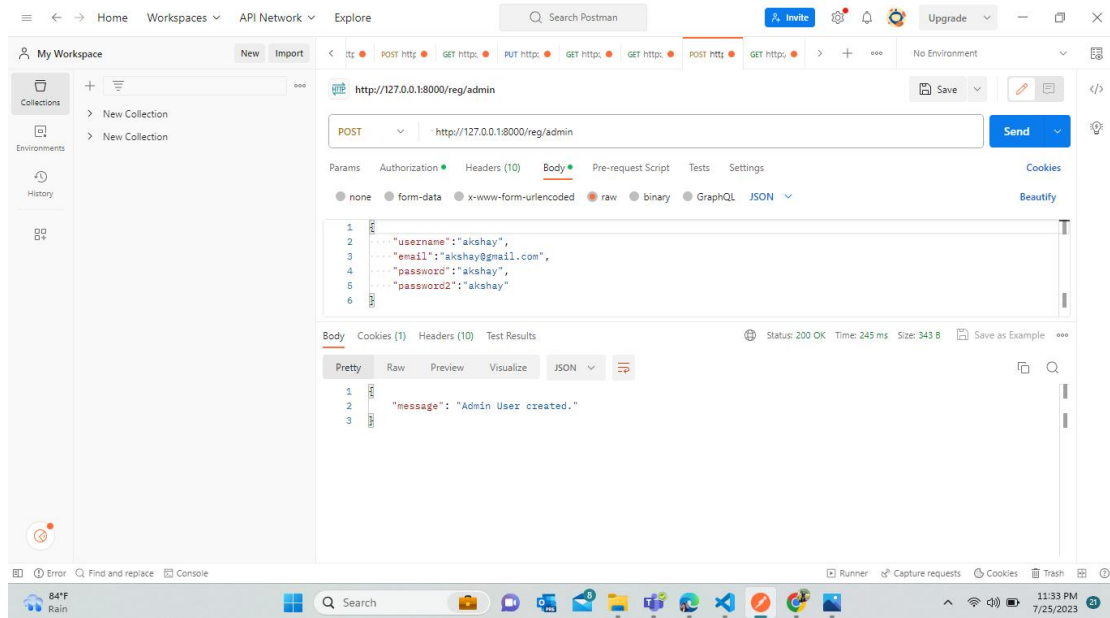


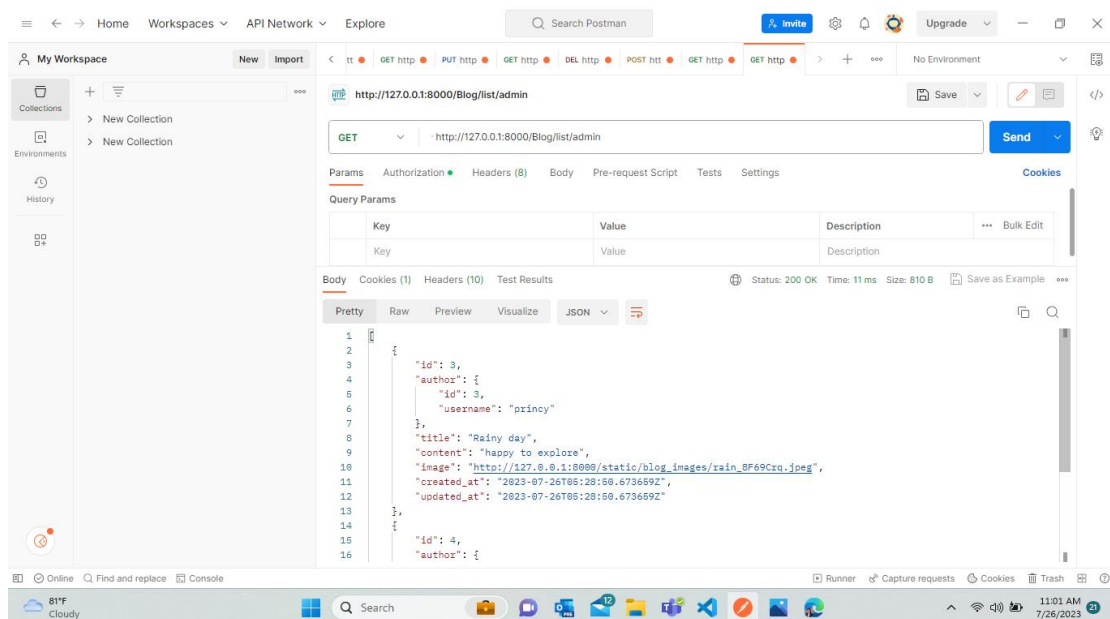
Admin

1. Authentication:

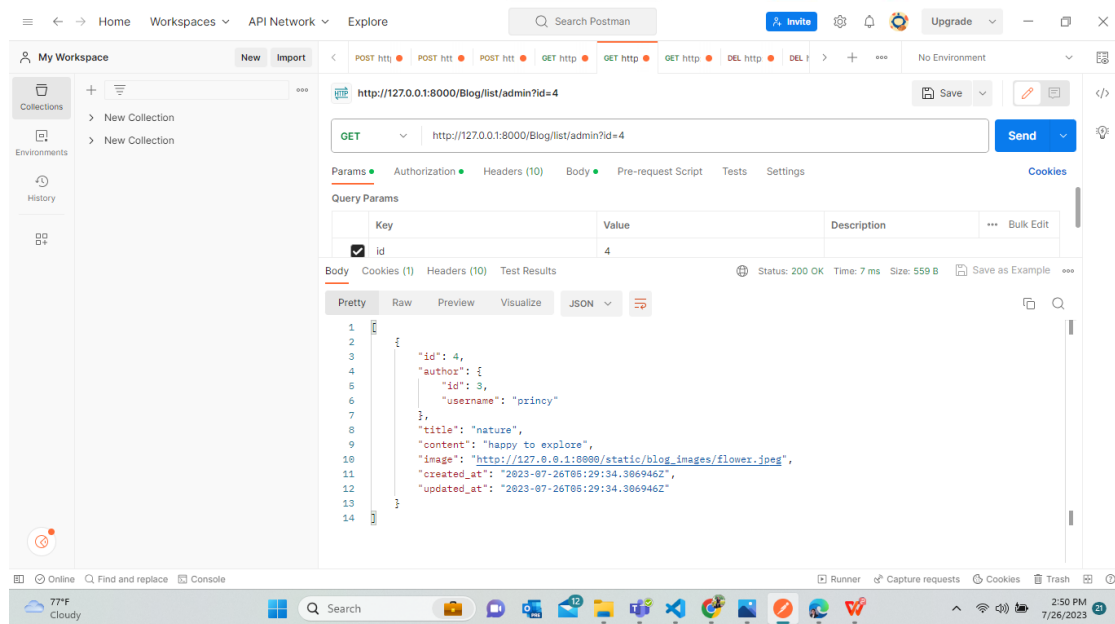


2. Blog Posts:

Blog list



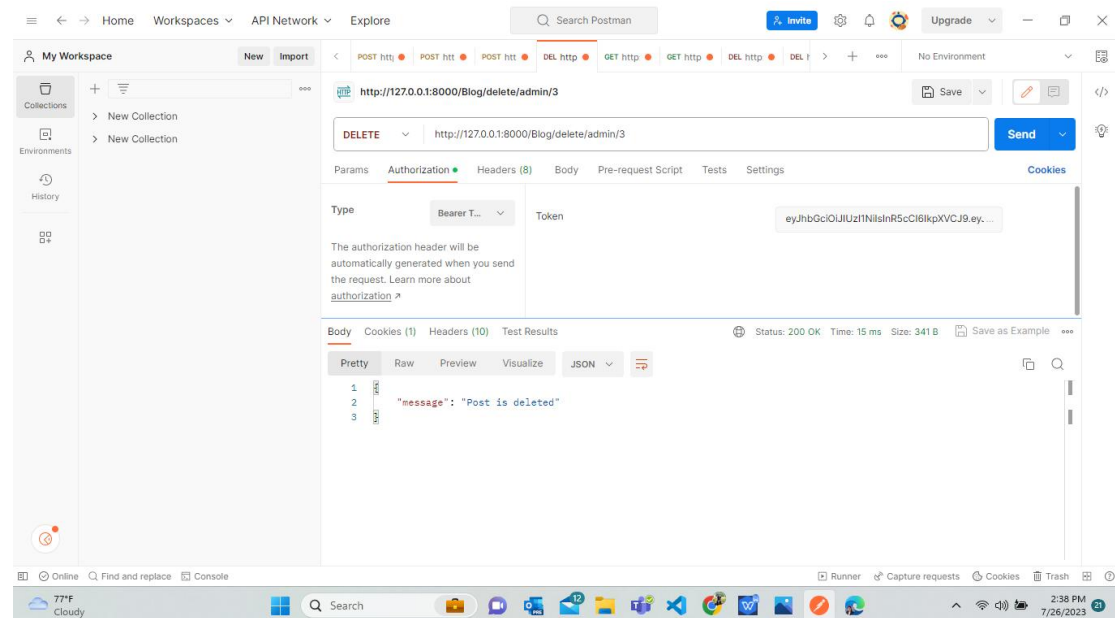
Blog detail:



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8000/Blog/list/admin?id=4`. The response is a JSON object with the following structure:

```
{
  "id": 4,
  "author": {
    "id": 3,
    "username": "princy"
  },
  "title": "nature",
  "content": "happy to explore",
  "image": "http://127.0.0.1:8000/static/blog_images/flower.jpeg",
  "created_at": "2023-07-26T06:29:34.386946Z",
  "updated_at": "2023-07-26T06:29:34.386946Z"
}
```

Delete Blog:

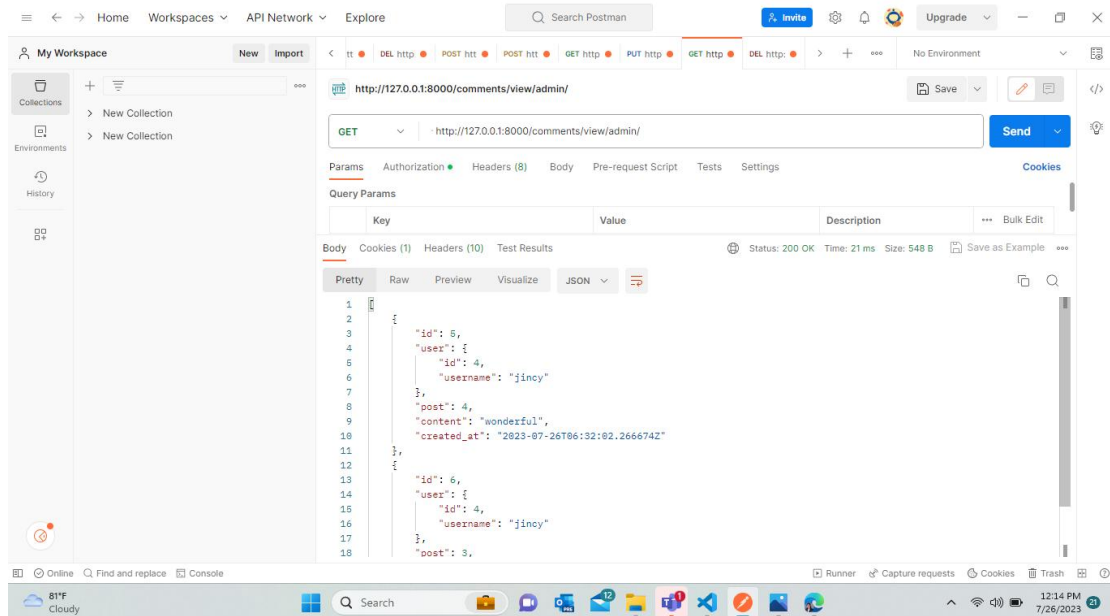


The screenshot shows the Postman interface with a DELETE request to `http://127.0.0.1:8000/Blog/delete/admin/3`. The request is authorized with a Bearer token. The response is a JSON object with the following structure:

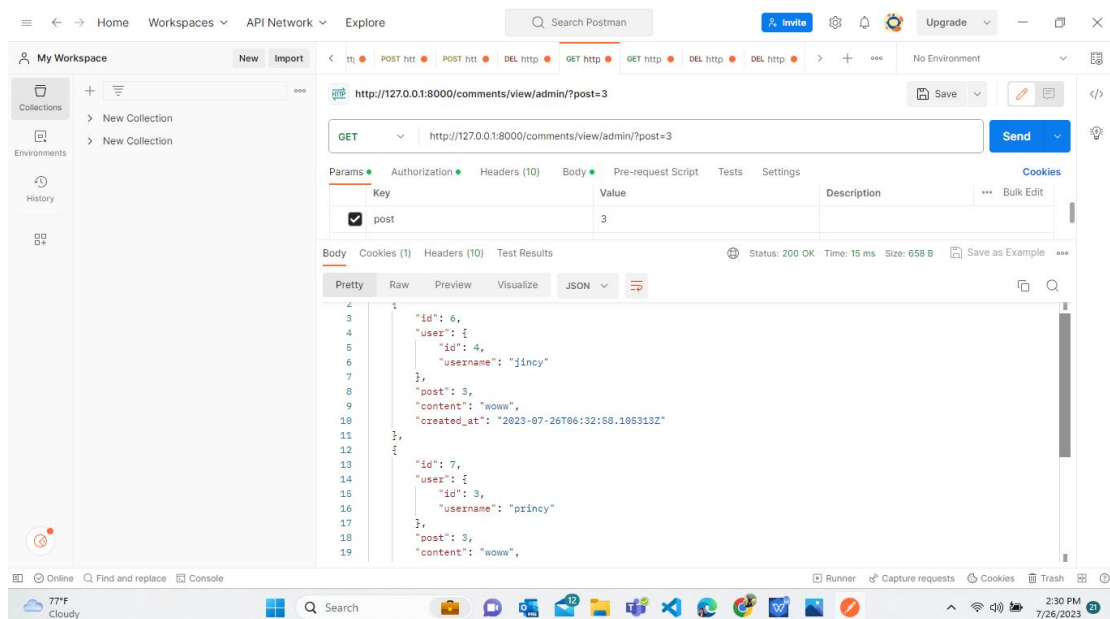
```
{
  "message": "Post is deleted"
}
```

