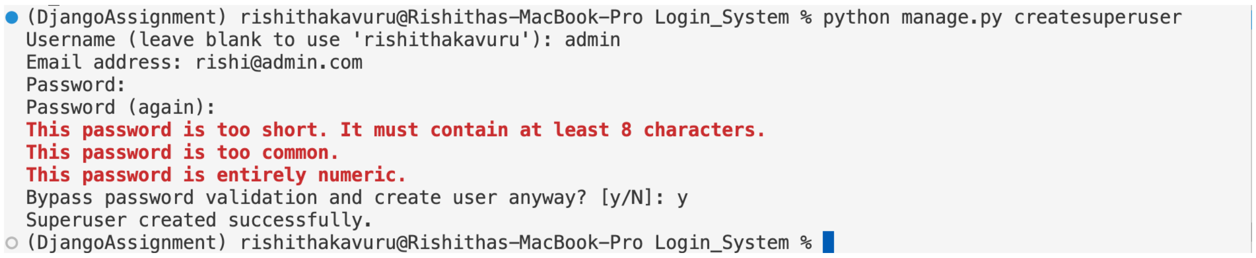
Django Assignment

Shell Screenshots

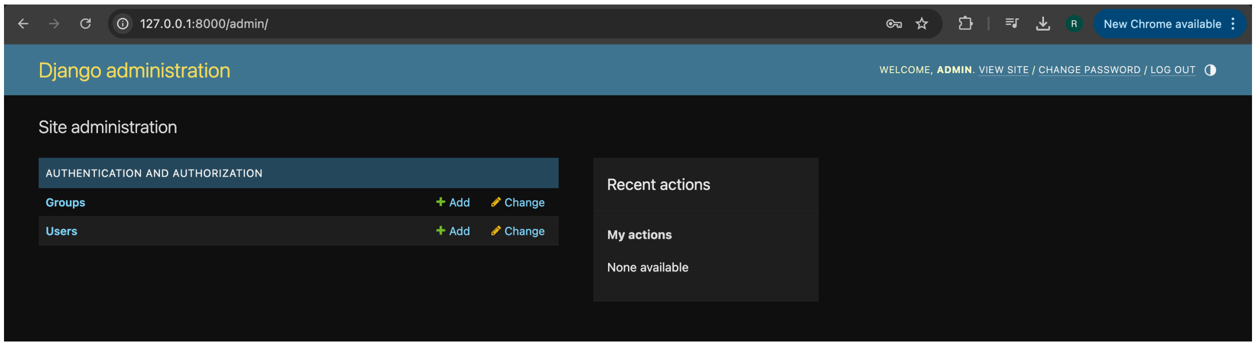
Task 3: MODELS & ADMIN

Created superuser



```
● (DjangoAssignment) rishithakavuru@Rishithas-MacBook-Pro Login_System % python manage.py createsuperuser
  Username (leave blank to use 'rishithakavuru'): admin
  Email address: rishi@admin.com
  Password:
  Password (again):
  This password is too short. It must contain at least 8 characters.
  This password is too common.
  This password is entirely numeric.
  Bypass password validation and create user anyway? [y/N]: y
  Superuser created successfully.
○ (DjangoAssignment) rishithakavuru@Rishithas-MacBook-Pro Login_System %
```

Verified superuser endpoint by visiting the admin interface.



127.0.0.1:8000/admin/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups + Add ✎ Change

