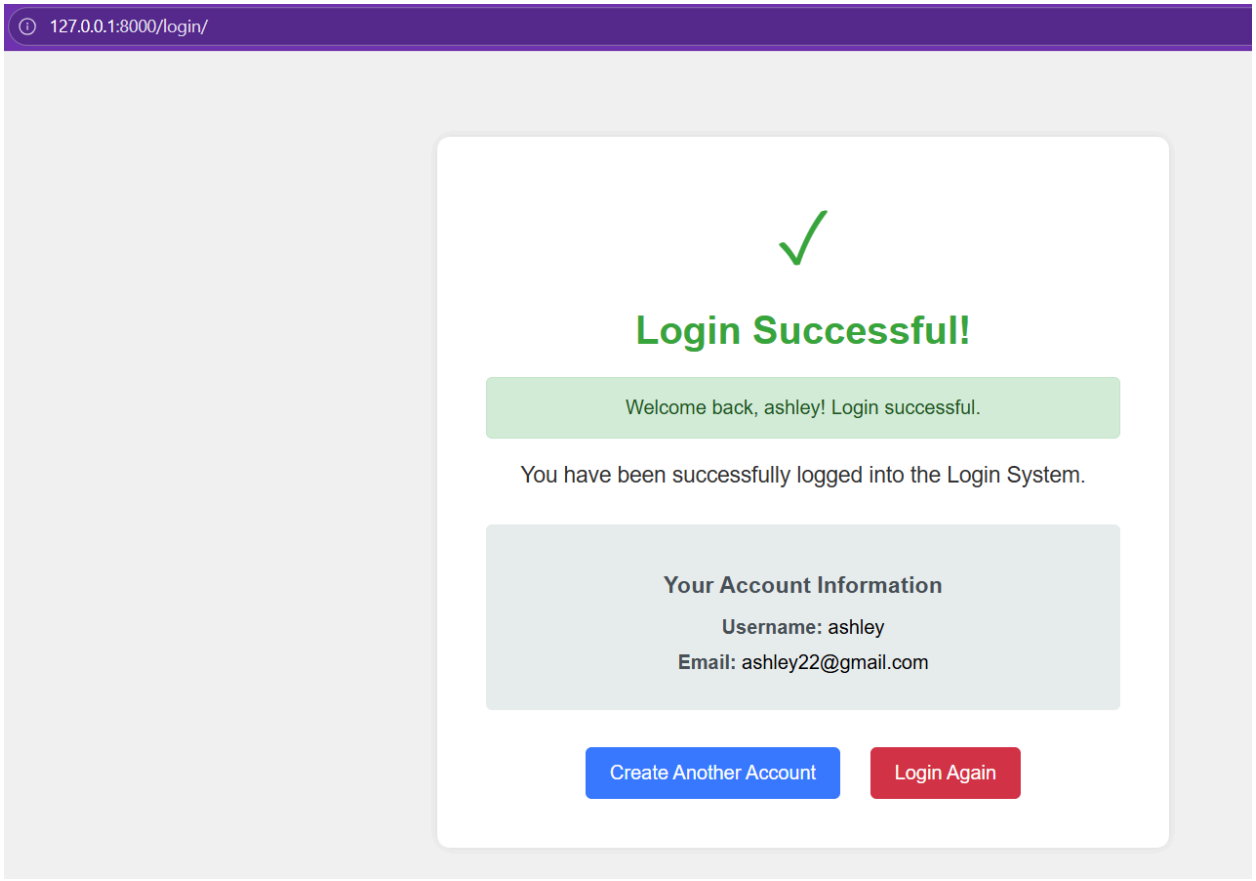
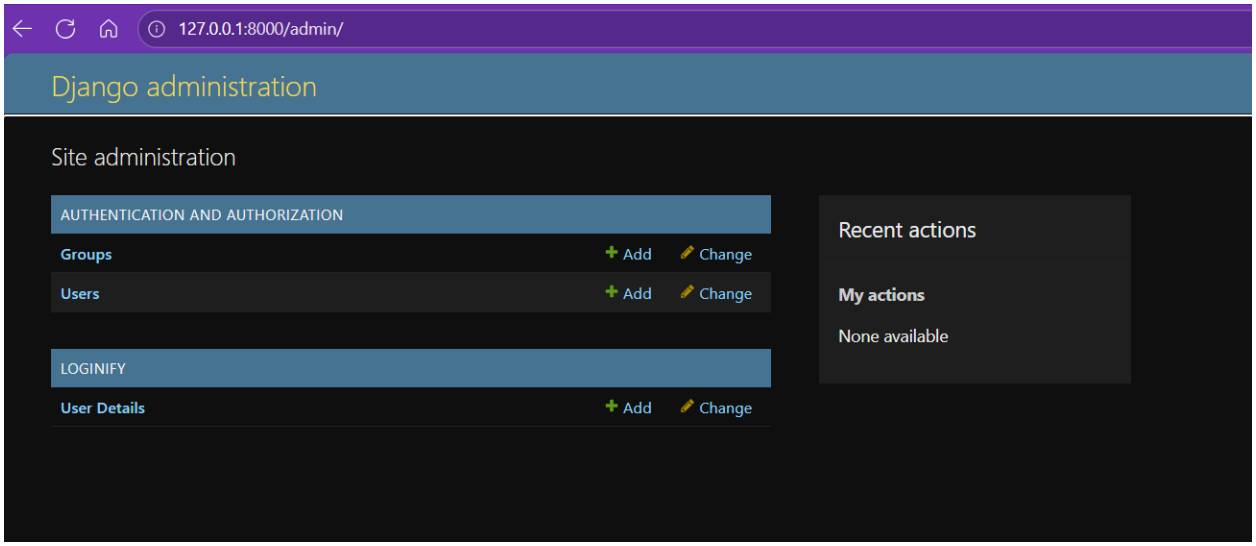
Django Assignment Screenshots

