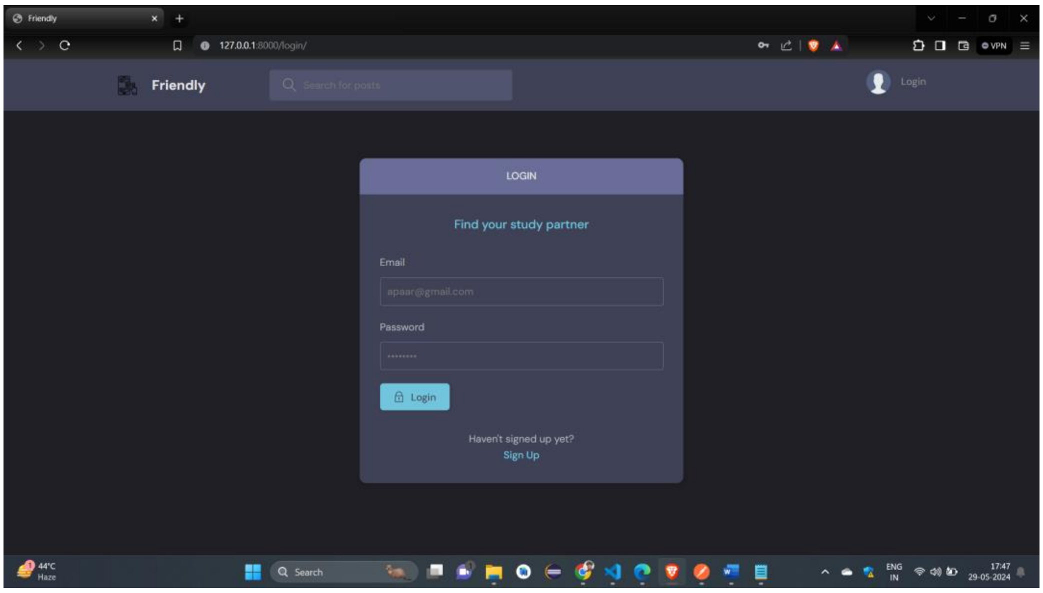
Friendly

Instead of an api I have created a social networking app, which uses Django and Django rest framework, to implement functionality, I have used frontend as well to make it look a bit interactive

I have used function based views to implement most of the functionality which makes it user friendly and easy to evaluate

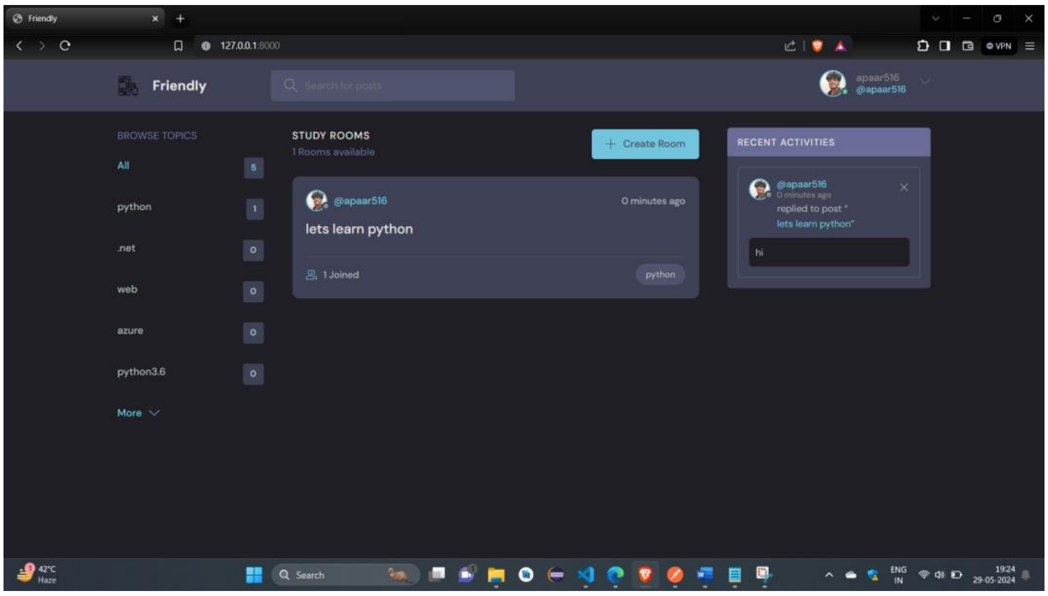
The name of the application is friendly and here are certain screenshots attached to make you understand before running it in your local machine