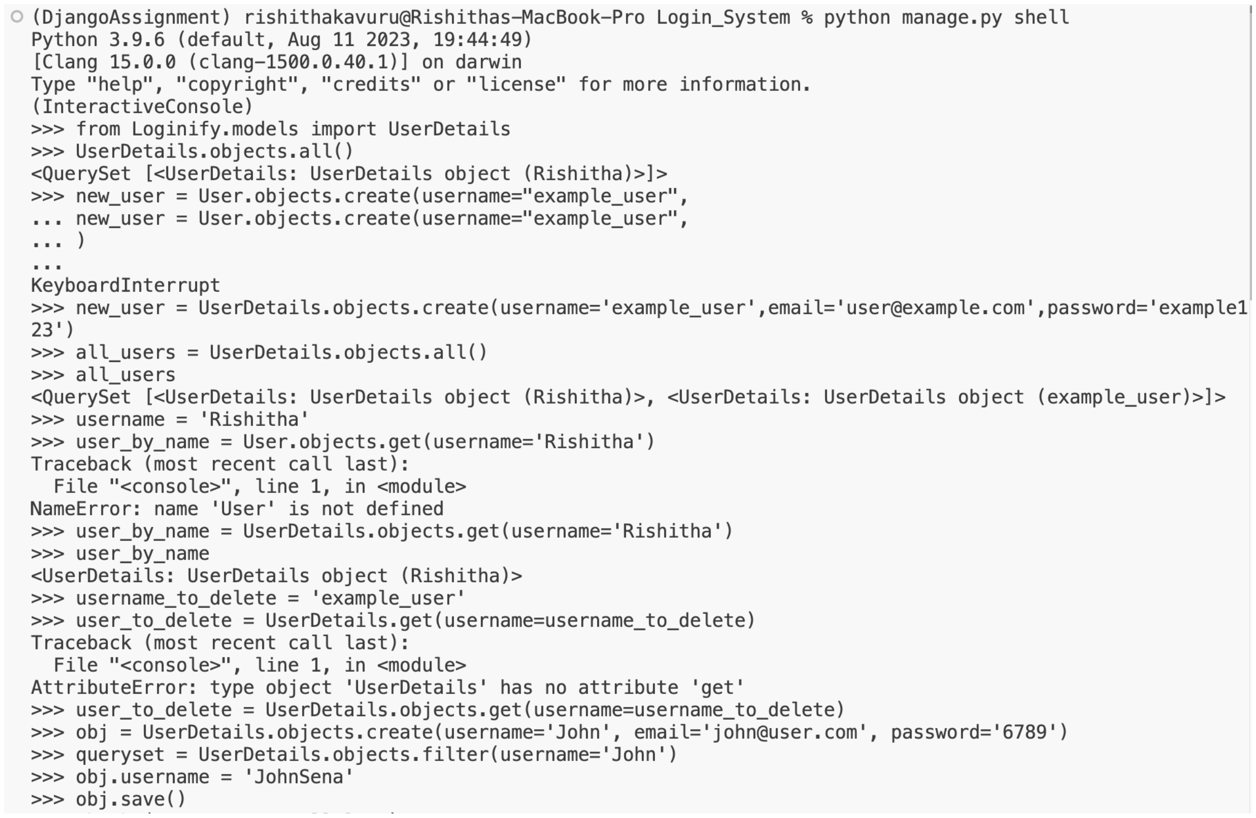
Users + Add ✎ Change

Recent actions

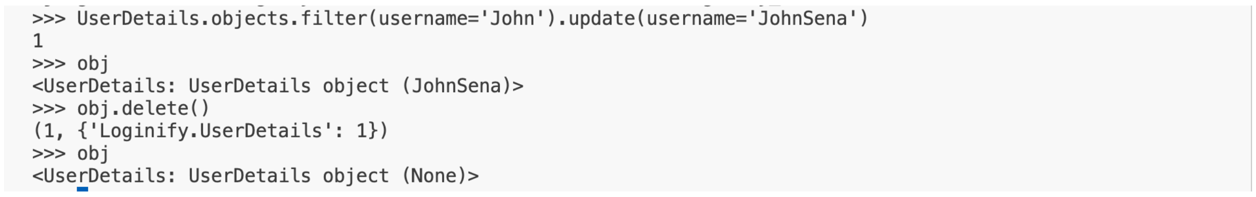
My actions

None available

Django Shell



```
(DjangoAssignment) rishithakavuru@Rishithas-MacBook-Pro Login_System % python manage.py shell
Python 3.9.6 (default, Aug 11 2023, 19:44:49)
[Clang 15.0.0 (clang-1500.0.40.1)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from Loginify.models import UserDetails
>>> UserDetails.objects.all()
<QuerySet [<UserDetails: UserDetails object (Rishitha)>]>
>>> new_user = User.objects.create(username="example_user",
... new_user = User.objects.create(username="example_user",
... )
...
KeyboardInterrupt
>>> new_user = UserDetails.objects.create(username='example_user',email='user@example.com',password='example1
23')
>>> all_users = UserDetails.objects.all()
>>> all_users
<QuerySet [<UserDetails: UserDetails object (Rishitha)>, <UserDetails: UserDetails object (example_user)>]>
>>> username = 'Rishitha'
>>> user_by_name = User.objects.get(username='Rishitha')