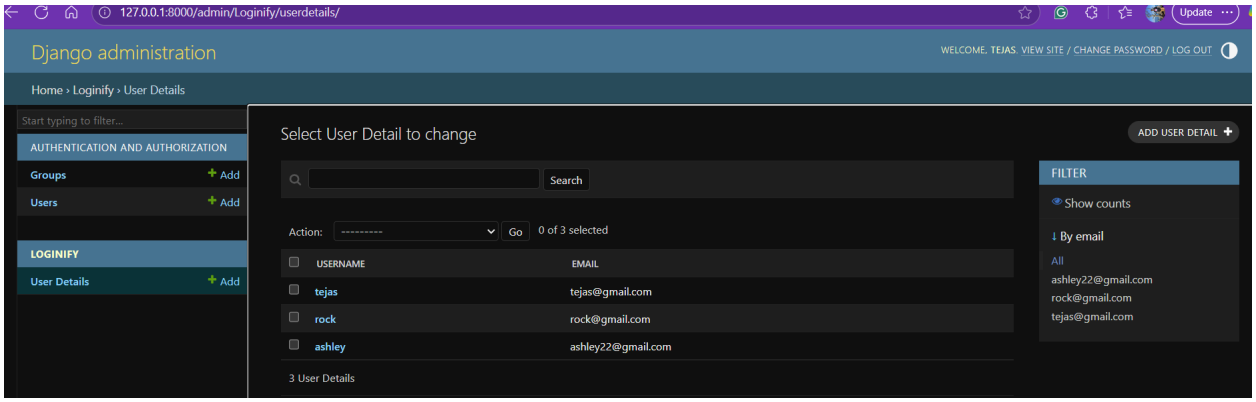
(tejas bachhav)

Login view success:



Django Admin panel:


127.0.0.1:8000/admin/
Django administration
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups
Add
Change
Users
Add
Change
LOGINIFY
User Details
Add
Change
Recent actions
My actions
None available


127.0.0.1:8000/admin/Loginify/userdetails/
Update
Django administration
WELCOME, TEJAS. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Loginify › User Details
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups
Add
Users
Add
LOGINIFY
User Details
Add
Select User Detail to change
ADD USER DETAIL
Search
Action:
Go
0 of 3 selected
USERNAME
EMAIL
tejas
tejas@gmail.com
rock
rock@gmail.com
ashley
ashley22@gmail.com
3 User Details
FILTER
Show counts
By email
All
ashley22@gmail.com
rock@gmail.com
tejas@gmail.com

