Docker Assignment

[Github link : https://github.com/piyu2409/Dockerassignment.git]

Software/Plugins/Language Used :

- Python
- Flask Web FrameWork
- Postman
- Mysql
- Postman
- Docker

Implementation :

- I have done all installations.
  e.g. flask and its depenedencies, postman

- Create the folder Structure in that one folder contain REST API(CRUD) operations and another folder contain mysql file. In first folder i have write all configuration and business logic for REST API. In second folder i have write sql file to create database and table.

- Write Dockerfile which contain all code and mysql.

- Build Dockerfile :

  ```
  sudo docker build -t sample .
  ```

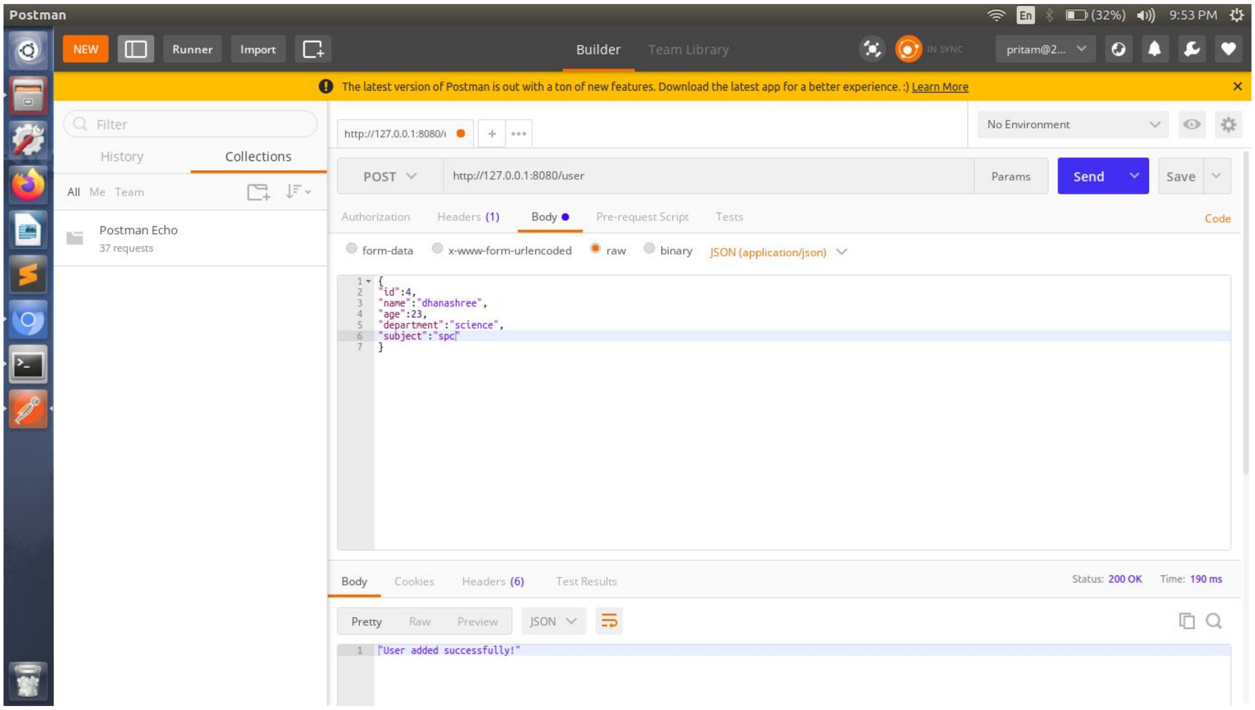
  Run Dockerfile :

  ```
  sudo docker run -it -p 8080:8080 sample
  ```

- In postman send the request -> POST,GET, PUT,DELETE

  POST = Create data
  GET = Read data
  PUT = Update data
  DELETE = Delete data

POST input :