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'User' is not defined
>>> user_by_name = UserDetails.objects.get(username='Rishitha')
>>> user_by_name
<UserDetails: UserDetails object (Rishitha)>
>>> username_to_delete = 'example_user'
>>> user_to_delete = UserDetails.get(username=username_to_delete)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
AttributeError: type object 'UserDetails' has no attribute 'get'
>>> user_to_delete = UserDetails.objects.get(username=username_to_delete)
>>> obj = UserDetails.objects.create(username='John', email='john@user.com', password='6789')
>>> queryset = UserDetails.objects.filter(username='John')
>>> obj.username = 'JohnSena'
>>> obj.save()
```

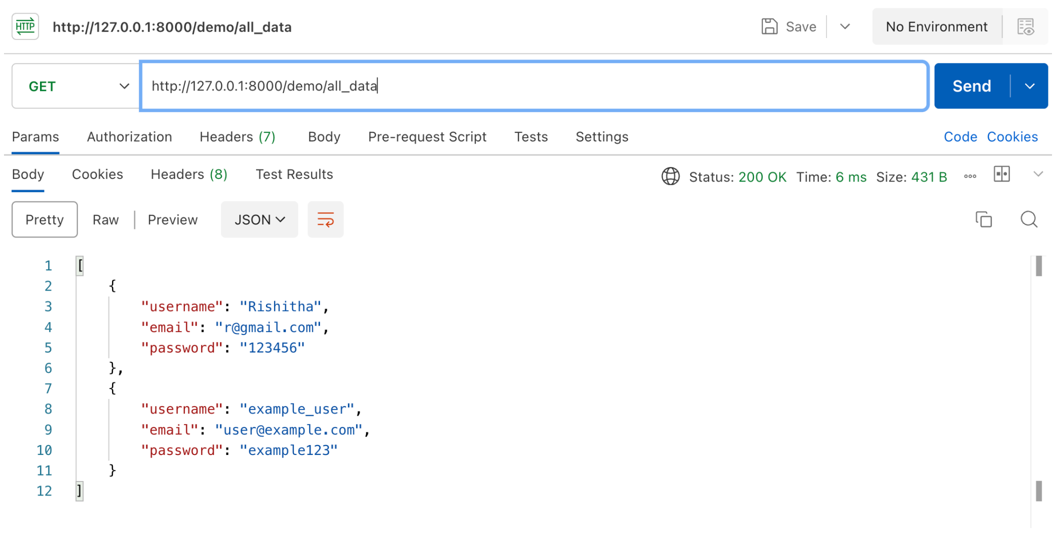


```
>>> UserDetails.objects.filter(username='John').update(username='JohnSena')
1
>>> obj
<UserDetails: UserDetails object (JohnSena)>
>>> obj.delete()
(1, {'Loginify.UserDetails': 1})
>>> obj
<UserDetails: UserDetails object (None)>
```

Postman Screenshots

Task 4: CRUD OPERATIONS

Get all user details view: Retrieves and displays details of all users.

