**V1**

import ballerina/grpc;

import ballerina/uuid;

import ballerina/sql;

import ballerina/mysql;

listener grpc:Listener grpcListener = new (9090);

mysql:Client dbClient = check new (host = "localost", user = "root", password = "password", database = "shopping\_db");

# Description.

@grpc:Descriptor {value: SHOPPING\_DESC}

service "Admin" on grpcListener {

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

        AddAdminResponse response = {

            user\_id: value.user\_id,

            role: value.role

        };

        return response;

    }

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

        boolean isAuthenticated = check authenticate(value.user\_name, value.password);

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

        AuthenticateAdminResponse response = {message : message};

        return response;

    }

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

        //ADD PRODUCT DETAILS FROM PRODUCT DATABASE

        string product\_id = generateProductId();

        sql:ExecutionResult result = check dbClient->execute(

            "INSERT INTO products (product\_name, product\_price, stock\_quantity, stock\_keeping\_unit) VALUES (?, ?, ?, ?)",

            value.product\_name, value.product\_price, value.stock\_quantity, value.stock\_keeping\_unit

        );

        ProductIdResponse response = {product\_id: generateProductId()};

        return response;

    }

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

        //ADD PRODUCT DETAILS FROM PRODUCT DATABASE

        sql:ExecutionResult result = check dbClient->execute(

            "UPDATE products SET product\_name = ?, product\_price=?, stock\_quantity=? where product\_id=?",

            value.product\_name,value.product\_price,value.stock\_quantity,value.product\_id

        );

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

        return response;

    }

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

        //REMOVE LOGIC

        sql:ExecutionResult result = check dbClient->execute(

            "DELETE FROM products WHERE product\_id = ?",

            value.product\_id,value.product\_name,value.status

        );

        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",

            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 "prod" + uuid:createTypeAsString{};

}