**GANPAT UNIVERSITY**

**U.V. PATEL COLLEGE OF ENGINEERING**

B.Tech 5th Semester CE/IT

2CEIT5PE4: Software Packages

**Practical: 7**

**Working with Express Framework**

**Que:1 Build a simple express js program to add update delete and display all products available in list of e-commerce application using get put post and delete methods. Assume that products are available on products object. Product properties are product\_id product\_name product\_size product\_brand product\_color.**

var express = require('express');

var bap = require('body-parser');

var app = express();

var products=[

    {

        product\_id:"123456789",

        product\_name:"HP Pavilion Gaming Laptop 15",

        product\_size:"15inch",

        product\_brand:"HP",

        product\_color:"Infrared"

    }

];

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

app.use(bap.json());

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

    res.send(products);

});

app.post("/Add\_Product/",(req,res)=>{

    var sdata = req.body;

    products.push(sdata);

    res.send(products);

});

app.put("/Edit\_Product/:product\_id",(req,res)=>{

    var old\_product\_id = req.params.product\_id;

    var newdata = req.body;

    var index = products.findIndex((element) =>

element.product\_id === old\_product\_id);

    if(index!=-1)

    {

        products[index].product\_id=newdata.product\_id;

        products[index].product\_name=newdata.product\_name;

        products[index].product\_size=newdata.product\_size;

        products[index].product\_brand=newdata.product\_brand;

        products[index].product\_color=newdata.product\_color;

    }

    res.send(products);

});

app.delete("/Delete\_Product/:product\_id",(req,res)=>{

    var old\_product\_id = req.params.product\_id;

    products = products.filter(item => item.product\_id !== old\_product\_id);

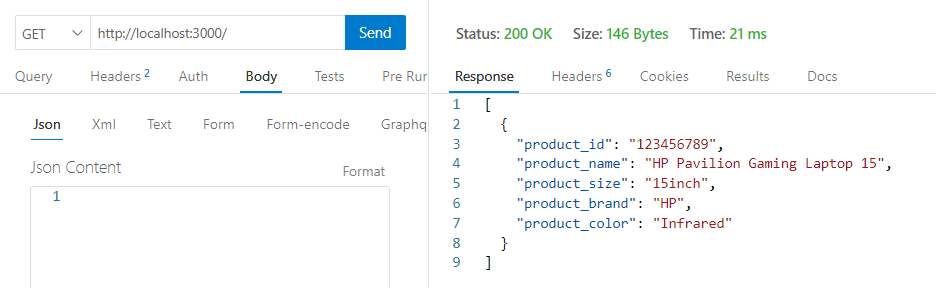
    res.send(products);

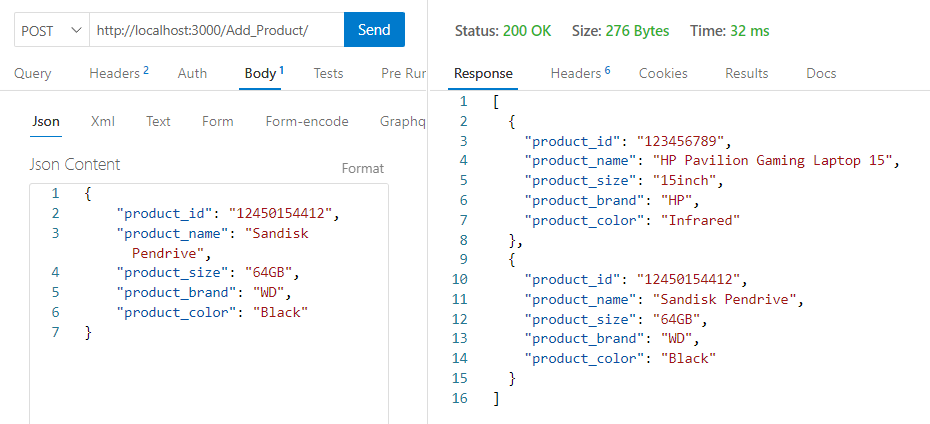
});

