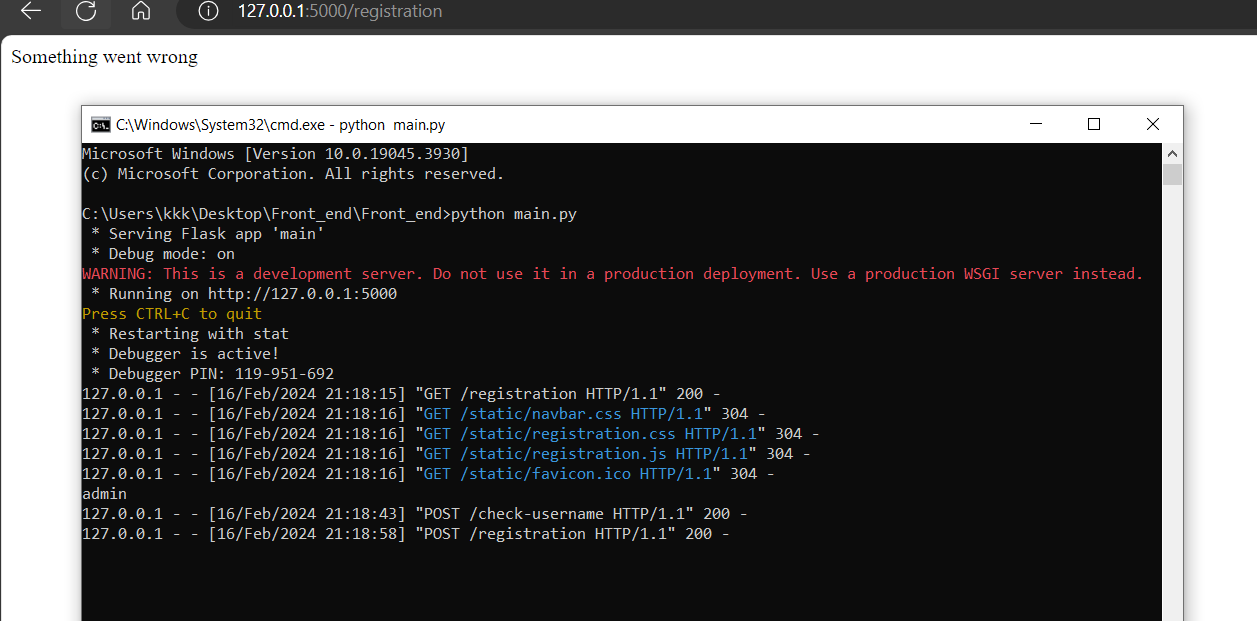
**HTTP Handlers for Practical Assignment No. 3**

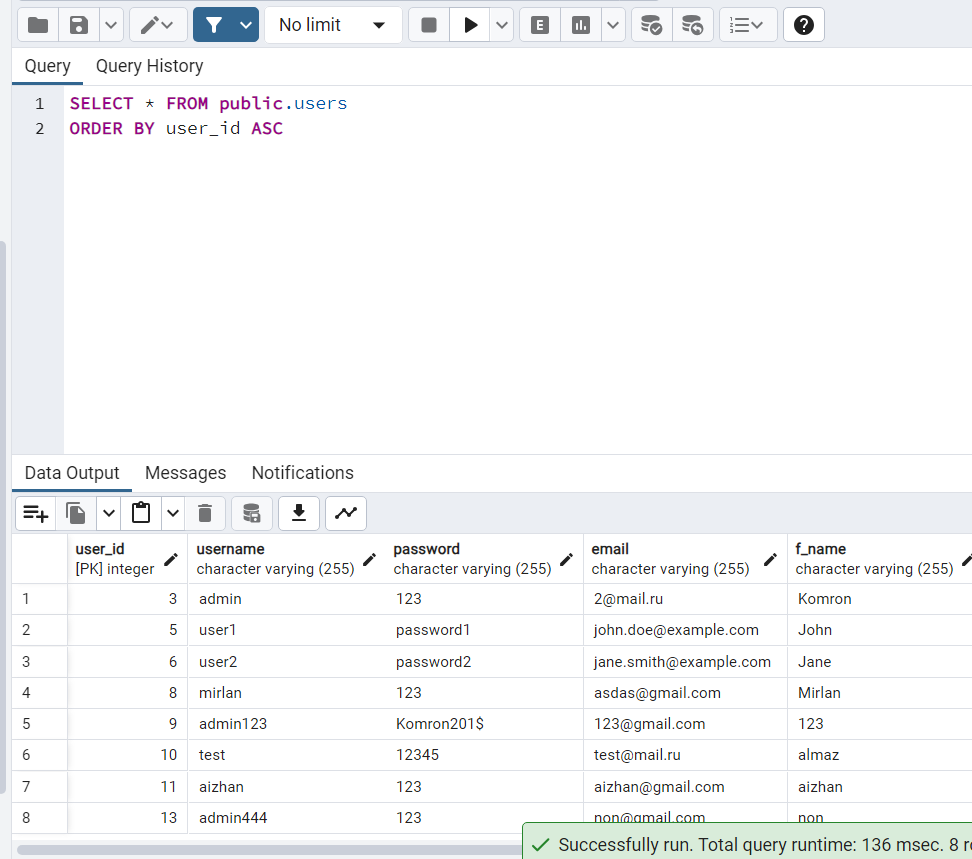
Task 1: Extract Data from Query String and Write to Response ("/registration")

* **Objective:** Extract data from the query string of the "/registration" endpoint and write the data attributes to the HTTP response.
* **Implementation:** Utilized Flask to create the "/registration" endpoint that retrieves data from the registration form and responds with a confirmation message or an error.