Here is the login page, and user can login with email which is case insensitive



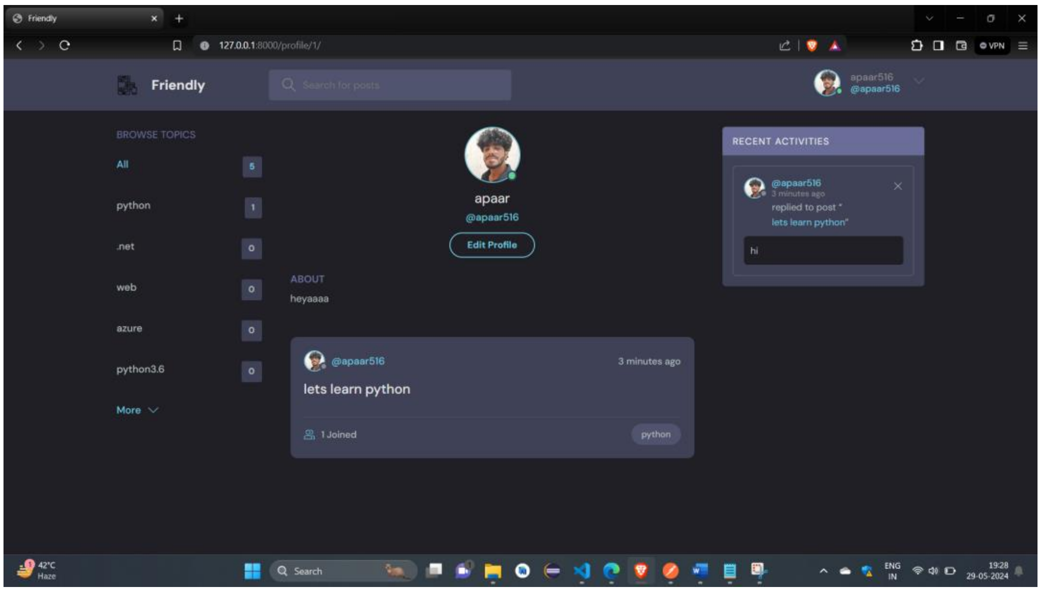
This is the home page of my application, its divided into three columns

First shows the topics, second shows the groups and third shows the recent actions



This is the user profile page for all the users,

Users can edit only their own pages, comments and messages and no anyone elses, I have taken care of that using the login_required decorator,



Now coming to the functionality part, the messages and rooms can be edited, and to show functionality and for this point

Users can not send more than 3 friend requests within a minute.

I have used a rate limiter which only allows the users to make only three three rooms per minute, after that it blocks the requests

And instead of listing friends, this is going to list which rooms has every user joined in his\her profile

Django Rest Framework

the Django REST Framework (DRF) is used primarily for creating APIs (Application Programming Interfaces) that allow communication between your web application and other systems. Let’s explore where DRF is utilized:

Authentication and Permissions:

In the loginPage view, authentication is handled using DRF.

Users can log in securely with their email and password.

API Endpoints:

DRF is used to define API endpoints for various functionalities.

For example, the createRoom view creates a chat room via an API endpoint.

The updateRoom, deleteRoom, and deleteMessage views also interact with the API.

Serializers and Views:

In the quickstart app, serializers and views are defined using DRF.

Serializers (such as UserSerializer and GroupSerializer) handle data conversion (e.g., from Django models to JSON).

Views (such as UserViewSet) provide CRUD functionality for API endpoints.

URL Routing:

In my project's urls.py, I have mapped URLs to DRF views and viewsets.

DRF helps define URL patterns for your API endpoints.

Permissions and Throttling:

For example, I have used the @ratelimit decorator to limit room creation requests.

A docker file is uploaded with the code files as well, and READ_ME contains simple steps for running this project in your local machine,

I have been working on this project since a week and this seemed like a good opportunity to present my backend skills with Django and a little frontend skills as well,

Hope you enjoy this description and looking forward to work with your esteemed company

Thank you