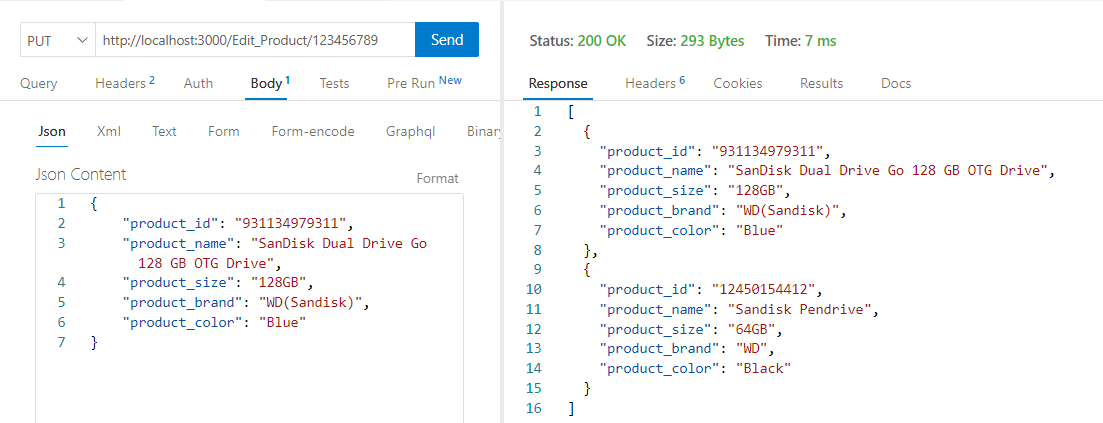
app.listen(3000);

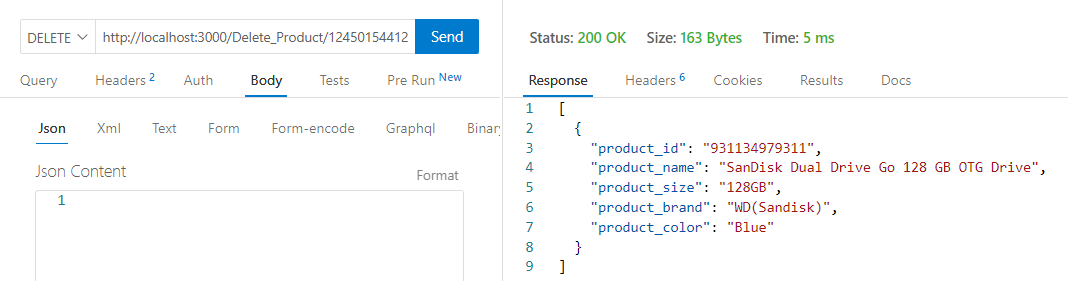
**Output:**











**Que:2 Create express js program using router for e-commerce application. This application has 4 modules: order user product and category. Each module has 4 methods namely get put post and delete and has path ‘/’ ‘/update-details’ ‘create-details’ and ‘delete-details respectively. For example: ‘localhost:8000/product’ will be accessed and the page‘/’ will displayed message “get method from user module” same for other 3 methods and modules.**

***q2.js***

var express = require('express');

var user = require('./user.js');

var order = require('./order.js');

var product = require('./product.js');

var category = require('./category.js');

var app = express();

app.use("/user",user.router1);

app.use("/order",order.router1);

app.use("/product",product.router1);

app.use("/category",category.router1);

app.listen(3000);

***user.js***

var express = require('express');

var router1 = express.Router();

router1.get('/', function (req, res, next) {

    res.send("This is get method of user");

});

router1.post('/create-details', function (req, res, next) {

    res.send("This is post method of user");

});

router1.put('/update-details', function (req, res, next) {

    res.send("This is put method of user");

});

router1.delete('/delete-details', function (req, res, next) {

    res.send("This is delete method of user");

});

exports.router1=router1

***order.js***

var express = require('express');

var router1 = express.Router();

router1.get('/', function (req, res, next) {

    res.send("This is get method of order");

