const express = require("express");

const mongoose = require("mongoose");

const User = require("./schemas/schema");

const Product = require("./schemas/prodSchema");

var path = require('path');

const port = 8000;

const app = express();

mongoose.connect("mongodb://localhost:27017/seebiz", {

    useNewUrlParser: true,

    useUnifiedTopology: true,

})

.then((res) => {

    console.log("Connected to Database.")

}). catch((error) => {

    console.log(error);

})

app.get('/', (req, res) => {

    res.sendFile(path.join(\_\_dirname + '/index.html'));

});

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

    const user = await User.find({});

    // console.log(user);

    res.send(user);

})

app.get("/user/:user\_id/products", async (req, res) => {

    var uId=mongoose.Types.ObjectId(req.params.user\_id);

    // const myProduct = await Product.populate((uId));

    const userProducts = await Product.find({merchant\_id: uId}).limit(10);

    res.send(userProducts);

});

app.post("/user/:user\_id/products", async (req, res) => {

   ///////////////////

});

app.delete("/delete/products/:pid", async(req,res)=>

{

    var \_pid=mongoose.Types.ObjectId(req.params.pid);

    console.log("pid : " ,\_pid)

    Product.deleteOne({

        \_id: \_pid

    })

    // await Product.save();

    res.send({value :"Deleted Successfully"});

});

app.listen(port, ()=> {

    console.log("Listening to ", port);

});

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- <link rel="stylesheet" type="text/css" href="./test.css"> -->

    <style>

      .dropdown{

        width: 100px;

        color:brown

      }

    </style>

    <title>Document</title>

</head>

<body>

  <span class="dropdown">

    <button type="button" id="dropdownMenuButton1" data-bs-toggle="dropdown" aria-expanded="false" onclick="display()">Dropdown</button>

    <span style="display: none;">

      <ul id="listID">

     </ul>

    </span>

  </span>

  <div>

    <table id="TableID">

      <tr>

        <th>Name</th>

        <th>Price</th>

      </tr>

    </table>

  </div>

 <span>

  <table id="updateRecord">

    <tr>

      <td>

        <form autocomplete="off" submit="event.prevetdefault();">

          <div>

            <div>

              <label>Name : </label>

              <input type="text" name="ProductName">

            </div>

            <div>

              <label>Price : </label>

              <input type="number" name="ProductPrice">

            </div>

          </div>

          <div class="form-button">

            <input type="submit" value="UPDATE">

          </div>

        </form>

      </td>

    </tr>

  </table>

 </span>

  <script>

/////////////////////for display nd fetching data from DB

      function display()

      {

          var displayU = document.getElementById("dropdownMenuButton1").nextElementSibling;

          if(displayU.style.display=="block")

          {

              displayU.style.display="none";

          }

          else{

              displayU.style.display="block";

          }

          fun();

      }

      function user\_display(users)

      {

        var ul = document.getElementById("listID");

                    //  arr = ["html","css","java","javascript","php","c++","node.js","ASP","JSP","SQL"];

             for(var i = 0; i < users.length; i++)

             {

                 var li = document.createElement("LI"),

                     txt = document.createTextNode(users[i].username);

                     var btn=document.createElement("BUTTON");

                    //  btn.getAttribute('id',users[i].\_id);

                    btn.appendChild(txt);

                    btn.onclick = (function(id){

                    return function()

                    {

                       getProducts(id);

                    }

                    })(users[i].\_id);

                     li.appendChild(btn);

                     ul.appendChild(li);

              }

      }

      /////////////update button onclick

     function updateTheRecords(pid)

     {

     }

     /////////////////record deletion in database (Products)

     function products\_display(products,user\_id)

     {

        // console.log("User ID",);

        var table=document.getElementById("TableID");

        for (var i=0;i<products.length;i++)

        {

          var tr=document.createElement("TR");

          var td1=document.createElement("TD");

          var td2=document.createElement("TD");

          var btn1=document.createElement("BUTTON");//////////delete button

          btn1.onclick = (function(id){

            return function()

            {

              updateTheRecords(id);

              fetchRecordInTable(this);

            }

          })(products[i].\_id,this);

          var btn2=document.createElement("BUTTON");//////edit button

          btn2.onclick=(function(id)

          {

            return function()

            {

              DeleteTheRecord(id);

              removeRecord(this);

            }

          })(products[i].\_id,this);

          ////////////changing text to a node value

          txt1=document.createTextNode(products[i].title);

          txt2=document.createTextNode(products[i].price);

          txt3=document.createTextNode("Edit");

          txt4=document.createTextNode("Delete");

          /////////////appending text to table data

          td1.appendChild(txt1);

          td2.appendChild(txt2);

          btn1.appendChild(txt3);

          btn2.appendChild(txt4);

          //////////////appending table data to table row

          tr.appendChild(td1);

          tr.appendChild(td2);

          tr.appendChild(btn1);

          tr.appendChild(btn2);

          //////////////////appending all rows data to a table

          table.appendChild(tr);

        }

      }

      async function fun()

     {

          await fetch("http://localhost:8000/user")

          .then( async(res) => {

              users = await res.json();

              user\_display(users);

          }).catch((error) => {

              console.log(error);

          })

      }

      async function DeleteTheRecord(pid\_id)

      {

        await fetch(`http://localhost:8000/delete/products/${pid\_id}`, {method:"DELETE"})

        .then(async(res)=>

        {

          myValue=await res.json();

          alert(myValue.value);

        }).catch((error)=>

        {

          console.log(error);

        });

      }

      function removeRecord(index)

      {

        var i=index.parentNode.rowIndex;

        document.getElementById("TableID").deleteRow(i);

      }

      async function getProducts(user\_id)

      {

        await fetch(`http://localhost:8000/user/${user\_id}/products`)

        .then( async(res) => {

          products = await res.json();

          products\_display(products,user\_id);

              // console.log(products[0].price);

            }).catch((error) =>

            {

              console.log(error);

            });

          }

          </script>

</body>

</html>

const  mongoose  = require("mongoose");

const { schema } = require("./schema");

const productSchema=new mongoose.Schema(

    {

        \_id:{

            type:String

        },

        title:{

            type:String

        },

        tierprice:{

            type: Number

        },

        merchant\_id:{

            type : mongoose.Types.ObjectId,

        },

        product\_images:{

            type:String,

            isTrashed: Boolean,

            isfeature:Boolean

        },

        users:[{type : mongoose.Schema.Types.ObjectId, ref : 'users'}]

    }

);

module.exports=mongoose.model("products",productSchema);

const  mongoose  = require("mongoose");

const userSchema=new mongoose.Schema({

    id\_ :{

        type:String

    },

    phone:{

        type:Number

    },

    merchant\_id :{

        type : Number

    },

    username: {

        type: String

    },

    email:{

        type:String

    },

    password:{

        type:String

    },

    api\_key:{

        type:String

    },

    unique\_url:{

        type:String

    },

    address:{

        country:String,

        city:String,

        zip:Number,

        street:String,

        state:String

    },

    products : [{type : mongoose.Schema.Types.ObjectId,ref:'products'}]

});

module.exports=mongoose.model("users",userSchema);