

# Day 0

Make a **single** web page with a read-only UI form for a booking appointment for a **predetermined** service provider (i.e. hairdresser). Data is retrieved from a fake database.

## GOALS:

- code is easy to read (two files for frontend, one file for the database)

## NON-GOALS:

- booking process
- multiple users
- multiple service providers
- login/registration
- server (backend application) or real database

## TASK:

1. Booking appointment UI form with React Spectrum
2. Fake database with Typicode server

## React Spectrum

React Spectrum is library with premade UI components and removes the need for styling the webpage (i.e. CSS).

Pre-Requisites:

- node
- yarn (the same tool as npm, use yarn everywhere)
- parcel (used to run the app)

See Getting Started for setup.

The “Build tooling” section is written with the expectation that you have a working environment and doesn’t tell you how to set it up. To setup Parcel you need to follow Build a web app with Parcel. Inside the ‘index.html’ file, you need to connect to React Spectrum by linking to the file containing it’s code (make sure to use the correct file name):

```
<script type="module" src="app.js"></script>
```

You don’t need to do anything else with Parcel except what’s on this page.

## Typicode Server

Typicode is our fake database. It reads a file and serves via a web app, mimicking a real app.

Install with ‘yarn add json-server’. Then follow the Usage section on the GitHub Readme. That’s all (except of course changing the data).

## Tips

1. Only the following components are enough to make the form (make it as simple as possible):
  - Form
  - TextField
  - Checkbox
  - Button
2. Read data from Typicode by using the fetch() function (inside the JS file)
3. For errors with JavaScript check node version (>18)
4. Don’t overthink, if in trouble read the official documentation and go step-by-step