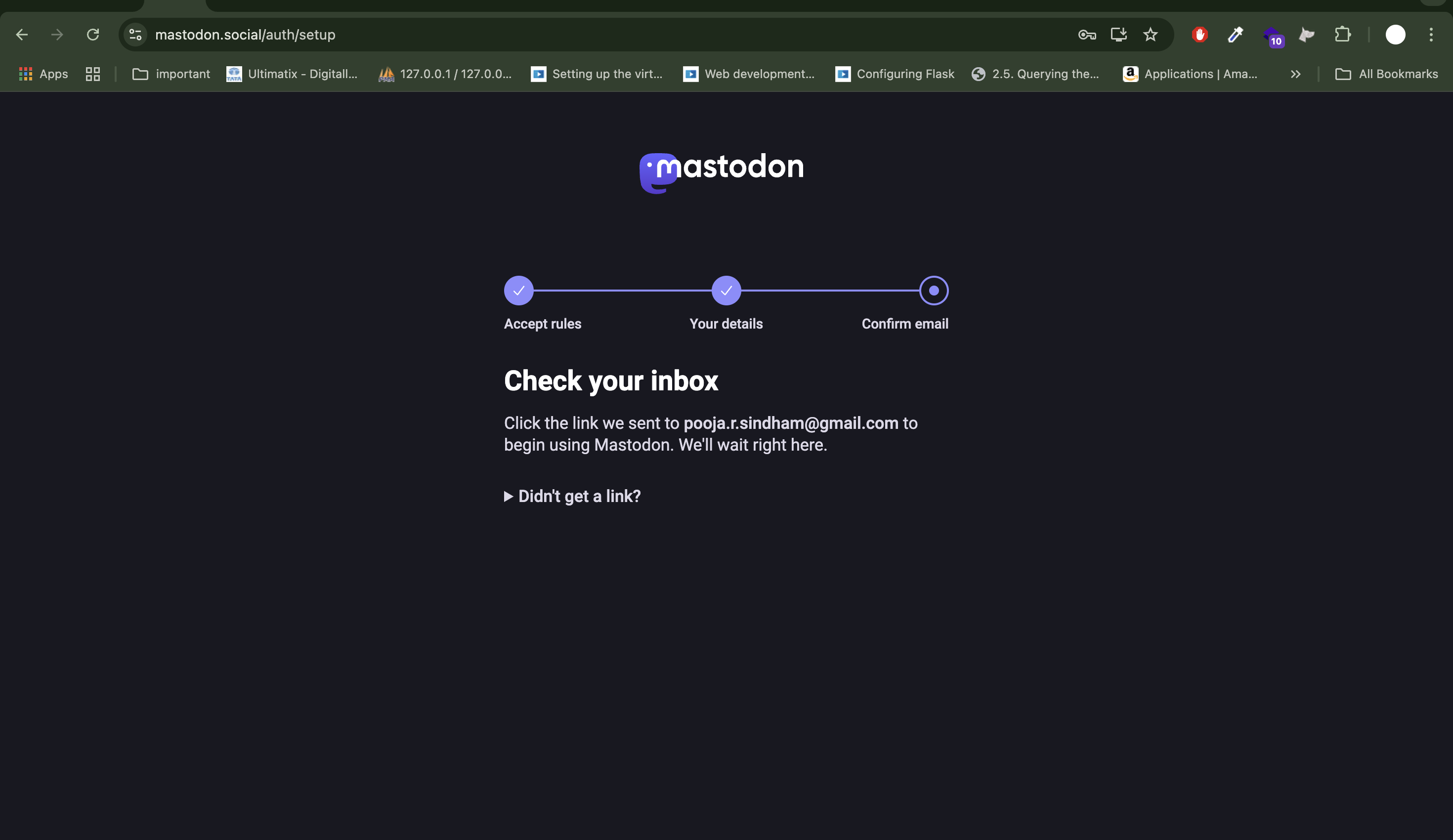
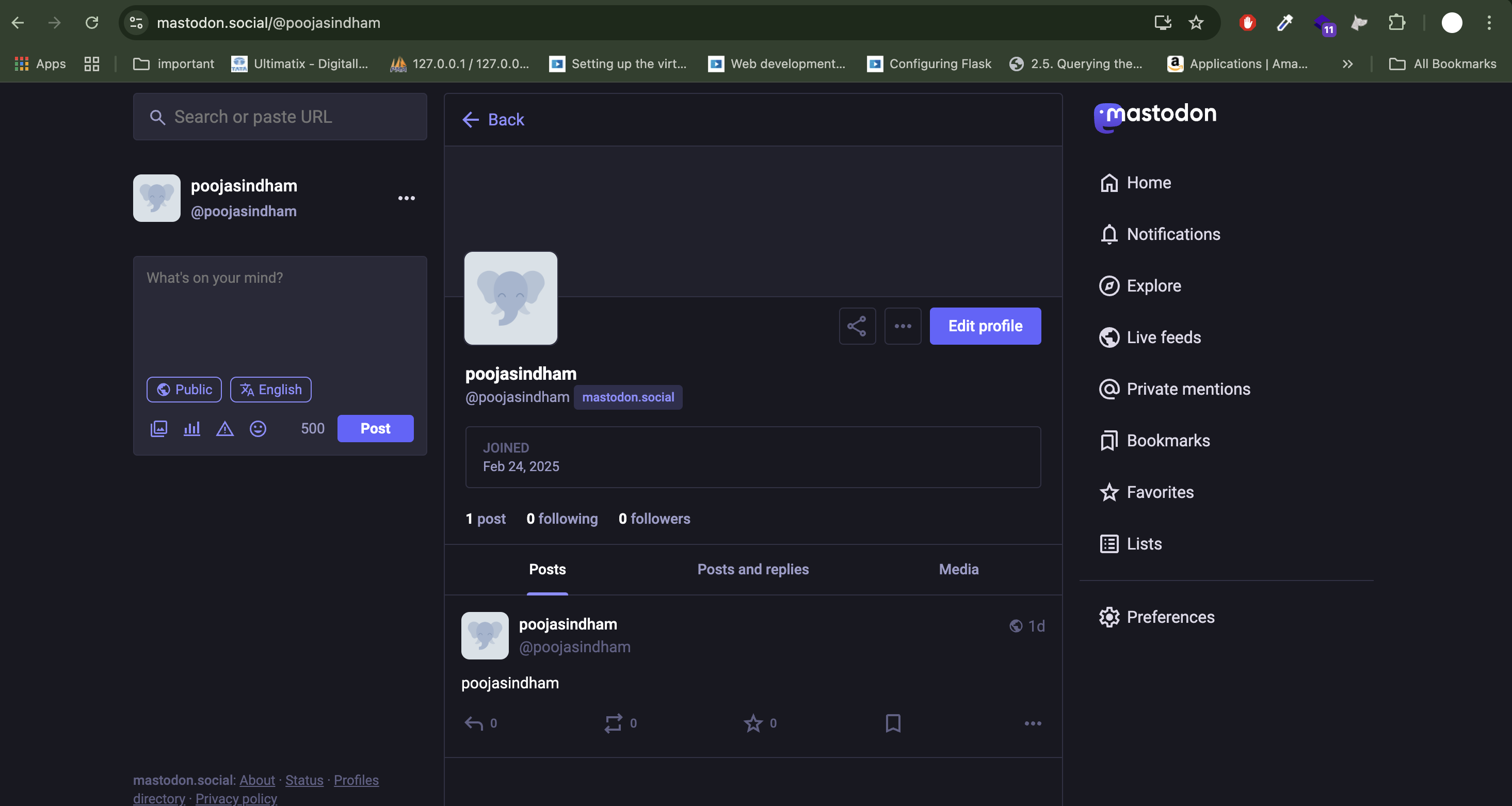
## **Homework #2 – Twitter Service**

***Aarsh Sheth || Pooja Sindham || Shivani Jariwala || Aishwariya Indi***

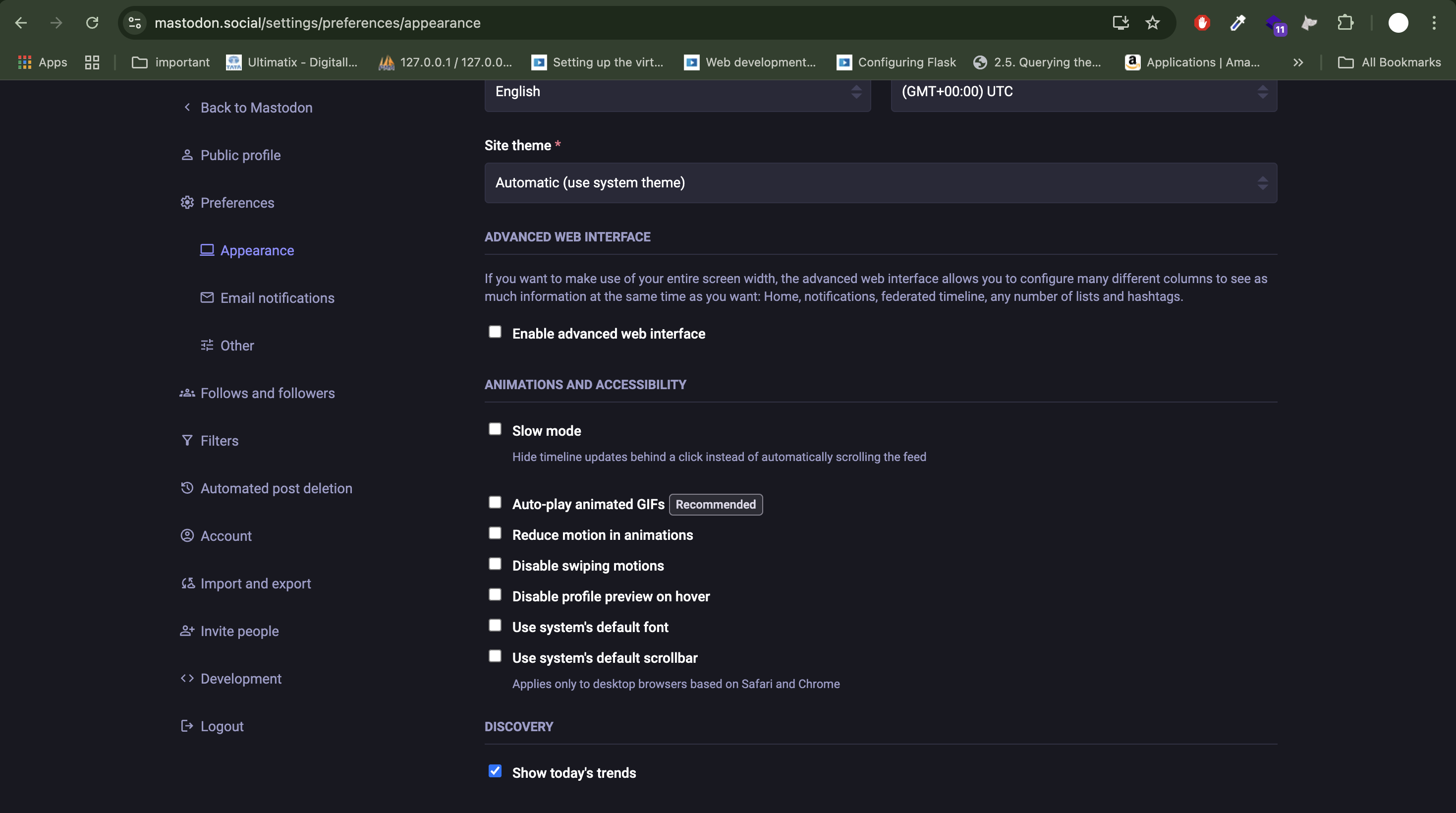
[**GitHub Link**](https://github.com/wh0th3h3llam1/cmpe-272/tree/main/HW_2_Twitter_Service)

