To interact with an API in Redux, we use a state which contains an isLoading property to follow the state of the request.

We use actions to follow the evolution of the request and update the state.

When the request is sent, we store.dispatch the fetching action.

If the request succeeds, we store.dispatch the resolved action with the data in payload.

If the request fails, we store.dispatch the rejected action with the error in payload.

To access the Redux store from a component, we use the useStore hook.

I stored the username & password in local state then on handlesubmit the request is sent, we store.dispatch the fetching action. which sets isLoading to true so a loading icon is displayed whilst we wait for the token to be collected & then waits for a response.

I used the post method with axios to the end point http://localhost:3001/api/v1/user/login with email, password as the payload in the body to get the token for authentication.

If the request succeeds, we store.dispatch the resolved action with the token in payload. Which stores the token in the store & sets isLoggedIn to true/ isLoading to false. This causes the signin page to redirect to the user profile page.

Otherwise, If the request fails, we store.dispatch the rejected action with the error in payload which then displays an error message

When the user profile page renders, the useEffect hook sends a request to get the user data with the JWT token from the store for authentication. Again,we store.dispatch the fetching action, which sets isLoading to true & waits for a response….

If the request succeeds, we store.dispatch the resolved action with the user data in payload. Which stores the user in the store & sets isLoading to false & and the page is rendered with the user profile & dummy transactions.

Otherwise, If the request fails, we store.dispatch the rejected action with the error in payload which then displays an error message

Used post method with axios to the end point http://localhost:3001/api/v1/user/profile & Bearer JWT token for authentication to get the user profile