3. Comments:

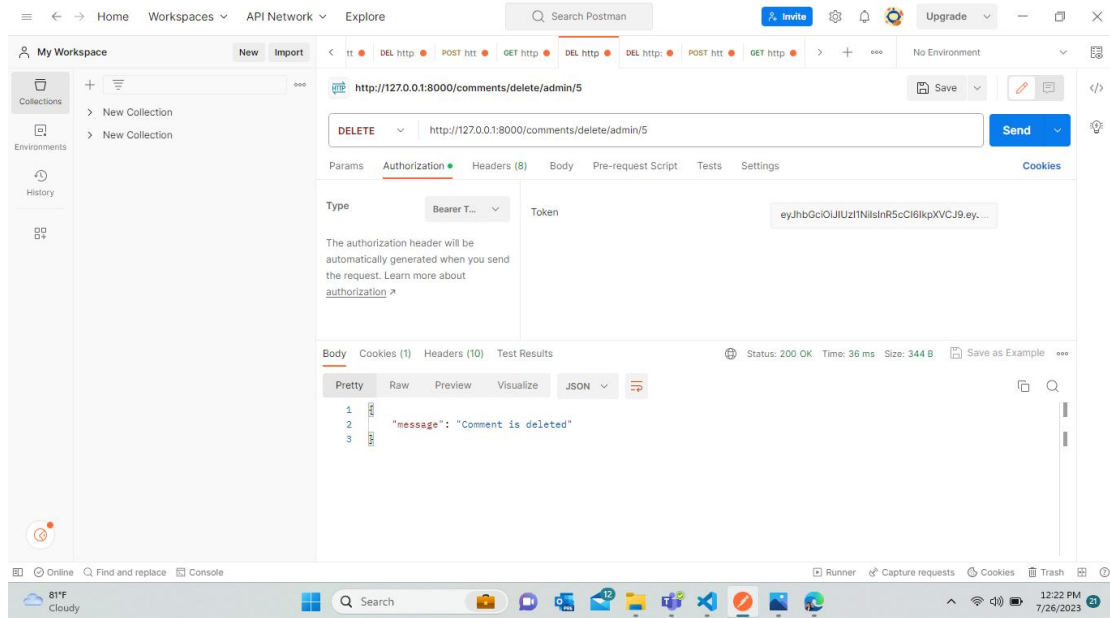
View comments list



Comment Detail



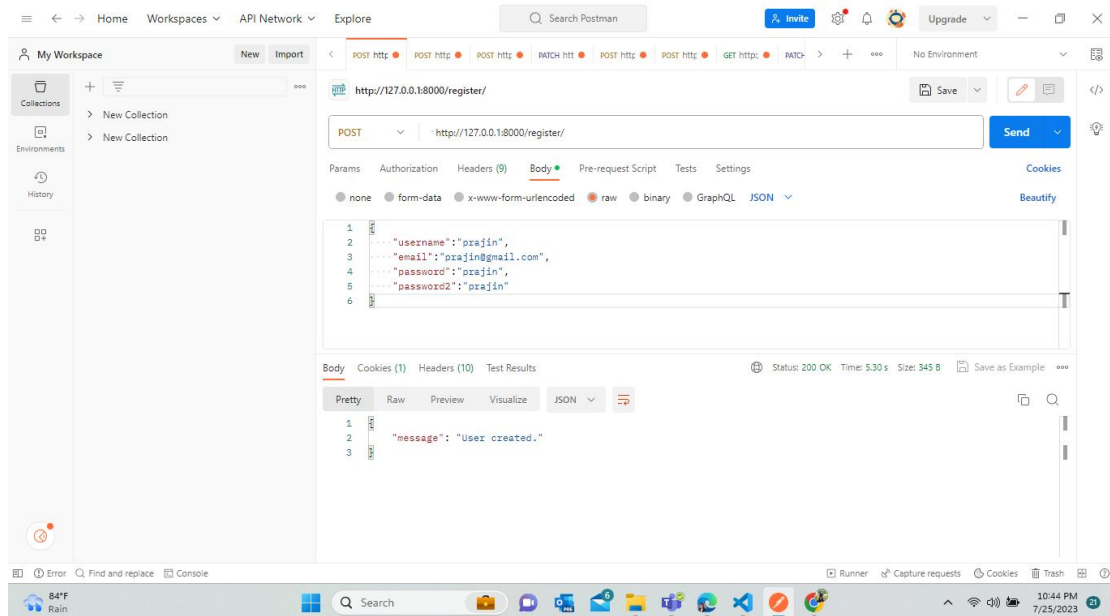
Comment Delete



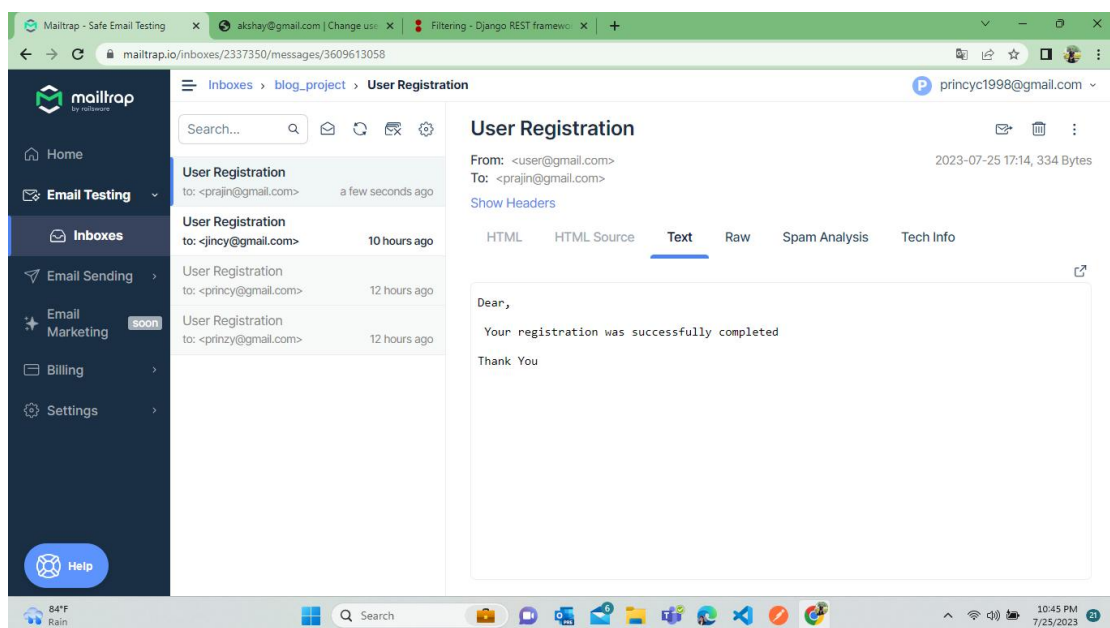
User

1. User Authentication:

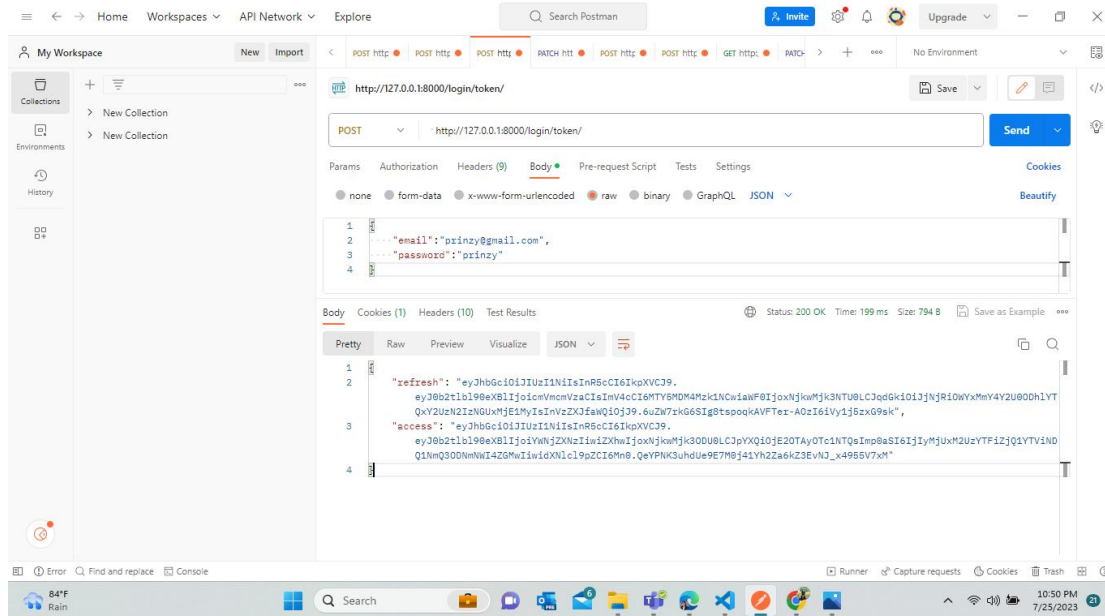
Register



Email sending

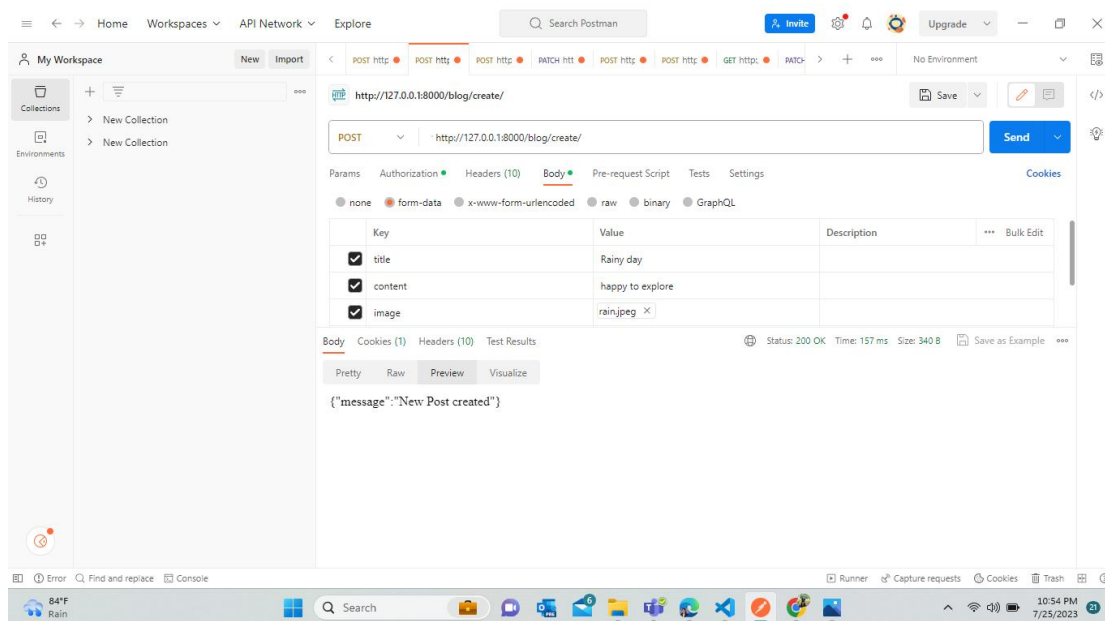


Login



2. Blog Posts:

Create post



Post list-view

The screenshot shows a REST client interface with a GET request to `http://127.0.0.1:8000/blog/list/`. The response is a 200 OK status with a JSON body containing two blog entries. The interface includes a sidebar with 'My Workspace', 'Collections', 'Environments', and 'History'. The main panel shows the request details, including headers, body, and test results. The JSON response is displayed in a 'Pretty' format.

```
1 {
2   "id": 1,
3   "author": {
4     "id": 4,
5     "username": "jincy"
6   },
