|  |  |
| --- | --- |
|  | **BAHRIA UNIVERSITY,**  **(Karachi Campus)**  *Department of Software Engineering*  **ASSIGNMENT#03 – Fall 2022**  **COMPLEX ENGINEERING PROBLEM** |

Course Title: **Web Engineering** Course Code: **SEN-310**

Class: **BSE-5(A,B)** Shift: **Morning**

Course Instructor: **Engr. M. Adnan Ur Rehman** Time Allowed:  **01 Week.**

Date: **02-Jan-2023** Max. Marks: **20 Points**

**Instructions:**

* This assignment is based on Complex Engineering Problem.
* Submit this assignment in a group consist up to 3 members. Submission is permitted on ***LMS***.
* **All** members of the group **MUST** upload the same solution in LMS.
* You have to submit a compressed file containing the following:
  1. ***This title page file*** **.docx** after filling the group names and output of your program
  2. ***Source files*** May include **.html** and **.js** files

|  |  |  |
| --- | --- | --- |
| **GROUP MEMBERS** | | |
| **S.No.** | **Student Name** | **Enrolment#** |
| 1 | Talha Zafar | 02-131202-080 |
| 2 | Muaz Shahzad | 02-131202-081 |
| 3 | Hassan Akhtar | 02-131202-019 |

**Task: *[CLO-3; 20 Points]***

1. Implement a RESTful API scenario having the following properties:

* User can manage the objects via CRUD operations.
* CRUD operations are supposed to be operated through HTTP verbs.

1. Write test cases along with test data and test results.

* You must use [Postman](https://www.postman.com) to test your API.

**Code:**

//Prepared By: TALHA ZAFAR , MUAZ SHAHAZAD , HASSAN AKHTAR

//CRUD USING REST API'S

//Tested in Postman

const express=require("express");

const app=express();

const port =8000;

app.use(express.json())

app.use(express.urlencoded({extended:false}))

var hotel = [

    { OrderId: 5, ItemName: "Pizza", Price: 999.9 },

    { OrderId: 10, ItemName: "Zinger", Price: 249.9 },

    { OrderId: 15, ItemName: "Fries", Price: 79.9 }

];

app.get("/hotel", function (req, res) {

    res.send(hotel)

});

app.post("/hotel", function (req, res) {

    console.log("req.query :" + JSON.stringify(req.query));

    console.log("req.body  :" + JSON.stringify(req.body));

    if (!req.body.ItemName || !req.body.Price) {

        res.status(404);

        res.send("\*\*\*INCOMPLETE Providing Data\*\*\*");

    } else {

        hotel.push({ OrderId: hotel.length + 1, ItemName: req.body.ItemName, Price: parseInt(req.body.Price) });

        res.send("\*\*Items Succesfully Added\*\*");

    }

});

app.delete("/hotel/:OrderId", function (req, res) {

    var std = hotel.find(function (stdt) {

        return stdt.OrderId == parseInt(req.params.OrderId);

    });

    console.log(req.params.OrderId);

    if (!std) {

        res.status(404);

        res.send("For Deletion! \*\*Items Not Found\*\*");

    }else{

        var index=hotel.indexOf(std);

        console.log(hotel.splice(index, 1));

        res.send(JSON.stringify(hotel));

    }

});

    app.put("/hotel/:OrderId", function (req, res) {

    var std = hotel.find(function (stdt) {

        return stdt.OrderId == parseInt(req.params.OrderId);

    });

    if (!std) {

        res.status(404);

        res.send(" For Deletion! \*\*Items Not Found\*\*");

    }else{

        std.ItemName=req.body.ItemName;

        std.Price=req.body.Price;

        res.send(JSON.stringify(hotel));

    }

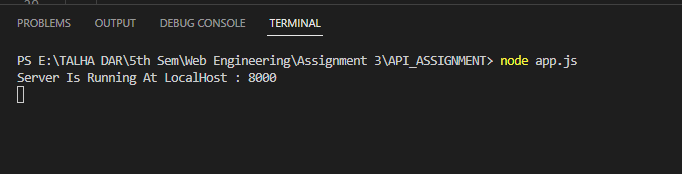
});

