Noting the Expected Behavior

In this lesson, we'll discuss the working of the app along with its expected behavior.

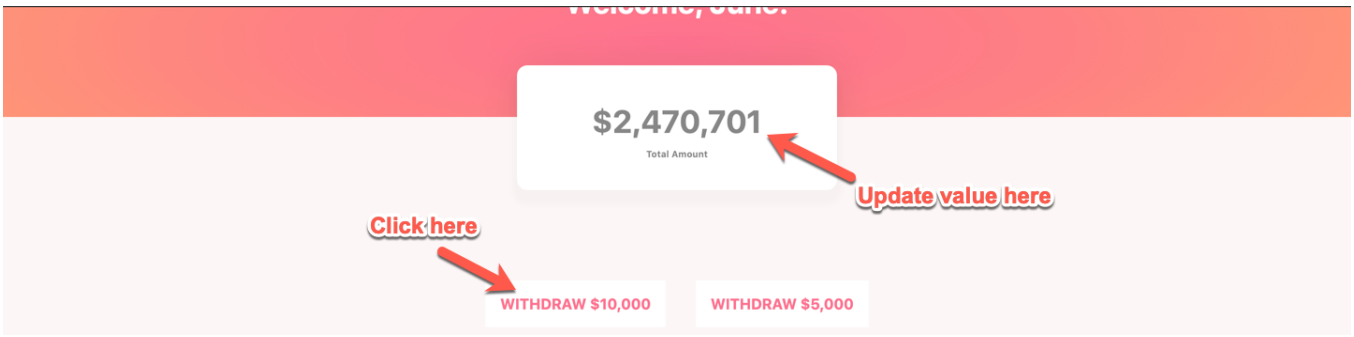
WE'LL COVER THE FOLLOWING

Expected Results

When I need to profile an application, specifically during a certain interaction within the app, I like to set the baseline for what I expect in terms of performance. This sort of expectation helps you retain your focus as you delve into interpreting the results from the Profiler.

Withdraw Functionality

Let's consider the bank application we want to profile. The interaction in the bank app is simple. You click a pair of buttons, and the withdrawal amount is updated.



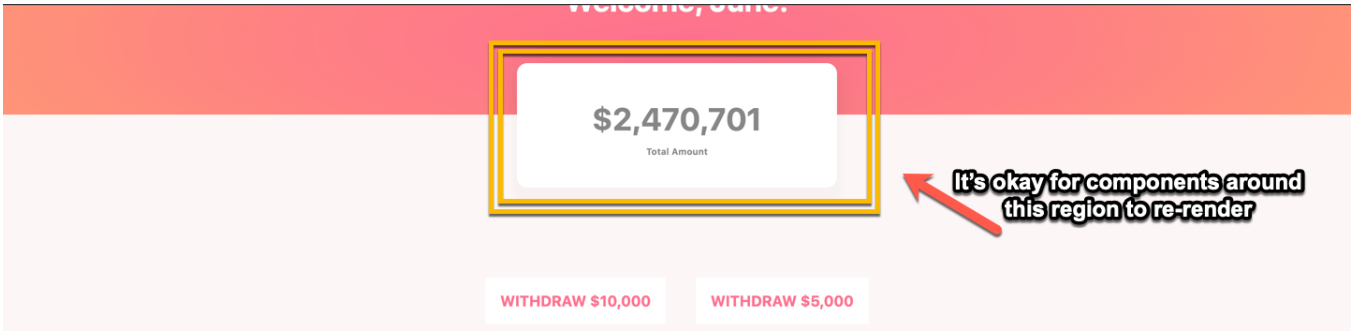
The basic interaction of the app

Now, what would you consider to be the expected behavior of this app with respect to re-renders and updates?

For me, it's quite simple.

Profiling Session

The only part of the app visually changing is the withdrawal amount. Before going into the profiling session, my expectation for an optimized application will be that no unnecessary components are re-rendered, just the component responsible for updating the total amount.



Where I expect re-renders to happen. Nowhere else.

