**PROJECT NAME:**

API for ShoppingCart

**INTRODUCTION:**

REST API for ShoppingCart to add ,delete and update products from the database (MONGODB).

**REQUIREMENTS:**

* + - Visual studio code
    - Postman
    - MongoDB
    - Node js

**INSTALLATION:**

* Install visual studio code from this link -><https://code.visualstudio.com/download>.
* Install postman from this link -><https://www.postman.com/downloads>
* Install mongoDB from this link -><https://www.mongodb.com/try/download/community>
* Install node js from this link -><https://nodejs.org/en/download>

**CONFIGURATION:**

**NPM:**

* After installing node js , initiate npm by ***npminit*** in VS code terminal.

**Visual studio code:**

* After installing the VS code, install required packages.
* For our project the required packages are,
  + **Express** ( *npmi express* to install the package).
  + **Mongoose** ( *npmi mongoos*e to install the package).

**MongoDB:**

* After installing mongoDB , Create a database using cmd or mongoDB compass.
* Create a database by giving the required informations (in mongoDB Atlas).

**Postman:**

* After installing postman,get and post your request

**Program:**

Create required schema:

* customerSchema –creating the schema for shopping cart using npm package mongoose

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const LineSchema = new Schema({

price:{

type:String,

required:true

},

quantity:{

type:String,

required:true

},

tax\_lines:{

type:Array,

price:{

type:String,

required:true

},

rate:{

type:String,

required:true

},

title:{

type:String,

required:true

}

}

});

const customerSchema = new Schema({

created\_at:{

type:Date,

default:Date.now()

},

customer\_id:{

type:String,

required:true

},

Line\_items:{

type:[LineSchema],

default:[]

}

});

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

**Creating Routers**

* **Add new customer-by using express.Router class ,post method requesting the body to add new customer and save it using save()**

const express = require('express');

const { findByIdAndDelete } = require('../model/customer');

const router = express.Router();

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

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

const customer = new Customer(req.body);

try {

await customer.save();

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

} catch (error) {

res.status(500).send();

}

});

**Expected Output:**

{

"created\_at": "2021-02-02T16:52:52.435Z",

"\_id": "601983a329a5154ee48af3c9",

"customer\_id": "C002",

"Line\_items": [

{

"tax\_lines": [

{

"price": "6.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "6.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3ca",

"price": "6.99",

"quantity": "1"

},

{

"tax\_lines": [

{

"price": "12.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "12.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3cb",

"price": "12.99",

"quantity": "4"

}

],

"\_\_v": 0

}

* **Add new product-by using post method,requesting the body form the postman , add new products to the existing customer by push() and save it using save()**

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

try {

const customer = await Customer.findOne({customer\_id:req.params.cid});

if (!customer) {

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

}

const newProduct = req.body;

console.log(newProduct);

console.log(typeof customer.Line\_items);

customer.Line\_items.push(newProduct)

customer.save();

res.status(200).send(customer);

} catch (error) {

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

}

});

**Expected output:**

{

"created\_at": "2021-02-02T16:52:52.435Z",

"\_id": "601983a329a5154ee48af3c9",

"customer\_id": "C002",

"Line\_items": [

{

"tax\_lines": [

{

"price": "6.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "6.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3ca",

"price": "6.99",

"quantity": "1"

},

{

"tax\_lines": [

{

"price": "12.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "12.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3cb",

"price": "12.99",

"quantity": "4"

},

{

"tax\_lines": [

{

"price": "5.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "5.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601985c9ad023a3548f465d4",

"price": "5.99",

"quantity": "2"

}

],

"\_\_v": 0

}

* **Get OrderdDetails using OrderId- by using get (),findById() , getting details of order items**

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

try {

const order = await Customer.findById(req.params.id);

if (!order) {

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

}

res.send(order);

} catch (error) {

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

}

});

**Expected Output:**

{

"created\_at": "2021-02-02T16:52:52.435Z",

"\_id": "601983a329a5154ee48af3c9",

"customer\_id": "C002",

"Line\_items": [

{

"tax\_lines": [

{

"price": "6.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "6.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3ca",

"price": "6.99",

"quantity": "1"

},

{

"tax\_lines": [

{

"price": "12.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "12.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3cb",

"price": "12.99",

"quantity": "4"

