## **PROJECT: TODO LIST**

### **Introduction**

At this point you’ve already had a fair amount of practice using the various techniques we’ve shown you. But we’ve been throwing a *lot* of information your way, so before we move on we’re going to take a minute to slow down and work on one more great portfolio project.

### **The Todo List**

Todo lists are a staple in beginning webdev tutorials because they can be very simple. There is, however, a lot of room for improvement and many features that can be added.

Before diving into the code, take a minute to think about how you are going to want to organize your project

### **Assignment**

1. Your ‘todos’ are going to be objects that you’ll want to dynamically create, which means either using factories or constructors/classes to generate them.
2. Brainstorm what kind of properties your todo-items are going to have. At a minimum they should have a title, description, dueDate and priority. You might also want to include notes or even a checklist.
3. Your todo list should have projects or separate lists of todos. When a user first opens the app, there should be some sort of ‘default’ project to which all of their todos are put. Users should be able to create new projects and choose which project their todos go into.
4. You should separate your application logic (i.e. creating new todos, setting todos as complete, changing todo priority etc.) from the DOM-related stuff, so keep all of those things in separate modules.
5. The look of the User Interface is up to you, but it should be able to do the following:
   1. view all projects
   2. view all todos in each project (probably just the title and duedate.. perhaps changing color for different priorities)
   3. expand a single todo to see/edit its details
   4. delete a todo
6. For inspiration, check out the following great todo apps. (look at screenshots, watch their introduction videos etc.)
   1. [Todoist](https://en.todoist.com/)
   2. [Things](https://culturedcode.com/things/)
   3. [any.do](https://www.any.do/)
7. Use localStorage to save user’s projects and todos between sessions.
8. Since you are probably already using webpack, adding external libraries from npm is a cinch! You might want to consider using the following useful library in your code:
   1. [date-fns](https://github.com/date-fns/date-fns) gives you a bunch of handy functions for formatting and manipulating dates and times.

### **Want MORE practice?**

There is plenty of content after this project but at this point you’ve been exposed to many foundational concepts. You really can’t spend too much time reviewing and practicing the fundamentals that we’ve covered and luckily for you there are several websites that offer just this type of thing. Seriously, the better you grasp and the more fluently you can use the fundamental building blocks, the easier the hard stuff is going to come to you when we get to it. Take a minute to check out the following links.

* [Exercism](http://exercism.io/) - an Odin favorite. Uses a neat command-line interface to download and submit exercises and allows you to give and get feedback on your solutions.
* [CodeWars](https://www.codewars.com/) - A neat gamified system with leveled exercises. As you complete more exercises you level up and access more difficult ones. To join the TOP clan: go to your account settings on the codewars website and add “The Odin Project” in the Clan field.
* [HackerRank](https://www.hackerrank.com/) - Similar in setup to CodeWars, but this site is more focused on interview question type tasks. Some companies even use this site to interview candidates.

### **Examples**

* <https://michalosman.github.io/todo-list/>
* <https://pret3nti0u5.github.io/To-Do-List/>
* <https://zakarya-mks.github.io/ToDoers/>
* <https://bscottnz.github.io/todo/>