const express = require('express');

const mongoose = require('mongoose');

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

const bodyParser = require('body-parser');

mongoose.connect('mongodb+srv://khannusrat8220:N9310487906@cluster0.iecmf.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0');

const db = mongoose.connection;

db.on('error', console.error.bind(console, 'mongodb connection error'));

const itemSchema = new mongoose.Schema({

Name: String,

Price: Number,

Description: String

});

const Item = mongoose.model('Item', itemSchema);

const app = express();

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

app.use(express.json());

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

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

app.use(express.static('public'));

app.use(bodyParser.json());

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

try {

const items = await Item.find();

res.render('item-list', { items: items });

} catch (err) {

res.status(500).send(err);

}

});

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

res.render('new-items', { errors: null });

});

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

try {

const item = await Item.findById(req.params.id);

if (!item) {

return res.status(404).send('Item not found');

}

res.render('edit-item', { item, errors: null });

} catch (err) {

res.status(500).send(err);

}

});

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

try {

const item = await Item.findByIdAndDelete(req.params.id);

if (!item) {

return res.status(404).send('Item not found');

}

res.redirect('/items');

} catch (err) {

res.status(500).send(err);

}

});

app.put('/items/:id', async (req, res) => {

try {

const item = await Item.findByIdAndUpdate(req.params.id, req.body, { new: true });

if (!item) {

return res.status(404).send('Item not found');

}

res.redirect('/items');

} catch (err) {

res.status(500).send(err);

}

});

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

try {

const newItem = new Item(req.body);

await newItem.save();

res.redirect('/items');

} catch (err) {

res.status(500).send(err);

}

});

const PORT = 3000;

app.listen(PORT, () => {

console.log('Server is running at port ' + PORT);

});