POST http://127.0.0.1:8080/user

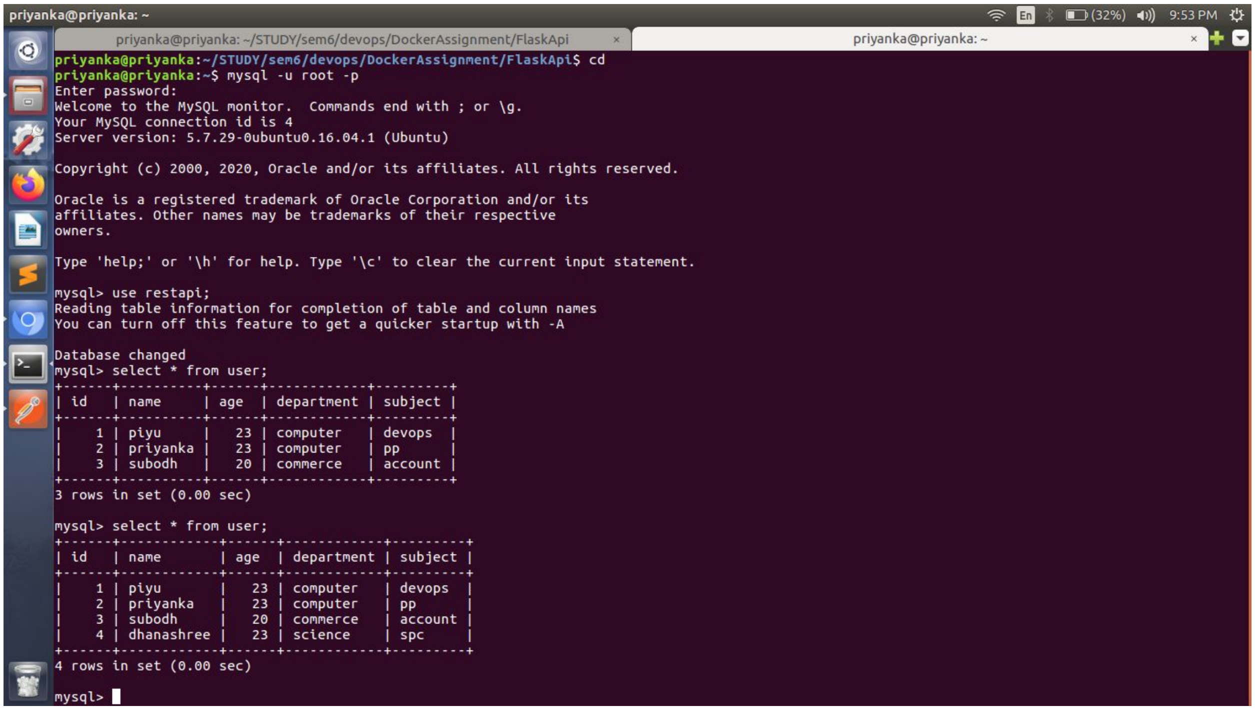
Body: raw JSON (application/json)

```
{
"id":4,
"name":"dhanashree",
"age":23,
"department":"science",
"subject":"spc"
}
```

Status: 200 OK Time: 190 ms

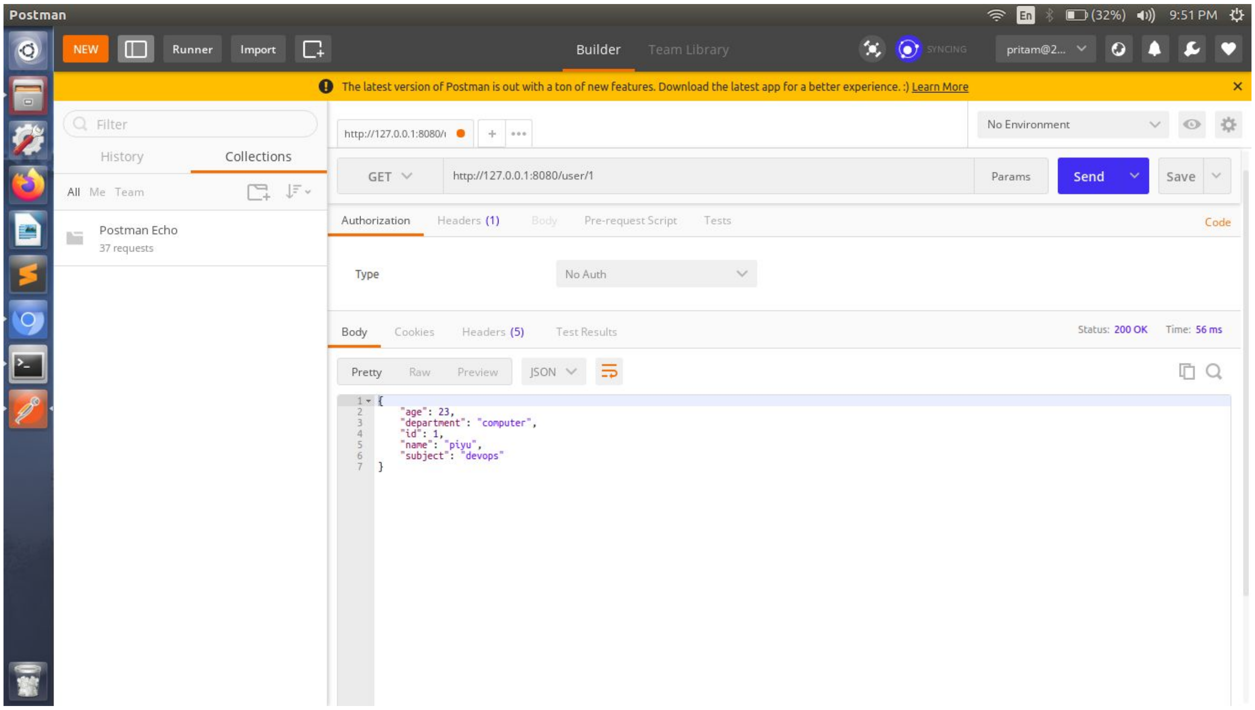
```
"User added successfully!"
```

POST Output :


