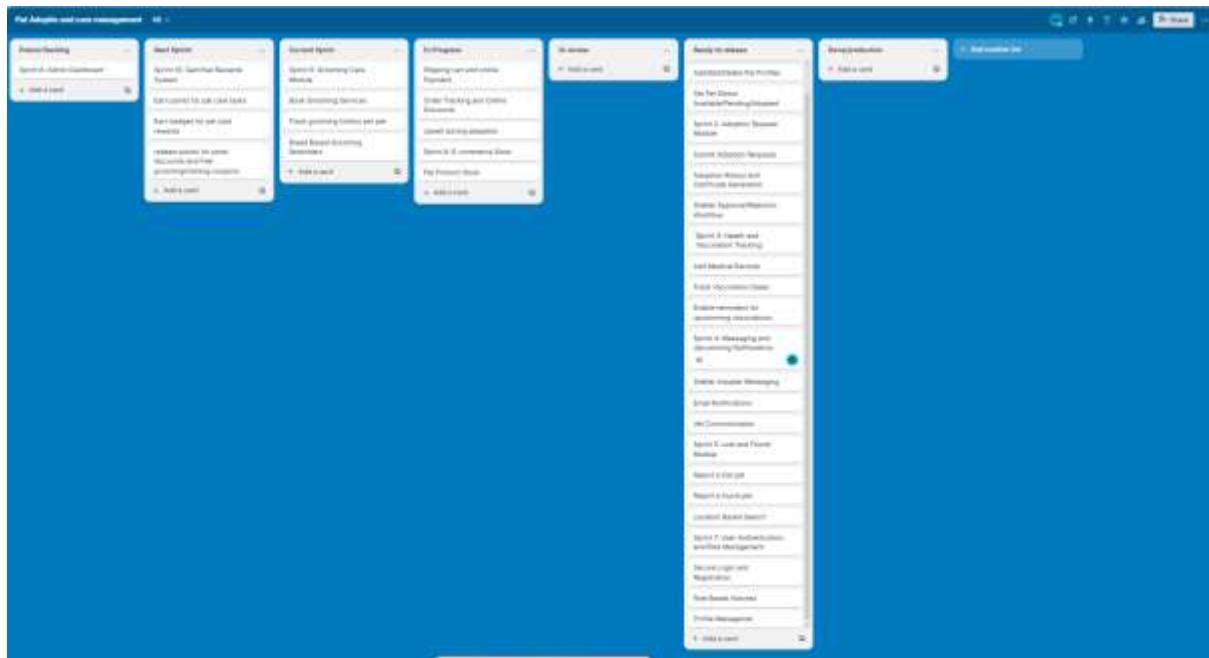


1. Sprint Planning & Backlog

1.1 Trello Sprint Planning Screenshot



1.2 Incomplete or Carried Forward Tasks from Previous Sprint

I completed the lost and found section of which the logic wasn't handled properly and made the chat system real time using the websocket

2. API Development & Testing

APIs Worked on

API Name	Method	Status	Tested
Send Message	Post	Completed	Sucessfull

API Testing Screenshots

```

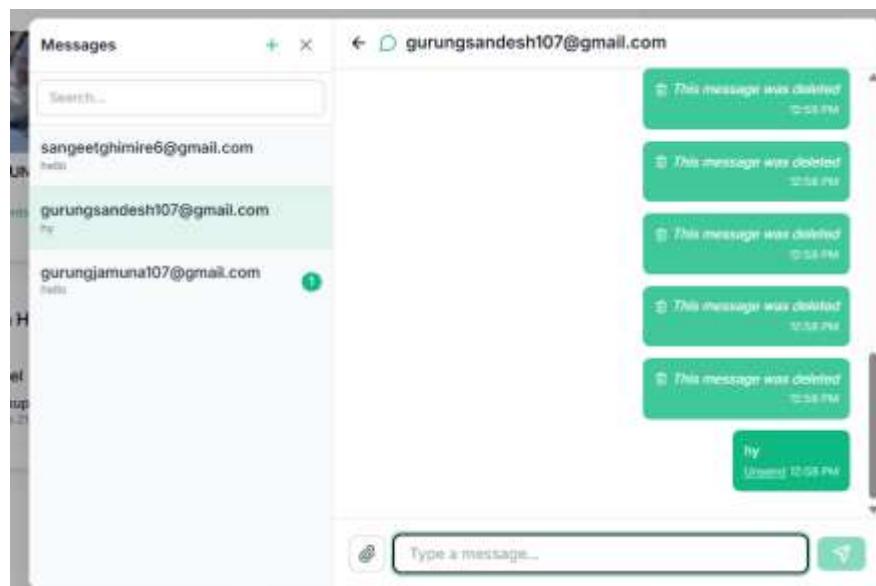
POST /api/messages/
{
  "recipient": "Shubham",
  "conversation_id": "123",
  "body": "Hello! I am interested in adopting this pet."
}
  