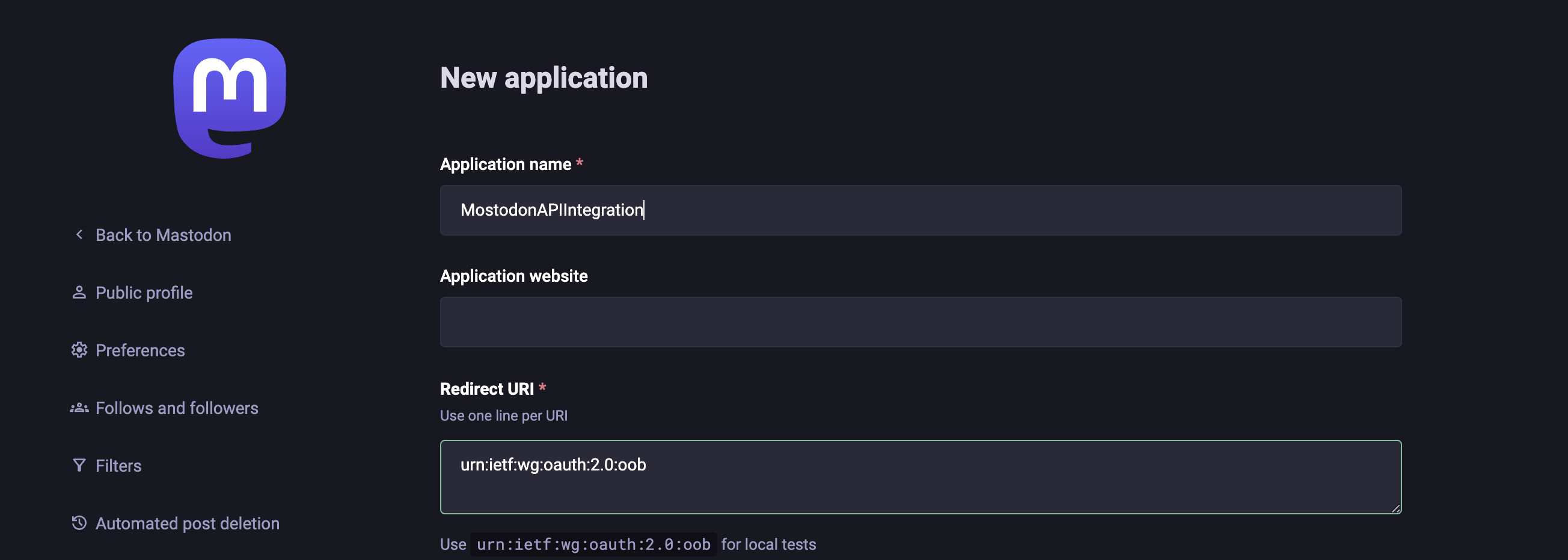
**Get Access token for Mostodon Social**

1. Signup for mastodon social.  
   
2. Login to your account and go to preferences.



1. Navigate to Preferences > Development.

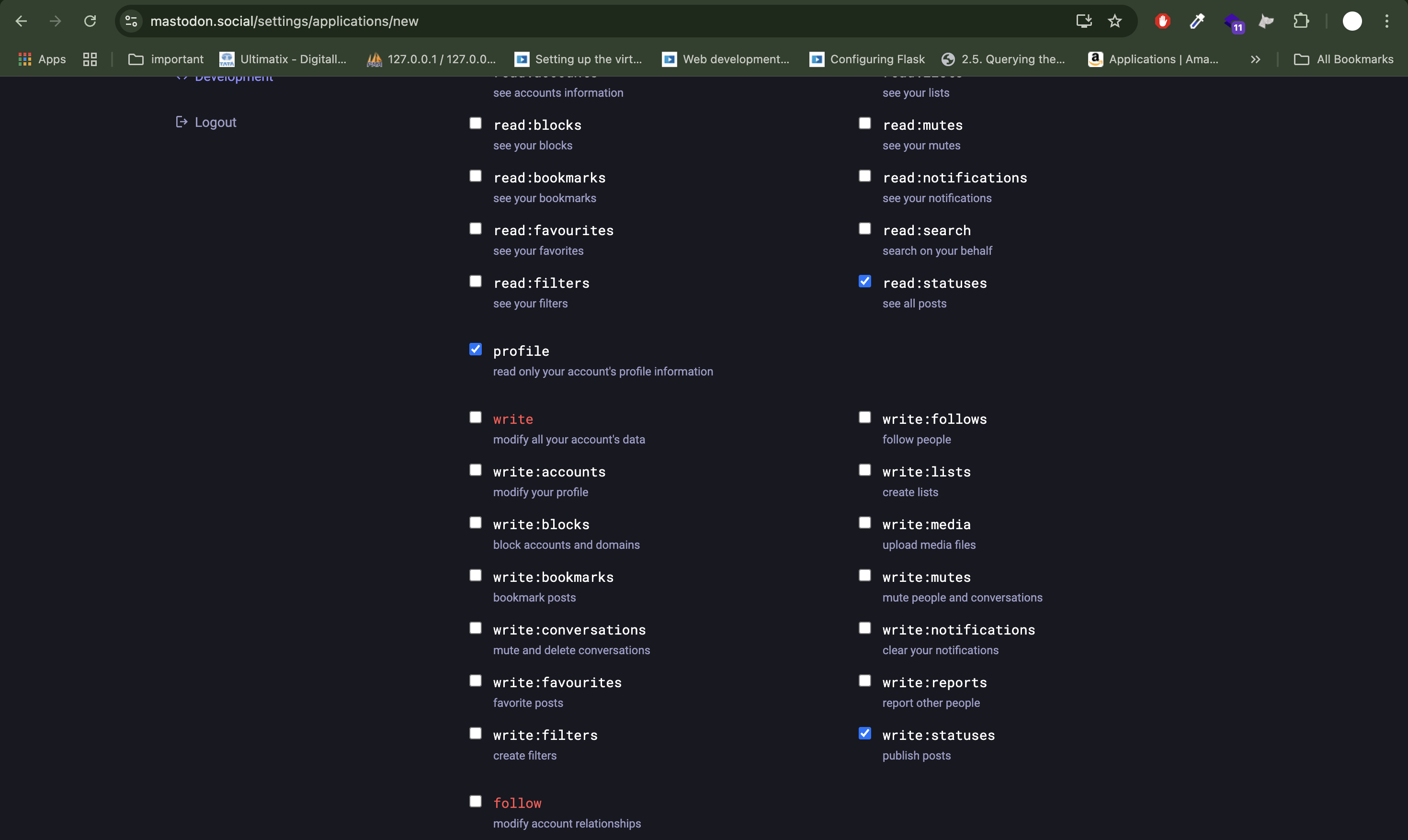


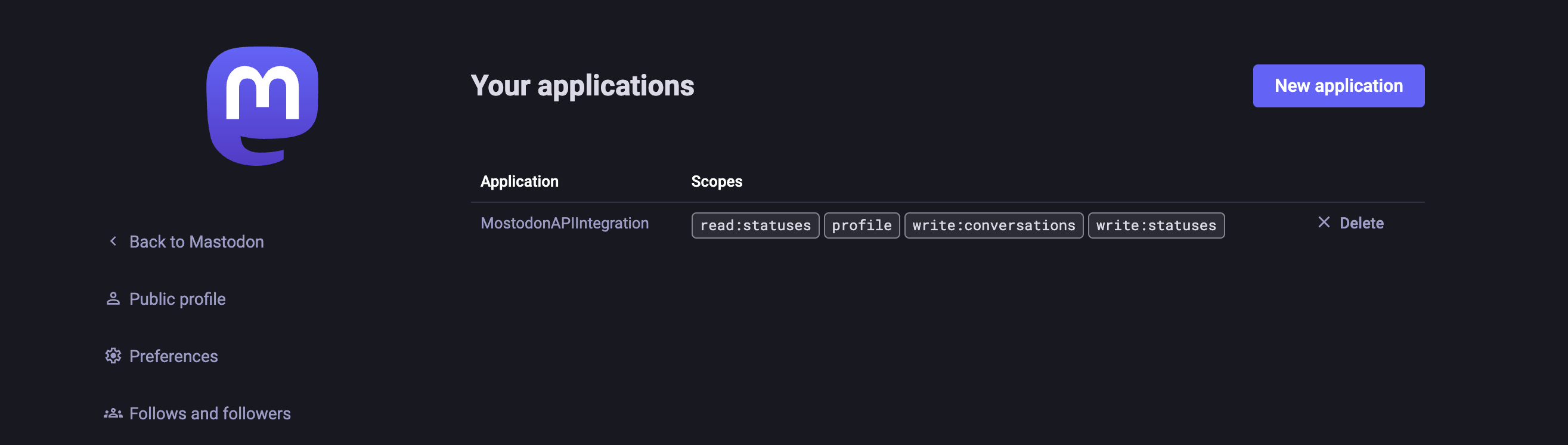
1. Create new application.  
   

Select the following options

*- read:statuses (Retrieve posts)*

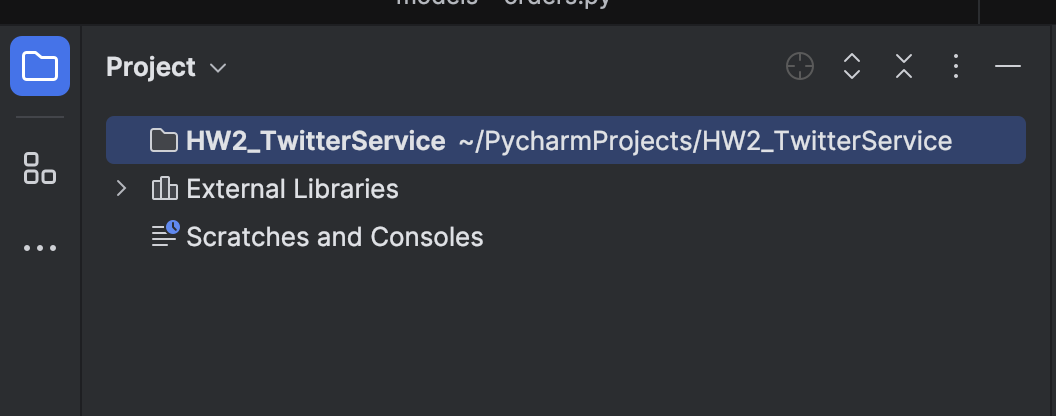
*- write:statuses (Create and* ***delete*** *posts)*



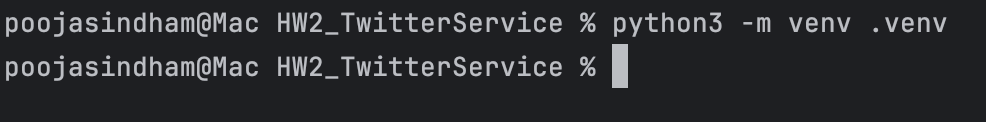
1. Click **Submit** and save the **access token**.  
   

