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 another great project that ties these techniques together. You should have the chance to show off most (if not all) of your newfound skills!

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 due date… 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
    2. Things
    3. any.do
7. We haven’t learned any techniques for actually storing our data anywhere, so when the user refreshes the page, all of their todos will disappear! You should add some persistence to this todo app using the Web Storage API.

1. localStorage allows you to save data on the user's computer. The downside here is that the data is ONLY accessible on the computer that it was created on. Even so, it's pretty handy! Set up a function that saves the projects (and todos) to localStorage every time a new project (or todo) is created, and another function that looks for that data in localStorage when your app is first loaded. Additionally, here are a couple of quick tips to help you not get tripped up:
    i. Make sure your app doesn't crash if the data you may want to retrieve from localStorage isn't there!
    ii. You can inspect data you saved in localStorage using DevTools! To do this, open the `Application` tab in DevTools and click on the `Local Storage` tab under `Storage`. Every time you add, update and delete data from localStorage in your app, those changes will be reflected in DevTools.
    iii. localStorage uses JSON to send and store data, and when you retrieve the data, it will also be in JSON format. Keep in mind you *cannot store functions in JSON*, so you'll have to figure out how to add methods back to your object properties once you fetch them. Good luck!