priyanka@priyanka: ~
En
(32%)
9:53 PM
priyanka@priyanka: ~/STUDY/sem6/devops/DockerAssignment/FlaskApi
priyanka@priyanka: ~

```
priyanka@priyanka:~/STUDY/sem6/devops/DockerAssignment/FlaskApi$ cd
priyanka@priyanka:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.29-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use restapi;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from user;
+------+----------+------+------------+---------+
| id   | name     | age  | department | subject |
+------+----------+------+------------+---------+
|    1 | piyu     |   23 | computer   | devops  |
|    2 | priyanka |   23 | computer   | pp      |
|    3 | subodh   |   20 | commerce   | account |
+------+----------+------+------------+---------+
3 rows in set (0.00 sec)

mysql> select * from user;
+------+------------+------+------------+---------+
| id   | name       | age  | department | subject |
+------+------------+------+------------+---------+
|    1 | piyu       |   23 | computer   | devops  |
|    2 | priyanka   |   23 | computer   | pp      |
|    3 | subodh     |   20 | commerce   | account |
|    4 | dhanashree |   23 | science    | spc     |
+------+------------+------+------------+---------+
4 rows in set (0.00 sec)

mysql> 
```

GET Input :



The latest version of Postman is out with a ton of new features. Download the latest app for a better experience. :) Learn More

