**V2**

import ballerina/grpc;

import ballerina/uuid;

import ballerina/sql;

type Admin record {|

 readonly string user\_id;

    string user\_name ;

    string password ;

    string role ;

|};

type Product record {|

   readonly string product\_id;

   string product\_name;

   float product\_price;

   int stock\_quantity;

   int stock\_keeping\_unit;

   string status;

|};

table <Admin> key(user\_id) adminTable = table[];

type <Product> key(product\_id) productTable = table[];

public function main() {

        adminTable.add({user\_id: "admin1",user\_name: "Joanna",password: "password1",role: "Admin"});

        adminTable.add({user\_id: "admin2",user\_name:"Anna",password:"password2",role:"Admin"});

        adminTable.add({user\_id:"admin3",user\_name:"Michal",password:"password3",role:"Admin"});

        adminTable.add({user\_id:"admin4",user\_name:"Joe",password:"password4",role:"Admin"});

    productTable.add({product\_id:"product1",product\_name:"Product 1",product\_price:200,stock\_quantity:300,stock\_keeping\_unit:20,status:"Available"});

    productTable.add({product\_id:"product2",product\_name:"Product 2",product\_price:150,stock\_quantity:40,stock\_keeping\_unit:200,status:"Not Available"});

    productTable.add({product\_id:"product3",product\_name:"Product 3",product\_price:1000,stock\_quantity:800,stock\_keeping\_unit:15,status:"Available"});

    productTable.add({product\_id:"product4",product\_name:"Product 3",product\_price:500,stock\_quantity:70,stock\_keeping\_unit:45,status:"Not Available"});

}

listener grpc:Listener grpcListener = new (9090);

# Description.

@grpc:Descriptor {value: SHOPPING\_DESC}

service "Admin" on grpcListener {

    remote function AddAdmin(AddAdminRequest value) returns AddAdminResponse|error {

        Admin newAdmin = {user\_name: value.user\_name,password: value.password,role: value.role,user\_id: ""};

        adminTable.add(newAdmin);

        AddAdminResponse response = {user\_id: value.user\_id,role: value.role};

        return response;

    }

    remote function AuthenticateUser(AuthenticateAdminRequest value) returns AuthenticateAdminResponse|error {

        boolean isAuthenticated = false;

        foreach Admin admin in adminTable {

            if admin.user\_name == value.user\_name && admin.password == value.password{

                isAuthenticated = true;

                break;

            }

        }

        string message  = isAuthenticated? "Authentication Successful" : "Authentication Failed";

        AuthenticateAdminResponse response = {message : message};

        return response;

    }

    remote function AddProduct(AddProductRequest value) returns ProductIdResponse|error {

        //ADD PRODUCT DETAILS

        string product\_id = generateProductId();

        Product newProduct = {product\_name: value.product\_name,product\_price: value.product\_price,stock\_quantity: value.stock\_quantity, stock\_keeping\_unit: value.stock\_keeping\_unit,product\_id: "",status: ""};

        productTable.add(newProduct);

        ProductIdResponse response = {product\_id: product\_id};

        return response;

    }

    remote function UpdateProduct(UpdateProductRequest value) returns Empty|error {

        if product is Product{

            product.product\_name=value.product\_name;

            product.product\_price=value.product\_price;

            product.stock\_quantity=value.stock\_quantity;

            productTable.put(product);

        }

        Empty response = {message: "The Product is Updated Successfully"};

        return response;

    }

    remote function RemoveProduct(RemoveProductRequest value) returns UpdatedListResponse|error {

        UpdatedListResponse response = {

            product\_id : value.product\_id,

            product\_name : value.product\_name,

            product\_price : 100.0,//AMOUNT

            stock\_quantity : 10,//AMOUNT

            stock\_keeping\_unit : "N/A", //CONFIRM

            status : "Removed"

        };

        return response;

    }

    function authenticate(string user\_name ,string password) returns boolean|error {

        if user\_name == "Admin" && password == "adminpassword"{

            return true;

        }else {

            return false;

        }

    }

}

//GENERATE PRODUCT IDs

function generateProductId() returns string {

    return "product" + uuid:createType1AsString();

}