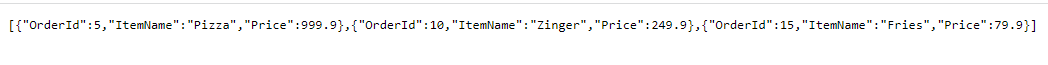
app.listen(port,()=>{

    console.log(`Server Is Running At LocalHost : ${port}`)

});

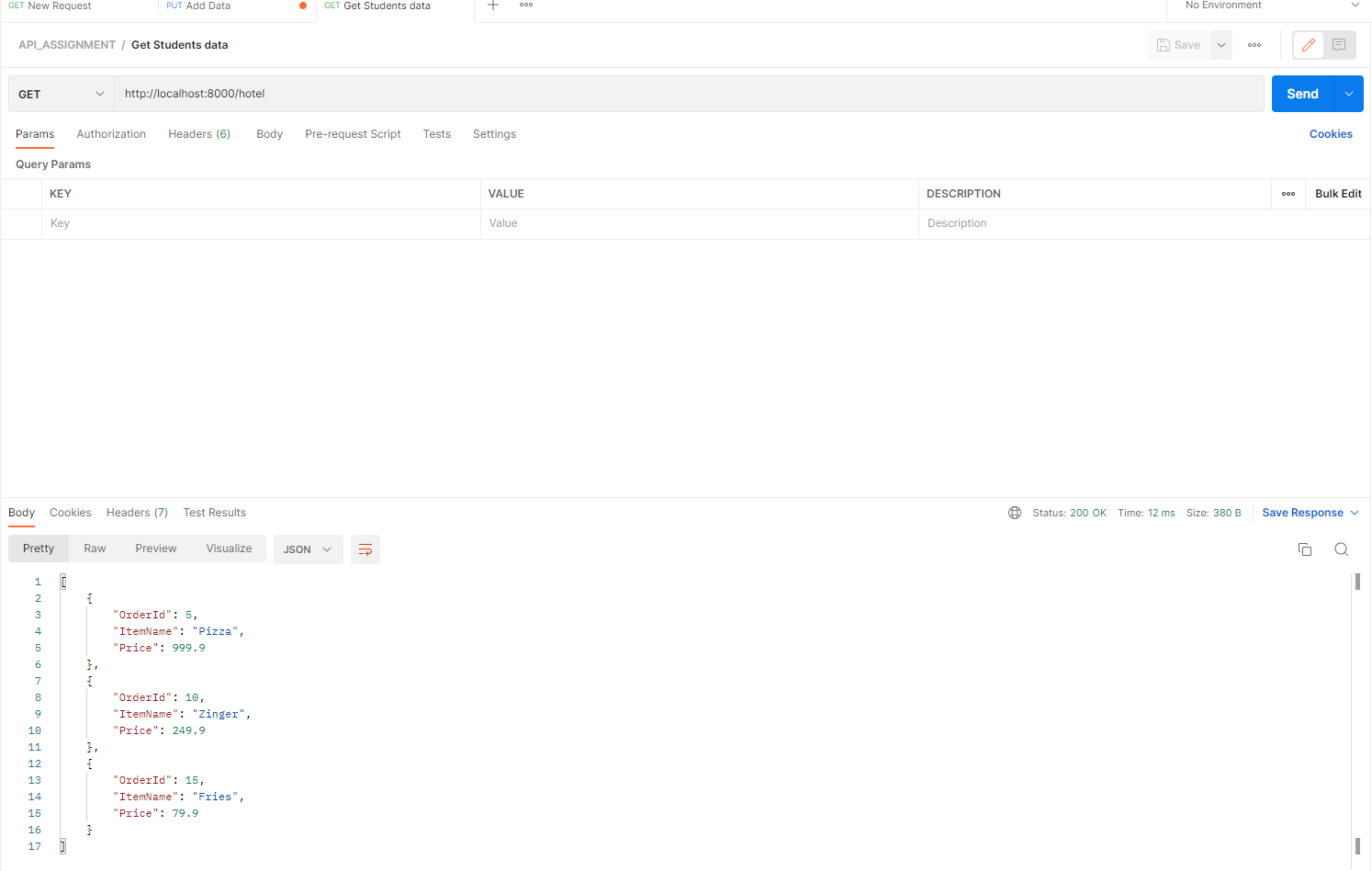
**Output**:



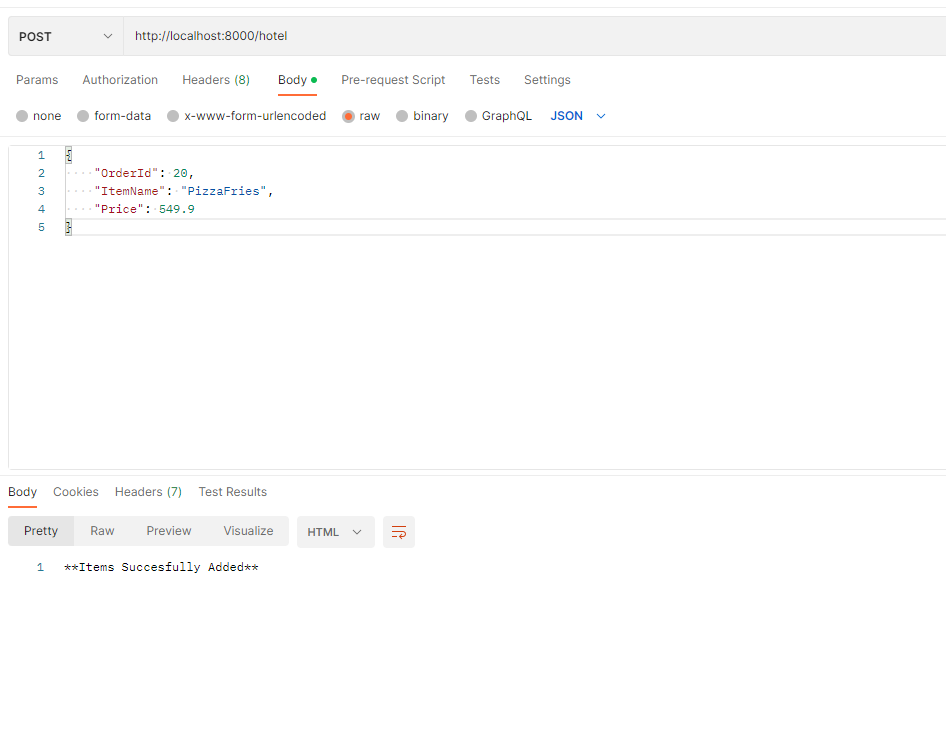
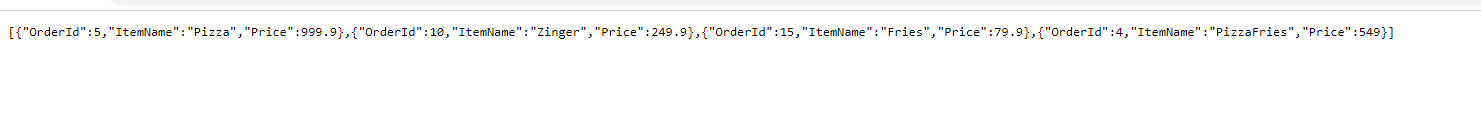


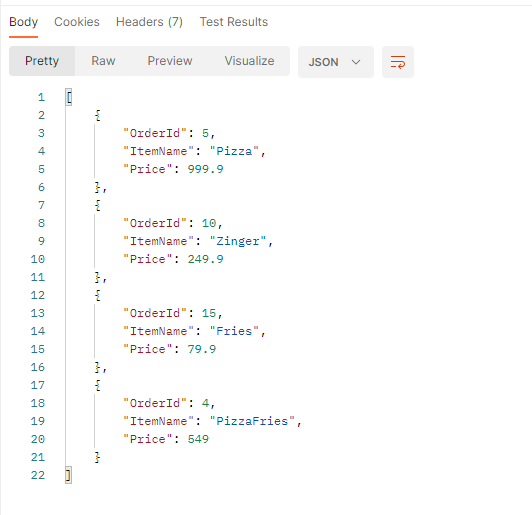
POSTMAN:

GET:

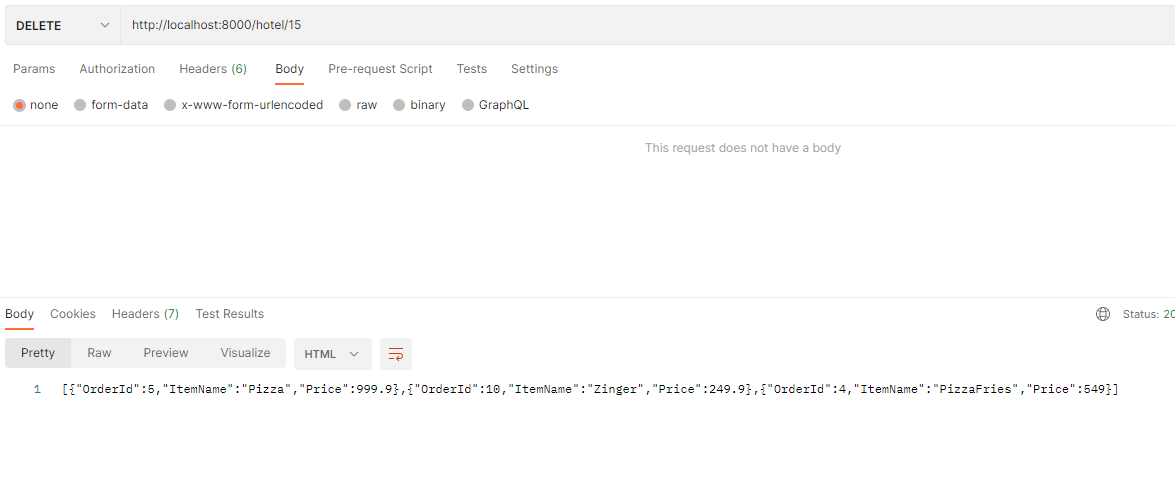


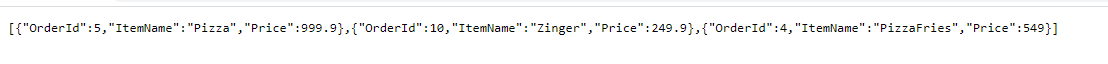
POST:



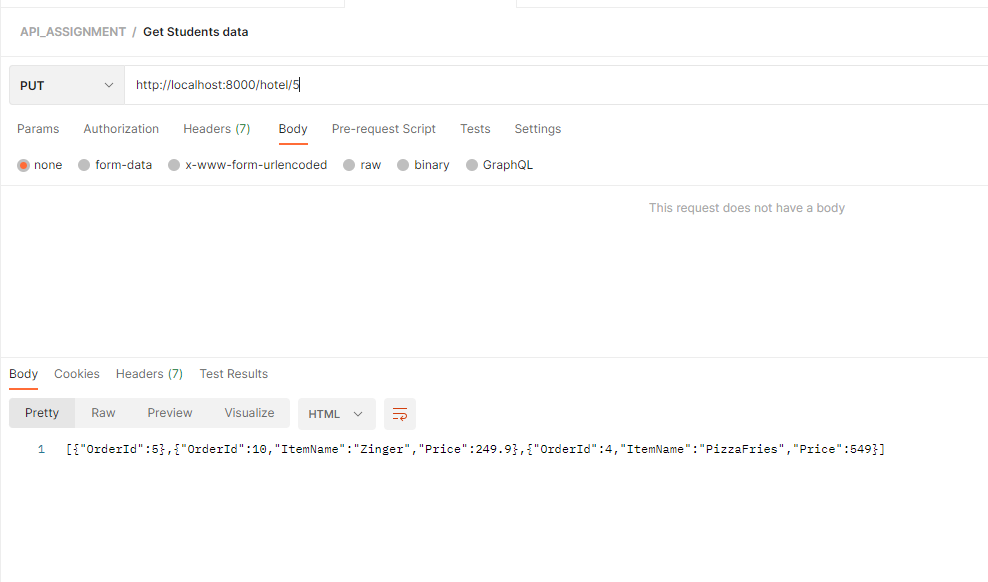


DELETE:





PUT:



THE END