

Docker image with PHP and MySql Microservice

Name :- Harshal Mohan Patil

Roll no :- M-18221 (IMCA)

Department :- Mathematics

Git :- <https://github.com/harshal316/Docker.git>

Implement microservice (CRUD Operation) :

In this assignment I used PHP & MySql to create API's for CRUD operations.

I created database "USERS"

and created a table "Students".

Our table "Students" schema:

```
Student Object {  
    Id : Integer  
    Name : String  
    Age : Integer  
    Department : String  
    Subjects : String  
}
```

How to run the application :-

There are two ways

I. First Method :-

1.0 Pull image from my docker hub repository

:- `docker pull harshal316/crud-api`

II. Second Method :-

1. Firstly clone this repository in your machine

:- `git clone https://github.com/harshal316/Docker.git`

2. Now move to the docker directory in terminal

:- `cd Docker`

3. Build an image using Dockerfile

:- `docker build -t IMAGE_NAME .`

4. Run your image (note that we exposed port 80 in dockerfile)

:- docker run -itd -p 80:80 IMAGE_NAME

5. To see the running application in the browser

:- https://localhost:80/api

or:- https://localhost/api

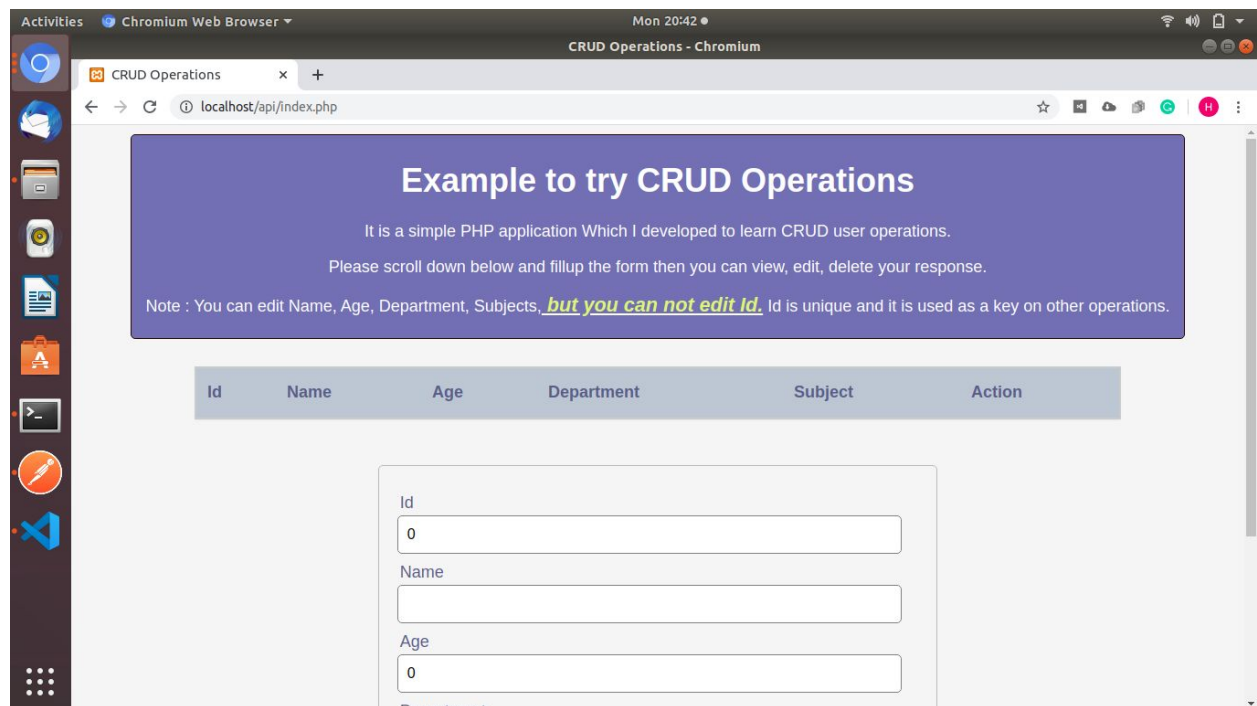
API Testing :-

0. Index Page (Optional)

:- To test the application is running successfully go to the following url.