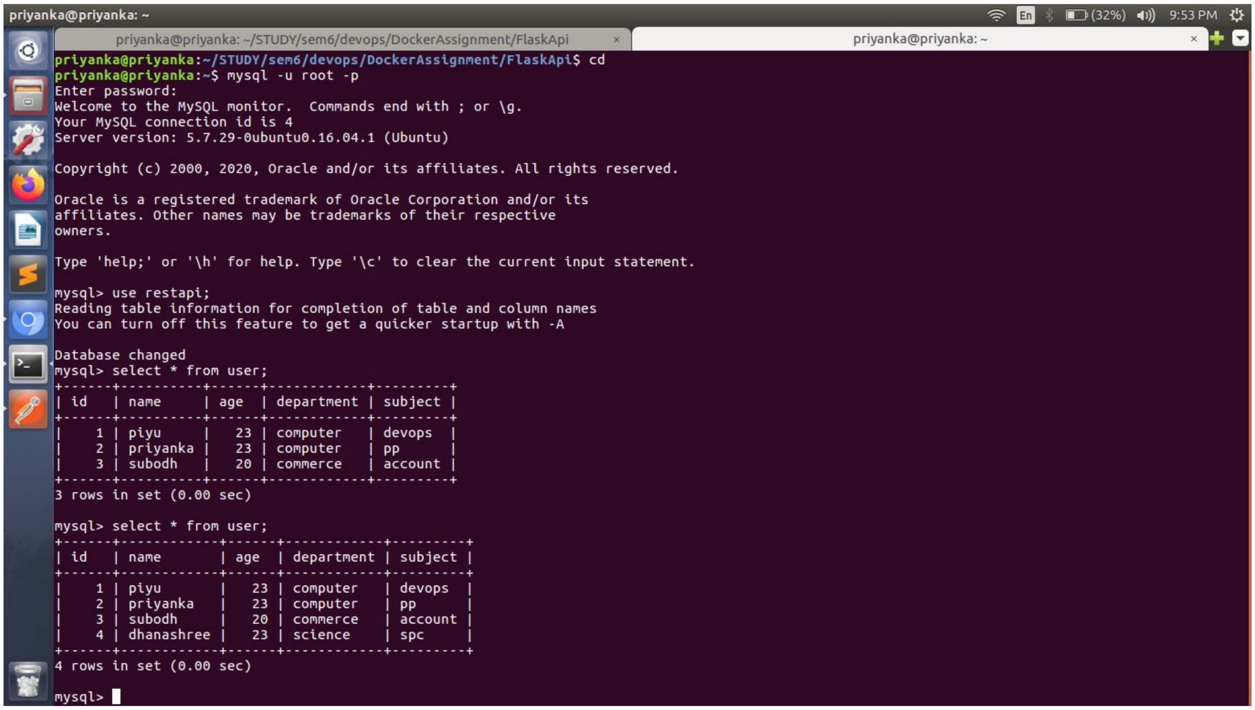
GET http://127.0.0.1:8080/user/1

Type: No Auth

Status: 200 OK Time: 56 ms

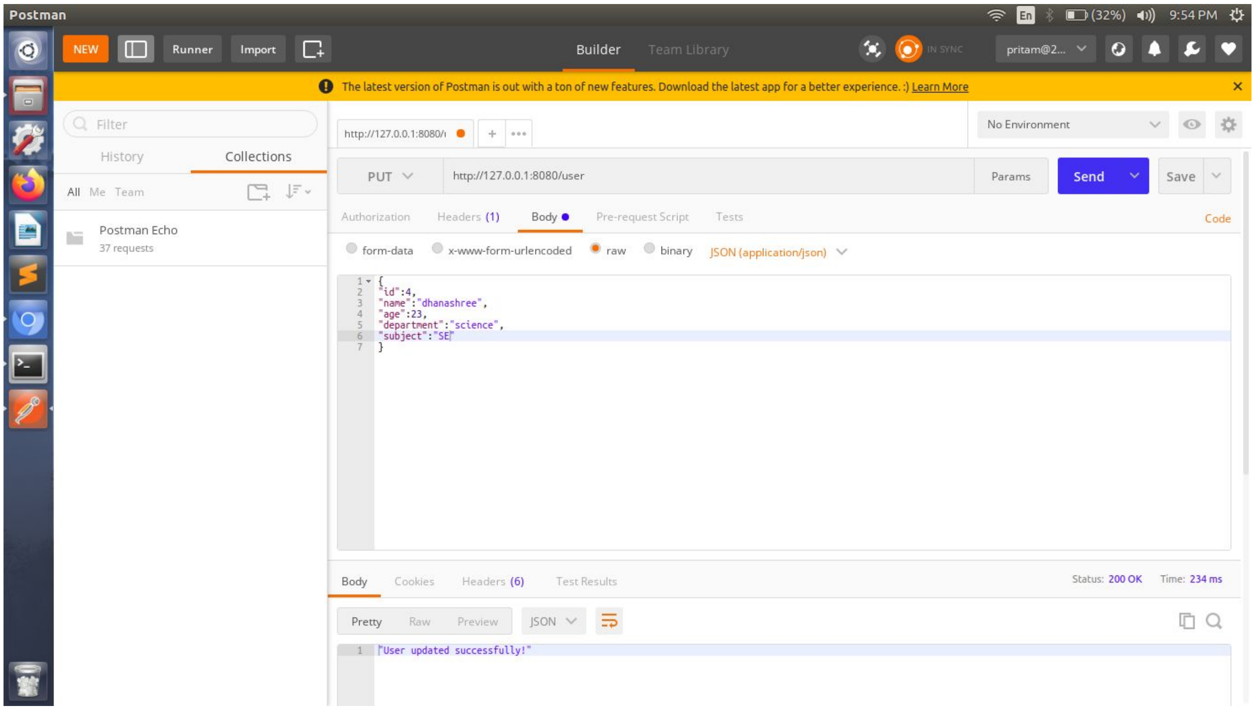
```
{
    "age": 23,
    "department": "computer",
    "id": 1,
    "name": "piyu",
    "subject": "devops"
}
```

GET Output :


priyanka@priyanka: ~
En
(32%)
9:53 PM
priyanka@priyanka: ~/STUDY/sem6/devops/DockerAssignment/FlaskApi
priyanka@priyanka: ~

```
priyanka@priyanka:~/STUDY/sem6/devops/DockerAssignment/FlaskApi$ cd
priyanka@priyanka:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.29-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use restapi;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from user;
+------+----------+------+------------+---------+
| id   | name     | age  | department | subject |
+------+----------+------+------------+---------+
|    1 | piyu     |   23 | computer   | devops  |
|    2 | priyanka |   23 | computer   | pp      |
|    3 | subodh   |   20 | commerce   | account |
+------+----------+------+------------+---------+
3 rows in set (0.00 sec)

mysql> select * from user;
+------+------------+------+------------+---------+
| id   | name       | age  | department | subject |
+------+------------+------+------------+---------+
|    1 | piyu       |   23 | computer   | devops  |
|    2 | priyanka   |   23 | computer   | pp      |
|    3 | subodh     |   20 | commerce   | account |
|    4 | dhanashree |   23 | science    | spc     |
+------+------------+------+------------+---------+
4 rows in set (0.00 sec)

mysql> 
```

PUT Input :



PUT http://127.0.0.1:8080/user

○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON (application/json)

```
{
"id":4,
"name":"dhanashree",
"age":23,
"department":"science",
"subject":"SE"
}
```