```

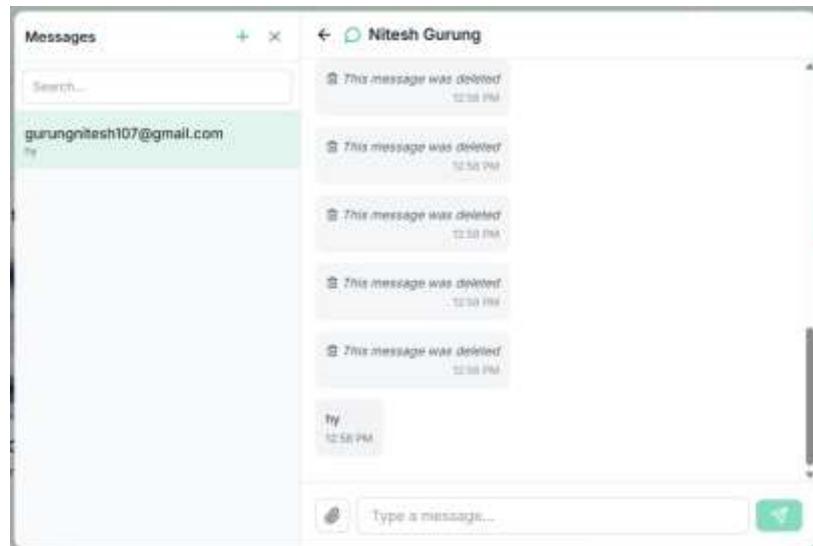
The screenshot shows the Postman interface with a successful response. The response body is displayed as follows:

```

{
  "id": 1,
  "sender_info": {
    "id": 10,
    "username": "gurungsandesh107@gmail.com",
    "email": "gurungsandesh107@gmail.com",
    "first_name": "Sandeep",
    "last_name": "Kumar"
  },
  "recipient": "Shubham",
  "recipient_info": {
    "id": 11,
    "username": "gurungjamuna107@gmail.com",
    "email": "gurungjamuna107@gmail.com"
  },
  "subject": "Message",
  "body": "Hello! I am interested in adopting this pet.",
  "attachment_ids": [],
  "attachment_urls": null,
  "is_read": false,
  "is_deleted": false,
  "created_at": "2018-02-22T00:07:00Z",
  "updated_at": null,
  "is_latest": true
}
  
```

Chat Testing Successful between adopter and shelter





Successfull

The screenshot shows a browser-based REST API tool with the following details:

- URL:** `http://127.0.0.1:8000/api/messages/`
- Method:** POST
- Headers:** None
- Body:** JSON (Raw)

