

The Simplest Kanban Board

In project management, there is a concept of a [kanban board](#) in which tasks are placed into columns that comprise a workflow. A simple example would be three columns: to-do, in progress, and complete. When you create a task such as "walk the dog", it begins in the first column and will then progress to the next column until it reaches completion.

This homework assignment will be to create an HTML page that will offer basic kanban board functionality that meets the following criteria.

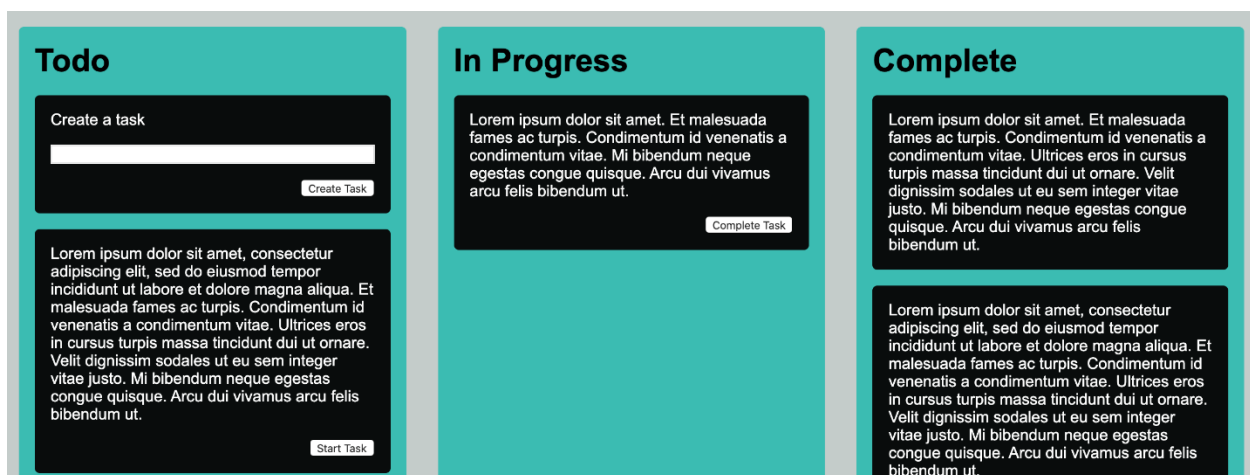
Technical Criteria

NO external libraries allowed. You can *not* use CSS framework such as Bootstrap. You can *not* use JavaScript libraries such as JQuery. The goal of the assignment is to get you comfortable with the core APIs and functionality of the browser before you leverage tools that abstract it away from you. You don't start learning how to drive in a Ferrari :)

Functional Criteria

Layout specifications:

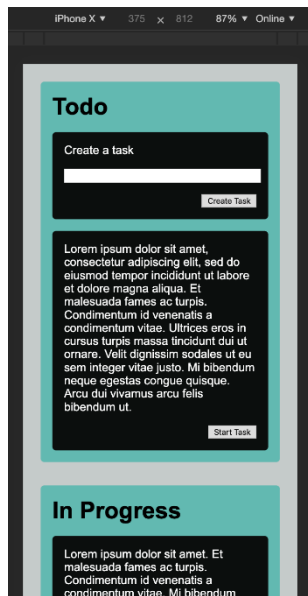
There will be 3 columns or "lanes." **Todo**, **In Progress** and **Completed**. On a screen with a lot of width (such as a computer), these 3 columns will expand to take up the entire width of the viewport. Each column should have a clear title at the top denoting which column it is. Use **flexbox** to achieve this layout.



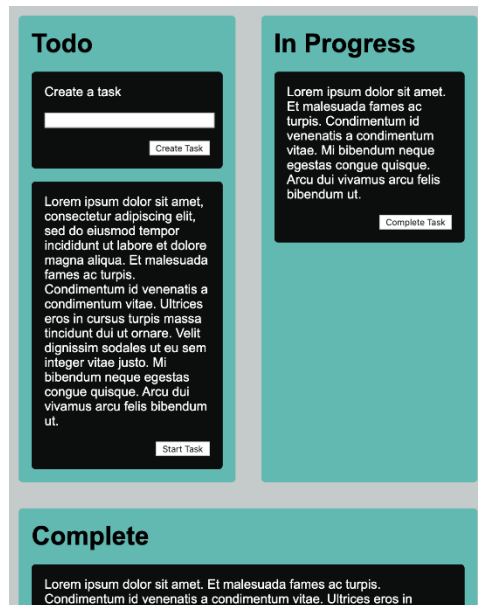
Responsive design:

This web page should be [responsive](#) and render appropriately based on different screen dimensions. For narrower screens (less pixel width) such as mobile phones, the 3 columns should re-flow to a single column where the "lanes" are stacked on top of each-other. Resizing your browser window and using [mobile device simulation in Chrome](#) will help you test. For grading purposes, your website will be tested on the iPhone X configuration.