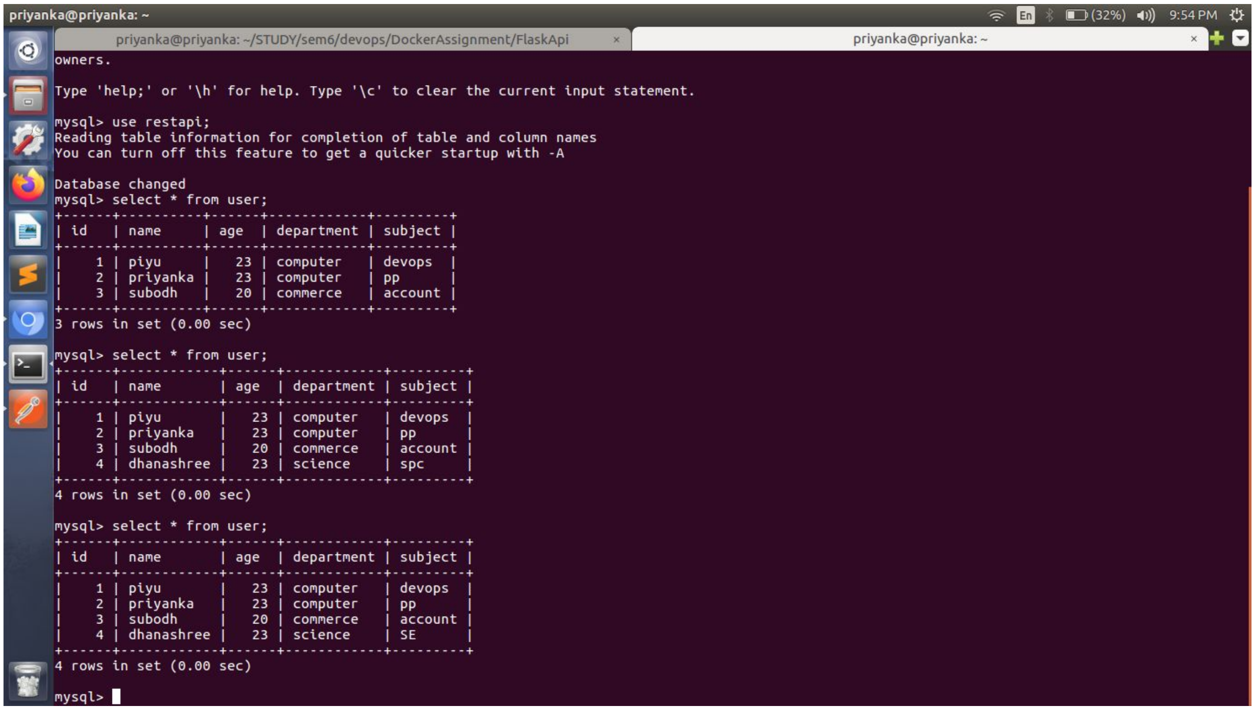
Body

Status: 200 OK Time: 234 ms

JSON

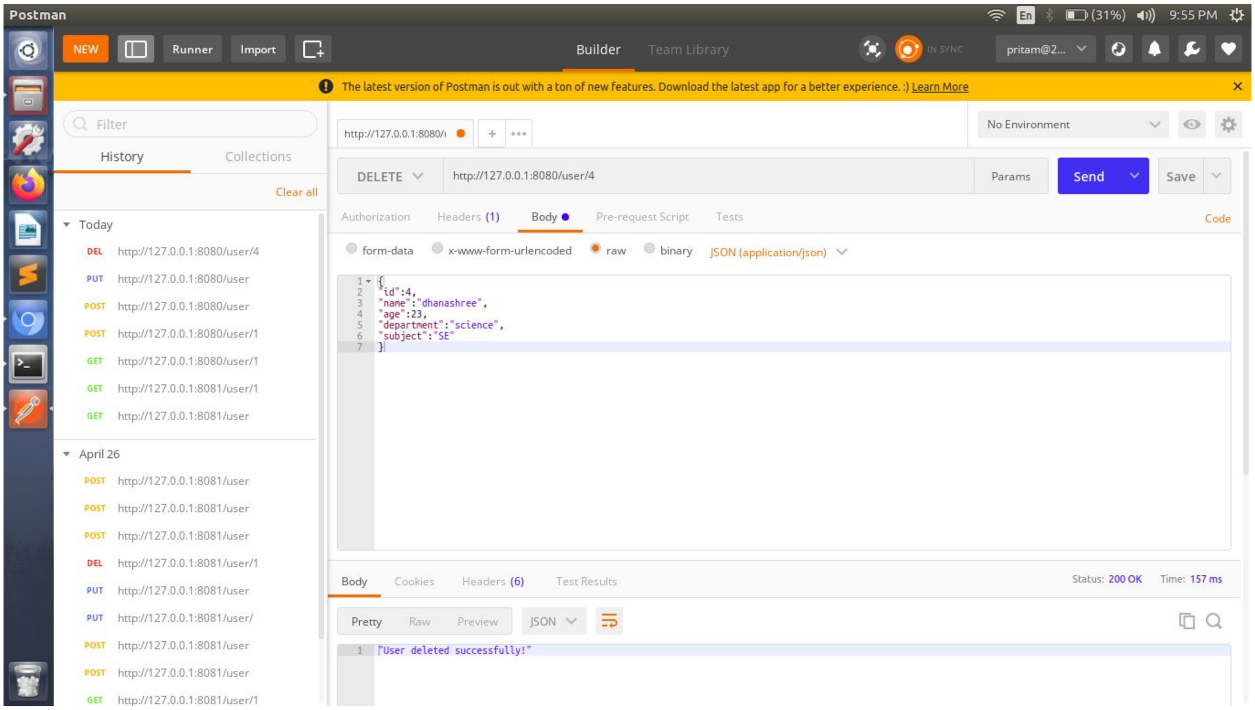
```
"User updated successfully!"
```

PUT Output :


