# Task Board

In this project, you will create a board of to-do items organized into columns. All of the HTML and most of the CSS have been written for you, but the grid property declarations have yet to be added.

In order to complete this project, you must know how to lay out the structure of a grid’s rows and columns and place items within that grid using CSS.

We recommend that you review our CSS Grid Lesson 1 before beginning.

The premade **index.html** and **style.css** files are displayed for you. Click [this link](https://s3.amazonaws.com/codecademy-content/courses/learn-css-grid/project-ii/index.html) to see the completed project in another tab. Have fun!

If you get stuck during this project or would like to see an experienced developer work through it, click “**Get Help**“ to see a **project walkthrough video**.

**Tasks**

**9/9Complete**

Mark the tasks as complete by checking them off

**1.**

We will be focusing on laying out the container <div> and the two card-column <div>s within it. Here’s an abbreviated version of those <div>s in **index.html**:

<div class="container">

<div class="card-column future-projects">

...

</div>

<div class="card-column active-projects">

...

</div>

</div>

Right now, the task board looks pretty cluttered. In **style.css**, let’s make the <div> with class container a grid container so that we can start to arrange these tasks better.

Hint

You can set the display property to grid inside the CSS rule for .container.

**2.**

That didn’t have any immediate effect, but it allows us to apply grid properties to the container <div>. Let’s start by giving the container three equally sized columns.

Hint

You can set the grid-template-columns property to split the grid into 3 equal 1fr pieces inside the CSS rule for .container.

**3.**

Now, let’s add a gap of 20 pixels in between the columns.

Hint

You can set the grid-gap property to add space between grid elements.

**4.**

We want the task cards themselves to be arranged neatly within the columns as well, so let’s make each card column a grid container. You can do this by adding a property to the .card-column rule in **style.css**.

Hint

To make the elements of class card-column grid containers, you can set the display property to grid.

**5.**

Each card column has a heading and 5 card elements. We want the heading to take up 20px and each card to take up 100px. Set the grid-template-rows property to have 1 20px row and 5 100px rows. Use the repeat function to make your code clean.

Hint

Inside the .card-column rule, declare grid-template-rows with a value of:

20px repeat(5, 100px);

**6.**

In order to have the cards spaced evenly in each column, use an appropriate [box alignment](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Box_Alignment_in_CSS_Grid_Layout) property. Choose the property that will put an even amount of vertical space in between each card.

Hint

Use align-content with a value of space-between to do this.

**7.**

Now we want to add a third column of Completed Projects to keep track of all of the tasks we’ve already done. Let’s add a third column that’s like the other two. We already have our container set to have space for three columns, so let’s just add the following HTML after your second column in **index.html**:

<div class="card-column completed-projects">

<div class="taskgroup-heading">

<h2>COMPLETED PROJECTS</h2>

<div class="ellipsis-icon">

<img src="https://s3.amazonaws.com/codecademy-content/courses/learn-css-grid/project-ii/Resources/oval-copy.svg">

<img src="https://s3.amazonaws.com/codecademy-content/courses/learn-css-grid/project-ii/Resources/oval-copy.svg">

<img src="https://s3.amazonaws.com/codecademy-content/courses/learn-css-grid/project-ii/Resources/oval-copy.svg">

</div>

</div>

<div class="card complete-i">

<div class="rectangle yellow"></div>

<div class="rectangle green"></div>

<div class="rectangle blue"></div>

<div class="rectangle orange"></div>

<p class="task-description">Research Trends</p>

<img class="list-icon" src="https://s3.amazonaws.com/codecademy-content/courses/learn-css-grid/project-ii/Resources/list\_icon.svg">

<p class="task-date">4/19/2017</p>

</div>

<div class="add-card-container">

<input class ="new-card-task" placeholder="Add a card...">

<button class="add-card">Add</button>

</div>

</div>

**8.**

We expect to complete a lot of projects, but we don’t know how many. Since those projects will be added as cards to the Completed Projects column, it will be helpful to use implicit grid properties instead of the explicit properties we have been using so far. We know the first row with the heading should be explicitly 20px, just as the other heading rows are. Every additional row, if it exists, should be set to 100px.

Hint

To size implicit rows, you can use the CSS Grid property grid-auto-rows.

**9.**

Good job! Now, feel free to play around with the layout of this site, using the other CSS properties you’ve learned. Once you’re satisfied with your work, press the Next Button to move on.