

3.1 Introduction

This is the third part of Building Tesla's Battery Range Calculator with React series.

In part 1, we've created the application with only the React core, and in part 2 we've transformed the application by introducing Redux, a state management solution.

Part 1, which focuses on React, [is here](#):

[Building Tesla's Battery Range Calculator with React \(Part 1\)](#)

[In this series of articles, I will walk you through the process of building Tesla's battery range calculator with React...](#)

Part 2, which focuses on Redux, [is here](#):

[Building Tesla's Battery Range Calculator with React \(Part 2: Redux version\)](#)

[This tutorial is the second part of building Tesla's battery range calculator with React.](#)

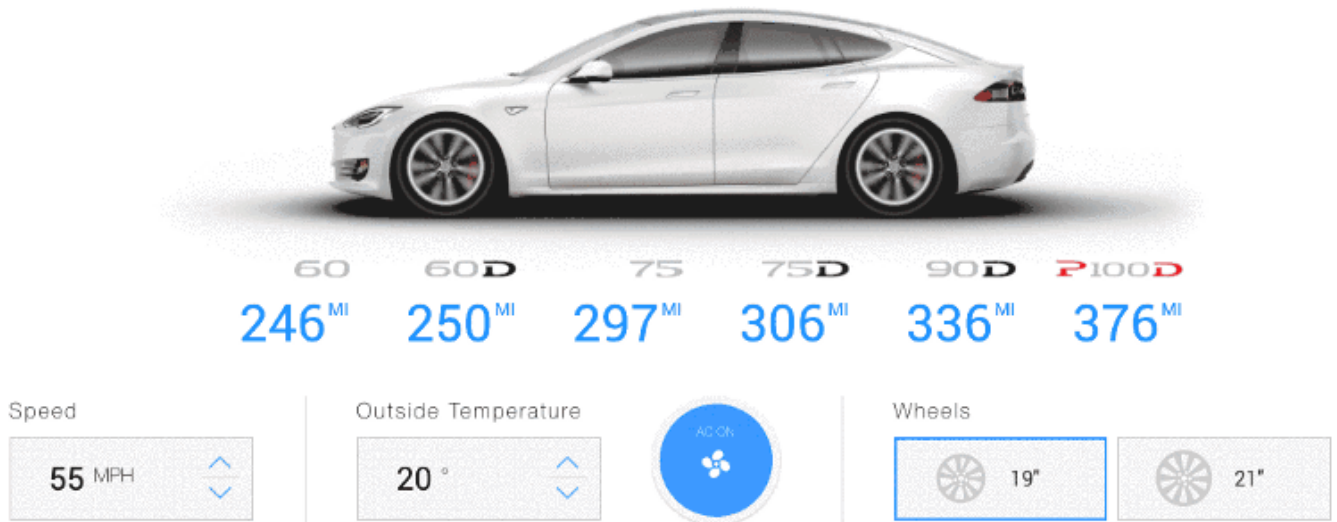
If you look at the application we've built so far, it's a bit plain and boring. It would look cooler if we could give it dynamic car showroom feel.

When the app is loaded, it would be nice if our stunning Tesla car images were in motion. With a little Transformer feel. The Tesla header is also shown as a fade-in effect, and the rest of the components should look a little nicer.

Even when the user event occurs after loading, it will be more impactful to show a noticeable visual effect instead of just changing the numerical value.

Just like this:

Range Per Charge



The actual amount of range that you experience will vary based on your particular use conditions. See how particular use conditions may affect your range in our simulation model. Vehicle range may vary depending on the vehicle configuration, battery age and condition, driving style and operating, environmental and climate conditions.

You can check out [live demo](#).

And here's the [full source code](#).

In **Part 3**, we'll take a step-by-step look at the process of making our finished apps more stylish using **CSS animation**.