**Create Mastodon Service**

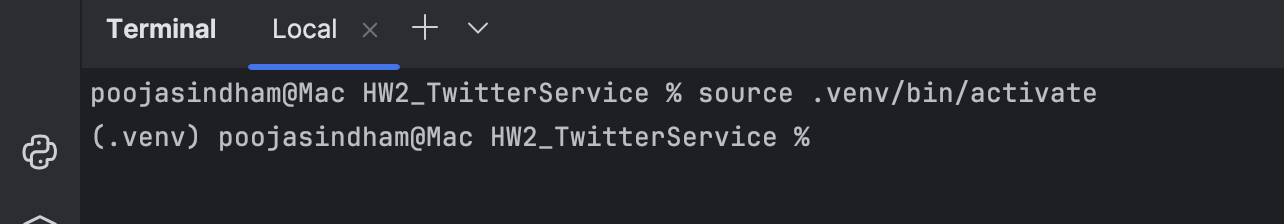
1. Create a new project - HM2\_TwitterService

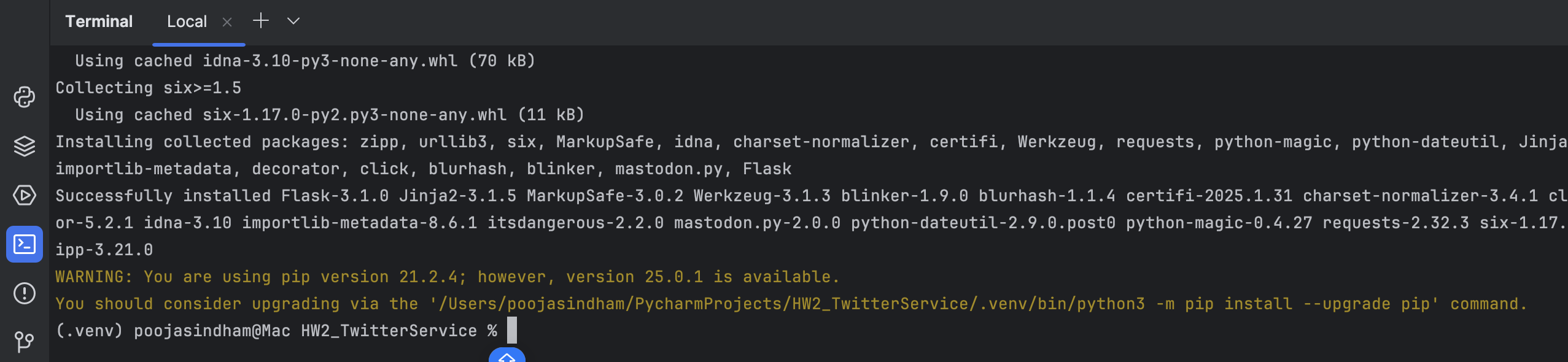
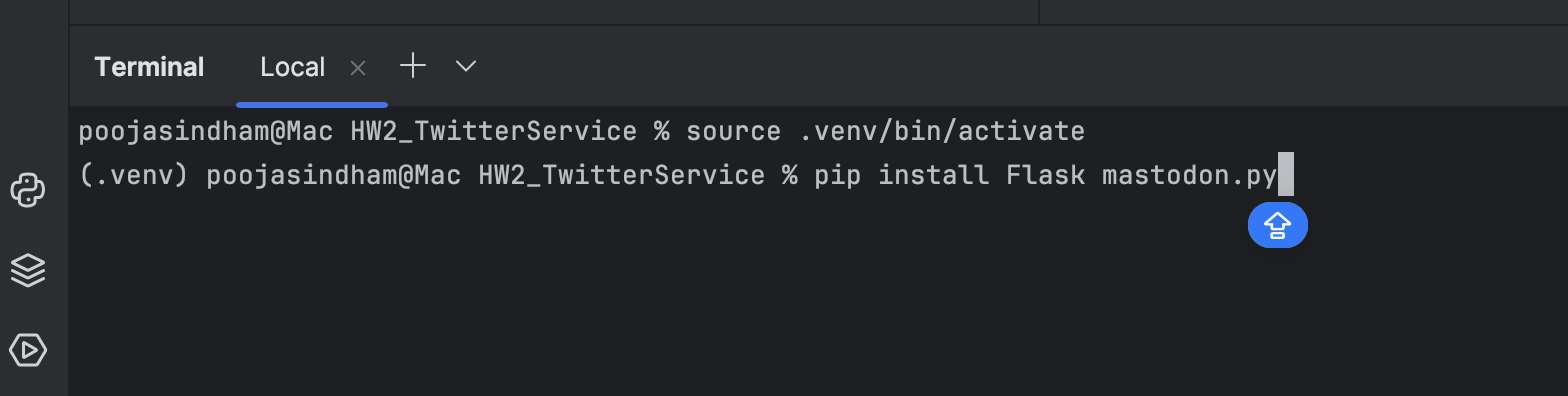
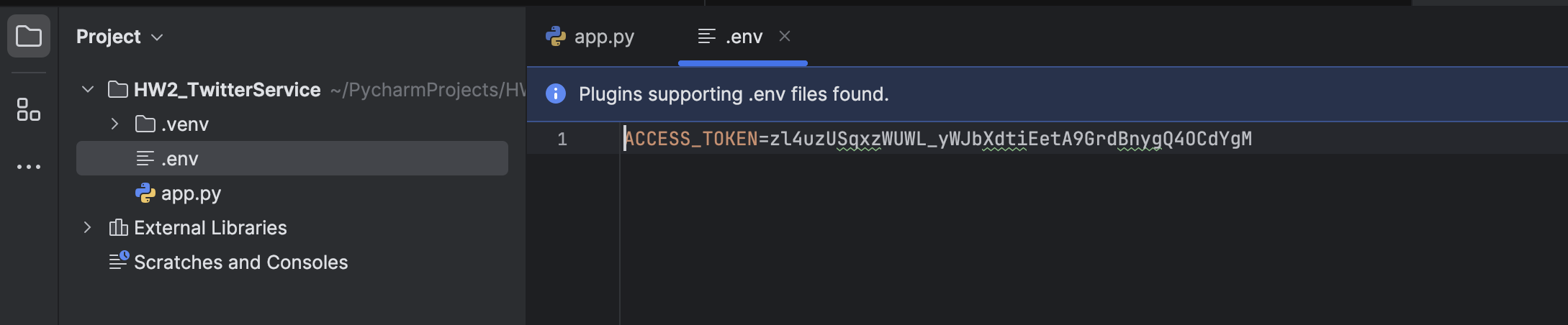


1. Create virtual environment.



1. Activate the virtual environment.



1. Install packages.  
   Here we are developing API through Flask and Mastodon API package.  
   
2. Create .env file to store Access Token.  
   
3. Set up flask server code to create, retrieve and delete post.  
   Create app.py
4. Import the required modules



**Flask**: Web framework for handling HTTP requests.

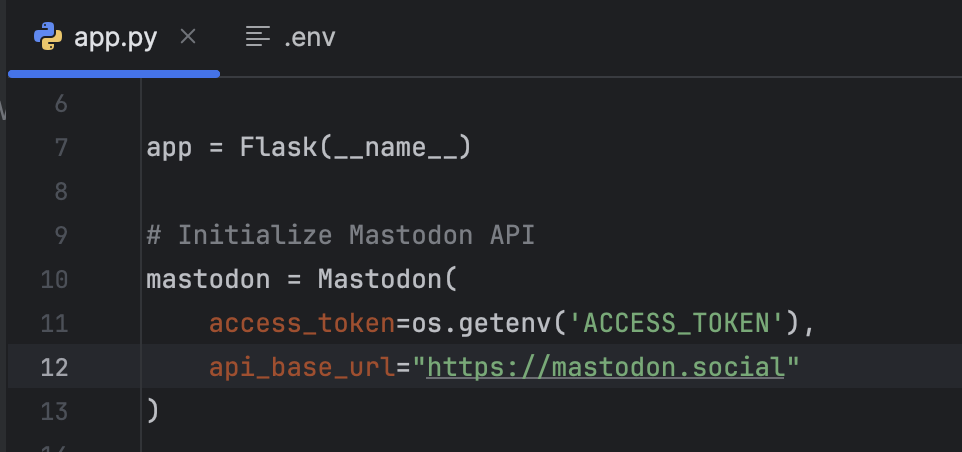
**render\_template**: Renders HTML templates.