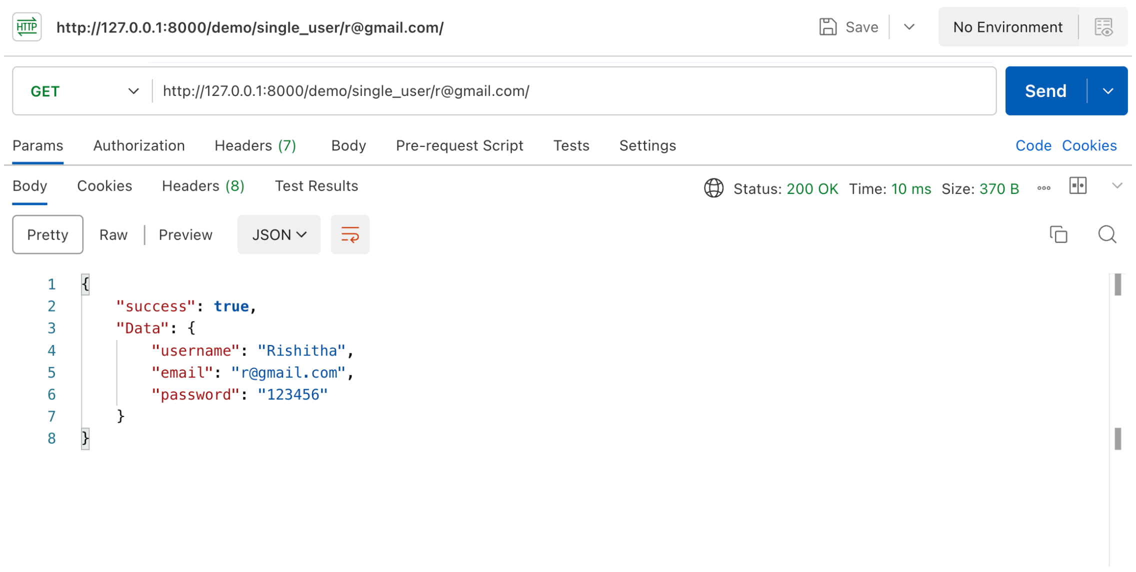

http://127.0.0.1:8000/demo/all_data

GET http://127.0.0.1:8000/demo/all_data

Status: 200 OK Time: 6 ms Size: 431 B

```
[
    {
        "username": "Rishitha",
        "email": "r@gmail.com",
        "password": "123456"
    },
    {
        "username": "example_user",
        "email": "user@example.com",
        "password": "example123"
    }
]
```

Get a single user using by email view: Retrieves and displays details of a specific user based on their name.

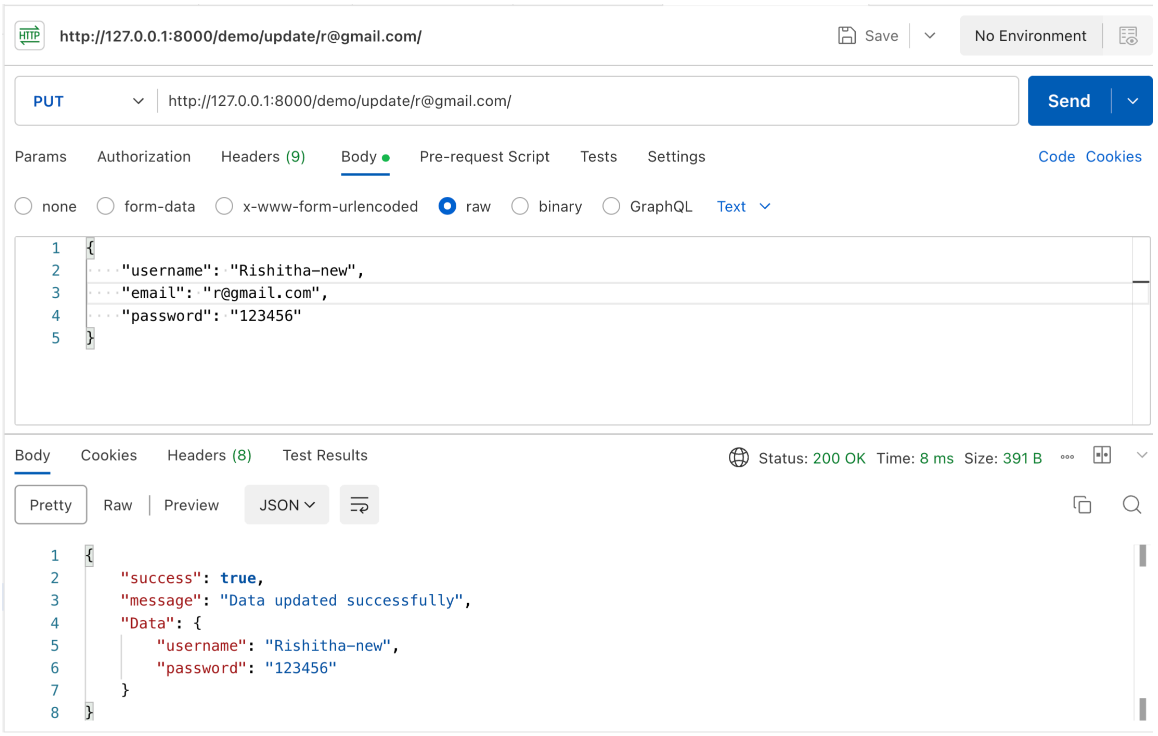

http://127.0.0.1:8000/demo/single_user/r@gmail.com/

GET http://127.0.0.1:8000/demo/single_user/r@gmail.com/

Status: 200 OK Time: 10 ms Size: 370 B

```
{
    "success": true,
    "Data": {
        "username": "Rishitha",
        "email": "r@gmail.com",
        "password": "123456"
    }
}
```

