

Flex Dashboard

Required Tools: Text Editor and Google Chrome Web Inspector

There was a lot of information to take in. There are so many ways to lay out our items, but the most modern way is the easiest and most...flexible! The best way to get the hang of laying out a page is to do it. Let's make a cool, modern dashboard using our cool, modern tool--flexbox!

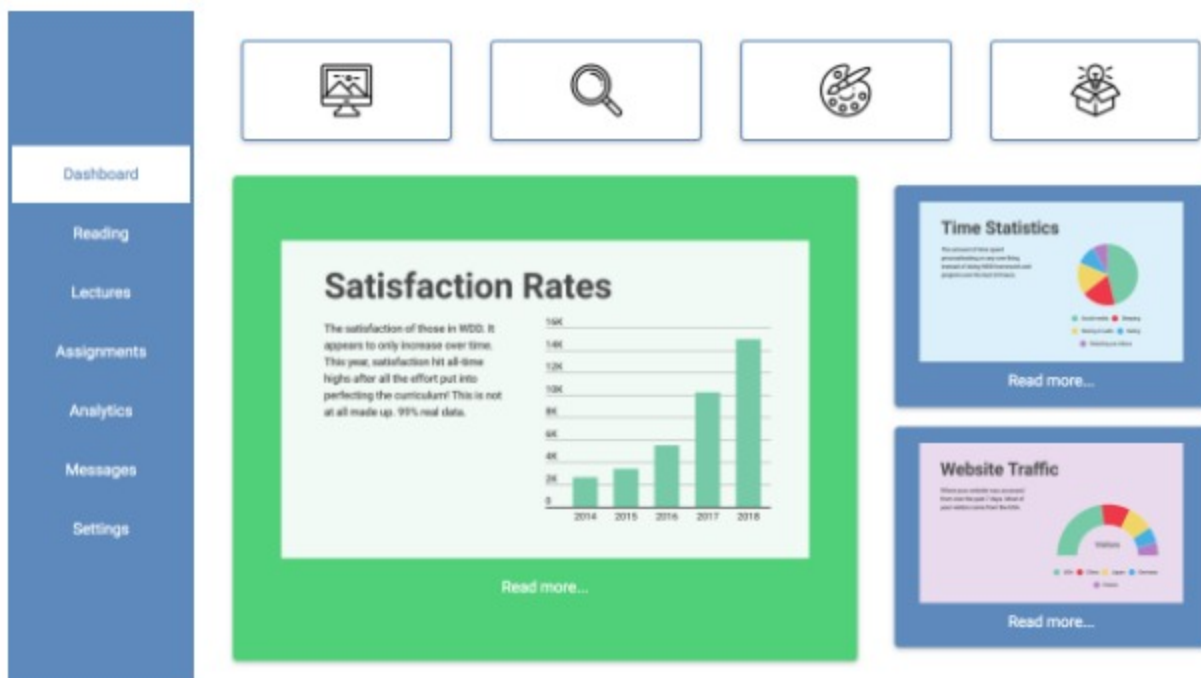
Assignment

You must NOT make any changes to HTML document in these questions. All of your changes should only be in `style.css`, except for the design response (you know the drill!).

Download Assignment

What we'll be doing

This time, the end product will be shown first, since it's much easier to use flexboxes when we have a visual of what we're going for. Below is the dashboard we'll be putting together in this assignment.

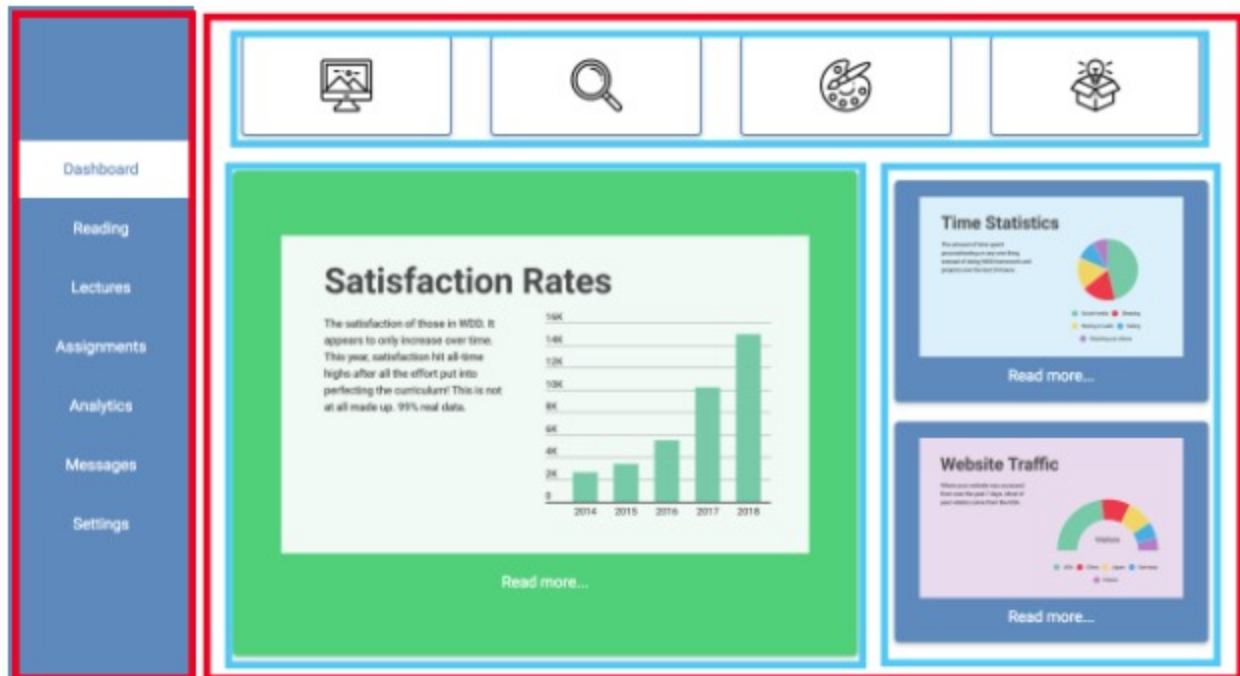


Hint: All we're going to be doing in this homework is positioning everything you see with flexbox. All other styling has been done, but you can always tweak whatever you want as long as it doesn't break things.

Part 1: Thinking About Structure

If you look closely, there are four main sections this design can be split into. Think about what these are for a moment. *Really* try to think about how we'd split this into boxes, rows, and columns yourself before you read the next paragraph or look at the picture so you can get used to thinking about constructing templates and boxes in CSS.

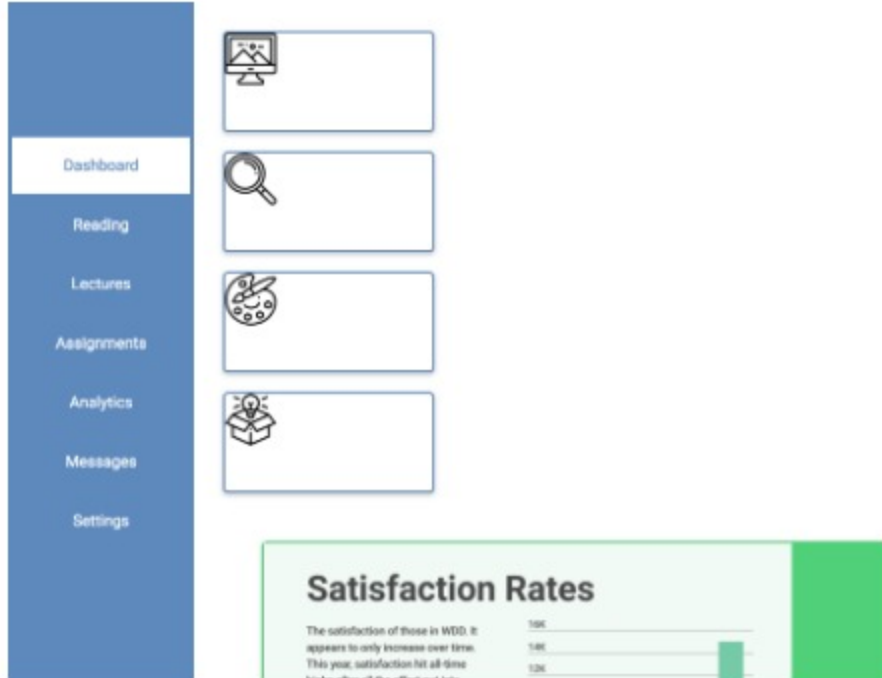
The four main sections are the sidebar, the row on top with four boxes, the large chart, and the column with small charts. However, the three of those sections (outlined in blue below) are all in one main container, which we will call content. So right now, we will just concentrate on the sidebar and the content (everything right of the sidebar).



Hint: The two main containers are outlined in red. The three sub-containers are outlined in blue. We'd want to split them this way because we're thinking about rows and columns. Note that the two bottom blue sub-containers, the charts, both also make up their own row.

Part 2: Tidy Up the Sidebar

Let's use flexbox on our sidebar so that the list of links is centered vertically and horizontally. You'll have to change two parts: `.sidebar` and `.sidebar-item`. Both will have to be flexboxes (use `display`). You'll have to use the flexbox centering methods you learned in class. After your changes, the sidebar should look like the one below.

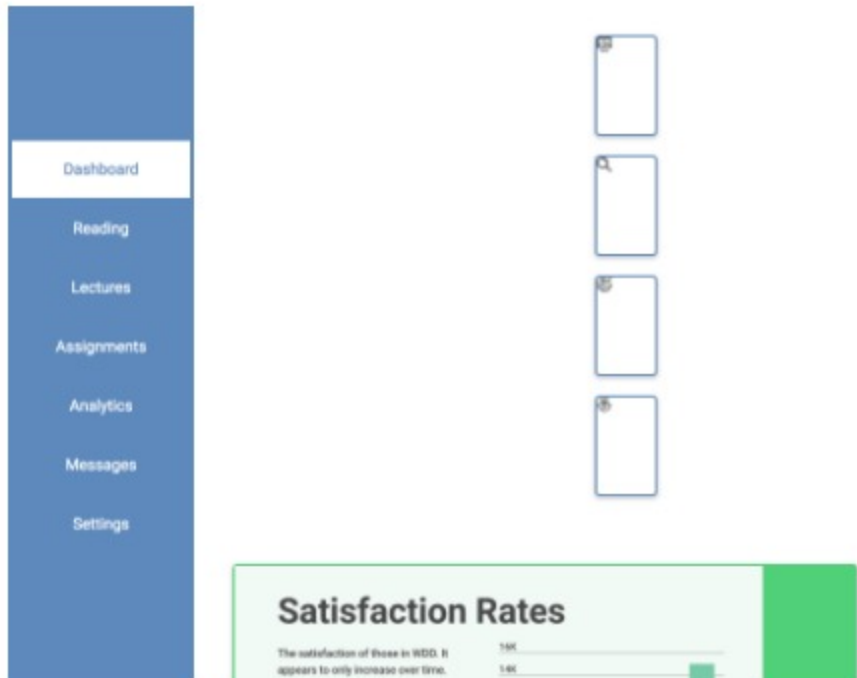


Hint: Remember the three main properties: `flex-direction`, `justify-content`, and `align-items`. Use some or all of them together to get your desired position.

Part 3: Setting Up the Content Box

Now that our sidebar is fixed up, we can start positioning based on the three boxes (or containers) within `.content` that we decided on earlier (the three blue boxes inside the large red box). To do this, you'll have to look at the HTML a bit in order to understand the structure. The `.content` div has two child divs: the one that contains the row of boxes that go on top and the one that contains the charts.

Knowing this and knowing that these two divs stack on top of each other, use the correct flexbox direction. Remember that you can only do this if you set its display to flex first. Afterwards, use `align-items: center` in `.content`. It seems to make it worse, but it'll be necessary to keep stuff centered.



Hint: It probably looks worse. Trust me, it's getting there.

Part 4: Sub-container Positioning

Now that we got the main container's (content's) structure done, let's move onto the sub-containers. First of all, let's fix the ugly `info-box-container` and `info-box`.