Python Shell Django implementation:

1. Create new user instance
2. Retrieve all users

```
PS D:\projects\training\DjangoProject\LoginSystem> python manage.py shell
7 objects imported automatically (use -v 2 for details).

Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on
win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>>
>>> new_user = User.objects.create(username="user1",email="user1@gmail.com", password="user
1@123")
>>> all_users = User.objects.all()
>>> print(all_users)
<QuerySet [<User: tejas>, <User: user1>]>
>>> print(all_users.count())
2
```

3 retrieve a single user by username

```
>>> username = "Mike"
>>> user_by_name = UserDetails.objects.get(username=username)
>>> print(user_by_name)
Mike
>>>
```

4username to delete

```
>>> username_to_delete = "Mike"
>>> user_to_delete = ""
>>> user_to_delete = UserDetails.objects.get(username=username_to_delete)
>>> print("deleted user is",username_to_delete)
deleted user is Mike
>>>
```

5 Create a new instance using an object

```
>>> obj = UserDetails.objects.create(username="shell_user", email="shell@example.com", password="shell123")
>>> print(obj)
shell_user
>>>
```

6 query objects

```
>>> queryset = UserDetails.objects.filter(email__contains="gmail")
>>> print(queryset)
<QuerySet [<UserDetails: tejas>, <UserDetails: rock>, <UserDetails: ashley>, <UserDetails: Mike>, <UserDetai
ls: kyle>]>
>>> queryset = UserDetails.objects.filter(email__contains="example")
>>> print(queryset)
<QuerySet [<UserDetails: shell_user>]>
```

7 update an object

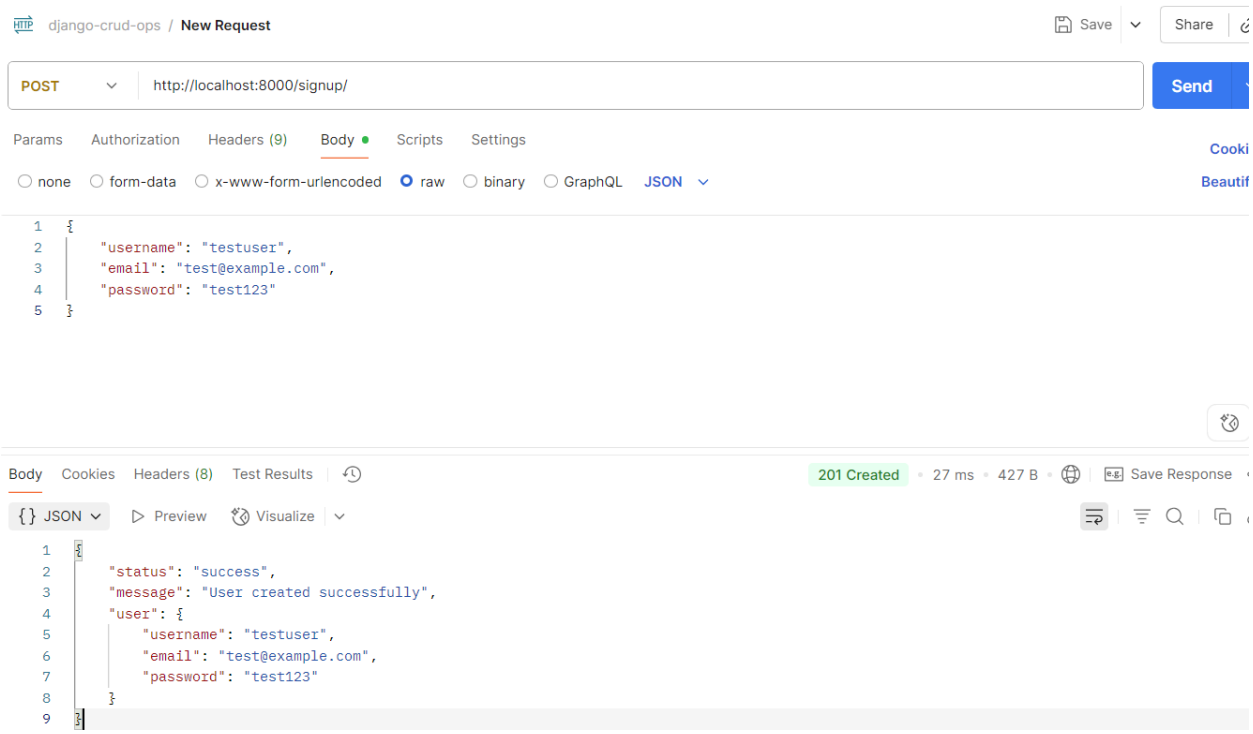
```
>>> obj.email = "updated_shell@example.com"
>>> obj.save()
>>> print("updated email to:", obj.email)
updated email to: updated_shell@example.com
>>>
```