});

router1.post('/create-details', function (req, res, next) {

    res.send("This is post method of order");

});

router1.put('/update-details', function (req, res, next) {

    res.send("This is put method of order");

});

router1.delete('/delete-details', function (req, res, next) {

    res.send("This is delete method of order");

});

exports.router1=router1

***product.js***

var express = require('express');

var router1 = express.Router();

router1.get('/', function (req, res, next) {

    res.send("This is get method of product");

});

router1.post('/create-details', function (req, res, next) {

    res.send("This is post method of product");

});

router1.put('/update-details', function (req, res, next) {

    res.send("This is put method of product");

});

router1.delete('/delete-details', function (req, res, next) {

    res.send("This is delete method of product");

});

exports.router1=router1;

***category.js***

var express = require('express');

var router1 = express.Router();

router1.get('/', function (req, res, next) {

    res.send("This is get method of category");

});

router1.post('/create-details', function (req, res, next) {

    res.send("This is post method of category");

});

router1.put('/update-details', function (req, res, next) {

    res.send("This is put method of category");

});

router1.delete('/delete-details', function (req, res, next) {

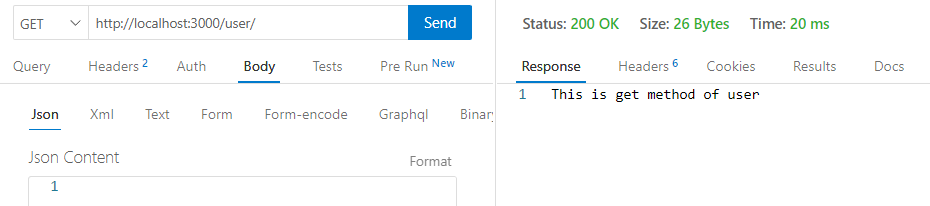
    res.send("This is delete method of category");

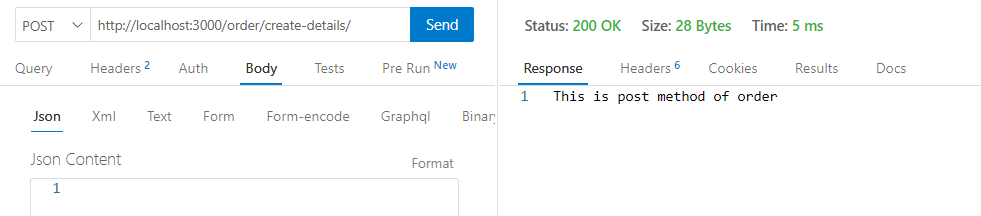
});

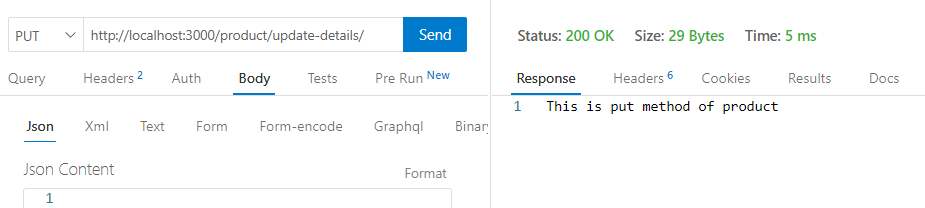
exports.router1=router1;

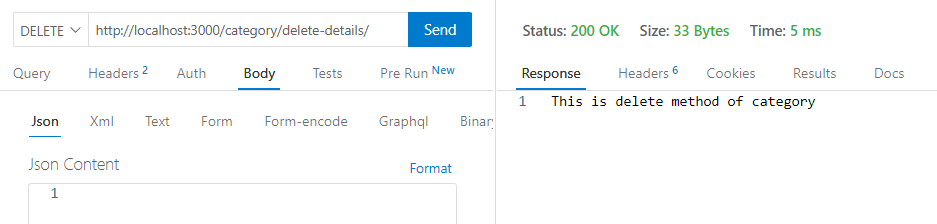
**Output:**









****

**Que:3 Create express js program that employ middleware using next() function.**

const express = require('express');

const app = express();

app.use(Employ);