<http://localhost/api/>

Probably it will not look good in Postman so try it in browser



1. CREATE :-

To test the create api, open POSTMAN and enter the following as the request URL.

Select Method "**POST**".

<http://localhost/api/product/create.php>

- Click the "Body" tab.
- Click "raw".
- Enter this JSON value:

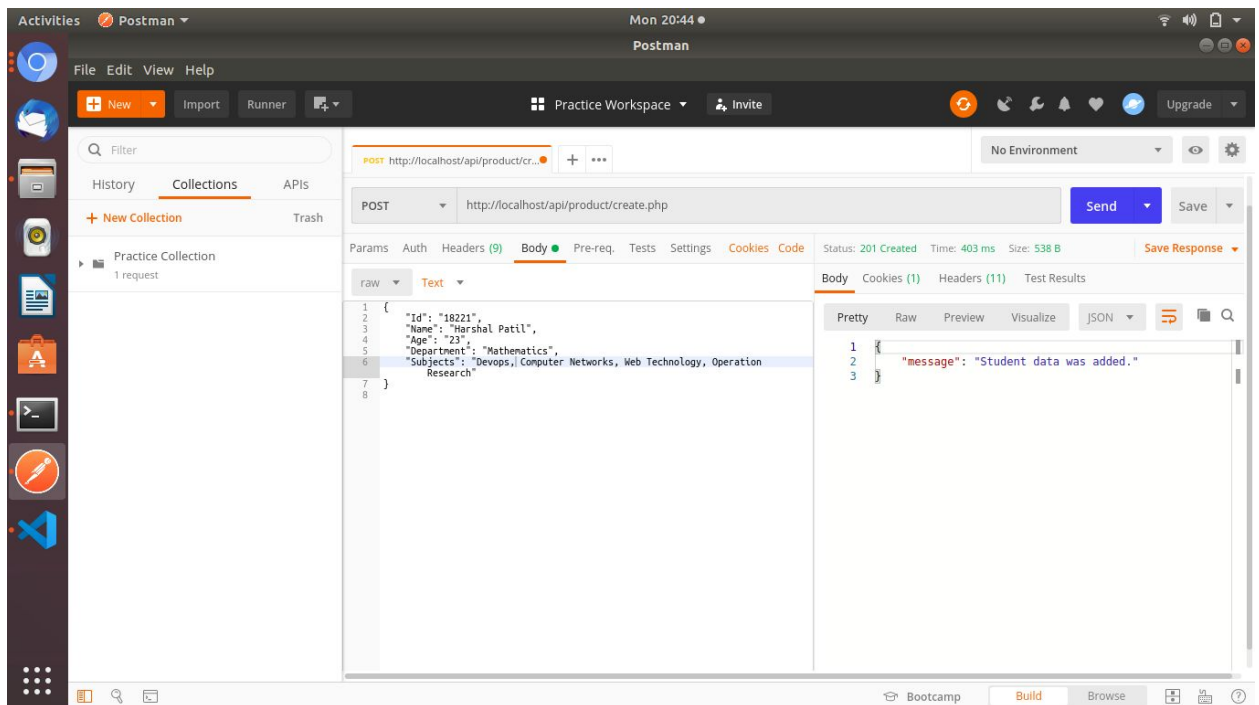
{

```

    "Id": "18221",
    "Name": "Harshal Patil",
    "Age": "23",
    "Department": "Mathematics",
    "Subjects": "Devops, Computer Networks, Web Technology, Operation Research"
}

```

- Click the blue 'Send' button
- It should look like this



- If the Id is already present in the database it gives the following message.


```

{
  "message": "Unable to add data. Student Id is already present"
}

