

# CS4530 Final Project: "NULL POINTER"

Group 111: Sutton Spindler, Danna Amitai, Wes Rising, and Miguel Hernandez

## Feature Overview

NULL POINTER was based upon the existing "Fake Stack Overflow." While this app had a lot of great functionalities, there were some features missing. During the development of NULL POINTER, we focused on three main features:

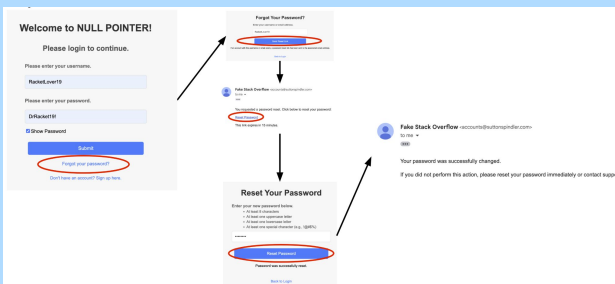
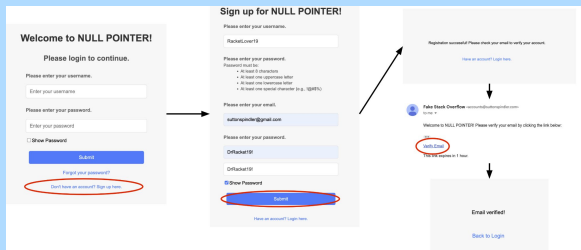
- Improved security: hashed passwords, email verifications, allowed for password reset, and implemented friends and message requests for non-friend conversations.
- Improved chat features: Implemented DMs to friends and approved users, community chat-rooms, allowed for emoji and image sending across chats and typing indicators.
- Leaderboard: Implemented a ranking according to users with most answer upvotes. Allowed for answer voting and hiding the total number of votes to increase privacy. Implemented a leaderboard page to display this ranking as well as in the user's profile.

## Demo and Source

Our demo site is available at

<https://fall25-project-group-111-client.onrender.com>, and our code

at <https://github.com/neu-cs4530/fall25-project-group-111>



Email verification upon signup and ability to reset password through an email link.

## Tech Stack:

We built our application with a React/TypeScript frontend and an Express/TypeScript backend. The client leverages React contexts and custom hooks to manage user authentication and dynamic UI updates, while the backend exposes REST routes for questions, messages, and user management, storing data in mongoose objects. The backend also sets up websocket event listeners for chats so that the frontend hooks use can socket event handlers for receiving live chat updates. Testing is automated using Cypress (for end-to-end) and Jest (unit and integration).

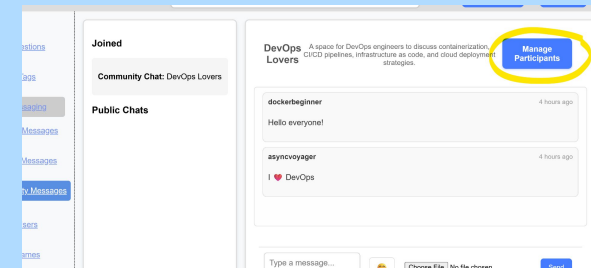
## Future Considerations:

Building upon the features we built for our three user stories, there are some improvements we could add. Firstly, for security purposes, we could allow users to use identification to get verification badges on their profile. We could also adjust message requests to allow users to send a message request that already has the message, so the other user could view the message before deciding to accept or reject. We could also include a page where users can view all pending requests within DMs. For chats, we could add more message features such as the ability to send polls or gifs. Additionally, we could improve the user interface such that users can customize their background or pick between themes. Finally, we could consider having the leaderboard update daily rather than on refresh to make change in rankings more meaningful, as users would be aware of the temporal aspect of the change.

The screenshot shows the 'Leaderboard' page with a search bar and a 'Refresh Ranking' button. Below is a table of the top 10 users ranked by total net upvote score.

User	Net Votes	Rank	Medal
frontendev	3	#1	🏆
devopsengineer	3	#2	🥈
reactexpert	2	#3	🥉
dbmaster	2	#4	🥈
datacenterist	2	#5	🥈
es6guy	2	#6	🥈
webpackward	1	#7	🥈
nodejsdev	0	#8	🥈
edgescavenger	0	#9	🥈
reactjsja	0	#10	🥈

Leaderboard view, where a user can refresh the rankings, observe changes, and see their own highlighted place if they have posted a solution.



Community chat page from an admin's view. Includes joined and public chats, and ability to manage participants.