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

    res.send("This is Home Page");

    console.log('This is Home Page');

});

function Employ (req, res, next) {

    console.log('This is Employ Middleware');

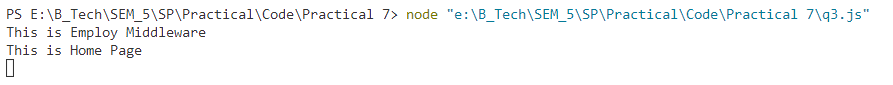
    next();

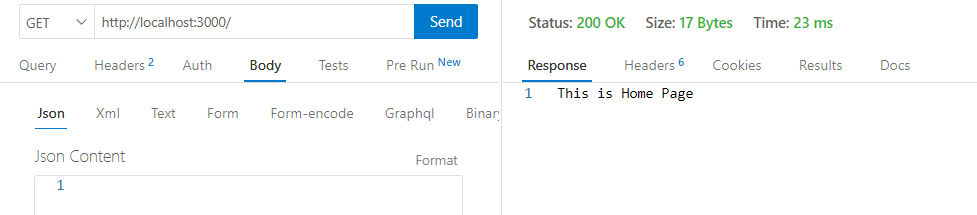
}

app.listen(3000);

**Output:**

****

****

****

**Que:4 Build a simple express js program to add update delete and display all products available in list of e-commerce application using get put post and delete methods. Using monodb database. Assume that products are available on ‘products’ collection in ‘ecom’ database. Product properties are product\_id product\_name product\_size product\_brand product\_color.**

var express = require('express');

var bap = require('body-parser');

var app = express();

const MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost/"

const dbname = "ecom";

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

app.use(bap.json());

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

    MongoClient.connect(url, (err1, db)=>{

        if(err1){

            throw err1;

        }

        else{

            var database = db.db(dbname);

            console.log("Database Successfully Connected!");

            database.collection('products').find({})

.toArray(function(err2,data\_result){

                if(err2)

                {

                    throw err2;

                }

                else{

                    res.send(data\_result);

                    db.close();

                }

            });

        }

    });

});

app.post("/Add\_Product/",(req,res1)=>{

    var sdata = req.body;

    MongoClient.connect(url, (err1, db)=>{

        if(err1){

            throw err1;

        }

        else{

            var database = db.db(dbname);

            console.log("Database Successfully Connected!");

            database.collection('products')

.insertMany(sdata,function(err2,res2){

                if(err2)

                {

                    throw err2;

                }

                else{

                    console.log(res2.insertedCount+" Data inserted");

                    database.collection('products').find({})

.toArray(function(err3,res3){

                        if(err3)

                        {

                            throw err3;

                        }

                        else{

                            res1.send(res3);

                            db.close();

                        }

                    });

                }

            });

        }

    });

});

app.put("/Edit\_Product/:product\_id",(req,res1)=>{

    var old\_product\_id = {"product\_id":req.params.product\_id};

    var data = req.body;

    var newdata = {$set:data};

    MongoClient.connect(url, (err1, db)=>{

        if(err1){

            throw err1;

        }

        else{

            var database = db.db(dbname);

            console.log("Database Successfully Connected!");

            database.collection('products')

.updateOne(old\_product\_id,newdata,function(err2,res2){

                if(err2)

                {

                    throw err2;

                }

                else{

                    console.log(res2.modifiedCount+" Data Updated");

                    database.collection('products').find({})

.toArray(function(err3,res3){

                        if(err3)

                        {

                            throw err3;

                        }

                        else{

                            res1.send(res3);

                            db.close();

                        }

                    });

                }

            });

        }

    });

});

app.delete("/Delete\_Product/:product\_id",(req,res1)=>{

    var old\_product\_id = {"product\_id":req.params.product\_id};

    MongoClient.connect(url, (err1, db)=>{

        if(err1){

            throw err1;