Queries are sent once the registration form is filled, if the email already exists in database it will return this: 

Otherwise, if it successfully in database it will return something like this:   
A screenshot of a computer program

Description automatically generated

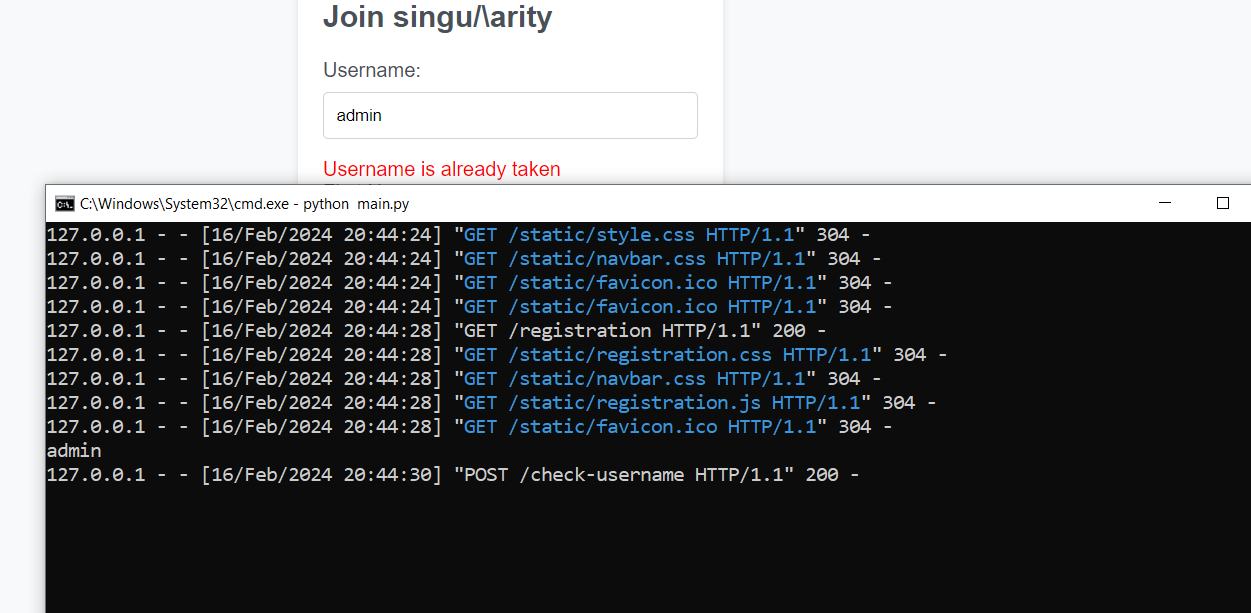


Here we have new users.

Task 2: Check Username Availability ("/check-username")

* **Objective:** Implement a check for username availability based on the query string parameter.
* **Implementation:** Developed the "/check-username" endpoint that receives a username in the query string, checks its availability, and responds with JSON indicating whether the username is available.

While we are creating profile, we filling “username” section and once the username is filled we send the request to the app to check whether the username is in database or not.



If the are not such username, when we get another message like this:   
A screenshot of a computer

Description automatically generated

**Conclusion**

This report outlines the two implemented HTTP handlers for the practical assignment. The "/registration" endpoint handles user registration data, while "/check-username" checks the availability of a username.

**Event Handling, Bindings and DOM Manipulation using JavaScript**

A screenshot of a computer program

Description automatically generated

Explanation:

* **$(document).ready(function () { ... });**: This ensures that the JavaScript code inside the function is executed only after the HTML document has been fully loaded.
* **$('#username').on('blur', function () { ... });**: This attaches an event listener to the 'blur' event of the 'username' input field. The event is triggered when the user moves the focus away from the input field.
* **var username = $(this).val();**: Retrieves the value entered in the 'username' input field.
* **$.ajax({ ... });**: Initiates an AJAX request to the server.
  + **url: '/check-username',**: Specifies the URL endpoint to which the AJAX request is sent.
  + **method: 'POST',**: Specifies that the request is a POST request.
  + **data: { 'username': username },**: Sends the username as data with the request.
  + **success: function (response) { ... }**: Defines a callback function to handle the response from the server.
    - **var availabilityDiv = $('#usernameAvailability');**: Retrieves the div element with the id 'usernameAvailability' where the availability message will be displayed.
    - **if (response.available) { ... } else { ... }**: Checks the response from the server to determine whether the username is available.
      * If available, it updates the content of **availabilityDiv** to indicate that the username is available and sets the text color to green.
      * If not available, it updates the content of **availabilityDiv** to indicate that the username is already taken and sets the text color to red.