

I have implemented additional functionality beyond the specification by adding a simple search engine inside the Cloudysky app. The goal of this feature is to let users enter a query and retrieve all posts and comments containing any of the words in that query. Matching posts and comments and comments are displayed along with any other associated data such as username, timestamp, and Post ID. This makes navigating the content much easier compared to manually scrolling or relying on curl commands.

To create this search engine, I decided to make it within the cloudysky_app by creating a new view in views.py, and adding a corresponding URL route in urls.py. I also designed a HTML page that allows for a more user-friendly way to submit search queries, and also so curl commands are not required.

The search engine works in 4 main steps:

1. It receives the user's query and splits it into individual words
2. Each word is compared against the text of every post and comment in the database
3. All matching posts and comments are displayed on a HTML results page
4. Current implementation supports case insensitivity for comments and case sensitivity for posts

The feature handles GET requests, does keyword-based matching over both Media_obj.content_text and Comments.comment_content. After a user submits a query, all matching posts and comments appear in a clean format. This feature can also be used and built upon for further applications such as content discovery, moderation, or even recommendation features. For future improvements of the search engine, I would layer it so posts that are most identical show up first, and least identical show up last.

In addition to the search engine, I enhanced the front-end to make CloudySky more visually appealing and user-friendly. I updated the HTML templates for the login page, home page, and forms. I also added buttons for creating posts, creating comments, viewing the feed, and accessing the search engine. These improvements make the app much more user-friendly as you no longer have to rely on curl commands for basic features.