

Layouts with Flexbox Lab

We have a few cool components. And these components may work okay but they can use some help in styling and layout. Why don't we work on the layout for a while?

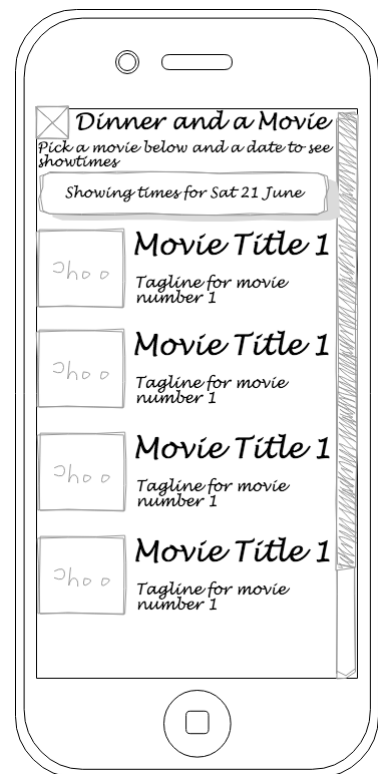
As you can see below, you've been given some simple wireframes of how our client's UX experts would like the views to lay out. Your job will be to use what you've learned about flexbox to make all of that happen. Once you've got them laid out in the code, try to look at them on a number of devices and don't forget to rotate your device to make sure they look good portrait and landscape.

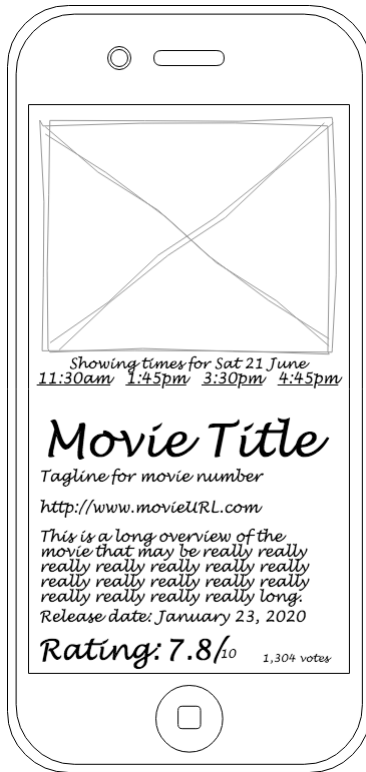
Note: You're not responsible to make the font sizes or centering work just yet. We'll have a chance to do that later. Just concern yourself with layout at this point.

Ready? Let's go!

The Landing Component

1. Add a company logo to the upper-left. Use one of your own or use daam.png in the starters folder. If you need to size it, set the style's height and/or width property. But the size of images should be the only styling you'll do in this lab.
2. Place the logo and our company name side-by-side
3. Make the movie poster lay out to the left of the movie title and the tagline.





The Movie Details Component

4. Make the movie poster centered and bigger
5. List your showing times side-by-side
6. Make a few additions:
 - Add a hardcoded "Rating" <Text> just before your vote_average
 - Add a hardcoded "/" after vote_average
 - Put a hardcoded "10" immediately after that
 - Then put the vote_count
 - Add a hardcoded "votes" after that
7. Then put all of those things on the same line

The Pick Seats Component

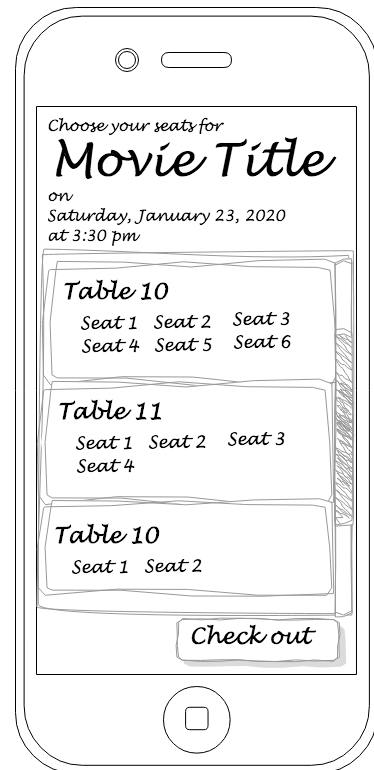
8. This is a new component. Go ahead and create it in PickSeats.js.
9. Place it in the JSX of App.js instead of <Landing> for testing the layout here.

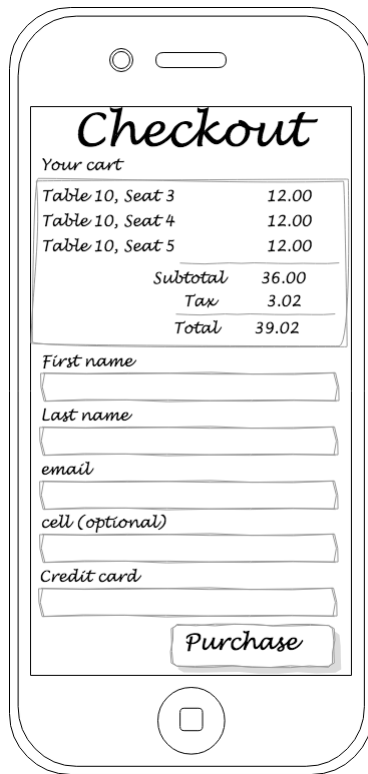
Pick Seats will eventually be shown when a user picks a movie, a date, and a showing. Then they will be able to reserve seats at a table. This scene lets them choose where they want to sit.

10. Make it look like the sketch to the right. ---->
11. Some notes:
 - Don't worry about making Movie Title big.
 - Each of the Tables will be in a View.
 - Each table has a table_number.
 - Each table has an array of seats. The tables will have a varying number of seats.
 - The table list will be in a <ScrollView>
 - If you want some testing data, copy starters/tables.json to the src folder and go:

```
import tables from './tables.json';
```

 at the top of PickSeats.js. That JSON file has a list of tables and seats that you can .map() through.





The Checkout Component

12. For testing your layout, go ahead and make `<Checkout>` the startup component in `App.js`. Don't forget to make `<Landing>` the startup component when this lab is finished.
13. Open `Checkout.js` in your IDE.
14. First, add a `<View>` that will serve as your cart.
15. If you want some data that you can `.map()` through, look in the `starters/json` folder for a file called `cart.json`. This is some hardcoded data that will work for now.
16. Note that you are going to want to calculate a subtotal, tax, and a grand total.
17. Also note the layout. You're going to want to replicate that as best as you can.

Got four laid-out components? Cool! You're done!