        }

        else{

            var database = db.db(dbname);

            console.log("Database Successfully Connected!");

            database.collection('products')

.deleteOne(old\_product\_id,function(err2,res2){

                if(err2)

                {

                    throw err2;

                }

                else{

                    console.log(res2.deletedCount+" Data Deleted");

                    database.collection('products').find({})

.toArray(function(err3,res3){

                        if(err3)

                        {

                            throw err3;

                        }

                        else{

                            res1.send(res3);

                            db.close();

                        }

                    });

                }

            });

        }

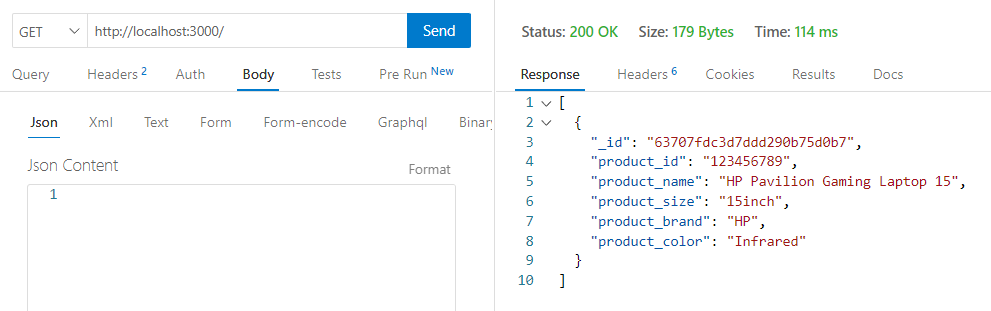
    });

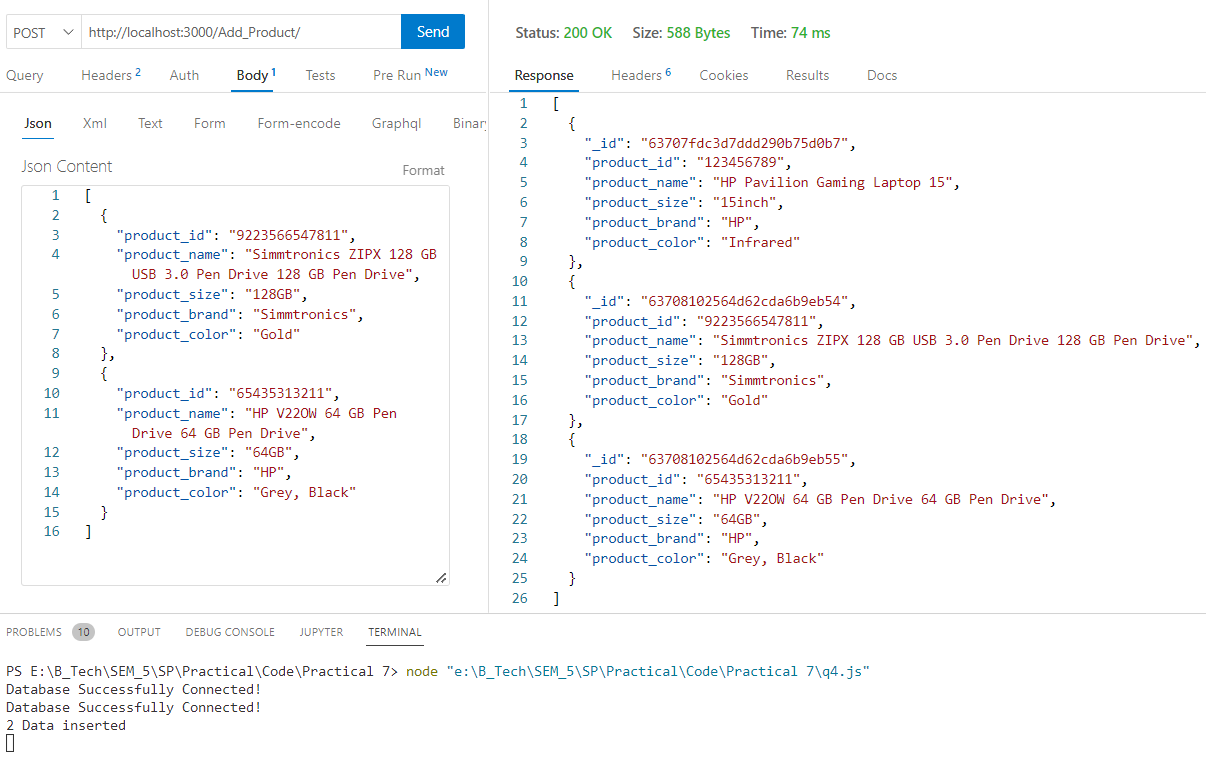
});