priyanka@priyanka: ~
En
(32%)
9:54 PM
priyanka@priyanka: ~/STUDY/sem6/devops/DockerAssignment/FlaskApi
priyanka@priyanka: ~

```
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use restapi;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from user;
+------+----------+------+------------+---------+
| id   | name     | age  | department | subject |
+------+----------+------+------------+---------+
|    1 | piyu     |   23 | computer   | devops  |
|    2 | priyanka |   23 | computer   | pp      |
|    3 | subodh   |   20 | commerce   | account |
+------+----------+------+------------+---------+
3 rows in set (0.00 sec)

mysql> select * from user;
+------+------------+------+------------+---------+
| id   | name       | age  | department | subject |
+------+------------+------+------------+---------+
|    1 | piyu       |   23 | computer   | devops  |
|    2 | priyanka   |   23 | computer   | pp      |
|    3 | subodh     |   20 | commerce   | account |
|    4 | dhanashree |   23 | science    | spc     |
+------+------------+------+------------+---------+
4 rows in set (0.00 sec)

mysql> select * from user;
+------+------------+------+------------+---------+
| id   | name       | age  | department | subject |
+------+------------+------+------------+---------+
|    1 | piyu       |   23 | computer   | devops  |
|    2 | priyanka   |   23 | computer   | pp      |
|    3 | subodh     |   20 | commerce   | account |
|    4 | dhanashree |   23 | science    | SE      |
+------+------------+------+------------+---------+
4 rows in set (0.00 sec)

mysql> 
```

DELETE Input :



The latest version of Postman is out with a ton of new features. Download the latest app for a better experience. :) Learn More

Today
- DEL http://127.0.0.1:8080/user/4
- PUT http://127.0.0.1:8080/user
- POST http://127.0.0.1:8080/user
- POST http://127.0.0.1:8080/user/1
- GET http://127.0.0.1:8080/user/1
- GET http://127.0.0.1:8081/user/1
- GET http://127.0.0.1:8081/user

April 26
- POST http://127.0.0.1:8081/user
- POST http://127.0.0.1:8081/user
- POST http://127.0.0.1:8081/user
- DEL http://127.0.0.1:8081/user/1
- PUT http://127.0.0.1:8081/user
- PUT http://127.0.0.1:8081/user/
- POST http://127.0.0.1:8081/user
- POST http://127.0.0.1:8081/user
- GET http://127.0.0.1:8081/user/1

DELETE http://127.0.0.1:8080/user/4

Body: ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON (application/json)

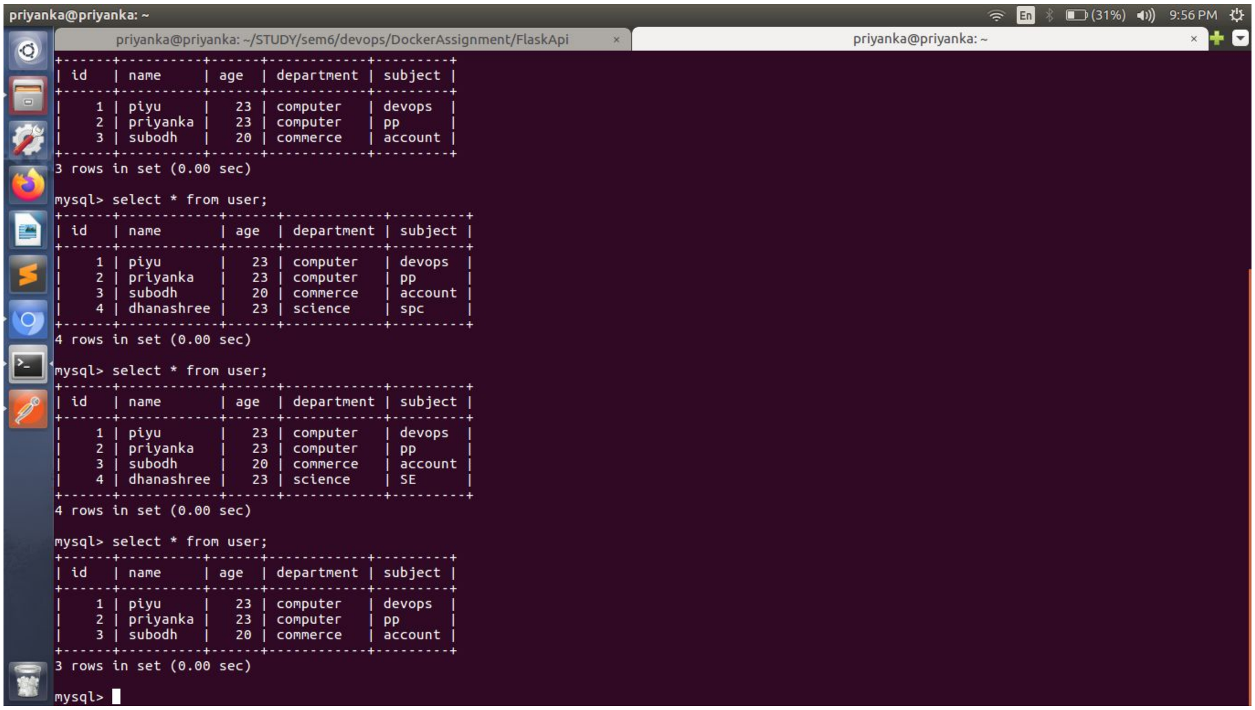
```
{
"id":4,
"name":"dhanashree",
"age":23,
"department":"science",
"subject":"SE"
}
```

