/review')

const catchAsync = require('../utils/catchAsync');

const ExpressError = require('../utils/ExpressError');

const { reviewSchema } = require('../schema.js');

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

    const { error } = reviewSchema.validate(req.body);

*if* (error) {

        const msg = error.details.map(el => el.message).join(',')

*throw* new ExpressError(400, msg)

    }

*else* {

        next();

    }

}

router.delete('/:rid', catchAsync(async (req, res) => {

    const { id, rid } = req.params;

*await* Review.findByIdAndDelete(rid);

*await* Campground.findByIdAndUpdate(id, { $pull: { reviews: { $in: [rid] } } })

    res.redirect(`/campgrounds/${id}`)

}))

router.post('/', validateReview, catchAsync(async (req, res, next) => {

    const { id } = req.params;

    const campground = *await* Campground.findById(id);

    const review = *await* new Review(req.body.review);

    campground.reviews.push(review);

*await* campground.save();

*await* review.save();

    res.redirect(`/campgrounds/${id}`)

}))

module.exports = router;

<% layout('layouts/boilerplate') %>

    <div *class*="row">

        <div *class*="col-md-6 offset-3">

            <h1 *class*="text-center">

                Edit Campground

            </h1>

            <form *action*="/campgrounds/<%= campground.\_id %>?\_method=PUT" *method*="POST" *novalidate*

*class*="validate-form">

                <div *class*="mb-3">

                    <label *for*="title" *class*="form-label">

                        Campground title

                    </label>

                    <input *type*="text" *class*="form-control" *id*="title" *name*="campground[title]"

*value*="<%=campground.title %>" *required*>

                </div>

                <div *class*="mb-3">

                    <label *for*="location" *class*="form-label">

                        Location

                    </label>

                    <input *type*="text" *class*="form-control" *id*="location" *name*="campground[location]"

*value*="<%=campground.location %>" *required*>

                </div>

                <div *class*="mb-3">

                    <label *for*="image URL" *class*="form-label">

                        Image URL

                    </label>

                    <input *type*="text" *class*="form-control" *id*="image" *name*="campground[image]"

*value*="<%=campground.image %>" *required*>

                </div>

                <div *class*="mb-3">

                    <label *for*="price" *class*="form-label">

                        Campground Price

                    </label>

                    <div *class*="input-group">

                        <span *class*="input-group-text">$</span>

                        <input *type*="number" *class*="form-control" *name*="campground[price]" *aria-label*="Price per Night"

*id*="price" *value*="<%=campground.price %>" *required*>

                        <span *class*="input-group-text">.00</span>

                    </div>

                </div>

                <div *class*="mb-3">

                    <label *for*="description" *class*="form-label"> Campground Description</label>

                    <textarea *name*="campground[description]" *id*="description" *cols*="30" *rows*="5" *class*="form-control"

*required*>

                    <%=campground.description %>

                </textarea>

                </div>

                <div *class*="mb-3">

                    <button *class*="btn btn-success">Submit</button>

                </div>

            </form>

        </div>

    </div>

    //

*<!-- <form action="/campgrounds/<%= campground.\_id %>?\_method=PUT" method="POST">*

*<div>*

*<label for="title">Campground title</label>*

*<input type="text" id="title" name="campground[title]" value="<%=campground.title %>">*

*</div>*

*<div>*

*<label for="image">Image</label>*

*<input type="text" name="campground[image]" id="image" value="<%=campground.image %>">*

*</div>*

*<div>*

*<label for="location">Location</label>*

*<input type="text" name="campground[location]" id="location" value="<%=campground.location %>">*

*</div>*

*<div>*

*<label for="price">Price</label>*

*<input type="number" name="campground[price]" id="price" value="<%=campground.price %>">*

*</div>*

*<div>*

*<label for="description">Description</label>*

*<textarea name="campground[description]" id="description" cols="30" rows="10">*

*<%=campground.description %>*

*</textarea>>*

*</div>*

*<button>Submit</button>*

*</form>*

*<footer>*

*<p>*

*<a href="/campgrounds">Back to All Campgrounds</a>*

*</p>*

*</footer> -->*

<% layout('layouts/boilerplate') %>

    <div *class*="card">

        <div *class*="row" *mb-3*>

            <div *class*="col-md-6">

                <img *src*="<%= campground.image %>" *class*="card-img-top" *alt*="...">

                <div *class*="card-body">

                    <div *class*="card-title">

                        <h4>

                            <%= campground.title %>

                        </h4>

                    </div>

                    <ul *class*="list-group list-group-flush">

                        <li *class*="list-group-item">

                            <%= campground.location %>

                        </li>

                        <li *class*="list-group-item">$<%= campground.price %> per day</li>

                        <li *class*="list-group-item">

                            <%= campground.description %>

                        </li>

                    </ul>

                    <form *action*="/campgrounds/<%= campground.\_id %>?\_method=DELETE" *method*="POST">

                        <button *class*="btn btn-danger">Delete</button>

                    </form>

                </div>

            </div>

            <div *class*="col-md-6">

                <div *class*="text-center">

                    <h2>Leaves a Review</h2>

                </div>

                <form *action*="/campgrounds/<%= campground.\_id %>/reviews" *method*="POST" *class*="validate-form"

*novalidate*>

                    <div *class*="card-text mb-3">

*<!-- <div class="mb-3"> -->*

                        <label *for*="rating" *class*="form-label">

                            <h5>Enter a Rating</h5>

                        </label>

                        <input *type*="range" *class*="form-range" *min*="1" *max*="5" *id*="rating" *name*="review[rating]">

                    </div>

                    <div *class*="card-text mb-3">

                        <label *for*="body" *class*="form-label">

                            <h5>Write a review</h5>

                        </label>

                        <textarea *name*="review[body]" *id*="body" *cols*="30" *rows*="5" *class*="form-control "

*required*></textarea>

                    </div>

                    <div *class*="card-text mb-3">

                        <button *class*="btn btn-primary">Submit</button>

                    </div>

                </form>

                <% *for*(let review of campground.reviews) {%>

                    <div *class*="card-body">

                        <h5 *class*="card-title mb-3">

                            Rating : <%= review.rating %>

                        </h5>

                        <div *class*="card-text mb-3">

                            Review : <%= review.body %>

                        </div>

                        <form *action*="/campgrounds/<%= campground.\_id %>/reviews/<%= review.\_id %>?\_method=DELETE"

*method*="POST" *mb-3*>

                            <button *class*="btn btn-danger btn-sm">Delete</button>

                        </form>

                    </div>

                    <% } %>

            </div>

        </div>

    </div>

    </div>

*<!-- <h1>*

*<%= campground.title %>*

*</h1>*

*<h2>*

*<%= campground.location %>*

*</h2>*

*<p>*

*<img src="<%= campground.image %>" alt="">*

*</p>*

*<p>*

*$<%= campground.price %> per day*

*</p>*

*<p>*

*<%= campground.description %>*

*</p>*

*<form action="/campgrounds/<%= campground.\_id %>?\_method=DELETE" method="POST">*

*<button>Delete</button>*

*</form>*

*<footer>*

*<p>*

*<a href="/campgrounds">Back to Full List</a>*

*</p>*

*<p>*

*<a href="/campgrounds/<%= campground.\_id %>/edit">Edit Campground</a>*

*</p>*

*</footer> -->*