**request**: Gets data from the frontend.

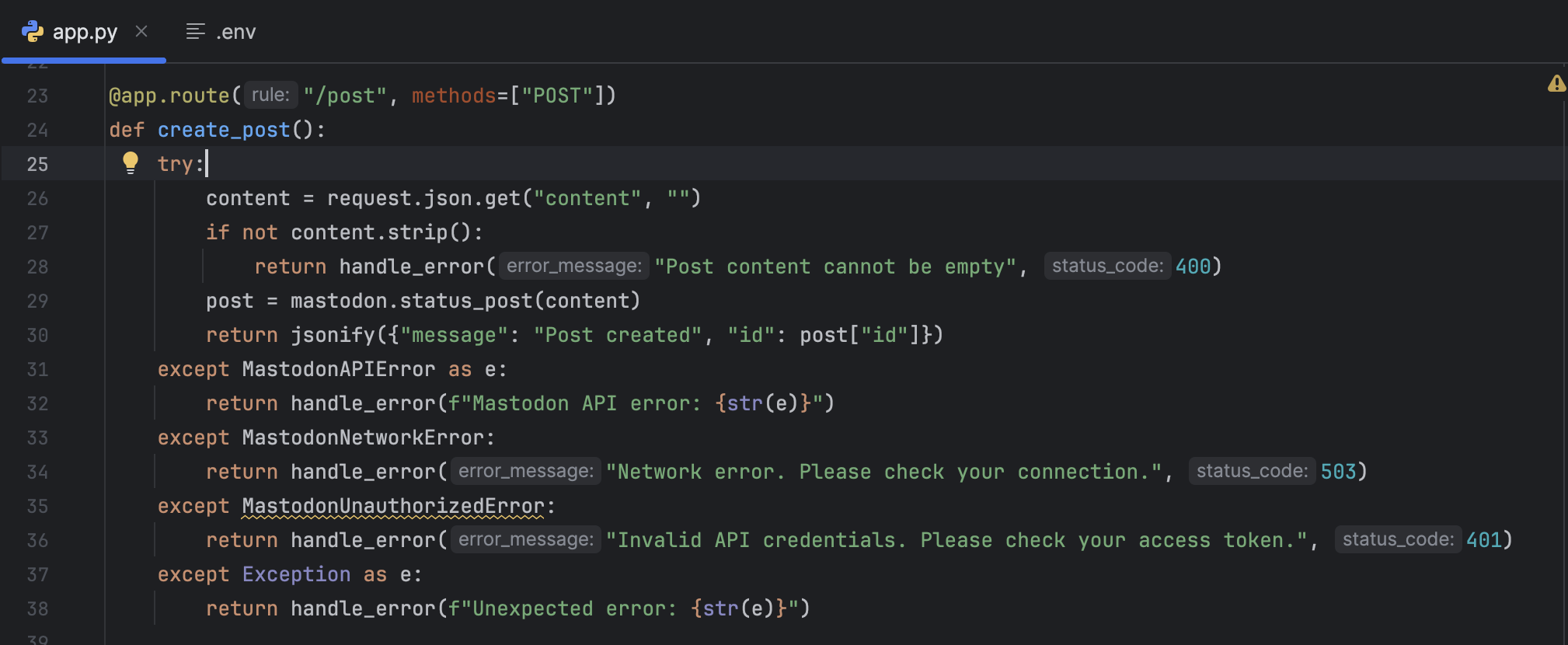
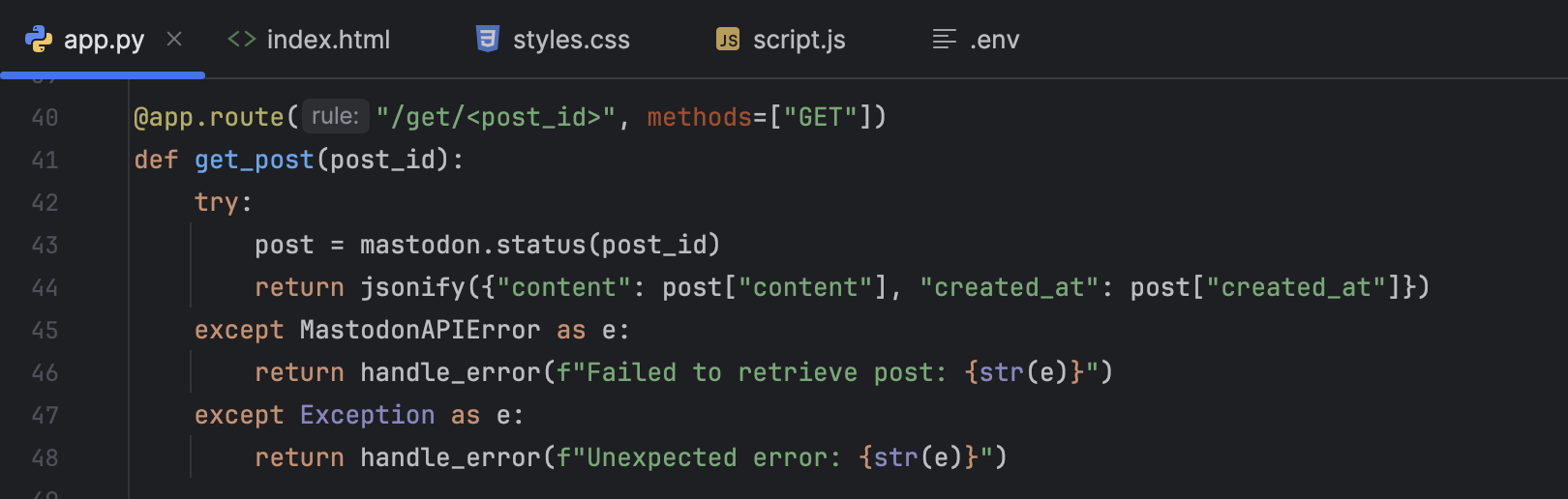
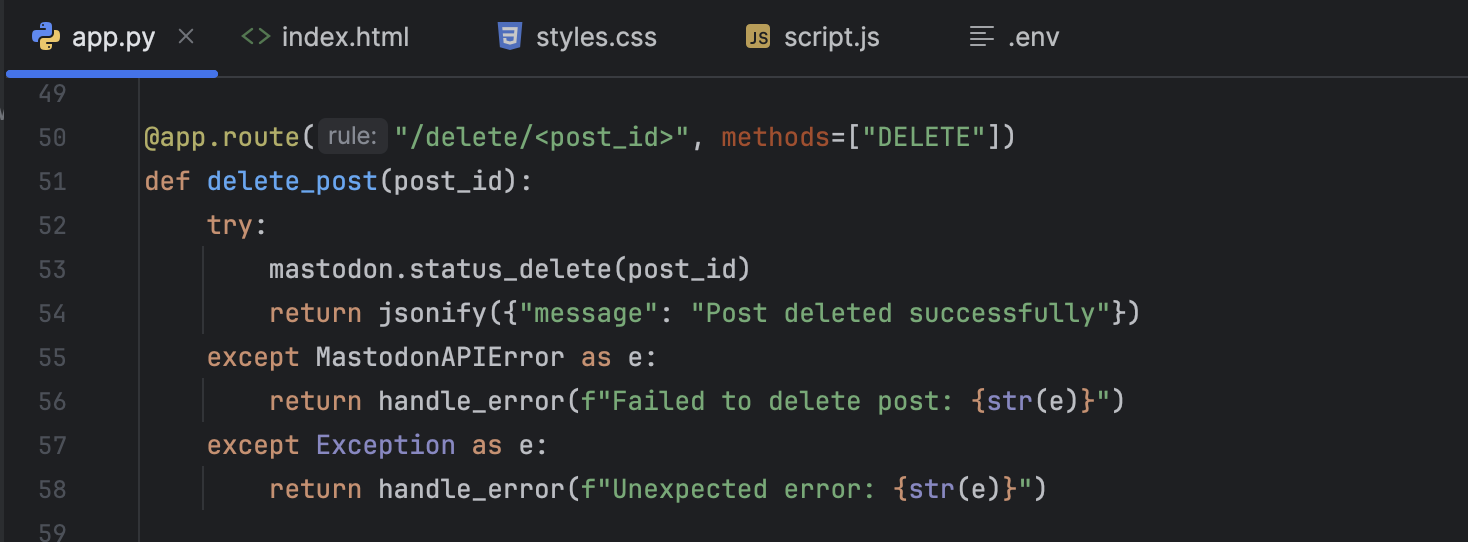
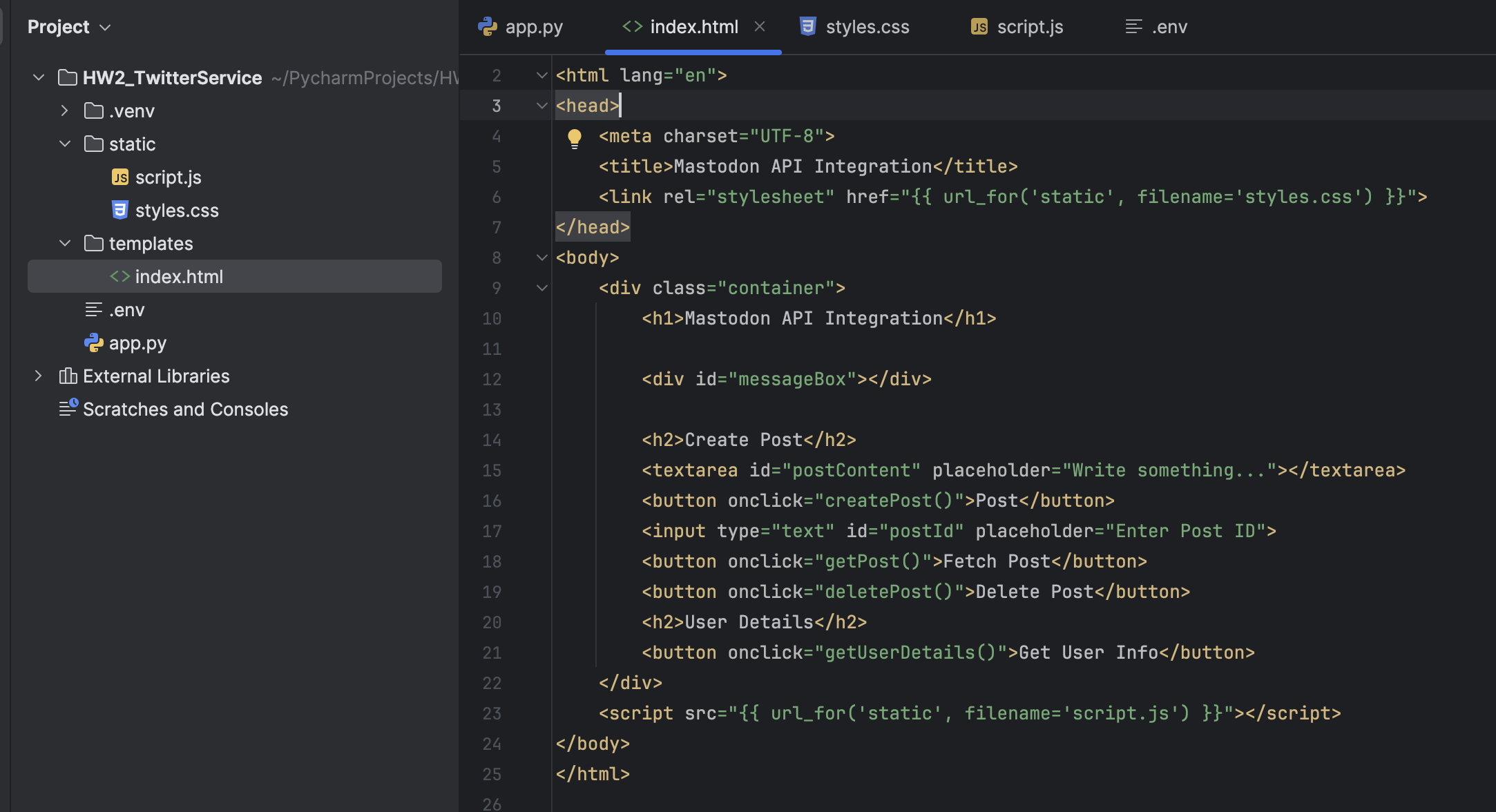
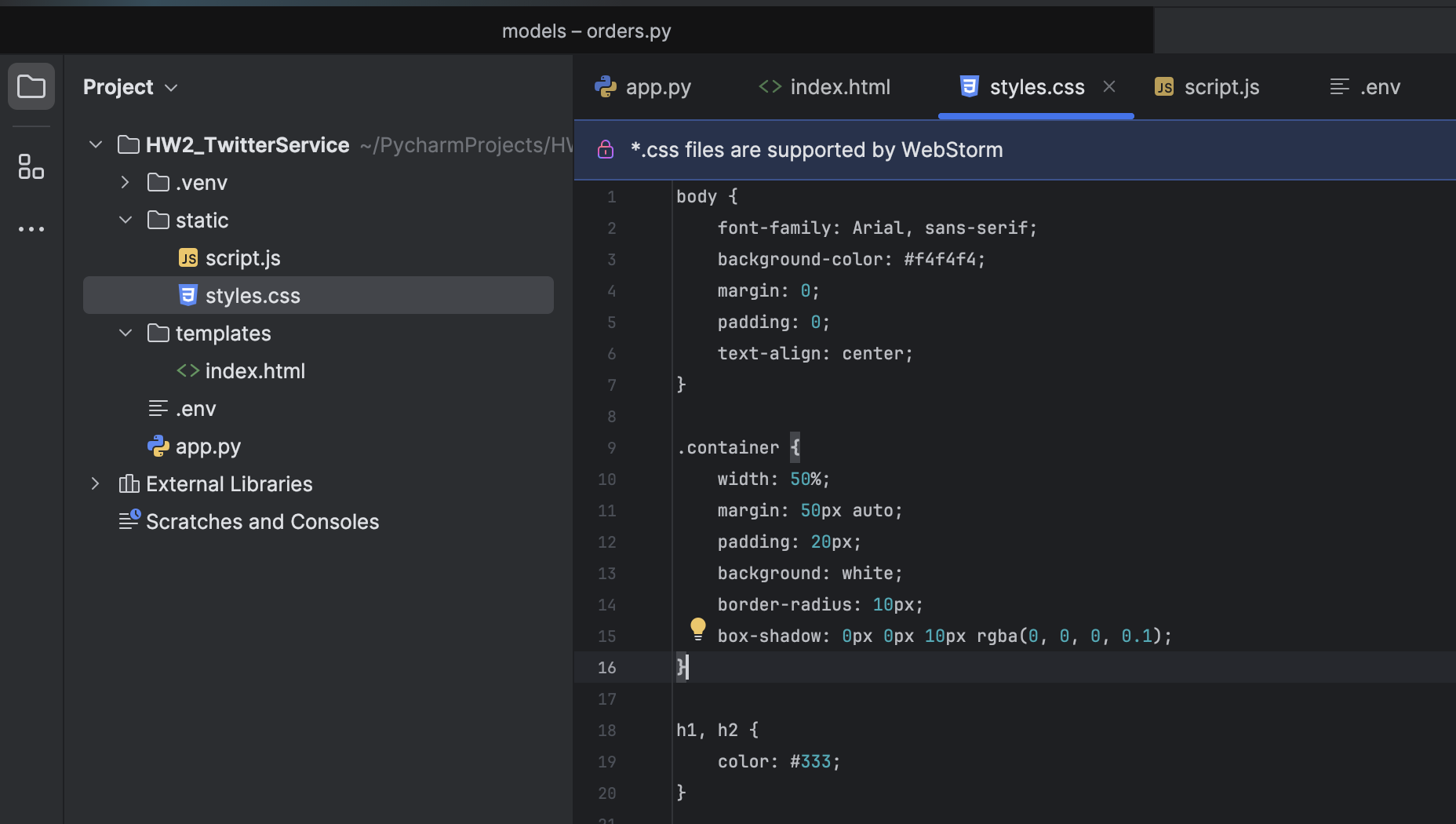
**jsonify**: Converts Python dictionaries to JSON format.