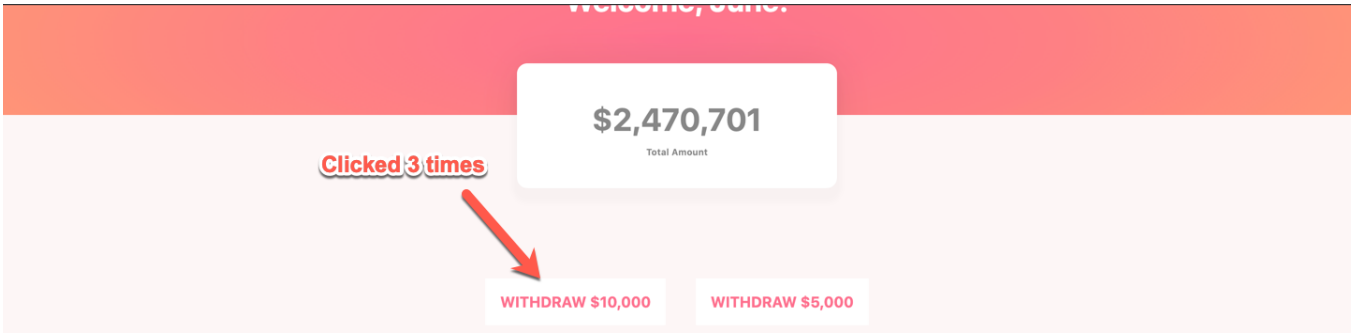
Recording Session

I'll expect just the `TotalAmount` component to be re-rendered and any other component directly connected with that update.

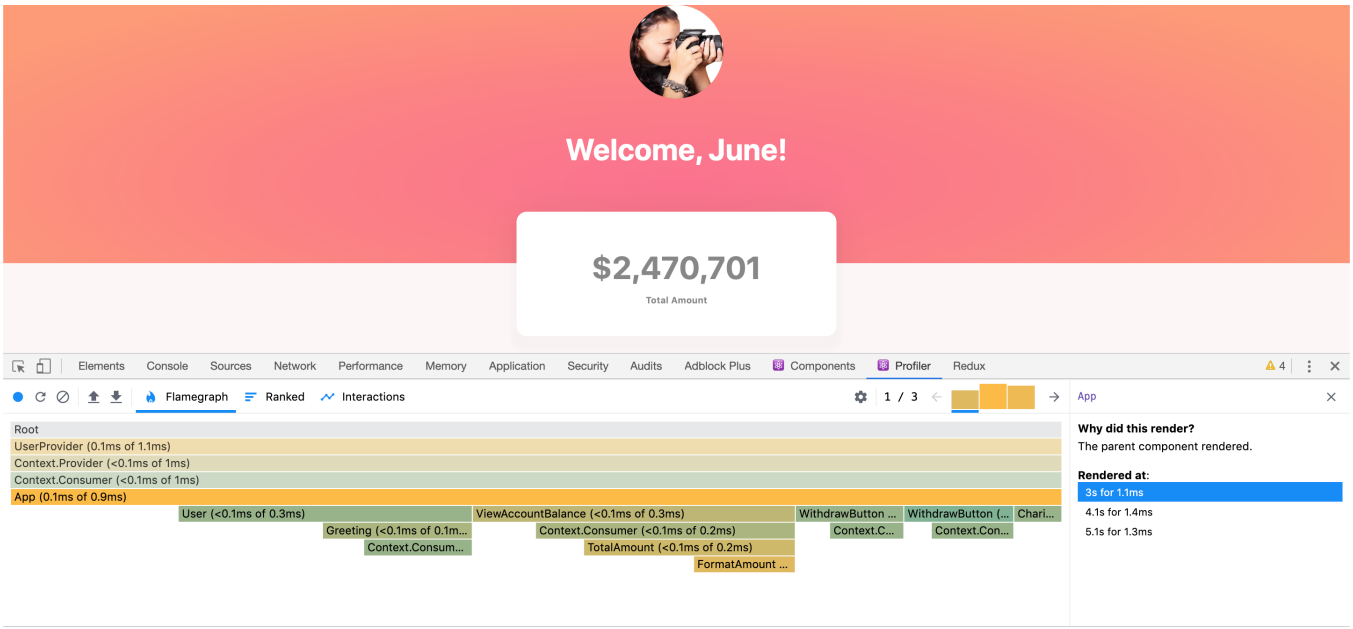
With this expectation set, let's go on and profile the application.

The steps are the same as we discussed earlier. You open your `devtools`, record an interaction session, and begin to interpret the results.

Now, I have gone ahead to record a session. In this session, all I did was click the "*withdraw $10,000*" button 3 times.



Here's the flame chart from the profiling session:



Flamegraph Results

From the chart above, so many components were re-rendered. Do you see the multiple bar colors represented in the flame chart?

This is far from ideal, so let’s begin fixing the problem.