Requirements Specification

Summary

Signup and login functionality are a must in terms of what you need for this MVP. When a user logs into the system, their feed is the first page they will see. There they can see the tickets and reviews of all users they follow. They should also see their own tickets and reviews, as well as any reviews in response to their own tickets - even if they do not follow the reviewer. (The logic around combining querysets of different model types can be complicated. Check the appendix at the end of this specification for some guidance on how to do this.)

You can think of a ticket as a request for a review from a user. They post their ticket requesting a review for a book or literature article. Users who follow them can then submit their reviews in response to the ticket. Users should also be able to post reviews for books and articles that do not have a ticket yet.

Users will be able to follow other users and should also have the option to unfollow them. As this is just an MVP, keep this functionality fairly simple. You won't need a search functionality or a Discover feed for new users. Keep it to a simple box where you enter the username of the user you wish to follow. You should also have a page that lists all the users the logged-in user follows with the option to unfollow.

You will also need another page from which users can review their own submissions. They should be able to see their posts and edit and delete them from this page.

Remember, this is an MVP, so try not to get too caught up on styling. Focus more on a clean and minimal UI. However, you should ensure things like date formats, styling, etc. are consistent across the site. Follow the layout of the wireframes provided, but don't be afraid to add some personal touches if you wish, remember, clean and minimal.

A user will need to:

- Log in and sign up - the site should not be accessible to a non-logged-in user.
- View a feed containing the latest tickets and reviews from users that they follow ordered by time with the latest first.
- Create new tickets requesting a review on a book/article.
- Create reviews as a response to tickets.
- Create reviews not in response to a ticket. As part of a one-step process, the user will create a ticket and then a review responding to their own ticket.
- Be able to view, edit, and delete their own tickets and reviews,
- Follow other users by entering their username,
- View who they follow and unfollow whoever they want.

A developer will need to:

- Create a local environment using venv and run the site based on the detailed documentation laid out in the README.md.

The site will need to:

- Have a UI matching those of the wireframes.
- Have a clean and minimal UI.
- Use server-side rendering to display information from the database on the page dynamically.

The codebase will need to:

- Use the Django framework.
- Use the Django template language for server-side rendering.
- Use SQLite as a local development DB (your db.sqlite3 file should be included in the repository).
- Have a database design that matches the database schema. Have syntax that meets PEP8 guidelines.

Appendix

Example for combining querysets of different models for the Feed

```
# in views.py
from itertools import chain

from django.db.models import CharField, Value
from django.shortcuts import render


def feed(request):
    reviews = get_users_viewable_reviews(request.user)
    # returns queryset of reviews
    reviews = reviews.annotate(content_type=Value('REVIEW', CharField()))

    tickets = get_users_viewable_tickets(request.user)
    # returns queryset of tickets
    tickets = tickets.annotate(content_type=Value('TICKET', CharField()))

    # combine and sort the two types of posts
    posts = sorted(
        chain(reviews, tickets),
        key=lambda post: post.time_created,
        reverse=True
    )
    return render(request, 'feed.html', context={'posts': posts})


# in feed.html
# Use the 'include' tag to reuse ticket and review elements between pages

...

{% for post in posts %}
    {% if post.content_type == 'TICKET' %}
        {% include 'ticket_snippet.html' %}
    {% elif post.content_type == 'REVIEW' %}
        {% include 'review_snippet.html' %}
{% endfor %}

...
```