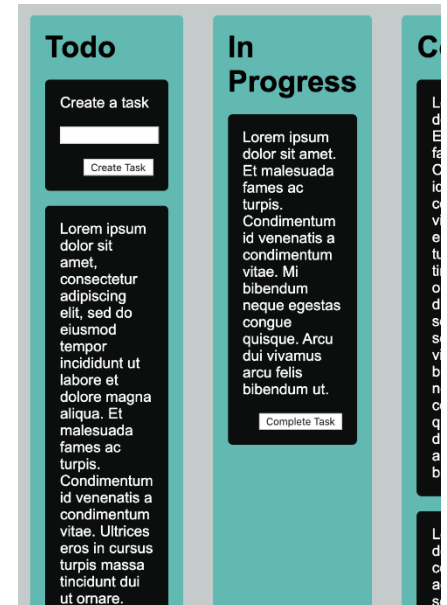
Good



Okay-ish (Confusing layout)



Bad (Too narrow)



Column specifications:

- **Todo:** The first item or "card" in this column will be a form where the user can enter text to represent the task. Below it will be a right-aligned button to append this task to the items in the "Todo" column. The "create task card" should always remain first in the list.
 - When a card is added to the "Todo" column, there should also be a right-aligned button below the task description that states "Start Task". When clicked, the corresponding card should be removed from the "Todo" column and placed within the "In Progress" column, appended to the bottom.
- **In Progress:** The cards will have only one functional difference from the ones in "Todo" - the button associated with each card will no longer show "Start Task", but "Compete Task" instead. When this button is clicked, it should be

removed from the "In Progress" column and appended to the cards present in the "Completed" column.

- **Completed:** The cards in the "Completed" column will no longer have a button, and will simply rest in their terminal state in this column.

Styling Criteria:

Make it look good! Effort really shows here. There are only a few strict guidelines:

1. There must be reasonable spacing between elements so they are not up against each-other. Keep your padding and margins consistent! It's an awkward and sloppy look if the text within a card has 1rem of space between the left edge of the card, and 1.5 from the right.
2. The buttons on the bottom of the cards (e.g. "Start Task", "Complete Task") should be right aligned.
3. Use some color. Try an online color palette generator and style your elements.
4. Use a custom font. Don't leave it at the default.

Suggestions & resources for completing the assignment

If you do not have a preferred editor/IDE, I recommend <https://code.visualstudio.com/>

[Familiarize yourself with Chrome's developer tools](#). This will be your *best* friend in this class. One of the killer features for this assignment would be to [see the padding/margin for elements, and see the applied styles](#).

Tackle the design & layout first. Start adding some dummy elements onto the page filled with random [lorem ipsum](#) text. Ensure that the structure of your HTML DOM and all of your various `<div>` and `<p>` tags work the way you want them too, before you start dynamically altering them with JavaScript.

When you're ready to take on the button behavior, experiment in the Developer tools Javascript *Console*. Write some code to grab a DOM element by its class or ID and assign it to a variable. See what API methods you have available to you. Working in this REPL will be much faster than trying to write your code in a separate javascript file and having to refresh the page every time you make a change!

Hints

[CSS lets you change a style based on parameters like screen width](#)

Documentation Resources (You need to skim these before you start)

Getting started with the web. See "HTML basics", "CSS basics", "JavaScript basics": [https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web)

Inline vs block behavior of HTML elements: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS Flow Layout/Block and Inline Layout in Normal Flow](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flow_Layout/Block_and_Inline_Layout_in_Normal_Flow)

Introduction to CSS layout: [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS layout/Introduction](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Introduction)

Flexbox: [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS layout/Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)

More Flexbox: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Manipulating the DOM with JavaScript: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side web APIs/Manipulating documents](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Manipulating_documents)

Submission Details

Send me a *zip* file with a *html* file that I can double click to open in my browser. It is highly recommended (but not required) to have your CSS and JavaScript in separate files.