7   "title": "Heaven",
8   "content": "LOVE",
9   "image": "http://127.0.0.1:8000/static/blog_images/nature.jpeg",
10  "created_at": "2023-07-25T09:10:29.760343Z",
11  "updated_at": "2023-07-25T10:10:30.194980Z"
12 },
13 {
14   "id": 2,
15   "author": {
16     "id": 3,
17     "username": "princy"
18   },
19   "title": "Rainy day",
20   "content": "happy to explore",
21   "image": "http://127.0.0.1:8000/static/blog_images/rain.jpeg",
22   "created_at": "2023-07-25T13:41:56.376913Z",
23   "updated_at": "2023-07-25T13:41:56.376913Z"
24 }
```

Post detail-view

The screenshot shows a REST client interface with a GET request to `http://127.0.0.1:8000/blog/detail/2`. The response is a 200 OK status with a JSON body containing details for a specific blog entry. The interface includes a sidebar with 'My Workspace', 'Collections', 'Environments', and 'History'. The main panel shows the request details, including headers, body, and test results. The JSON response is displayed in a 'Pretty' format.

```
1 {
2   "id": 2,
3   "author": {
4     "id": 3,
5     "username": "princy"
6   },
7   "title": "Rainy day",
8   "content": "happy to explore",
9   "image": "http://127.0.0.1:8000/static/blog_images/rain.jpeg",
10  "created_at": "2023-07-25T13:41:56.376913Z",
11  "updated_at": "2023-07-25T13:41:56.376913Z"
12 }
```

Update

Postman interface showing a PATCH request to `http://127.0.0.1:8000/update/1`. The request body is a JSON object:

```
{
  "title": "Heaven",
  "content": "LOVE"
}
```

The response is a JSON object:

```
{
  "id": 1,
  "author": {
    "id": 4,
    "username": "jincy"
  },
  "title": "Heaven",
  "content": "LOVE",
  "image": "http://127.0.0.1:8000/static/blog_images/nature.jpeg",
  "created_at": "2023-07-25T09:18:29.760343Z",
  "updated_at": "2023-07-25T17:28:31.177721Z"
}
```