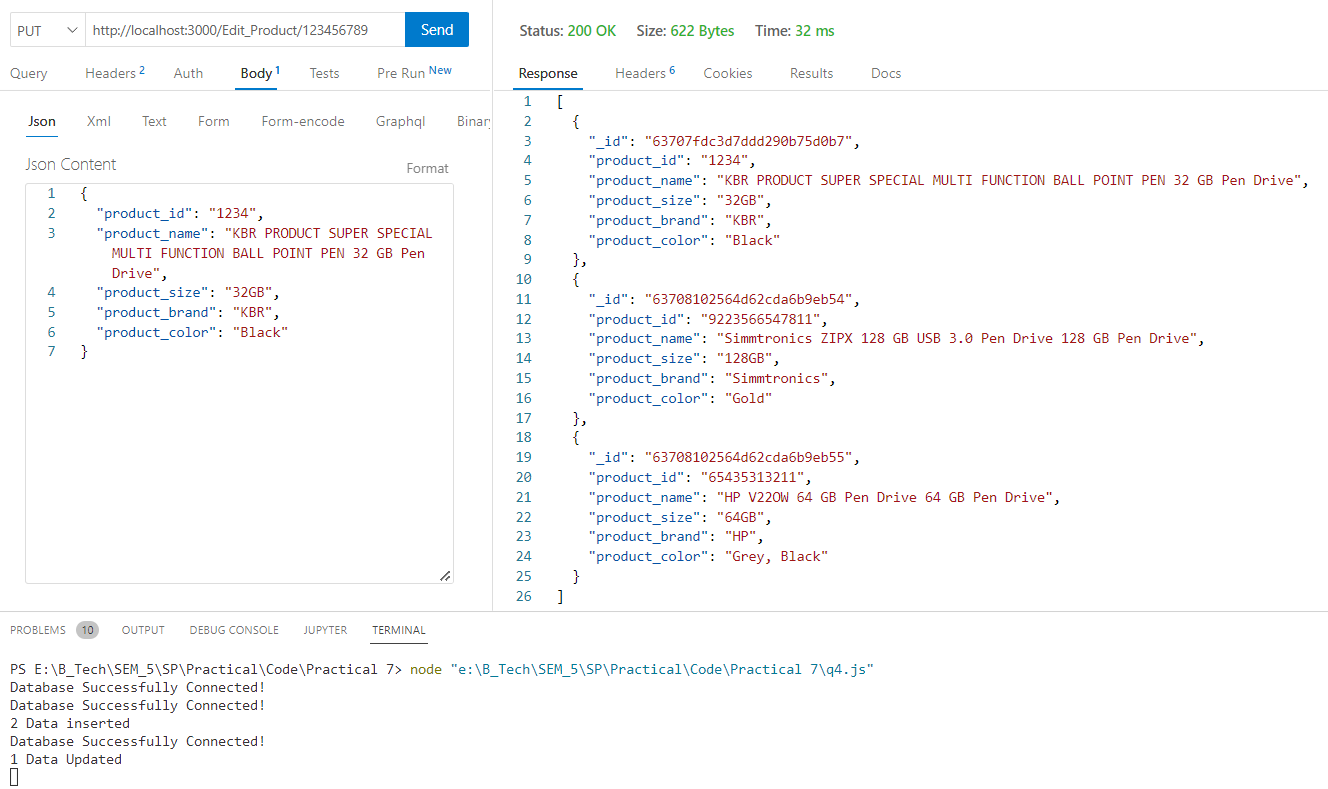
app.listen(3000);