The body of the request contains the following JSON data:

```

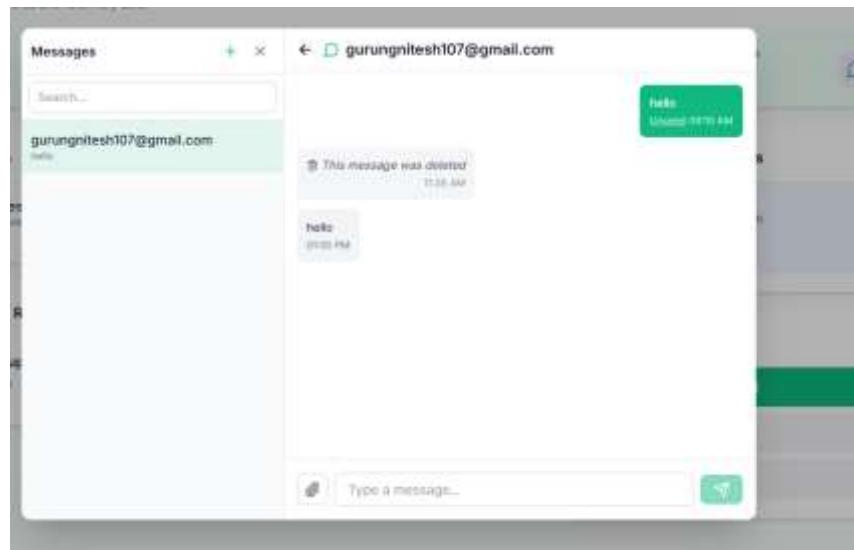
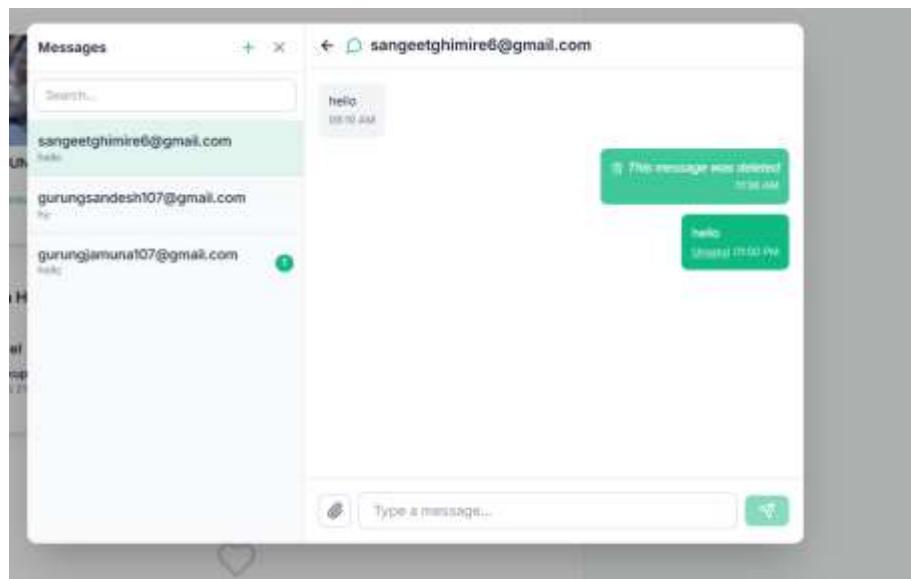
1. {
2.     "recipient": "g",
3.     "recipients_id": 12,
4.     "body": "Hello can you provide me a medical record of my pet."
5. }

```

The response section shows a successful response with the following details:

- Status:** 201 Created
- Time:** 1.33 s - 400.4 kB
- Content-Type:** application/json
- Content:** (Large JSON object redacted)

Chat testing successful between adopter and veterenerian



Successfull

The screenshot shows a Postman request to `http://127.0.0.1:8000/api/messages/`. The request method is POST. The body is a JSON object:

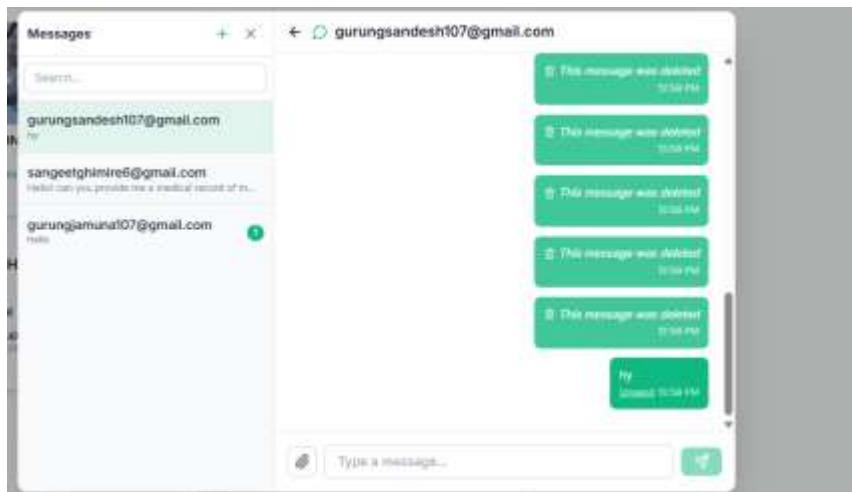
```

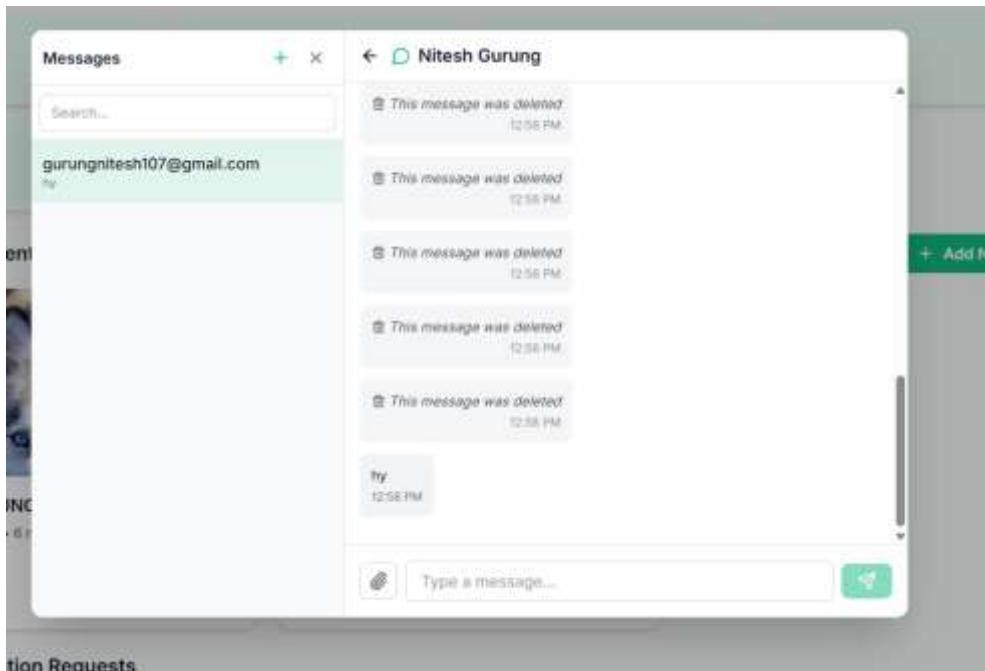
1  {
2     "from": "1234567890",
3     "to": "Hello! this pet is available.",
4     "recipients": [
5         {
6             "id": 29,
7             "username": "gurungsandesh107@gmail.com",
8             "email": "gurungsandesh107@gmail.com",
9             "first_name": "Sandeep",
10            "last_name": "Kumar"
11        }
12    ],
13    "subject": "Message",
14    "body": "Hello! this pet is available.",
15    "attachments": [
16        {
17            "id": 29,
18            "username": "gurungsandesh107@gmail.com",
19            "email": "gurungsandesh107@gmail.com"
20        }
21    ],
22    "is_read": false,
23    "is_deleted": false,
24    "is_trashed": false,
25    "created_at": "2020-02-21T09:31:29.043Z"
26  }

```

The response status is 201 Created.

Chat testing successful between shelter and adopter

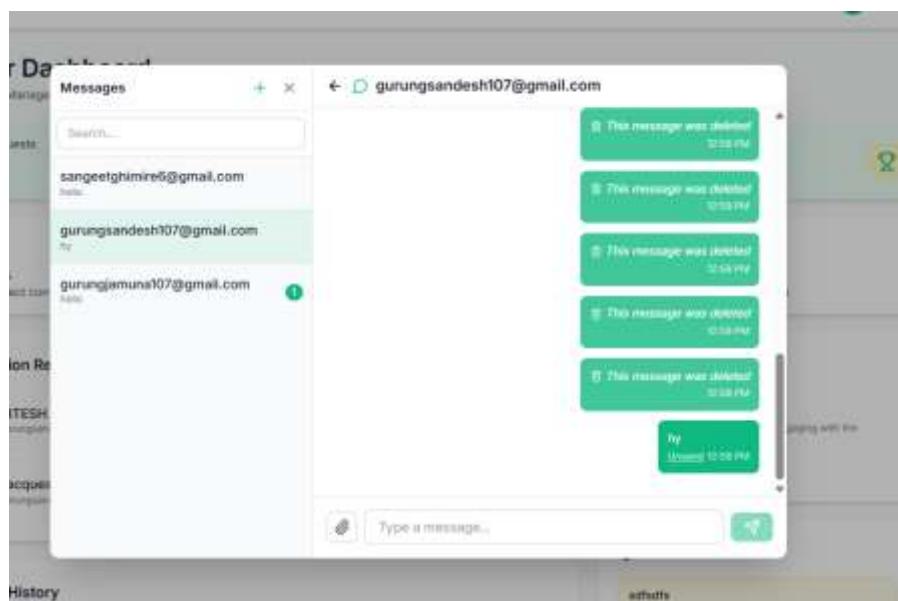




Successful

3. Frontend Integration Progress

Frontend Integration Screenshots



Chat Frontend Integration

```

websocket consumer for real-time chat, one code per conversation (pair of users).
Authenticates via JWT in query string; only allowed chat partners can join the room.
```
import json
from channels.generic.websocket import AsyncWebsocketConsumer
from channels.db import database_sync_to_async
from django.contrib.auth import get_user_model

