**Description**

In this post, we see how to build a module using HTML, CSS and JS in React/Redux environment that allows user to post topics and upvote/downvote them.

**Overview of concepts:**

* **create-react-app**: Create React App is a popular toolchain for building simple apps. The goal of this post is to explain the demand of Redux and React at a deeper level thus providing me with better control and understanding of the skill further filling me with confidence in the development environment.

Create React App is also a toolchain. It is specifically recommended by the React community for building single-page applications (SPAs) and for learning React (for building “hello, world” applications). It sets up your development environment so that you can use the latest JavaScript features, provides a nice developer experience, and optimizes your app for production.

* **Redux**: Redux is a predictable state container for JavaScript applications. It is a state management tool. With Redux, the state of your application is kept in a store, and each component can access any state that it needs from this store.

**React-boilerplate**: Boilerplate code refers to blocks of code used over and over again.

**Building the feature:**

We see this module as the submit button along with a text area that takes an input from the user and then allows them to upvote and downvote them accordingly.

**Step 1: Breaking up the feature into components**

We broke it into several sections:

* A section that contains an input field
* A section that contains a submit button
* A component that contains a list of posts
* A subcomponent that represents a post
* A subcomponent that represents a square
* A subcomponent that represents text

**Step 2: Creating the PostList Component**

To start off we created the component that represents the square buttons that are a part of each of the posts. Then we add some style to it.

We test this by rendering it to the page.

**Creating the PostText Component**

We created the component that represents the text areas that are a part of each of the posts.

**Creating the Post Component**

We created the components that represent the list of posts. We accomplished this by using the map() method to generate a Post component for each item in the component postList property array.

**Step 3: Creating the Controlled Component from the input field**

To start off we create an App component that will hold all of the other components.

Then we initialize the App component’s state. The App component has two state attributes: one to hold the value of input element and one to hold the array of post data.

**Step 4: Adding values to the Post List**

We add items to the state array through this function.

**Step 5: Updating the Post score**

This allows us to add the functionality to update the post score when the + or – buttons are pressed.

**Step 6: Removing posts**

This functionality allows the post to be removed when the x button is clicked.

**Commands to run:**

$npx create-react-app demo

<!-- This command allows the user to create a folder or project -->

$npm start

<!-- This command makes the application to start and run on the local server -->