},

{

"tax\_lines": [

{

"price": "5.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "5.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601985c9ad023a3548f465d4",

"price": "5.99",

"quantity": "2"

}

],

"\_\_v": 1

}

* **Get ordered products using CustomerId- by using get (),findOne() , getting all ordered products**

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

try {

const customer = await Customer.findOne({customer\_id:req.params.cid});

if (!customer) {

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

}

res.send(customer.Line\_items);

} catch (error) {

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

}

});

**Expected Output:**

[

{

"tax\_lines": [

{

"price": "6.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "6.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3ca",

"price": "6.99",

"quantity": "1"

},

{

"tax\_lines": [

{

"price": "12.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "12.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3cb",

"price": "12.99",

"quantity": "4"

},

{

"tax\_lines": [

{

"price": "5.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "5.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601985c9ad023a3548f465d4",

"price": "5.99",

"quantity": "2"

}

]

* **Get total number of products and total price by customerId-by using get(),findOne() to get the customer and getting the length of Line\_items for Total no of products and mapping through Line\_items to add the price of each products to get Total price**

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

try {

const customer = await Customer.findOne({customer\_id:req.params.cid});

if (!customer) {

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

}

const products=customer.Line\_items.length;

const price = customer.Line\_items;

var y = 0;

price.forEach((e)=>{

y += parseFloat(e.price);

});

console.log(y);

const total = [{"total products":products,"totalprice":y}]

total.push();

res.send(total);

} catch (error) {

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

}

});

**Expected Output:**

[

{

"total products": 3,

"totalprice": 25.97

}

]

* **Updating quantity using order id and product id-by using patch(),requesting body from the post man ,finding the order by findById() ,mapping the products using map(),finding the product by comparing the params id and product id and updating using req.body.quantity**

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

const updates = Object.keys(req.body)

console.log(updates);

try {

const orderId = req.params.id;

const product\_id = req.params.pid;

const customer = await Customer.findById(orderId)

if (!customer) {

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

}

customer.Line\_items.map((e)=>{

if(e.\_id==product\_id)

{

e.quantity = req.body.quantity

}

})

await customer.save()

res.send(customer);

} catch (error) {

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

}

});

**Expected Output:**

{

"created\_at": "2021-02-02T16:52:52.435Z",

"\_id": "601983a329a5154ee48af3c9",

"customer\_id": "C002",

"Line\_items": [

{

"tax\_lines": [

{

"price": "6.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "6.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3ca",

"price": "6.99",

"quantity": "5"

},

{

"tax\_lines": [

{

"price": "12.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "12.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3cb",

"price": "12.99",

"quantity": "4"

},

{

"tax\_lines": [

{

"price": "5.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "5.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601985c9ad023a3548f465d4",

"price": "5.99",

"quantity": "2"

}

],

"\_\_v": 1

}

* **Delete a product in Line\_items-by using delete(),findById,get the order and find the product using product id deleteing it using remove()**

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

const orderId = req.params.id;

const product\_id = req.params.pid;

try {

const customer = await Customer.findById(orderId);

if (!customer) {

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

}

customer.Line\_items.map((e)=>{

if(e.\_id == product\_id)

return e.remove()

});

customer.save()

res.send(customer);

} catch (error) {

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

}

});

module.exports = router;

**Expected Output:**

{

"created\_at": "2021-02-02T16:52:52.435Z",

"\_id": "601983a329a5154ee48af3c9",

"customer\_id": "C002",

"Line\_items": [

{

"tax\_lines": [

{

"price": "6.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "6.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3ca",

"price": "6.99",

"quantity": "5"

},

{

"tax\_lines": [

{

"price": "12.99",

"rate": "0.06",

"title": "State Tax"

},

{

"price": "12.99",

"rate": "0.06",

"title": "Central Tax"

}

],

"\_id": "601983a329a5154ee48af3cb",

"price": "12.99",

"quantity": "4"

}

],

"\_\_v": 1

}

**Running the server:using express,mongoose to connect db and using a middleware (/orders)**

const express = require("express");

const mongoose = require("mongoose");

const bodyParser = require("body-parser")

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

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

const port = 3000;

const app = express();

app.use(express.json());

app.use(bodyParser.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);

});

**To run :Type nodemon start in terminal**