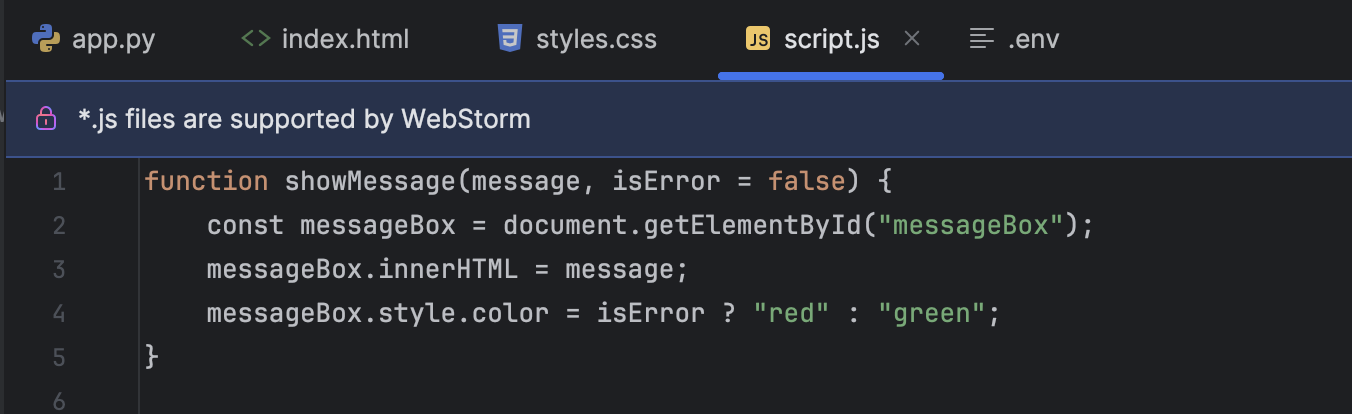
**mastodon**: Mastodon API wrapper for Python.

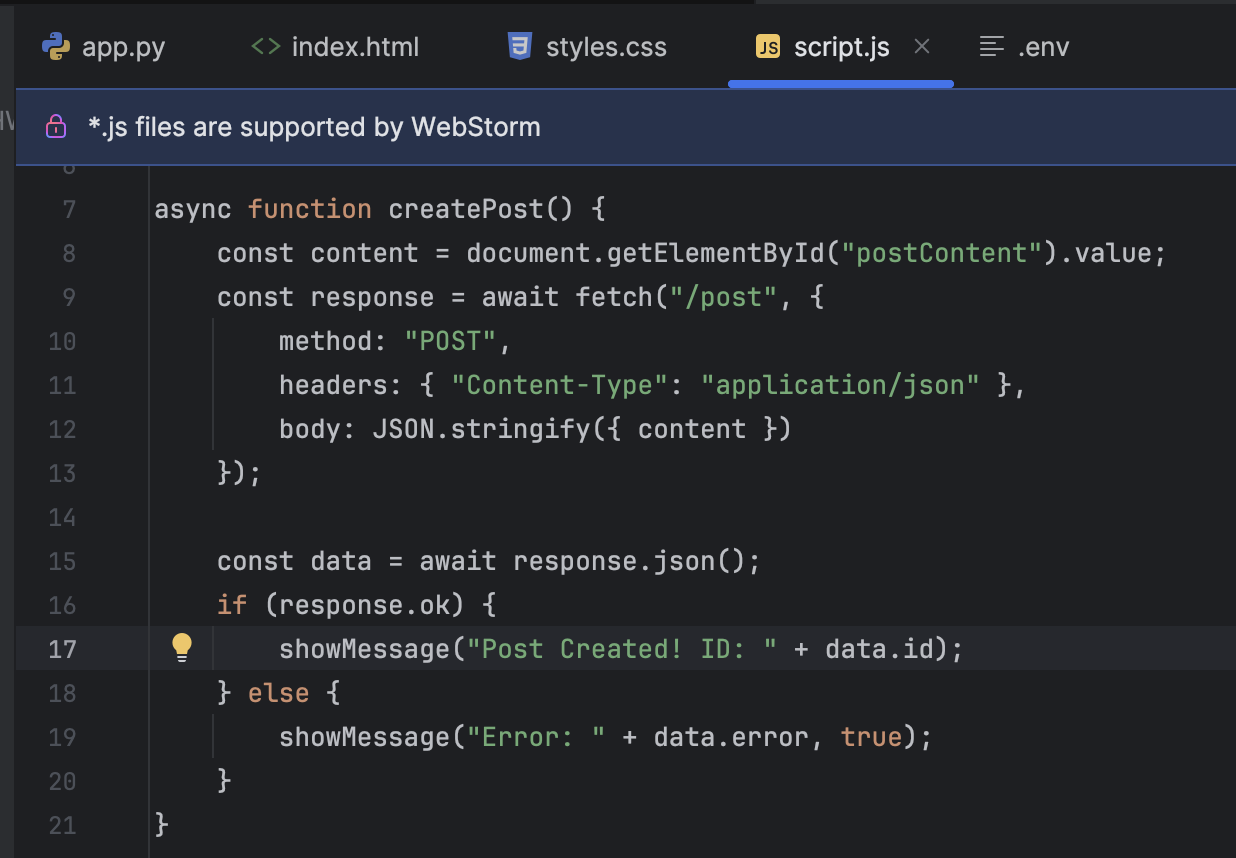
**MastodonAPIError, MastodonNetworkError, MastodonUnauthorizedError**: Handles different API errors.