Body — Status: 200 OK Time: 157 ms

JSON

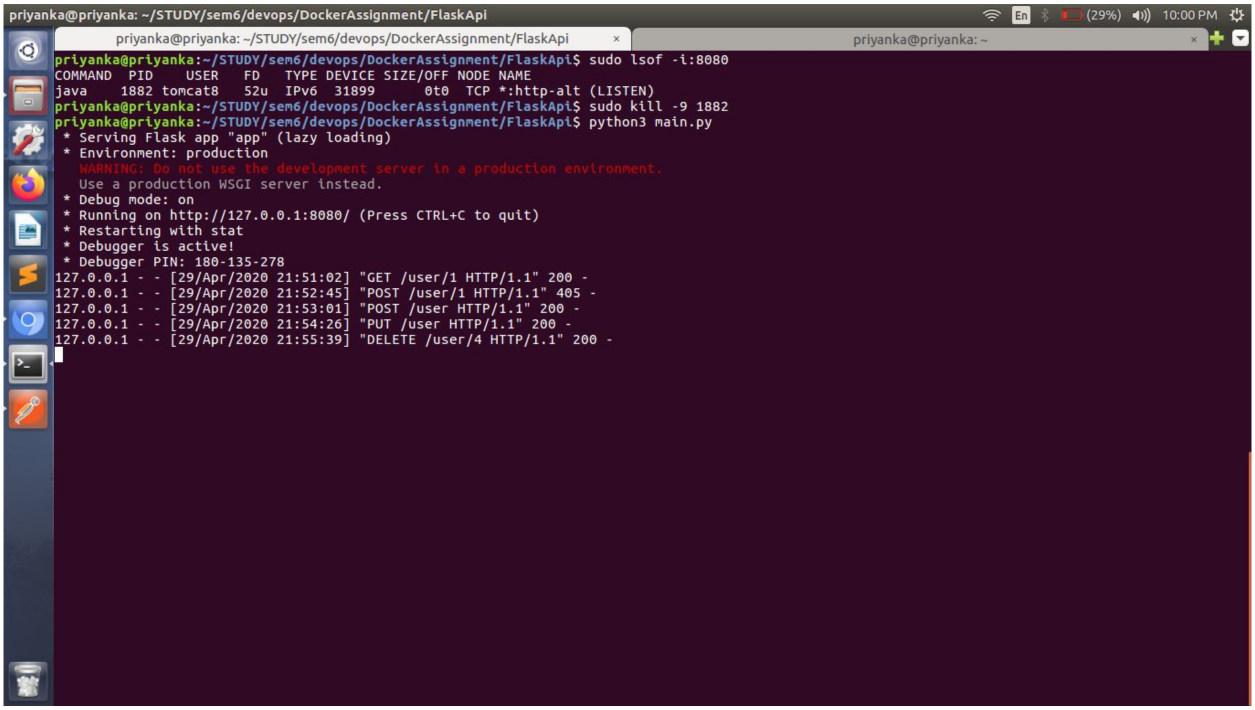
```
"User deleted successfully!"
```

DELETE Output :


priyanka@priyanka: ~
priyanka@priyanka: ~/STUDY/sem6/devops/DockerAssignment/FlaskApi
priyanka@priyanka: ~
En
(31%)
9:56 PM