Postman interface showing a PUT request to `http://127.0.0.1:8000/update/1`. The request is authenticated with a Bearer token. The response is a JSON object:

```
{
  "id": 1,
  "author": {
    "id": 4,
    "username": "jincy"
  },
  "title": "Heaven",
  "content": "LOVE",
  "image": "http://127.0.0.1:8000/static/blog_images/nature.jpeg",
  "created_at": "2023-07-25T09:18:29.760343Z",
  "updated_at": "2023-07-25T17:28:31.177721Z"
}
```


Delete

The screenshot shows the Postman interface with a DELETE request configured. The URL is `http://127.0.0.1:8000/delete/2`. The request is authenticated using a Bearer token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t...`. The response status is 200 OK, and the body contains a message: "Post is deleted".

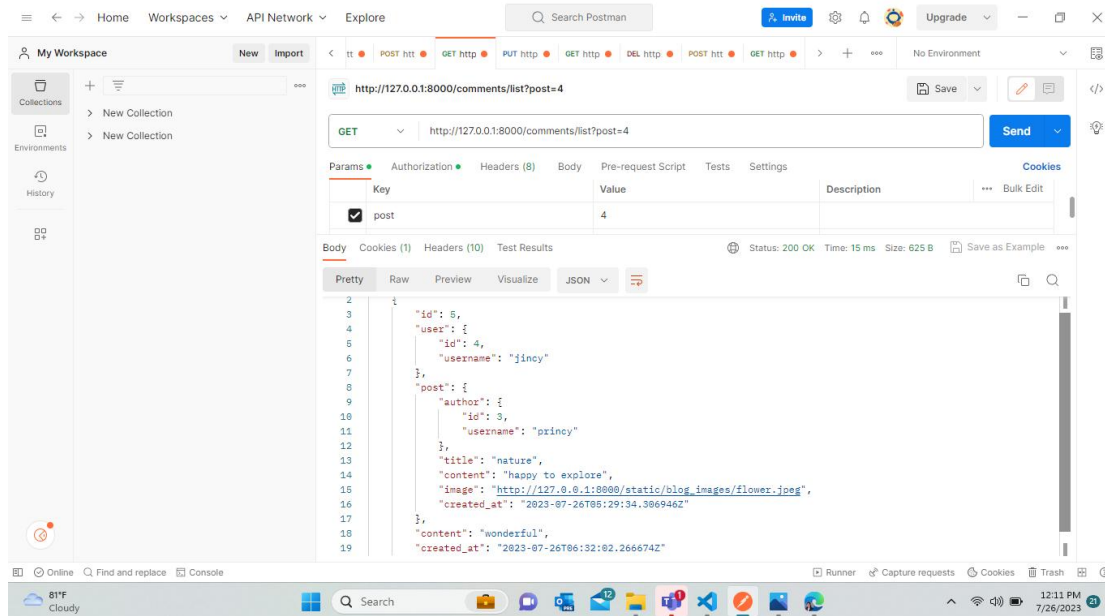
3. Comments:

Create

The screenshot shows the Postman interface with a POST request configured. The URL is `http://127.0.0.1:8000/comments/add/`. The request is authenticated using a Bearer token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t...`. The response status is 201 Created, and the body contains a JSON object representing a comment:

```
{
  "id": 7,
  "user": {
    "id": 3,
    "username": "princy"
  },
  "post": 3,
  "content": "wow",
  "created_at": "2023-07-26T08:16:02.177464Z"
}
```