```
- If the data is incomplete it gives following message


```

{
  "message": "Unable to add data. Data is incomplete."
}

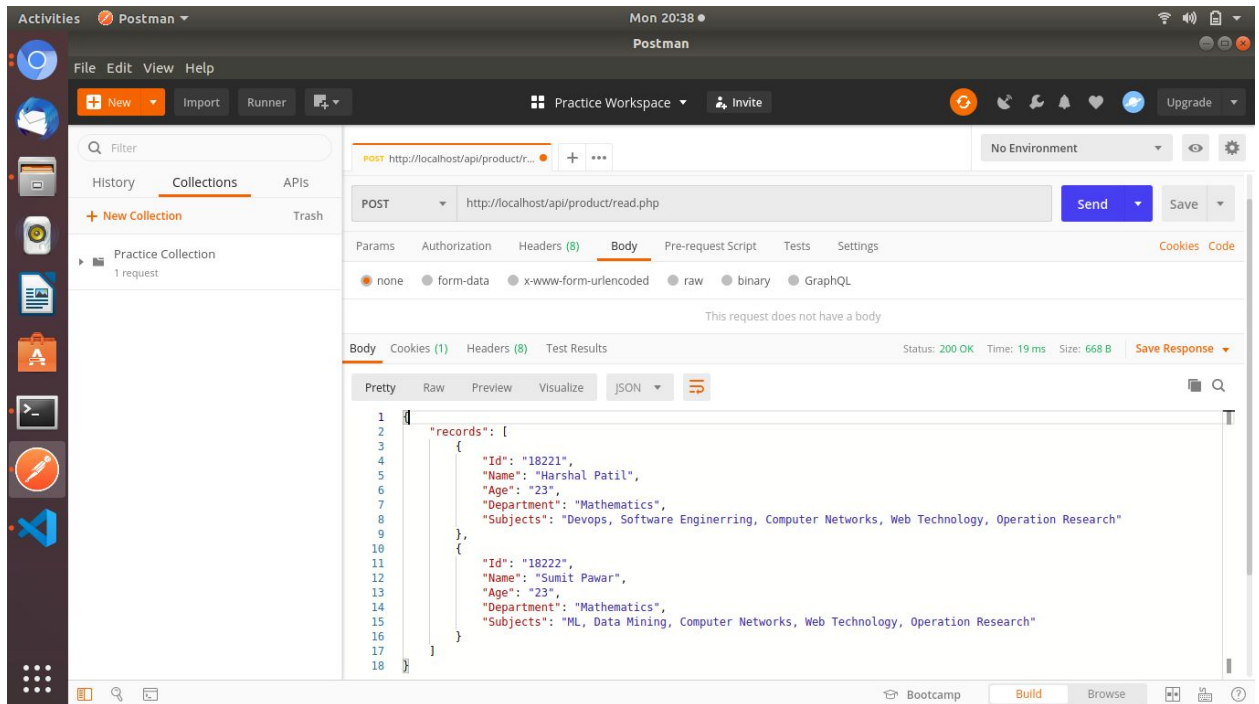
```

