# ASSIGNMENT | STUDENT DASHBOARD

A screenshot of a computer

Description automatically generated with low confidence

## #Assignment\_10: Final assignment: React Student Dashboard

For this final assignment, you will make a "real-life" project. This will be a project of which the end result will actually be used by Winc Academy: a Student Dashboard!

**Goal**: to make it easy for Winc teachers to see how the students evaluate the assignments

**What is the current situation**: Excel, Excel, Excel.....

**Design**: We would like to see, per assignment, the evaluation of each student in a "Bar Chart".

**Tools:** Use a JavaScript framework such as React.

### **The data**

The link below is to a spreadsheet file (in Google Docs) with a mountain of fake data (also called mock data), with fake students. These are the results of students once they have completed all assignment evaluations.

[Winc Final Assignment – ​​Student Mock data](https://docs.google.com/spreadsheets/d/1BHjq5MjpuSItvVbnQcEdQt_v956-Ks1lr3f_nEFkTks/edit?usp=sharing)

As you can see, this data has the following structure:

* Name of student
* Name of the assignment / project (including the project code)
* Rate how much fun the assignment was
* Rate how difficult the assignment was

It is up to you to model and structure the data correctly. Don't go over typing the data, it's way too much data for that.

A good way to work with data is a [CSV file](https://en.wikipedia.org/wiki/Comma-separated_values). You can download the data from Google Sheets as CSV. You can also work directly with the data from Google Sheets.

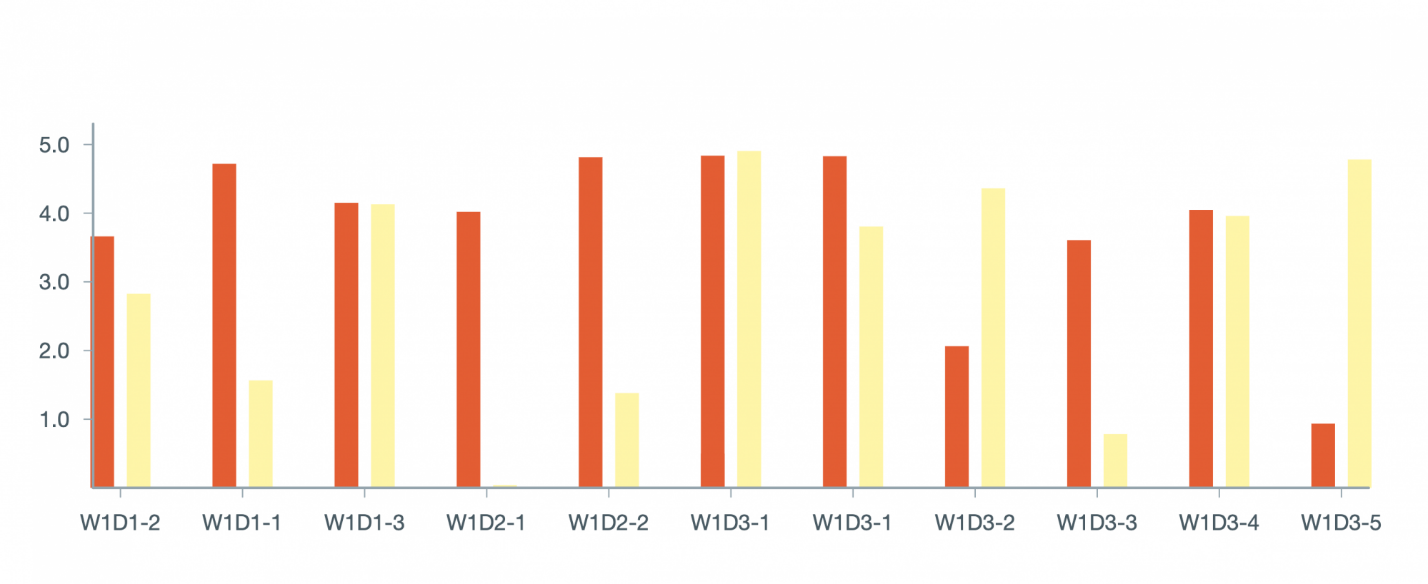
Use a search engine to find a smart way to work with the data. It can also be done in several ways. As an example, Node has a [list of packages](https://www.npmjs.com/search?q=CSV) to process CSV.

### Requirements

Your WebApplication **must display the following**

1. **Dashboard Overview User-story:** As a user, when I open the homepage of the application I want to see an overview in the form of a **bar chart** of the evaluations (fun & difficult) of **all students**.