8 Delete object

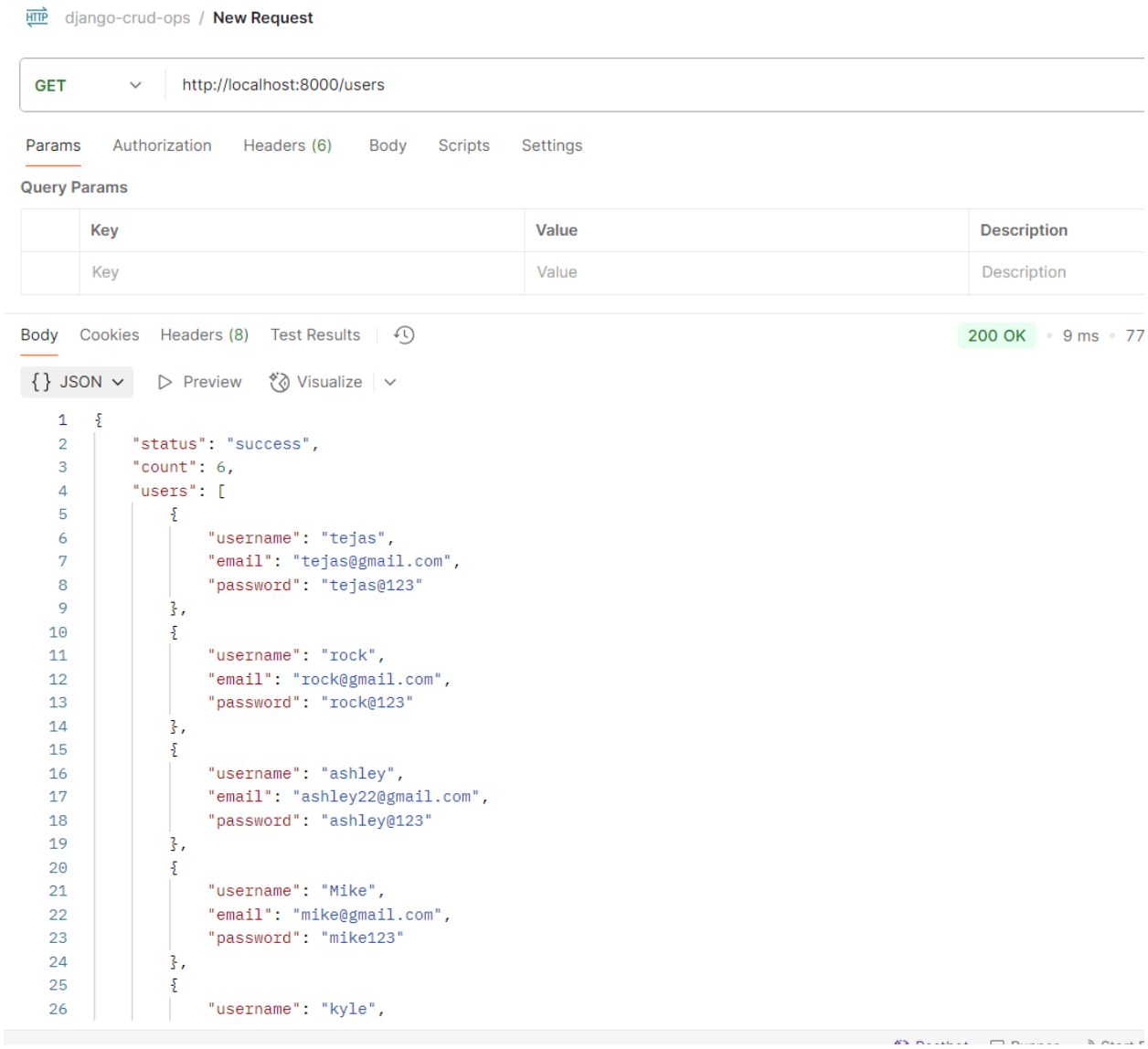
```
>>> obj.delete()
(1, {'Loginify.UserDetails': 1})
>>>
>>>
```