2. READ :-

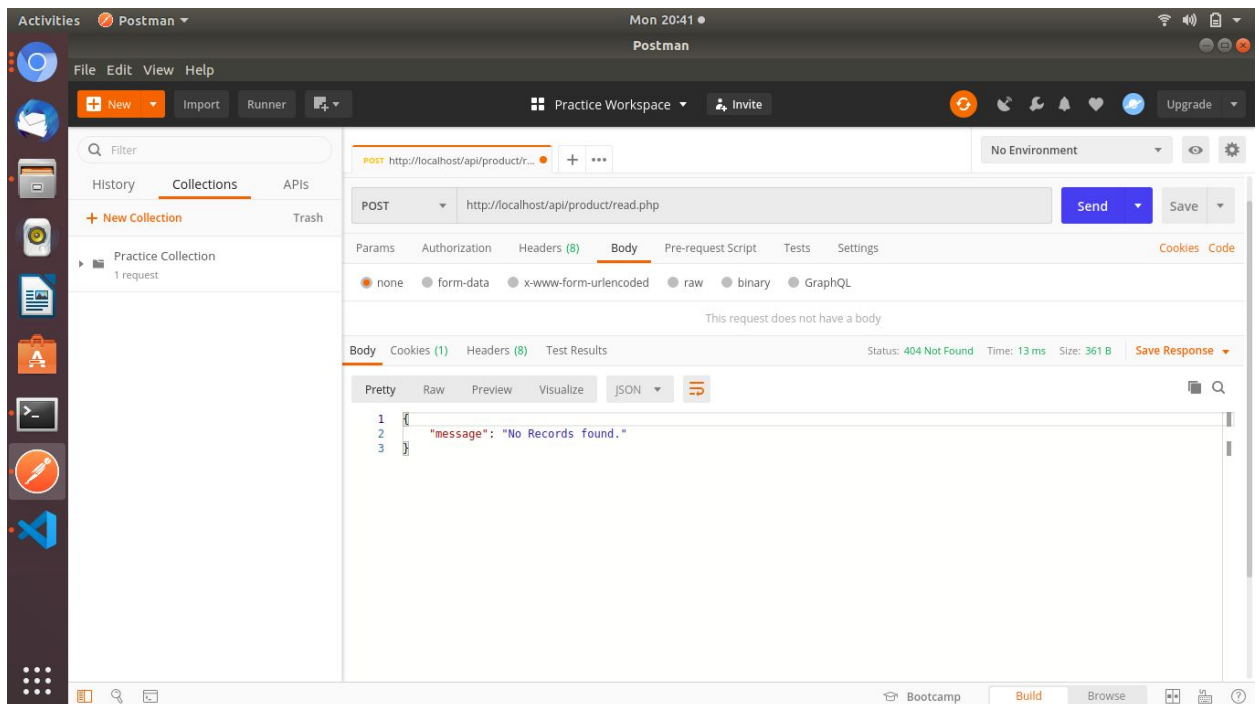
To test the read api, open POSTMAN and enter the following as the request URL. Select Method “**GET**”.

<http://localhost/api/product/read.php>

- Click the blue 'Send' button
- Output if students records are present in database :-



- If student record is not present :-



3. UPDATE :-

To test the update api, open POSTMAN and enter the following as the request URL.

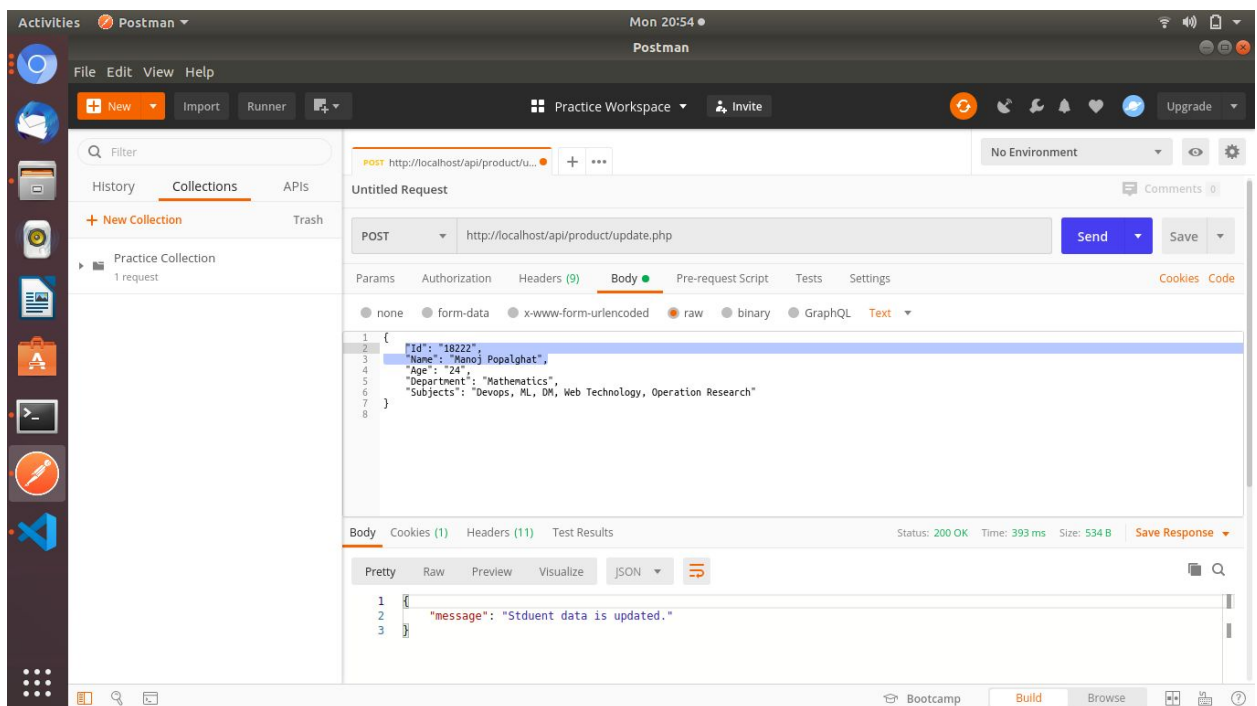
Select Method **"PUT"**.