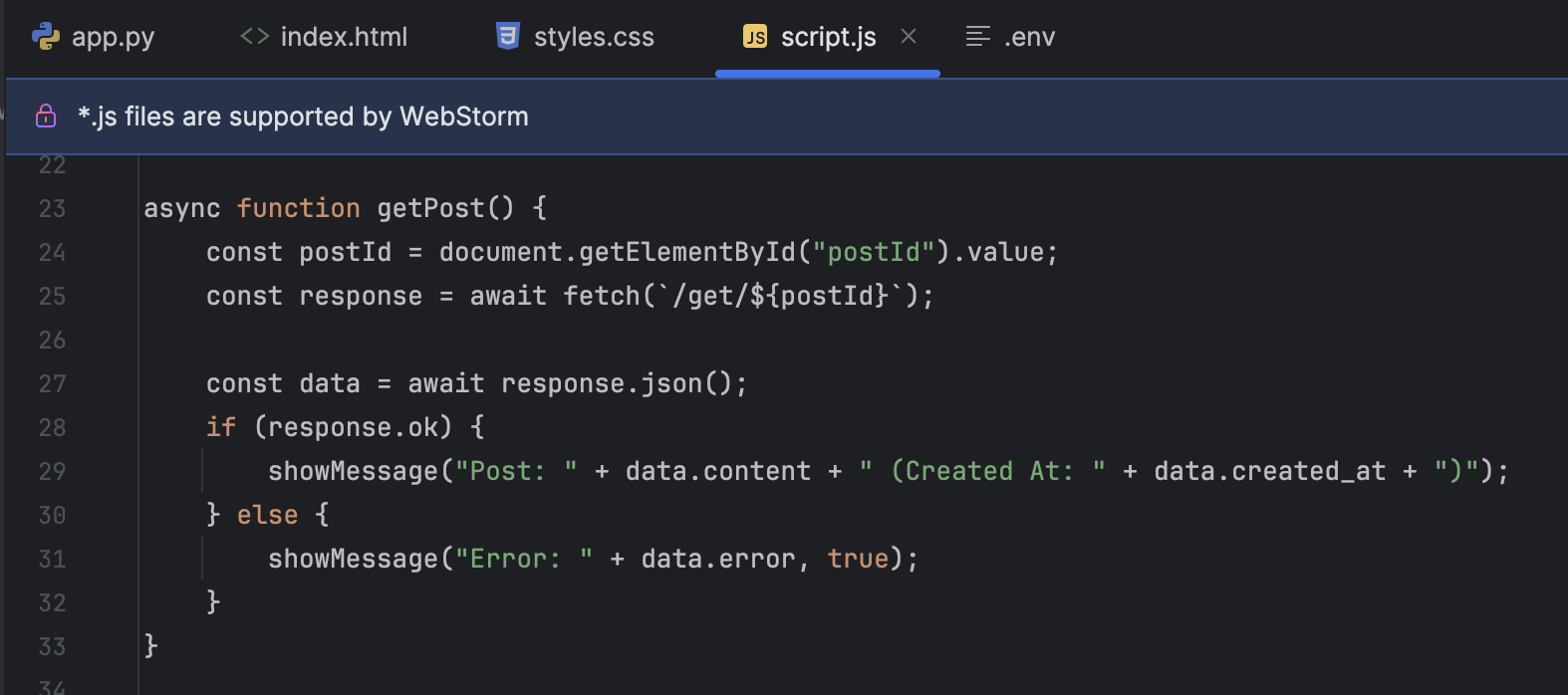
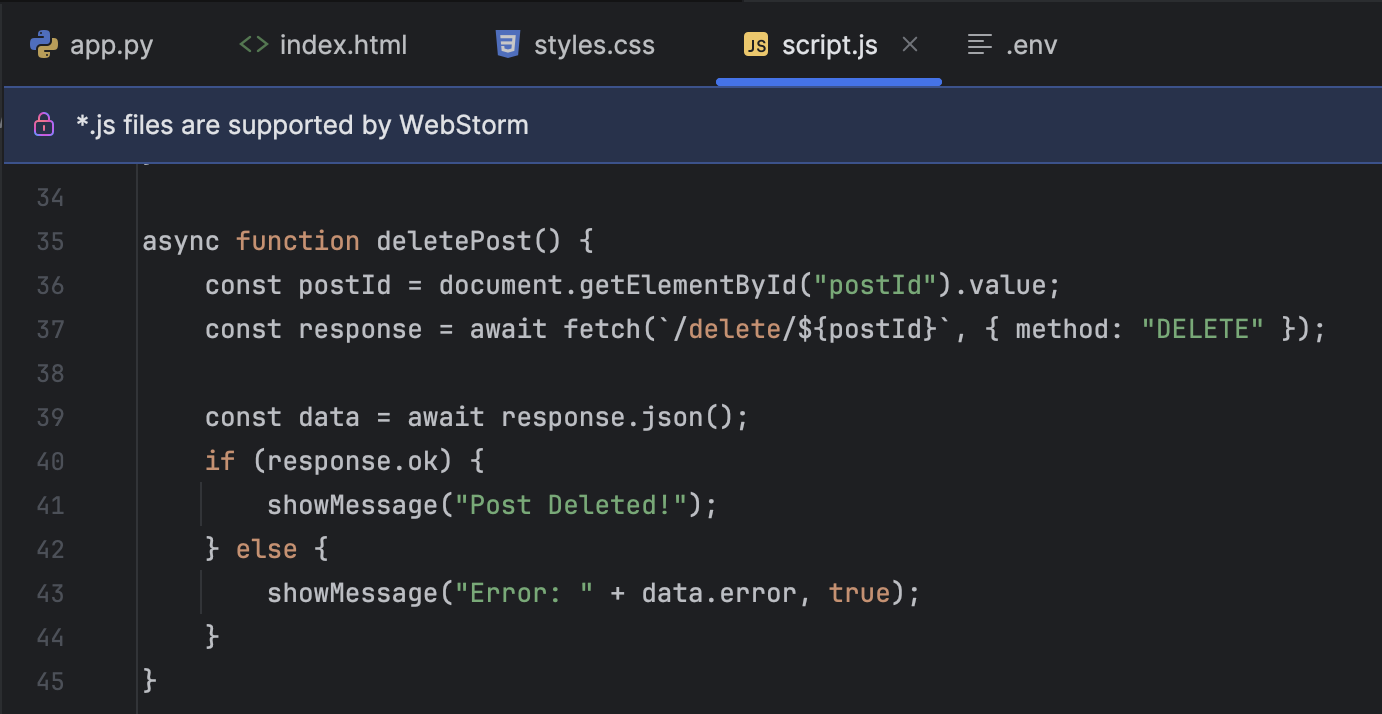
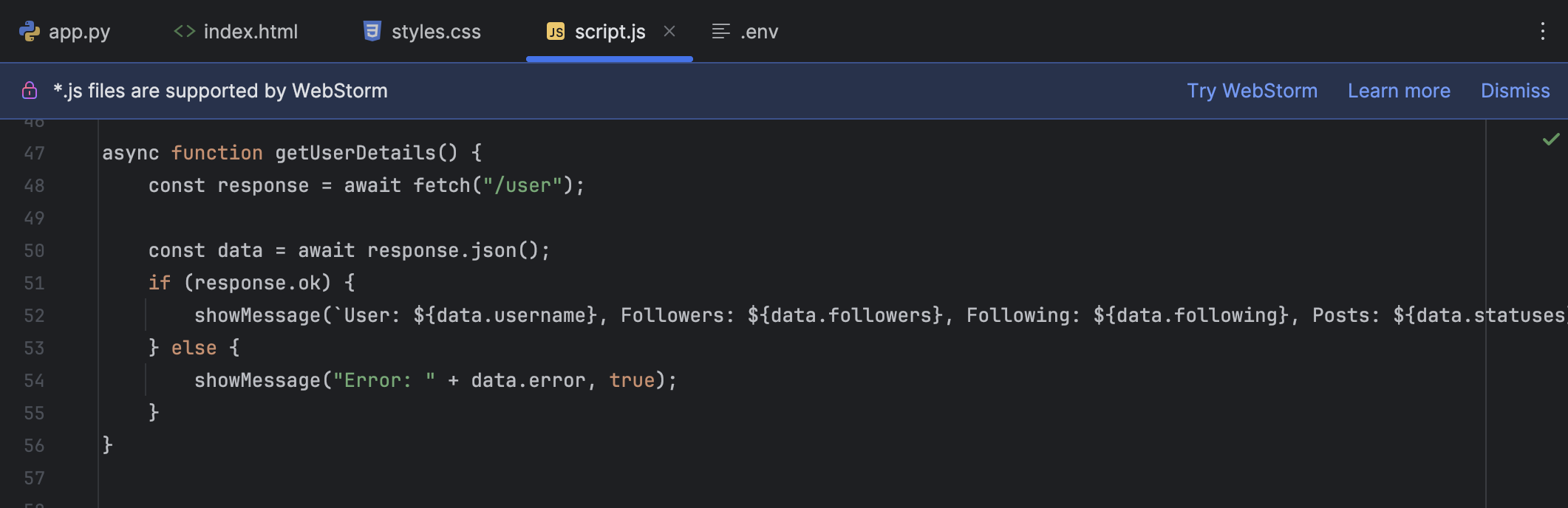
1. Initialize Flask and Mastodonapp = Flask(\_\_name\_\_): Creates a Flask app instance.

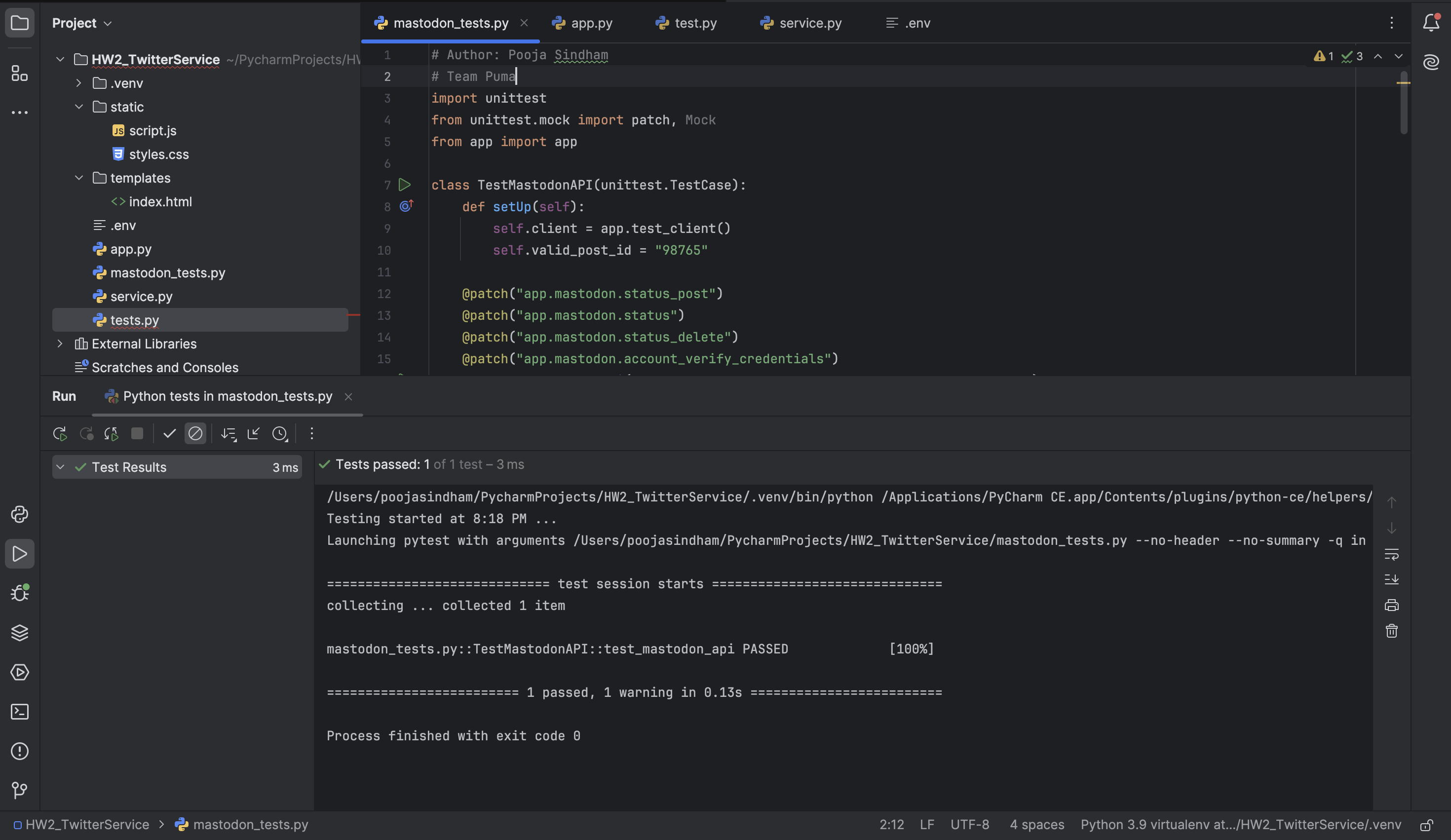
Mastodon(...): Connects to the Mastodon API using an access token.