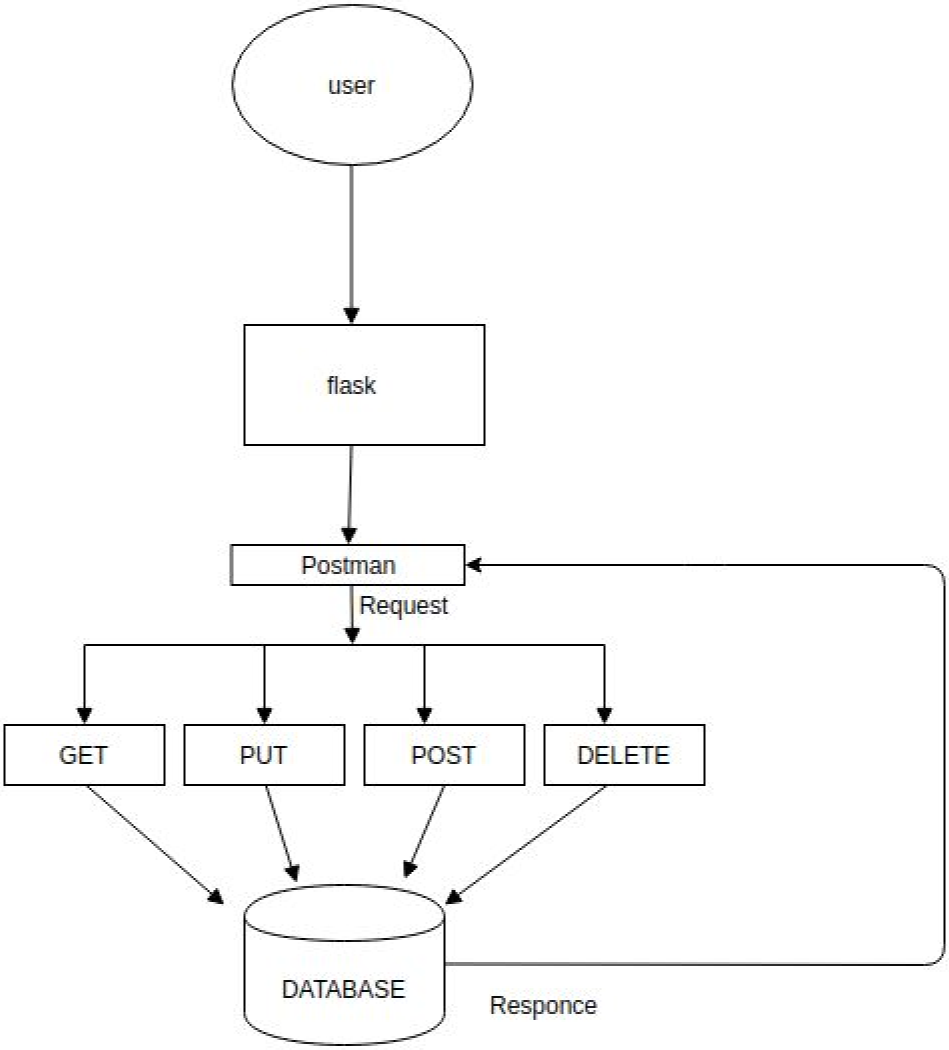
```
+------+----------+------+------------+---------+
| id   | name     | age  | department | subject |
+------+----------+------+------------+---------+
|    1 | piyu     |   23 | computer   | devops  |
|    2 | priyanka |   23 | computer   | pp      |
|    3 | subodh   |   20 | commerce   | account |
+------+----------+------+------------+---------+
3 rows in set (0.00 sec)

mysql> select * from user;
+------+------------+------+------------+---------+
| id   | name       | age  | department | subject |
+------+------------+------+------------+---------+
|    1 | piyu       |   23 | computer   | devops  |
|    2 | priyanka   |   23 | computer   | pp      |
|    3 | subodh     |   20 | commerce   | account |
|    4 | dhanashree |   23 | science    | spc     |
+------+------------+------+------------+---------+
4 rows in set (0.00 sec)

mysql> select * from user;
+------+------------+------+------------+---------+
| id   | name       | age  | department | subject |
+------+------------+------+------------+---------+
|    1 | piyu       |   23 | computer   | devops  |
|    2 | priyanka   |   23 | computer   | pp      |
|    3 | subodh     |   20 | commerce   | account |
|    4 | dhanashree |   23 | science    | SE      |
+------+------------+------+------------+---------+
4 rows in set (0.00 sec)

mysql> select * from user;
+------+----------+------+------------+---------+
| id   | name     | age  | department | subject |
+------+----------+------+------------+---------+
|    1 | piyu     |   23 | computer   | devops  |
|    2 | priyanka |   23 | computer   | pp      |
|    3 | subodh   |   20 | commerce   | account |
+------+----------+------+------------+---------+
3 rows in set (0.00 sec)

mysql> 
```

REST API :


priyanka@priyanka: ~/STUDY/sem6/devops/DockerAssignment/FlaskApi
priyanka@priyanka: ~/STUDY/sem6/devops/DockerAssignment/FlaskApi
priyanka@priyanka: ~
En
(29%)
10:00 PM

```
priyanka@priyanka:~/STUDY/sem6/devops/DockerAssignment/FlaskApi$ sudo lsof -i:8080
COMMAND  PID    USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
java    1882 tomcat8   52u  IPv6  31899      0t0  TCP *:http-alt (LISTEN)
priyanka@priyanka:~/STUDY/sem6/devops/DockerAssignment/FlaskApi$ sudo kill -9 1882
priyanka@priyanka:~/STUDY/sem6/devops/DockerAssignment/FlaskApi$ python3 main.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 180-135-278
127.0.0.1 - - [29/Apr/2020 21:51:02] "GET /user/1 HTTP/1.1" 200 -
127.0.0.1 - - [29/Apr/2020 21:52:45] "POST /user/1 HTTP/1.1" 405 -
127.0.0.1 - - [29/Apr/2020 21:53:01] "POST /user HTTP/1.1" 200 -
127.0.0.1 - - [29/Apr/2020 21:54:26] "PUT /user HTTP/1.1" 200 -
127.0.0.1 - - [29/Apr/2020 21:55:39] "DELETE /user/4 HTTP/1.1" 200 -
```

FlowDiagram :



******************************** Thanks *********************************