Task 5 - CRUD operations - Screenshots

1. Create test user

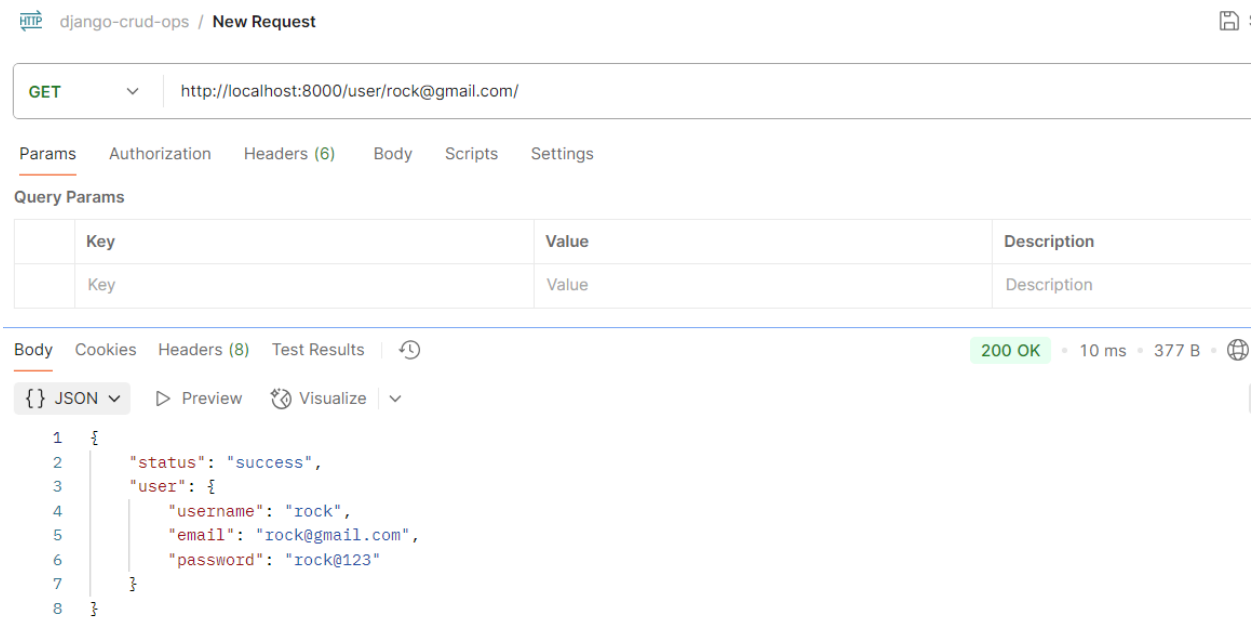


1. Get all users views:

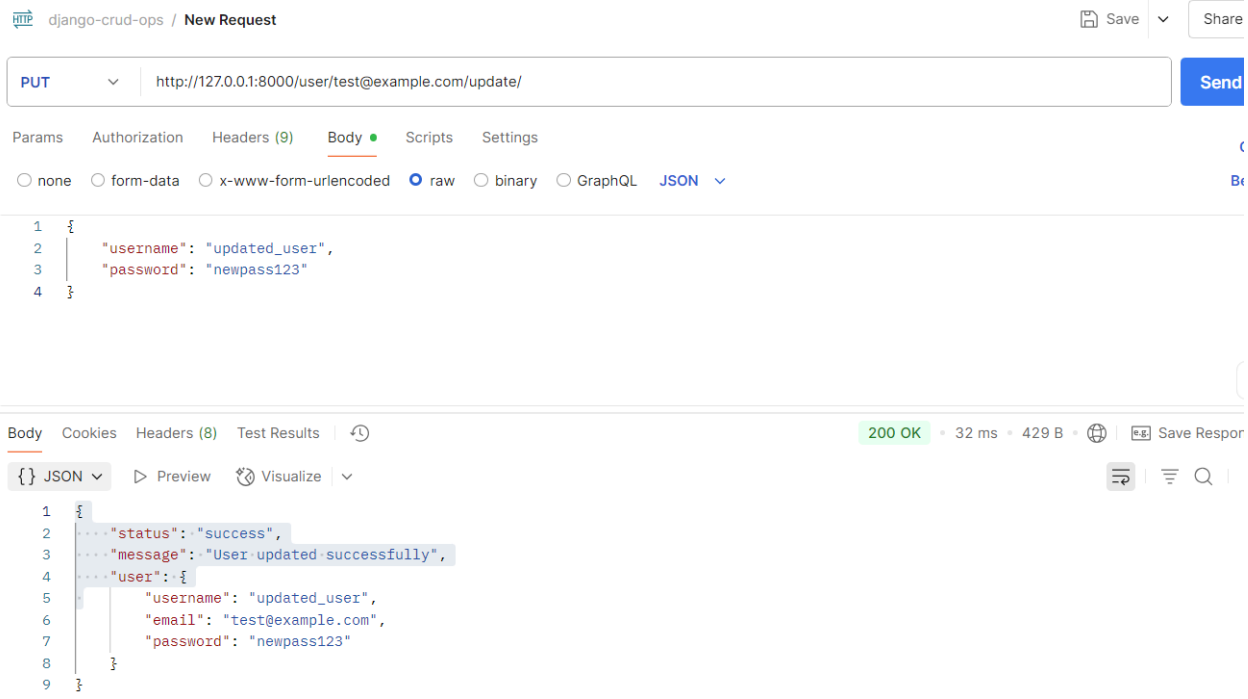