Update User details



PUT http://127.0.0.1:8000/demo/update/r@gmail.com/

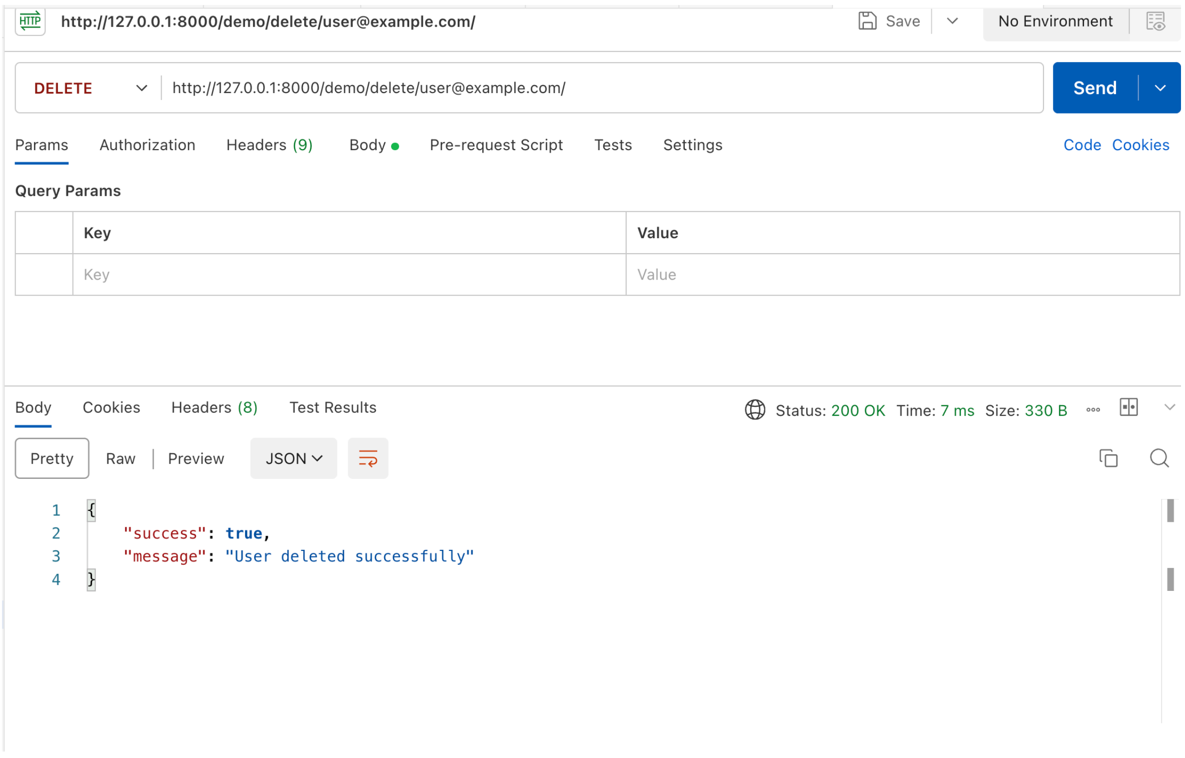
Body: raw

```
{
    "username": "Rishitha-new",
    "email": "r@gmail.com",
    "password": "123456"
}
```

Status: 200 OK Time: 8 ms Size: 391 B

```
{
    "success": true,
    "message": "Data updated successfully",
    "Data": {
        "username": "Rishitha-new",
        "password": "123456"
    }
}
```

Delete user



DELETE http://127.0.0.1:8000/demo/delete/user@example.com/

Key	Value
Key	Value

Status: 200 OK Time: 7 ms Size: 330 B

```
{
    "success": true,
    "message": "User deleted successfully"
}
```