DOCUMENTATION

INSTALL:

* node
* npm
* npm I express
* npm I mongoose
* npm I –D nodenom

SERVER.JS:

const express = require("express");

const mongoose = require("mongoose");

const order = require('./router/customer');

const db = "mongodb+srv://priyadharshini:Priya\_7140@cluster0.hx9z8.mongodb.net/shopping\_cart?retryWrites=true&w=majority";

const port = 4000;

const app = express();

app.use(express.json());

mongoose.connect(db,{

    useNewUrlParser: true,

  useUnifiedTopology: true,

  useCreateIndex: true

}).then(()=>{

    console.log("mongodb connected");

}).catch((err)=>{

    console.log({err: err });

})

app.use('/orders',order);

app.listen(port,(err)=>{

if(err){

    console.log( {err : err});

}

console.log("server running on port" +port);

});

SCHEMA:

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const ObjectId = mongoose.Types.ObjectId;

const lineSchema = new Schema({

product\_id:{

    \_id:ObjectId

},

price:{

    type:Number,

    required:true

},

quantity:{

    type:Number,

    required:true

},

tax\_lines:{

    type:Array,

    price:{

    type:Number,

    required:true

},

rate:{

    type:Number,

    required:true

},

title:{

    type:String,

    required:true

}

}

});

const customerSchema = new Schema({

    order\_id:{

        \_id:ObjectId

    },

    created\_at:{

        type:Date,

        default:Date.now()

    },

    customer\_id:{

        \_id:ObjectId

    },

    line\_items:[lineSchema]

          });

module.exports = Customer = mongoose.model('customers',customerSchema);

CRUD(operations):

The acronym CRUD stands for create, read, update and delete.

TO CREATE:

const express = require('express');

const router = express.Router();

const Customer = require('../model/customer');

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

    console.log('Users route');

    const newCustomer = new Customer(req.body);

    try {

        await newCustomer.save();

        res.status(201).send(newCustomer);

    } catch (err) {

        res.status(500).send();

    }

});

TO READ:

const express = require('express');

const router = express.Router();

const Customer = require('../model/customer');

router.get('/', async (req, res) => {

    try {

        const customers = await Customer.find({});

        res.send(customers);

    } catch (error) {

        res.status(404).send({ error: 'Path not found' });

    }

});

TO READ A SINGLE CUSTOMER:

const express = require('express');

const router = express.Router();

const Customer = require('../model/customer');

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

    const \_id = req.params.id;

    try {

        const customer = await Customer.findById(\_id);

        if (!customer) {

            return res.status(404).send({ error: 'customer not found' });

        }

        res.send(customer);

    } catch (error) {

        res.status(500).send({ error: 'Internal server error' });

    }

});

TO UPDATE:

const express = require('express');

const router = express.Router();

const Customer = require('../model/customer');

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

    const updates = Object.keys(req.body);

    console.log(updates);

    const allowedUpdates = ['quantity'];

    const isValidOperation = updates.every((update) => {

        return allowedUpdates.includes(update);

    });

    if (!isValidOperation) {

        return res.status(400).send({ error: 'Invalid Operation' });

    }

    try {

        const \_id = req.params.id;

        const customer= await Customer.findById(\_id);

        console.log(\_id);

        if (!customer) {

            return res.status(404).send({ error: 'customer not found' });

        }

        updates.forEach((update) => {

            customer[update] = req.body[update];

        });

        await customer.save();

        res.send(customer);

    } catch (error) {

        res.status(500).send({ error: 'Internal server error' });

    }

});

TO DELETE:

const express = require('express');

const router = express.Router();

const Customer = require('../model/customer');

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

    try {

        const customer = await Customer.findByIdAndDelete(req.params.id);

        if (!customer) {

            return res.status(404).send({ error: 'User not found' });

        }

        res.send(customer);

    } catch (error) {

        res.status(500).send({ error: 'Internal server error' });

    }

});

DATABASE:

