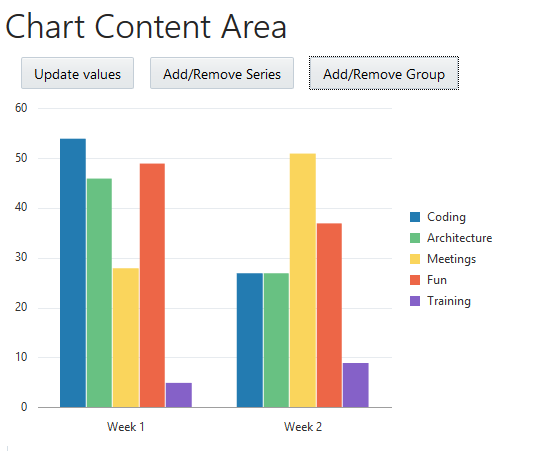
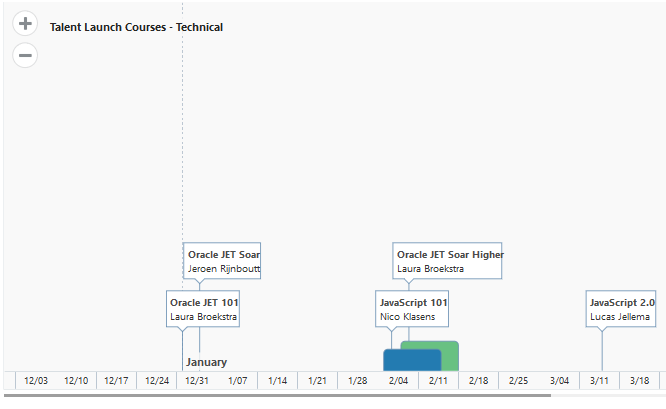
**Assignment: Visualization**

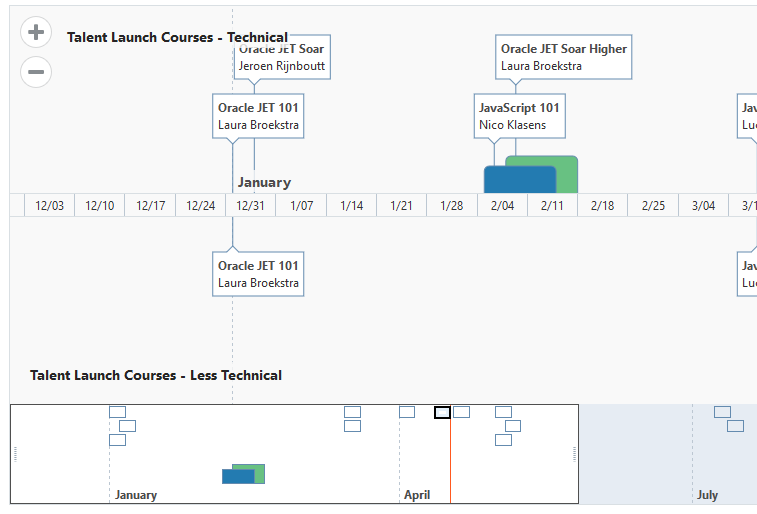
This assignment is about using the JET Cookbook and adapting that code to your own ideas. We are going to build a chart representing the number of hours you spend each week on different activities. Use the “Dashboard” tab to build something like this:



* Use the Bar Chart component from the cookbook.
* Extract the hard-coded types of work and put them in a separate JSON file.
* Use the RequireJS text plugin to load this data in your application.

Second component is a timeline that you can fill with events. You can scroll through time and zoom on a specific period. Try to make a timeline like this.

* Try to use start and end dates to see periods in the component.
* If you have time, take a look at the other options for the timeline. Can you make one that looks like this:



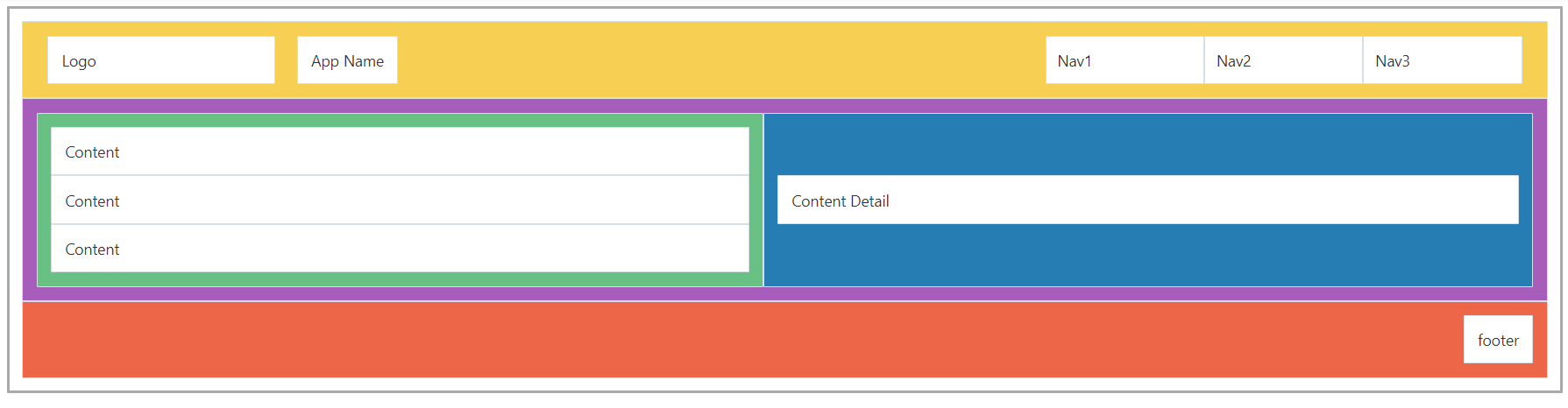
* At the bottom is a highlevel overview where you can scroll and select as well.

**Additional**

* Check the cookbook and the JSdoc to make other improvements to your chart and timeline.

**Result: Project Visualization on GitHub**

**Assignment: Flex Layout**

The flex layout is used to create a page layout that is flexible and responsive. Go to the Layout section in the JET Cookbook to discover more about the Flex Layout. Use the JS Doc, there you will find the different classes that can be used for flexibility and alignment. In this exercise you will use the different classes from Flex Layout, Grid Layout and oj-panels (also found in the Layout section of the Cookbook) to create the following design:

A few things before you start:

* Don’t worry about content, it’s more important that all panels are in the right place.
* Create a new tab in your application.
* Give the panels different background-colours so you can tell them apart (use CSS with id-selectors).
* The headerpanel also uses the Grid Layout - 12 column principle and flex-item padding.
* Not all oj-flex divs have oj-flex-items.
* The design contains 16 panels.

**Additional**

* Change the size of the browser or device (possible in the browser dev tools – F12) to see how the panels respond to size changes.
* Play around with the classes in the code to see how the panels respond.
* Fun when you’re done: <https://flexboxfroggy.com/>

**Result: Project LayOut on GitHub**

**Assignment: Composite Components**

**Build a badge**

Build a composite component to mark your progress in this course. For each (successfully) completed course you get a badge. The badge will have three properties. A URL for an image, a String for the name of the course, a Number for the grade. The badge will look like this:



**When you are done with this Composite Component and before you start with the additional part or before experimenting: copy the files and zip them. You will use this in your training for the Visual Builder Cloud Service.**

To help you with the styling here’s part of the CSS-classes:

course-badge{

display : block;

width : 210px;

height: 310px;

margin : 10px;

}

course-badge .badge-face {

height : 100%;

width : 100%;

background-color : #80C3C8;

border-radius: 5px;

text-align: center;

padding-top: 25px;

}

course-badge .badge-image {

height : 130px;

width : 130px;

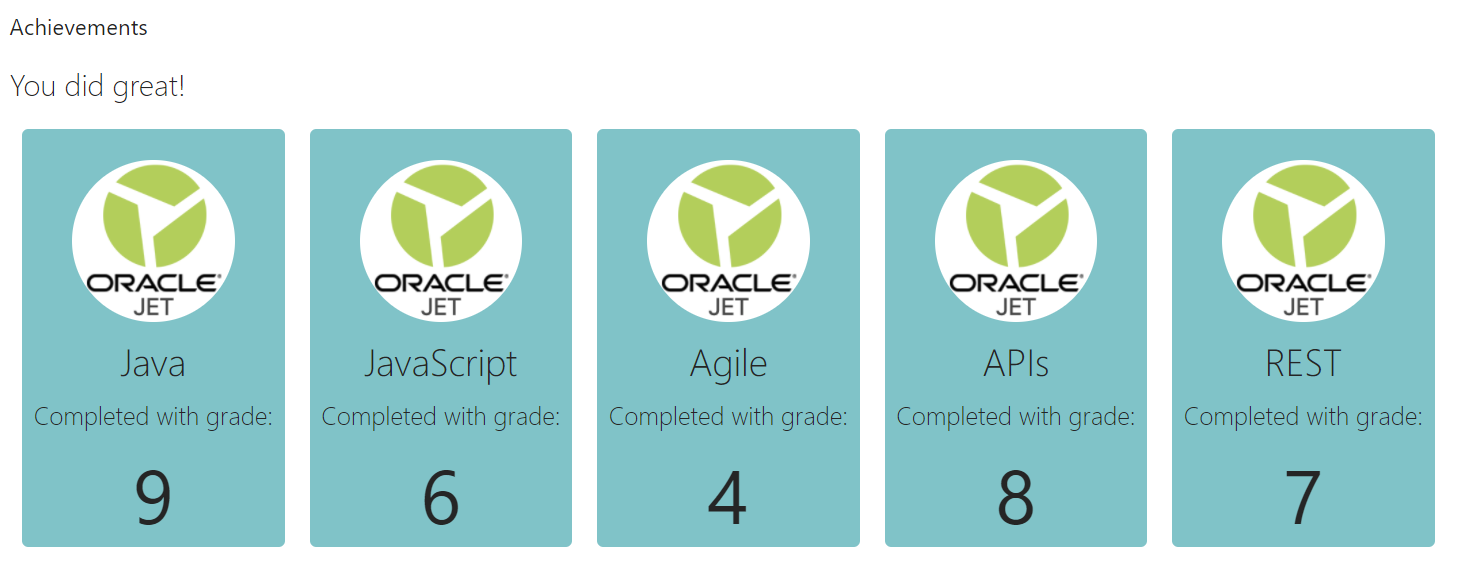
background-color : white;

border-radius: 50%;

}

**Additional**

* Add a button to the component (or make the whole badge clickable). Add a property to your component that will receive a function. Link the button to the function. A user can add his own functionality to the component. Don’t show the button when no function is provided!
* Create another composite component that includes multiple badges in a predefined slot. It can look something like this:



**Result: Project CompositeComponent on GitHub**