Considering the boxes are all on one line next to each other, decide on the correct flex direction and add that property along with the correct display to `info-box-container`. Then, horizontally center them on the page. Which property centers horizontally given the flex direction you had to use?

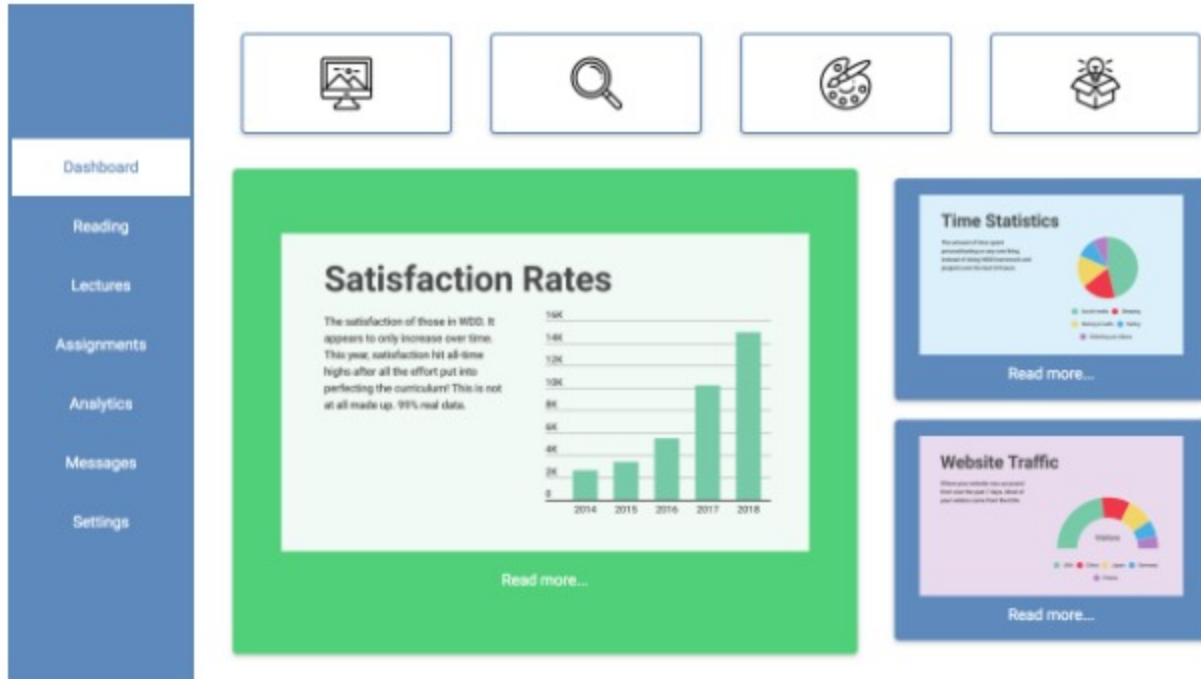
Now, center the items inside each individual `info-box` also using flexbox. Your row of boxes should look like the below image.



Not really a good hint: Flexboxes in flexboxes! Center, center, center, center, center!

Part 5: Charting out the Charts

We'll be looking at `.chart-container`, `.chart`, and `.small-charts-container` for this last part. You're on your own here. Think about how you would go about making it look like the below picture using all your knowledge about flexboxes gained thus far. Look at the HTML structure and refer back to the picture with the outlined box structure in Part 1 if you get stuck. At the end you'll have the completed dashboard!



Hint: This should take no more than 10 lines of CSS, and we are only using the properties we've been using during this homework.

Design Response

Click "Reading" on your new Dashboard to get to this week's reading! Or do it the boring way and go to the link in `design_responses.html`.

Submission

Submit the following files as a zip file through Assignments under Resources tab of the WDD Portal:

- index.html
- design_responses.html
- assets/css/styles.css
- assets/wdd.css
- assets/images/

Your assignment must be submitted as a zip file. Submission will automatically fail if your submission does not contain the index.html or if is not a zip file.

This assignment is due by Monday, March 18th at 7:00PM.