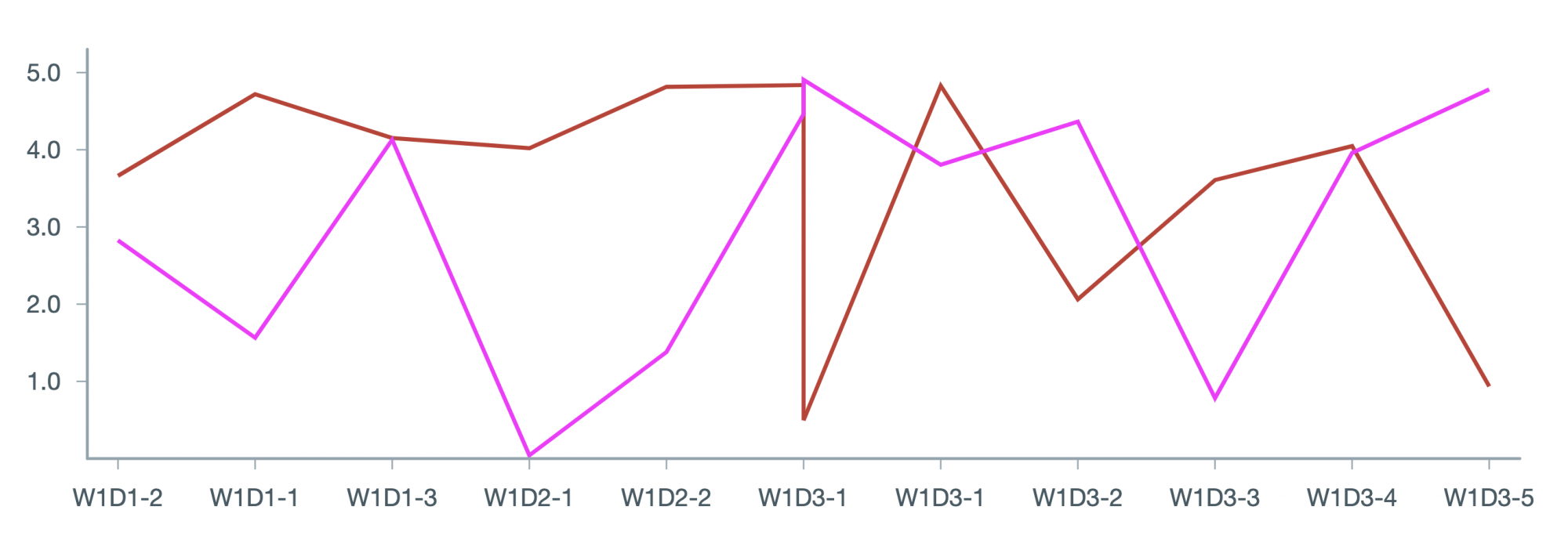
As a user, I must be able to distinguish at a glance between the assignments and the fun/difficult evaluation. Make sure that a clear distinction is made visually, for example by working with clear colours. See the example with red and yellow below.



1. **Separate routing per student** As a user I want to see a list of the names of all students and be able to click on one of these students. When I click on a student name I am taken to the route /{name-of-student}. The bar chart adjusts with the data of only this student.

Tip: the chart remains the same on the X and Y axes, only gets "less" data, namely the data of 1 student.

1. **Design**: Create a tool that you are proud of and that you would like to show to a future employer. We pay particular attention to: legibility of the graphs.
2. **Slicing and dicing. - Choose one of the methods below:** As a user of the tool you can "slice and dice" the data in [a number of ways](http://www.hypertextbookshop.com/dataminingbook/public_version/contents/chapters/chapter003/section004/blue/page004.html)".
   * Option 1: As a user, I want to be able to indicate by means of a checkbox whether I only want to show in the bar chart how nice the assignment was, only want to see how difficult the assignment was, or both.
   * Option 2: As a user, in addition to filtering on 1 person, I also want to be able to filter on multiple people. I, therefore, want to see a checkbox in the overview of my students that I can do
   * **check** if I want to **include** the data of this specific student in my chart
   * **uncheck** if I want to **exclude** the data of this specific student from my chart.
   * Option 3: As a user, I want to see a line-chart representation of my data showing the average grade for "fun" and the average grade for "difficult".

**

### Create Graphs with a Library!

You are free to choose and use your own chart library, a relatively simple library is:

[Victory](https://formidable.com/open-source/victory/). Here's an example of using the Victory graph library:

[WincAcademy/StudentDashboardExample](https://github.com/WincAcademy/StudentDashboardExample)

### Bonus features

* Make sure that the data can be "sliced and diced" in more than 1 way (see requirement 4).
* **Table overview** of all data ⇒ so as an Excel spreadsheet. You can decide for yourself how you want to structure the columns / rows:
  + *You can still filter in all the above ways*
  + *Add sort by data column*

- **User Profiles**. By using [Mockaroo](https://www.mockaroo.com/) you can retrieve objects with fake data in them. You can then add a profile for each student page and further enrich the fictitious students with:

* *Last name*
* *Age*
* *Phone number*
* *E-mail address*
* *Photo (URL)*
* *Note the rate limiting of this API - if you call too often and too much, you will be (temporarily) blocked: fetch the data once and then save it locally in a JSON file*
* Store and manage everything in Redux.
* **Sort** the bar charts of assignments by average grade (high to low or low to high).
* **Sort** the students by average grades (high to low or low to high).

### **Tips & Tricks**

* Make a plan: don't dive right into the code. First, carefully consider what is being asked of you:
  + What data do I have available?
  + What should I eventually do with this data?
    - *Keeping that in mind: in what form do I have to "cast" the data to enable the features?\**
  + Draw out which components you need (think) – try to make a subdivision right away between (smart) container components and visual (dumb) components
  + Also think about the structure and files of the different parts (don't forget that an appropriate and good name is worth gold!)
* "Copying" is allowed - help each other(!) and use every source you can find (online). But, of course, do not copy code (= plagiarism).
* You may use a CSS Framework (such as [Bootstrap](https://getbootstrap.com/)).
* If you need a reminder on how to start a React project: look at the lesson you've had about this.