Message Board Web Service

You are tasked with developing a web service for a message board. The service should allow users to register, post messages, vote up or down for any message on the board, remove messages, and view all their messages and their corresponding votes.

General guidelines:

- You can use the internet (for references)
- You can download any package/framework you want.
- Read all the questions carefully, if you have any question – ASK, this is not a test in reading comprehension.
- Try to write your code well designed (but not over-design), testable and simple.

Requirements:

1. Implement a RESTful API using any web framework of your choice (e.g., Flask, Django, Express.js, etc.) to handle the following endpoints:

   ✓ - **/register** (POST): This endpoint should allow users to register by providing a username and password. Store the user information securely.

   ✓ - **/login** (POST): This endpoint should allow registered users to log in using their credentials.

   ✓ - **/logout** (POST): This endpoint should log out the currently logged-in user.

   format txt ✓ - **/messages** (GET): This endpoint should return all messages on the board, including the author and vote count for each message.

   ✓ - **/messages** (POST): This endpoint should allow logged-in users to post a message to the board.

   ✓ - **/messages/{message_id}/vote** (POST): This endpoint should allow logged-in users to vote up or down for a specific message. The **message_id** parameter should indicate the message to vote for.

   ✓ - **/messages/{message_id}** (DELETE): This endpoint should allow logged-in users to remove their own message. Only the author of the message should be able to delete it.

   ✓ - **/user/messages** (GET): This endpoint should return all messages posted by the currently logged-in user, including their vote count.

2. Bonus points:

   ✓ - Error handling
   ✓ - Unit tests
   ✓ - docerfile
   - Using OpenAPI

Notes:

- You are free to use any programming language and frameworks to implement this assignment.

- Document any assumptions you make during the implementation process.

- You are encouraged to write clean, modular, and maintainable code.

- Avoid storing sensitive information (such as passwords) in plain text.

- Ensure your code is well-documented and include comments where necessary.

Submission:

Please provide your solution as a well-organized codebase along with clear instructions on how to set up and run the application locally. Additionally, provide any necessary database schema or migration files.

Good luck!

S