The Header Component

In this lesson, we'll access the updated state from the store (the update was made after we dispatched the setUserID action), and use it to display the selected contact's information in the header.

Let’s begin with the Header component.

The current content of the chatWindow component is this:

```
import React from "react";
const ChatWindow = ({ activeUserId }) => {
  return (
    <div className="ChatWindow">Conversation for user id:
      {activeUserId}</div>
  );
};
export default ChatWindow;
```

Not very helpful. Update the code to this:

```
import React from "react";
import store from "../store";
import Header from "../components/Header";
const ChatWindow = ({ activeUserId }) => {
  const state = store.getState();
  const activeUser = state.contacts[activeUserId];
  return (
    <div className="ChatWindow">
      <Header user={activeUser} />
    </div>
  );
};

export default ChatWindow;
```

ChatWindow.js

What’s changed?

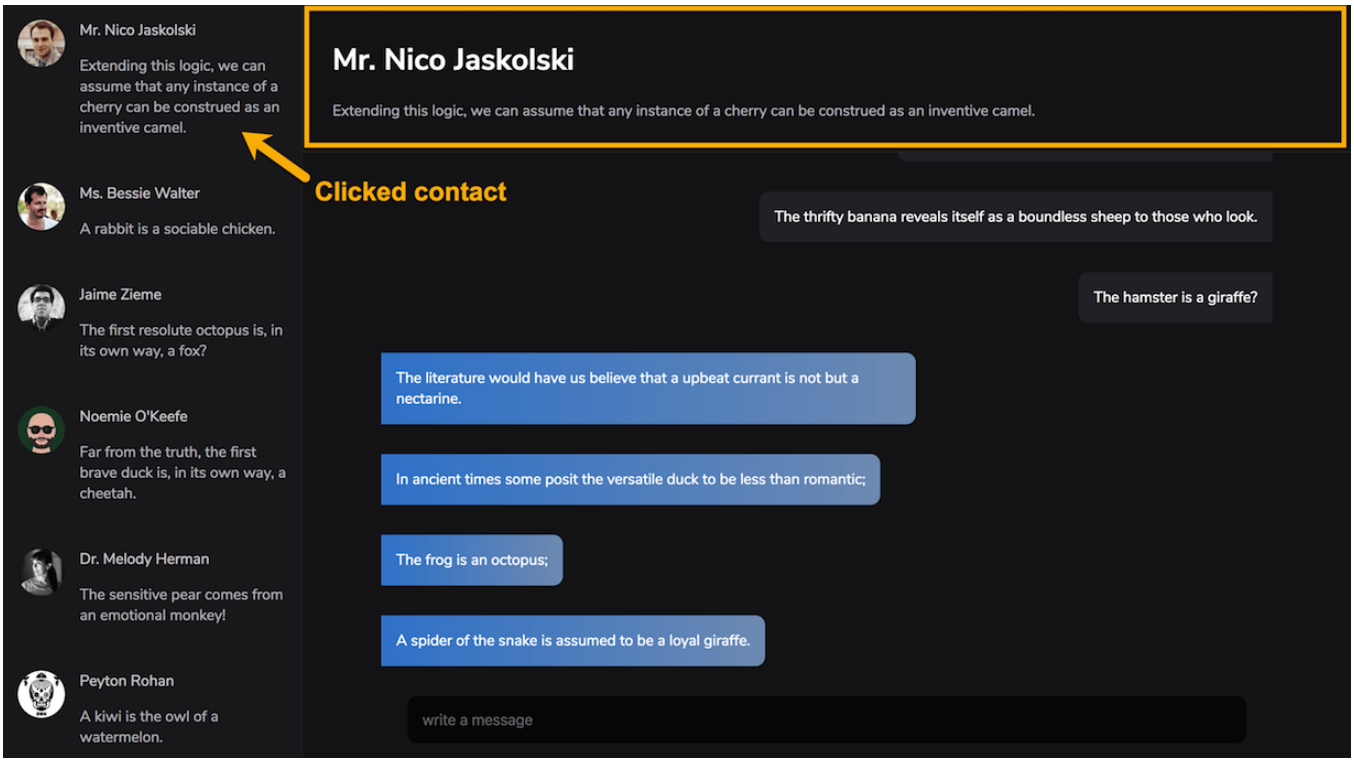
Remember that the activeUserId is passed as props into the ChatWindow component.

Now, instead of rendering the text, “Conversation for user id: ...”, render the

Header component.

The Header component cannot be rendered properly without having a knowledge of the clicked user. Why?

The name and status rendered in the Header are those of the clicked user.



To keep track of the active user, a new variable, activeUser is created, and the value is retrieved from the state object like this:

```
const activeUser = state.contacts[activeUserId];
```

How does this work?

First, we grab the state from the Redux store:

```
const state = store.getState();
```

Now, remember that every contact of the application user is stored in the contacts field. Also, every user is mapped by their user_id.

Thus, the active user can be retrieved by fetching the user with the corresponding id field from the contacts object:

```
state.contacts[activeUserId].
```

All good?

At this point we need to build out the rendered Header component.