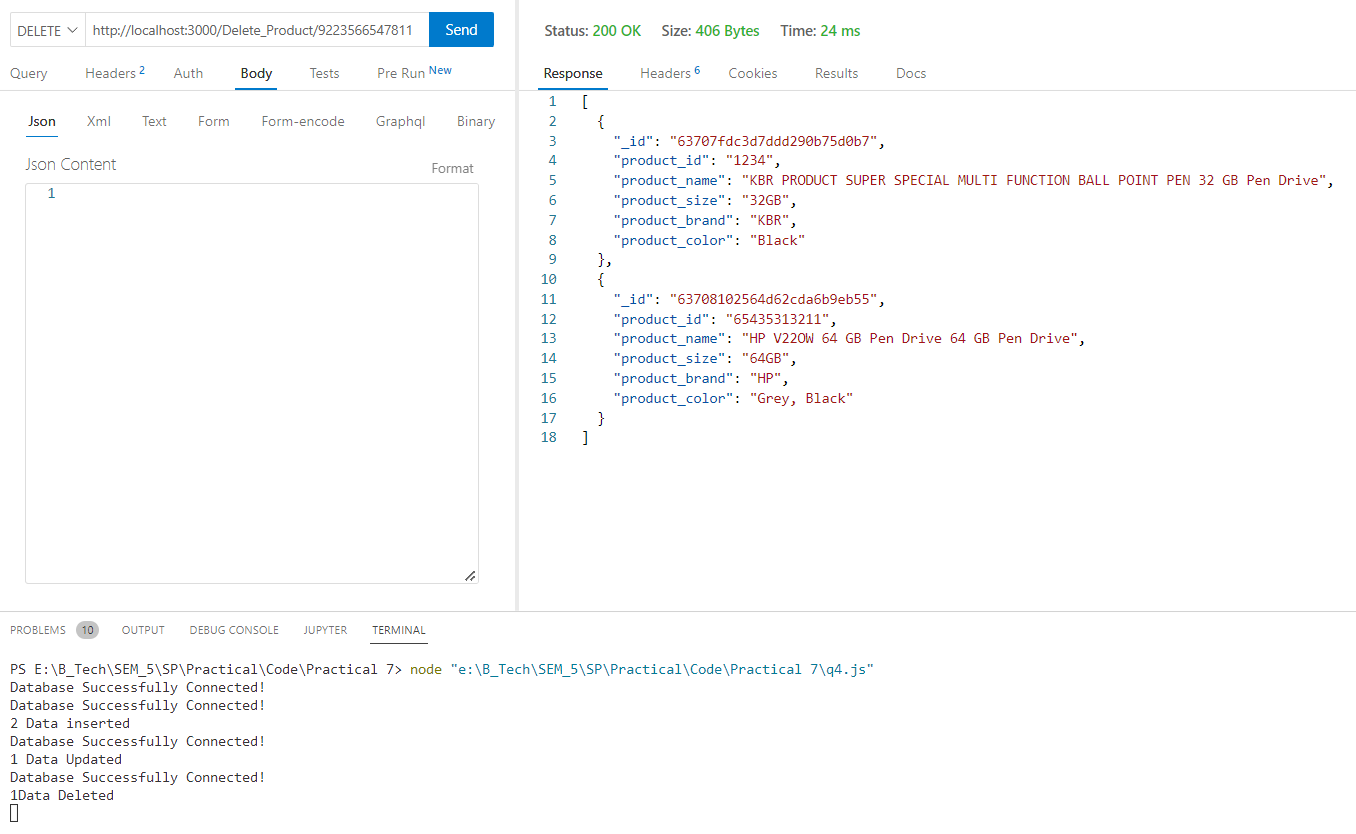
**Output:**

****

****

****

****

****