2.

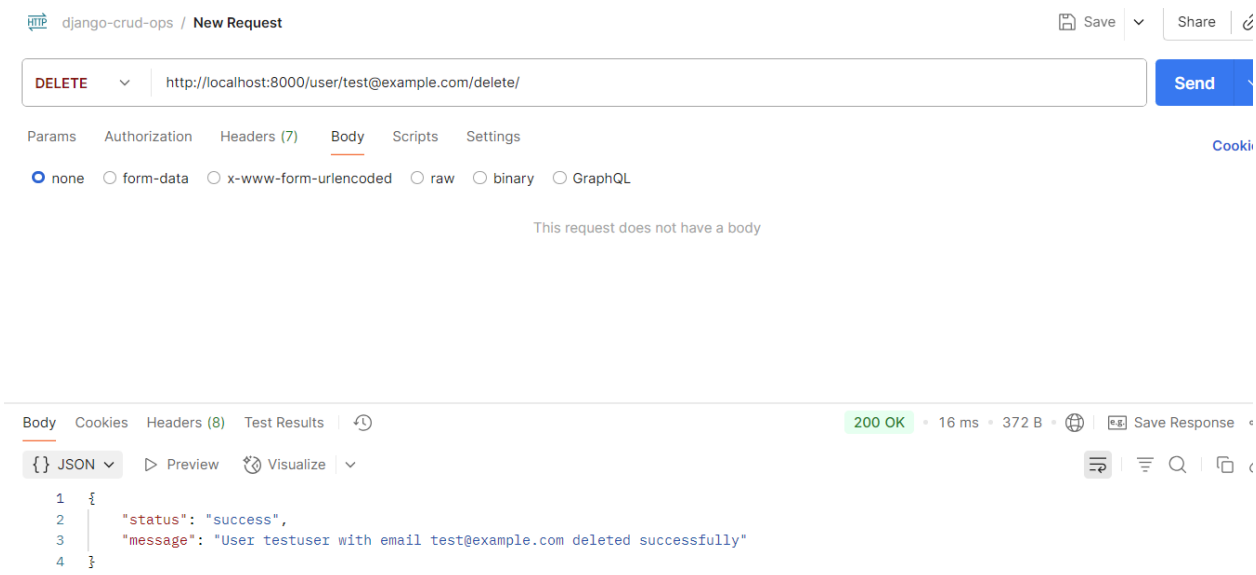
Get a single user using by email view:



Update User details



To delete a user using its email.



Thank you