<http://localhost/api/product/update.php>

- Click the "Body" tab.
- Click "raw".
- Enter this JSON value:

```
{
  "Id": "18222",
  "Name": "Manoj Popalghat",
  "Age": "24",
  "Department": "Mathematics",
  "Subjects": "Devops, ML, DM, Web Technology, Operation Research"
}
```

- Make sure the Id exist in the database
- Then click the blue 'Send' button
- Output look like :-



- If you specify an ID that does not exist in the database, it might still say that the data is updated. It does not update anything on the database but the query was executed successfully without any syntax errors.

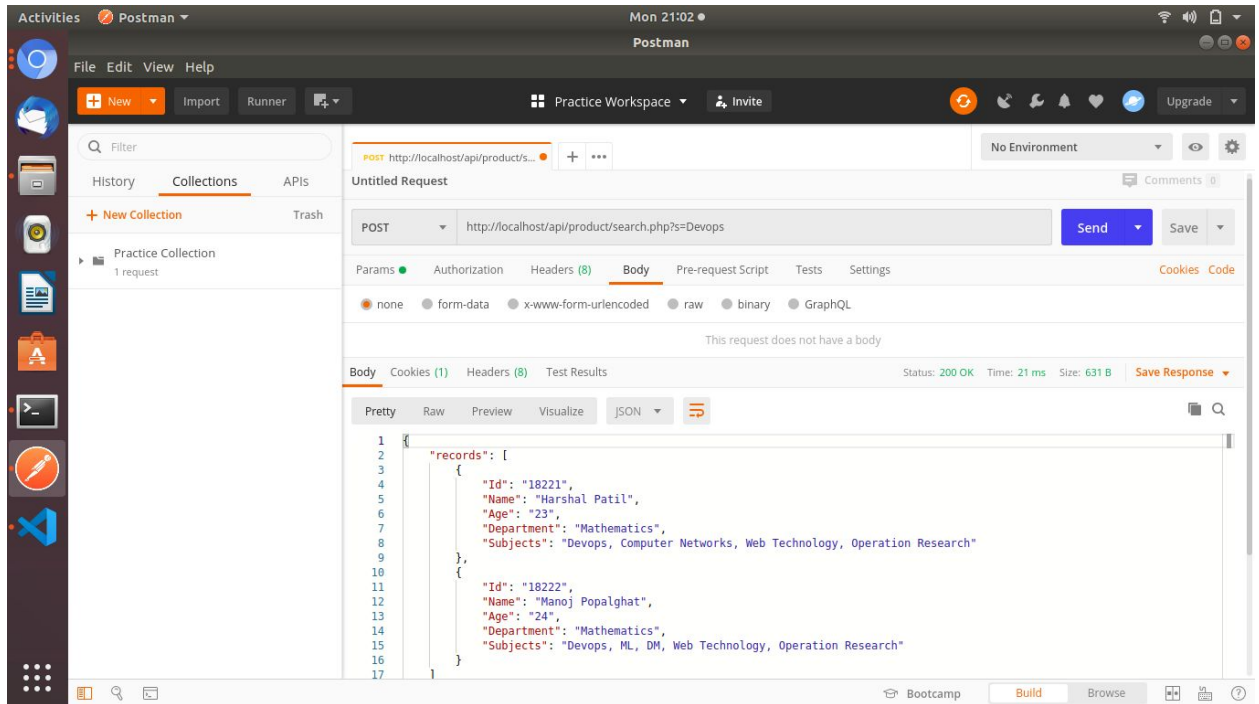
4. SEARCH :-

To test the search api, open POSTMAN and enter the following as the request URL.