1. Function for handling errors  
   This function will return JSON responses for errors with a message and HTTP status code.
2. Home Page  
   This will render index.html.
3. CreatePost  
   
4. Get Post  
   
5. Delete Post  
   
6. Get User Details  
   
7. Simple UI to interact with Flask API
8. index.html in templates directory  
   
9. Add Simple CSS for look and feel of UI. styles.css files under static folder.  
   
10. JavaScript to handle API request and error handling

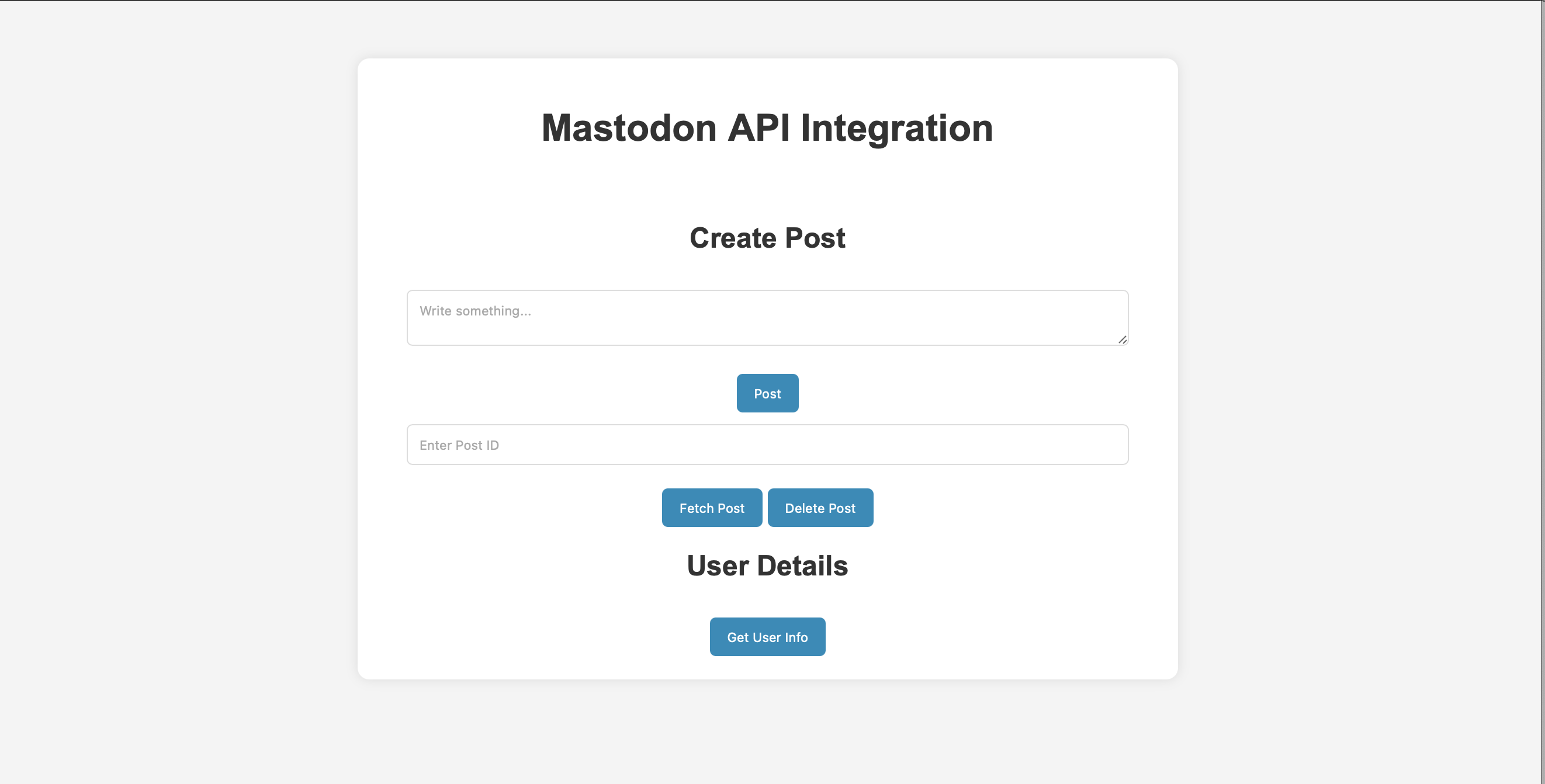
* Display Message  
  
* Create Post

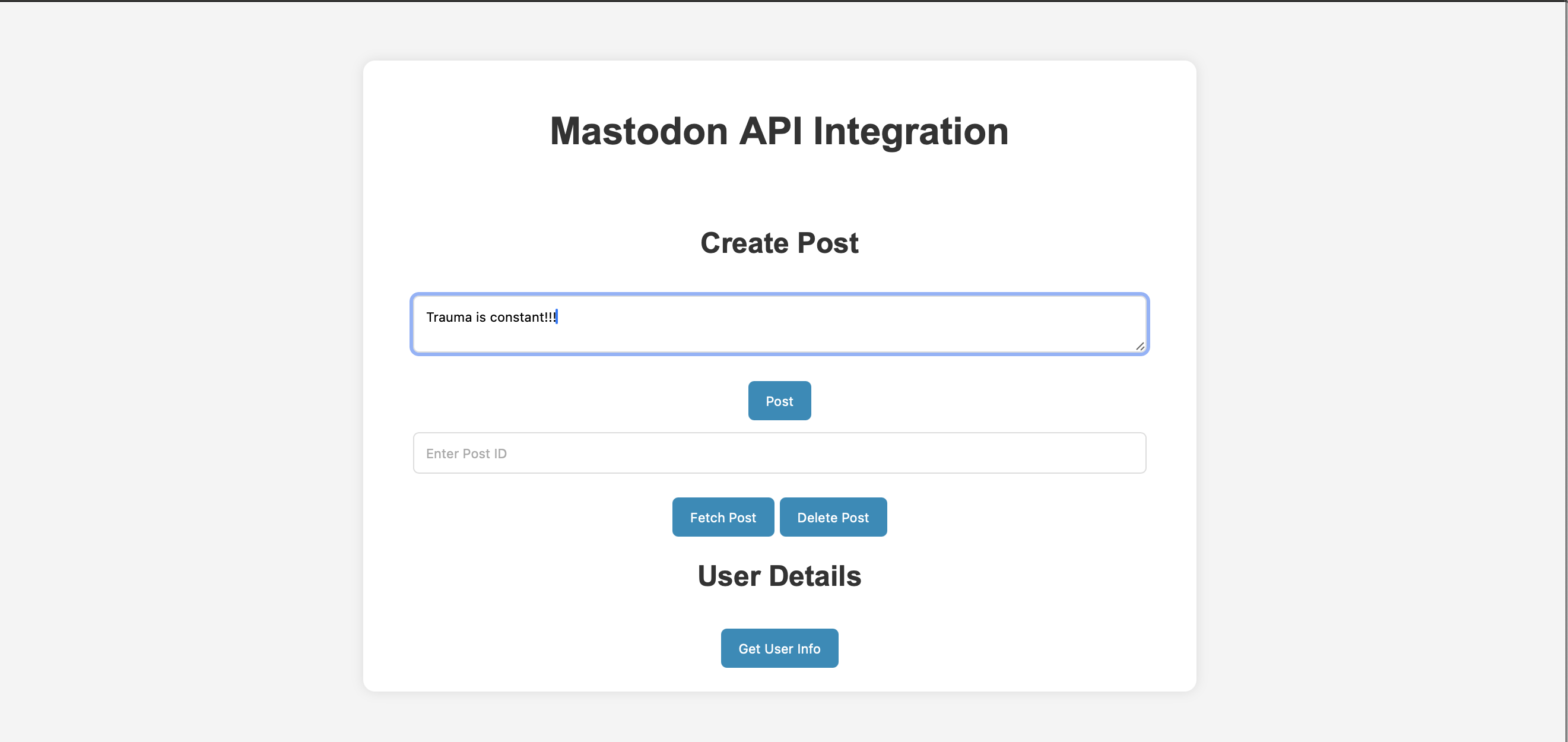


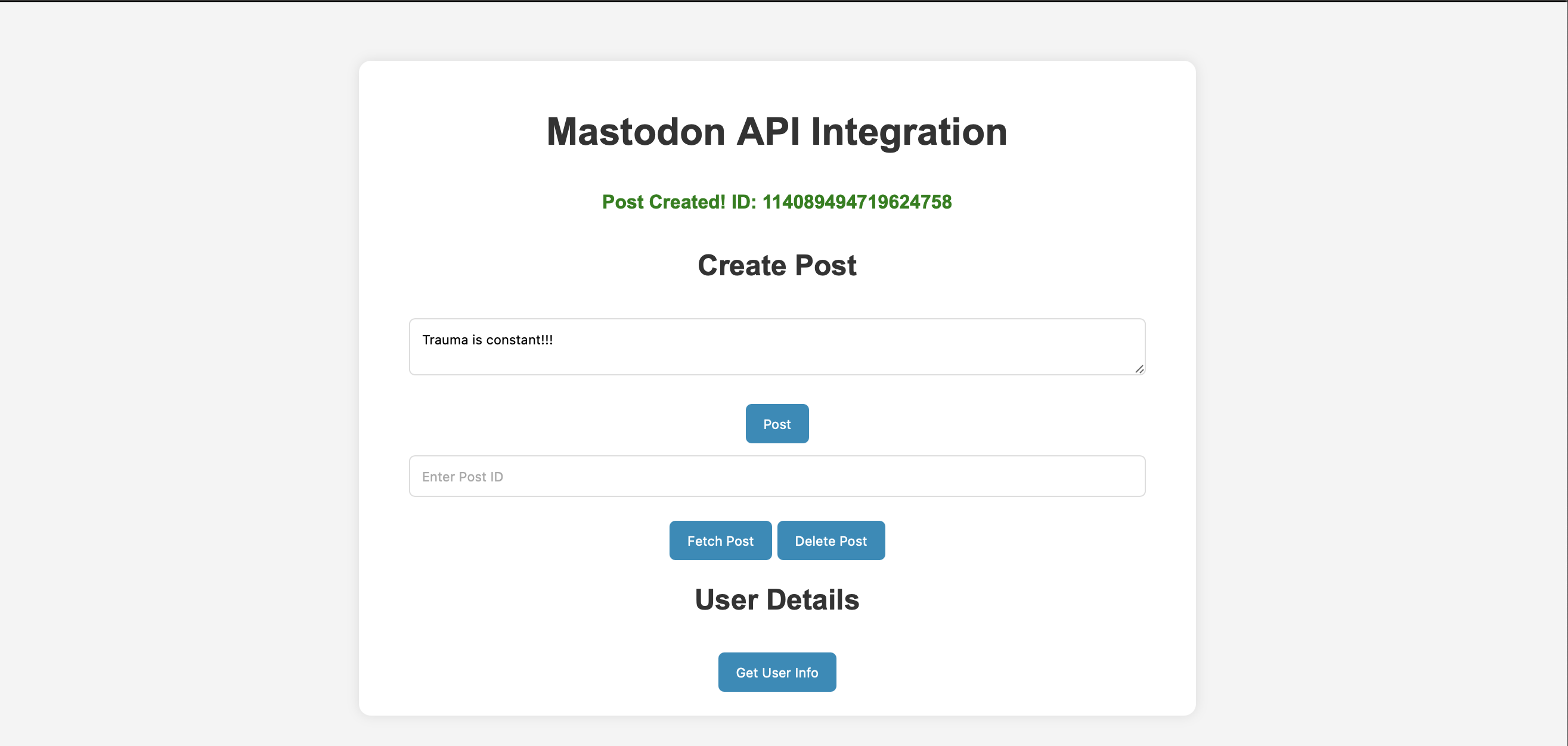
* Get Post  
  
* Delete Post  
  
* Get User Details  
  

1. Unit tests.  
   