User = get_user_model()

class ChatConsumer(AsyncWebsocketConsumer):
 ...
 Room name: chat_(min_user_id)_(max_user_id) so both participants join the same room.
 URL: ws/chat/<other_user_id>/?token=<jwt>
    ```

    @sync def connect(self):
        self.room_group_name = None
        self.user = None

        # Parse other_user_id from URL
        other_user_id = self.scope["url_route"]["kwargs"].get("other_user_id")
        try:
            other_user_id = int(other_user_id)
        except (TypeError, ValueError):
            await self.close(code=400)
            return

        # JWT from query string
        qs = self.scope.get("query_string", b"?").decode()
        token = None
        for part in qs.split("&"):
            if part.startswith("token="):
                token = part[6:]
                break
```

```

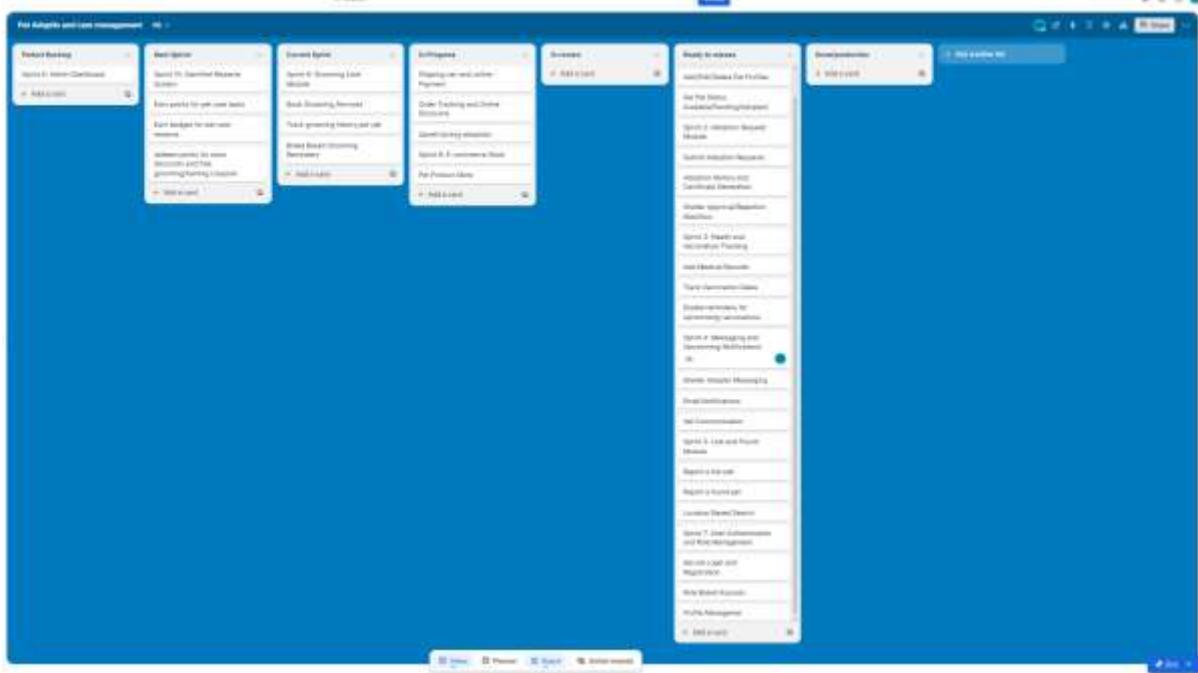
Backend code

#### 4. Unfinished tasks, Issues & Blockers

- left to make the logic of shelter and veterinarian chat

#### 5. Sprint Status & Next Steps

##### 5.1 End of Sprint Trello Screenshot



## 5.2 Next Sprint Focus:

Sprint 8 focuses on building the E-Commerce Store module of PetCare system. In this sprint, it will create a pet product store where users can browse items like food, toys, grooming supplies, and health products, add them to a shopping cart, and complete purchases using secure payment options such as Khalti, eSewa, or cash on delivery. The system will generate orders, allow users to track delivery status, and support discount coupons. Additionally, the store can recommend essential pet products (like starter kits or beds) during the adoption process. This sprint enhances the platform by providing a complete pet care ecosystem beyond adoption, making the system more practical and realistic.

**Overall Sprint Status: On Track**