Select Method “GET”.

<http://localhost/api/product/search.php?s=Devops>

- If there is data present for search key :-



- If data is not exist for search key it gives follwing message :-

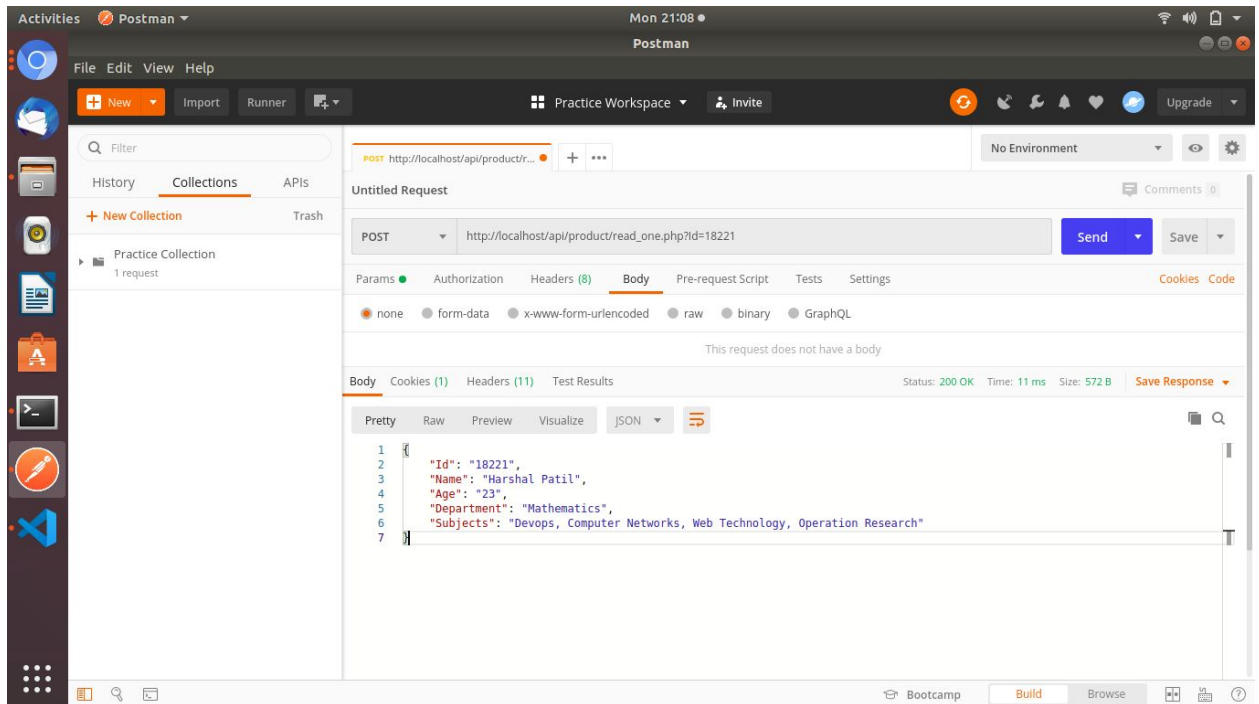
```
{
  "message": "No Records found."
}
```

5. READ_ONE :-

To read the data of a specific student by Id, open POSTMAN and enter the following as the request URL. Select Method “GET”.

http://localhost/api/product/read_one.php?Id=18221

- If data is present for that Id :-



- If data is not present for that Id :-

```
{
  "message": "Studnet Id does not exist."
}
```

6. DELETE :-

To test the delete api, open POSTMAN and enter the following as the request URL.
Select Method “**DELETE**”.

<http://localhost/api/product/delete.php>

- Click the "Body" tab.
- Click "raw".
- Enter this JSON value:

```
{
  "Id" : "18222"
}
```
- Make sure the Id exist in the database
- Then click the blue ‘Send’ button
- Output look like :-