**UI (Frontend)**

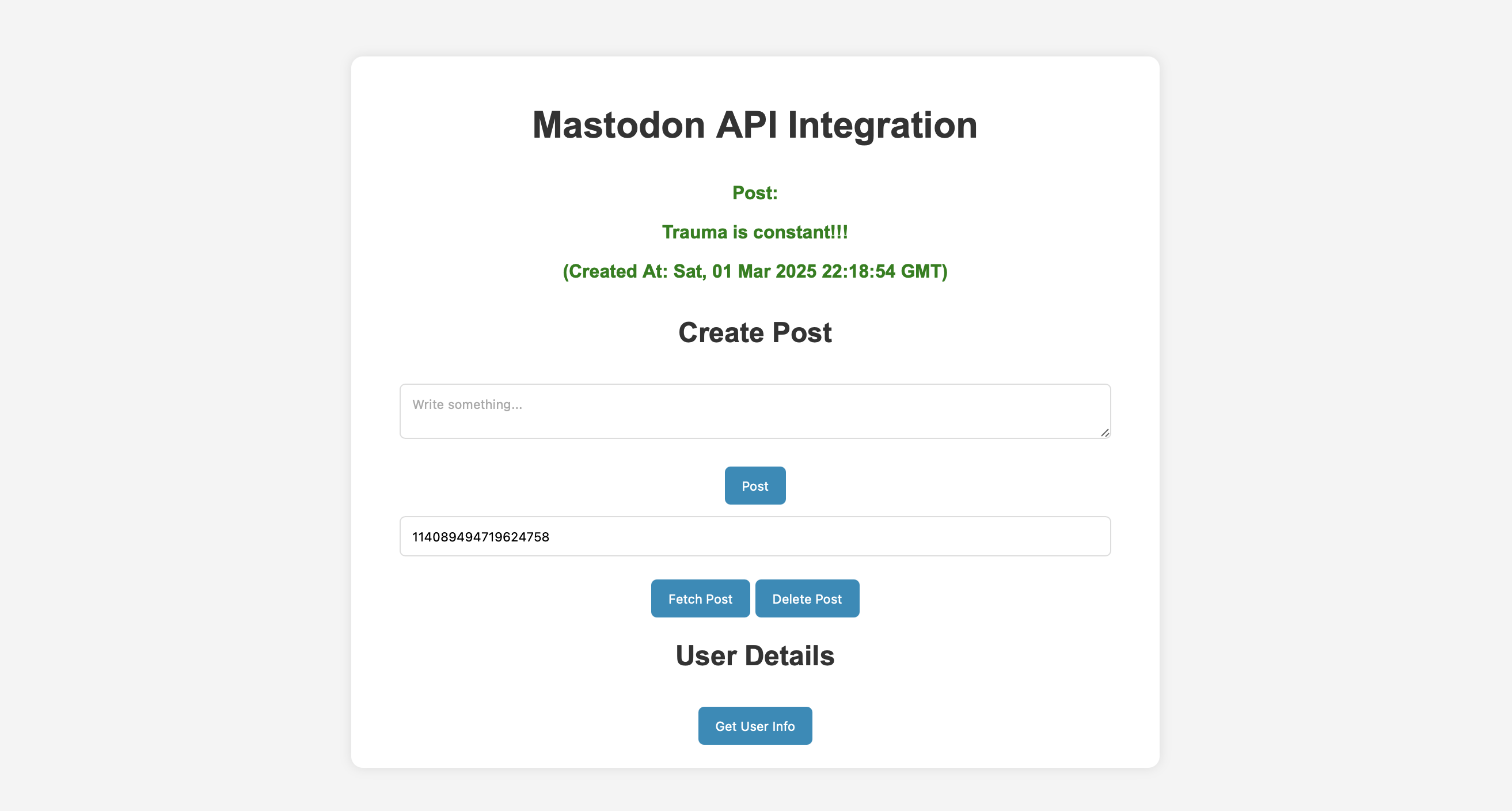
1. Home Page  


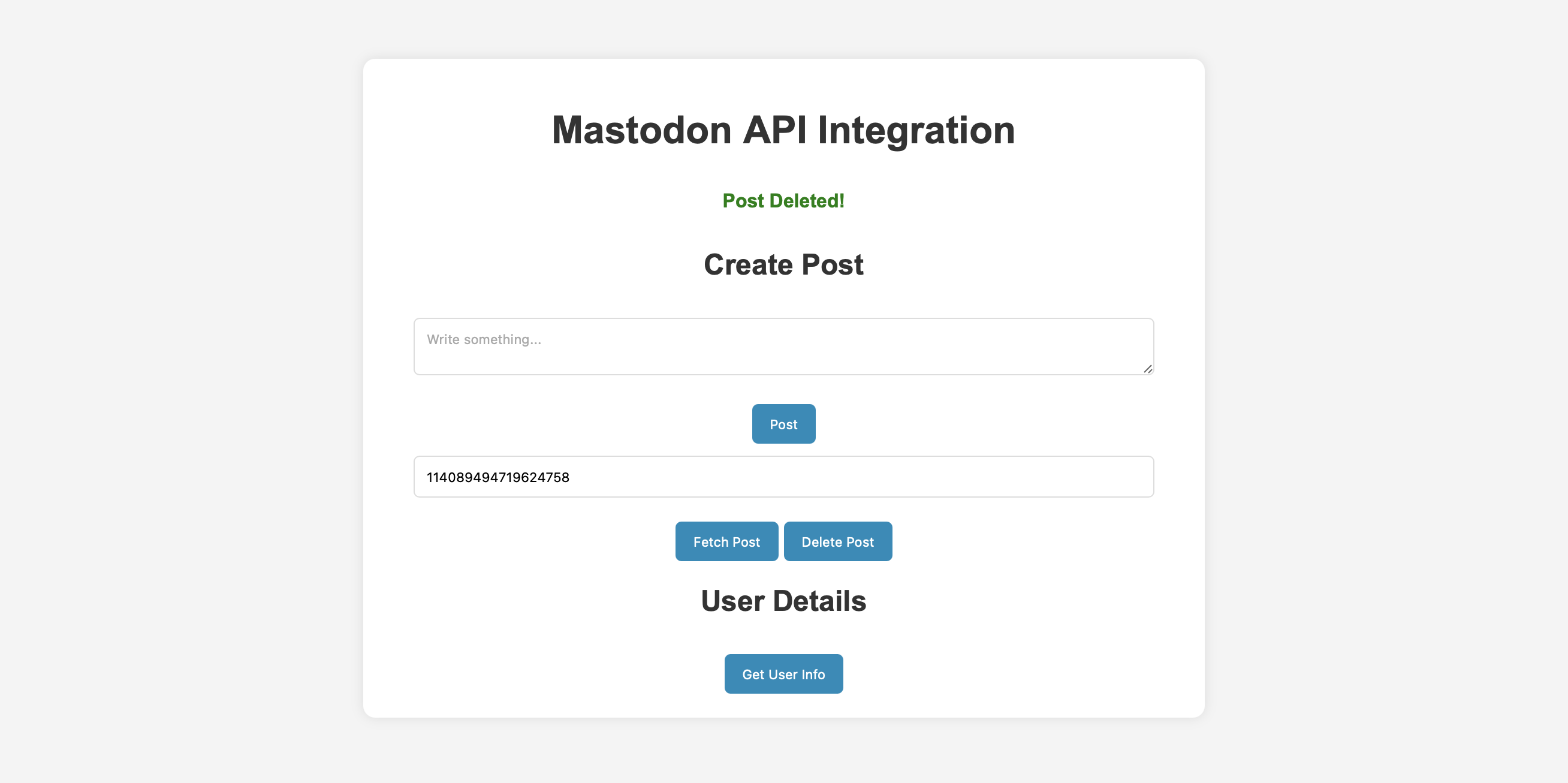
2. To post a text on mastodon social write something in the textbox and click on post.  


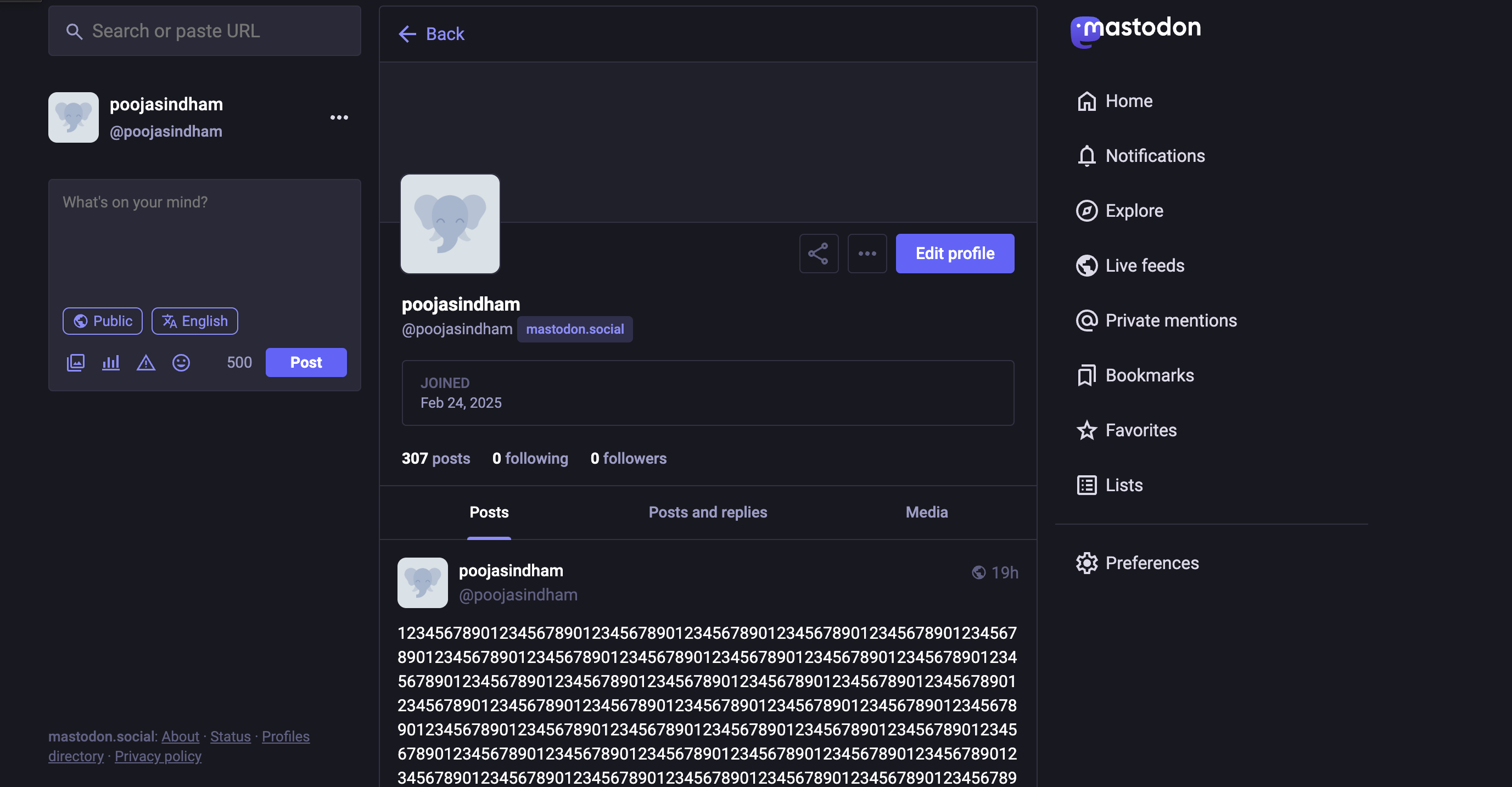
3. Once you click on post. You will see the post ID that got posted.  


You can see it posted on mastodon social.  


4. Now you can use this post id to retrieve or delete the post.

5. Retrieve the post. Give the post id in the second text box and click on Fetch post.  


6. Delete the post. Give the post id in second text box and click delete post.  


It got reflected in mastodon API, where the post will be deleted.  


7. Get User Info will fetch the user details like followers, following and number of posts.  