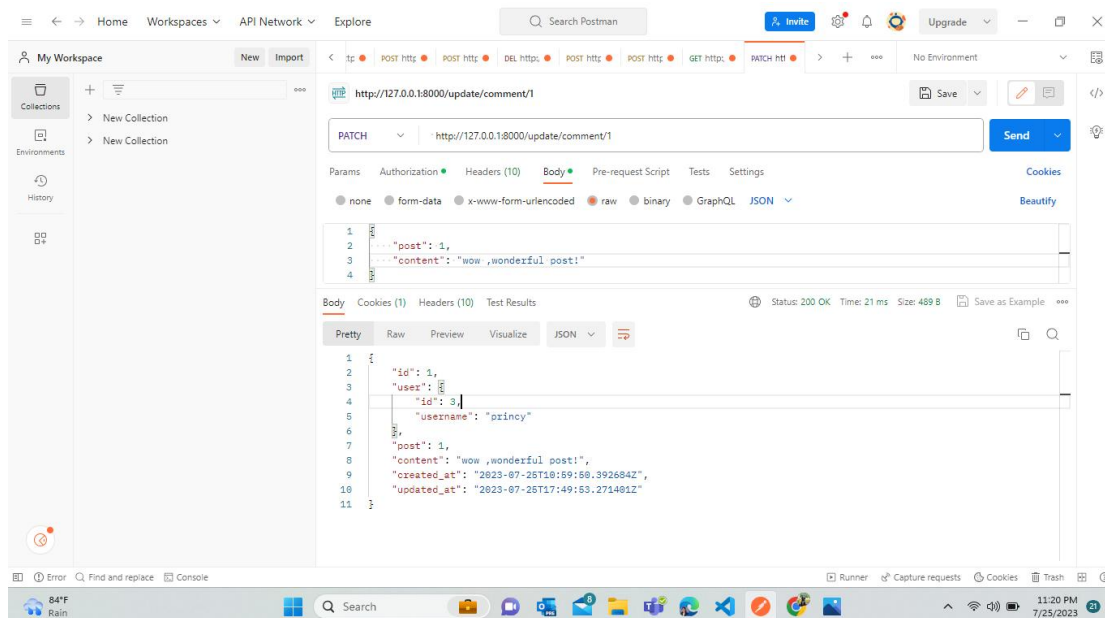
Comments of a post



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8000/comments/list?post=4`. The request has a parameter `post` with the value `4`. The response status is `200 OK` with a time of `15 ms` and a size of `625 B`. The response body is a JSON array of two comment objects.

```
1 [{"id": 5,
2   "user": {
3     "id": 4,
4     "username": "jincy"
5   },
6   "post": {
7     "author": {
8       "id": 3,
9       "username": "princy"
10    },
11    "title": "nature",
12    "content": "happy to explore",
13    "image": "http://127.0.0.1:8000/static/blog_images/flower.jpeg",
14    "created_at": "2023-07-26T06:29:34.386946Z"
15  },
16  "content": "wonderful",
17  "created_at": "2023-07-26T06:32:02.266674Z"
18 }]
```

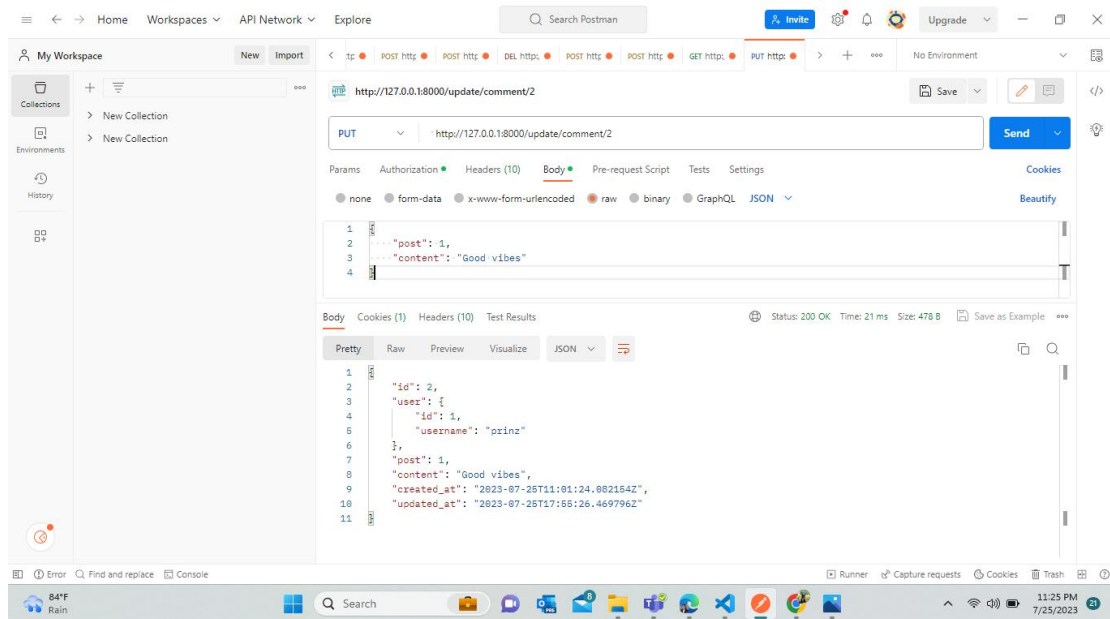
Update



The screenshot shows the Postman interface with a PATCH request to `http://127.0.0.1:8000/update/comment/1`. The request body is a JSON object with `post` and `content` fields. The response status is `200 OK` with a time of `21 ms` and a size of `489 B`. The response body is a JSON object with updated comment details.

```
1 {
2   "post": 1,
3   "content": "wow ,wonderful post!"
4 }

1 {
2   "id": 1,
3   "user": {
4     "id": 3,
5     "username": "princy"
6   },
7   "post": 1,
8   "content": "wow ,wonderful post!",
9   "created_at": "2023-07-25T10:59:50.392684Z",
10  "updated_at": "2023-07-25T17:49:53.271491Z"
11 }
```



Delete

