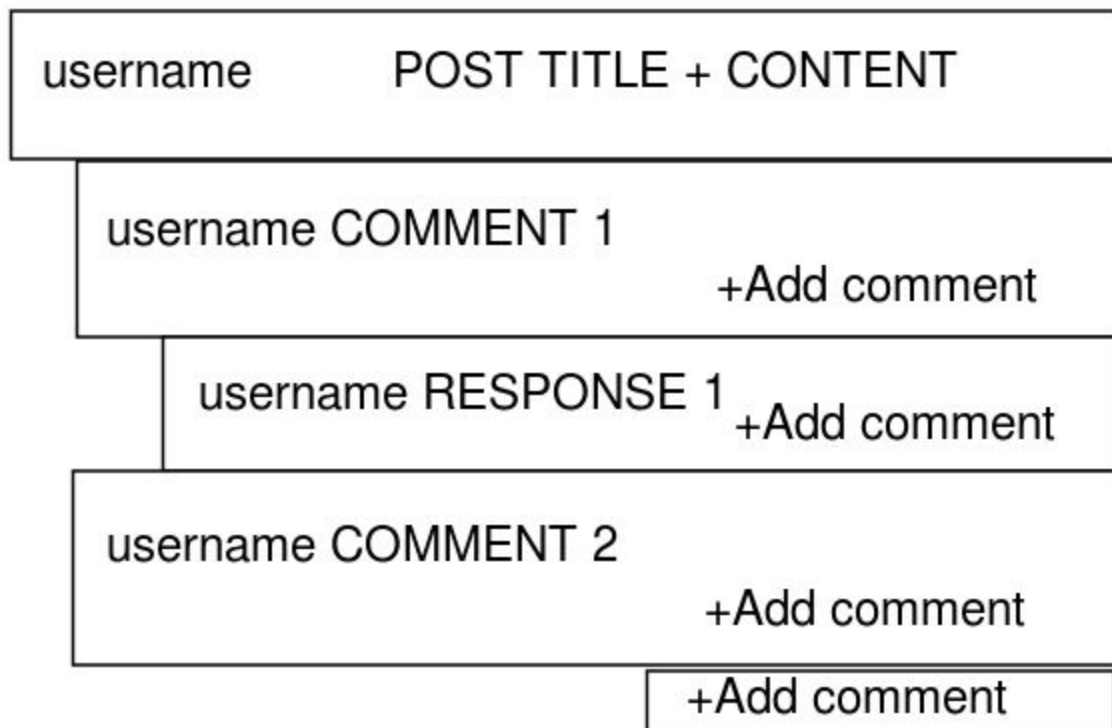


We'd like you to write a single-page React app: a comment-thread for a single post in a social news site. The page should look something like this sketch (in essence, but ideally prettier):



The first thing the user should see when they load the page is a default, hardcoded post at the top of the page, right above an “Add comment” button. The post should have

- (a) Username of the poster
- (b) Title of the post
- (c) Content of the post

Note that we do NOT need you to implement a database/storage/hosting layer for this problem, NOR do we need you to have a login system. In particular, this means that every time the page is loaded, it will only show (a) the initial post and (b) the Add comment button. You do not need to preserve data about comments if the page is refreshed.

When the user clicks the “Add comment” button, there should be a way for them to create a comment about the post. Every comment has

- (a) Username of the commenter
- (b) Text of the comment

After a user submits their first comment, a comment item should show up below the initial post. It should contain the username of the commenter, the content of the comment, and an “Add comment” button of its own.

In this manner, each comment has an “Add comment” button, and the overall page should have an “Add comment” button at the bottom. This way, the user can either add a comment about a comment, or add a comment about the original post. You wind up with a tree-like comment thread.

We would like some sort of indentation the deeper the thread goes, as in the initial sketch. The exact details of how to do this are up to you.

We do NOT need you to support editing of comments/posts, or deletion of comments/posts.

We’re looking for:

- 1) Correctness. The app should function as described above. Also, there are some details we didn’t specify, such as what to do if the comment thread gets very nested/deep. We’d like you to deal with all of these cases nicely.
- 2) Speed of implementation - we’d like you to
 - a) Email the source code you have after three hours of working on the task
 - b) If the task takes longer than three hours, email us the final source code in addition to the source code you have after three hours.
- 3) Code style - proper use of React, good documentation/naming, etc.
- 4) UI/UX - you can use any CSS framework you’d like to make the app look good. It should feel clean and easy to use.