

## CS393 - Homework (2)

Assignment Date : 25.11.2021

Deadline : 2.12.2021 11.00 PM

- This assessment must be prepared and finalized individually.
- You must upload your project into LMS as an archive file.

### Homework Details:

You are responsible for developing Restful Web Services in this homework. You can continue with your project which you already developed for Homework-1. If you don't want to use your project, you can use my project. You can download it from my GitHub address.

### Main Rules:

- Your project's database configuration should be done correctly. All services should read, update, delete, or edit data from the database. Using dummy data is not allowed.
- Using **cascade** attribute for entity-to-entity relationship is **not allowed**. You should write custom query methods for getting shop's products, etc.
- You must follow the rest design & naming principles. You can refer to the week8 presentation and the following URL:

<https://restfulapi.net/resource-naming/>

- You should insert the necessary records into the database. For initializing database, you can develop "JUnit tests" or write basic SQL scripts:

<https://docs.spring.io/spring-boot/docs/2.5.7/reference/html/howto.html#howto.data-initialization>

- Write a simple text file (readme.txt) in your project and explain the steps on how I can run your application (eg: creating a database, adding records, starting the application, etc.)

### Develop following Restful Web Services:

1. Listing all shops. Your service must return only shop data, not related customers, or products, etc.

This is a sample response. Your actual response can contain more attributes:

```
[
  {
    "id": 4,
    "name": "SHOP-1"
  },
  {
    "id": 6,
    "name": "SHOP-2",
  }
]
```

2. Creating a new shop. Saving only shop data is enough.

Request Body:

```
{
  "name": "SHOP-3"
}
```

Response Body:

```
{
  "id": 8,
  "name": "SHOP-3"
}
```

3. Listing a **specific** shop and all its products that has price >200. The shop's ID must be provided as part of service URL. Your service should return both shop data and products.

```
{
  "id": 4,
  "name": "SHOP-1",
  "products": [
    {
      "id": 21,
      "name": "Lenovo E14",
      "price": "1350"
    },
    {
      "id": 15,
      "name": "Canon Camera",
      "price": "1100",
    }
  ]
}
```

4. Delete a **specific** shop. The shop's ID must be provided as part of service URL.