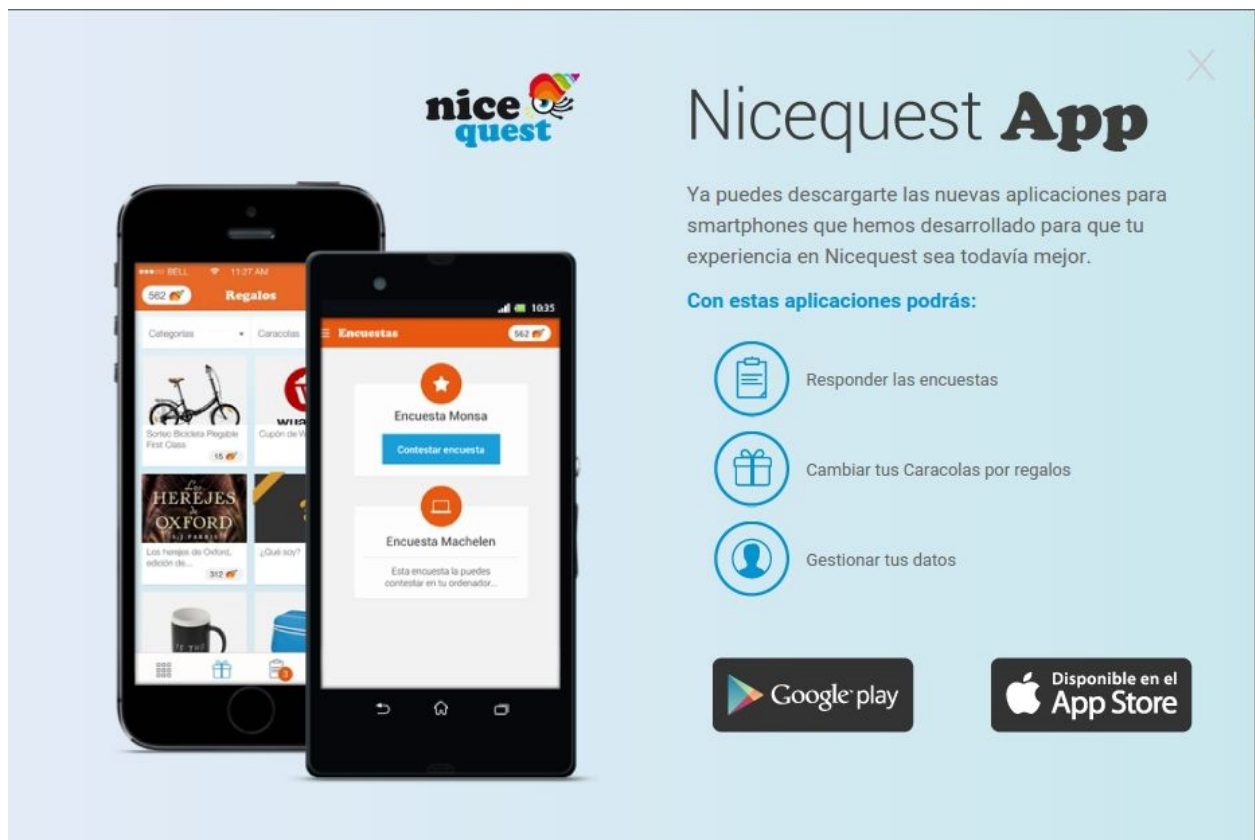# Instructions

The purpose of this assignment is to see how would you develop code. It's important that the UI components adapt properly to different screen sizes.

Regarding the code you must develop, do it like it was to be on production. Follow your own coding conventions but make it coherent through all the code you develop. Write clean, maintainable, quality code using the all components you need.

Assignment 1

Provide the HTML & CSS code for a page like this:

When the screen size is small enough it should adapt responsively as shown below:

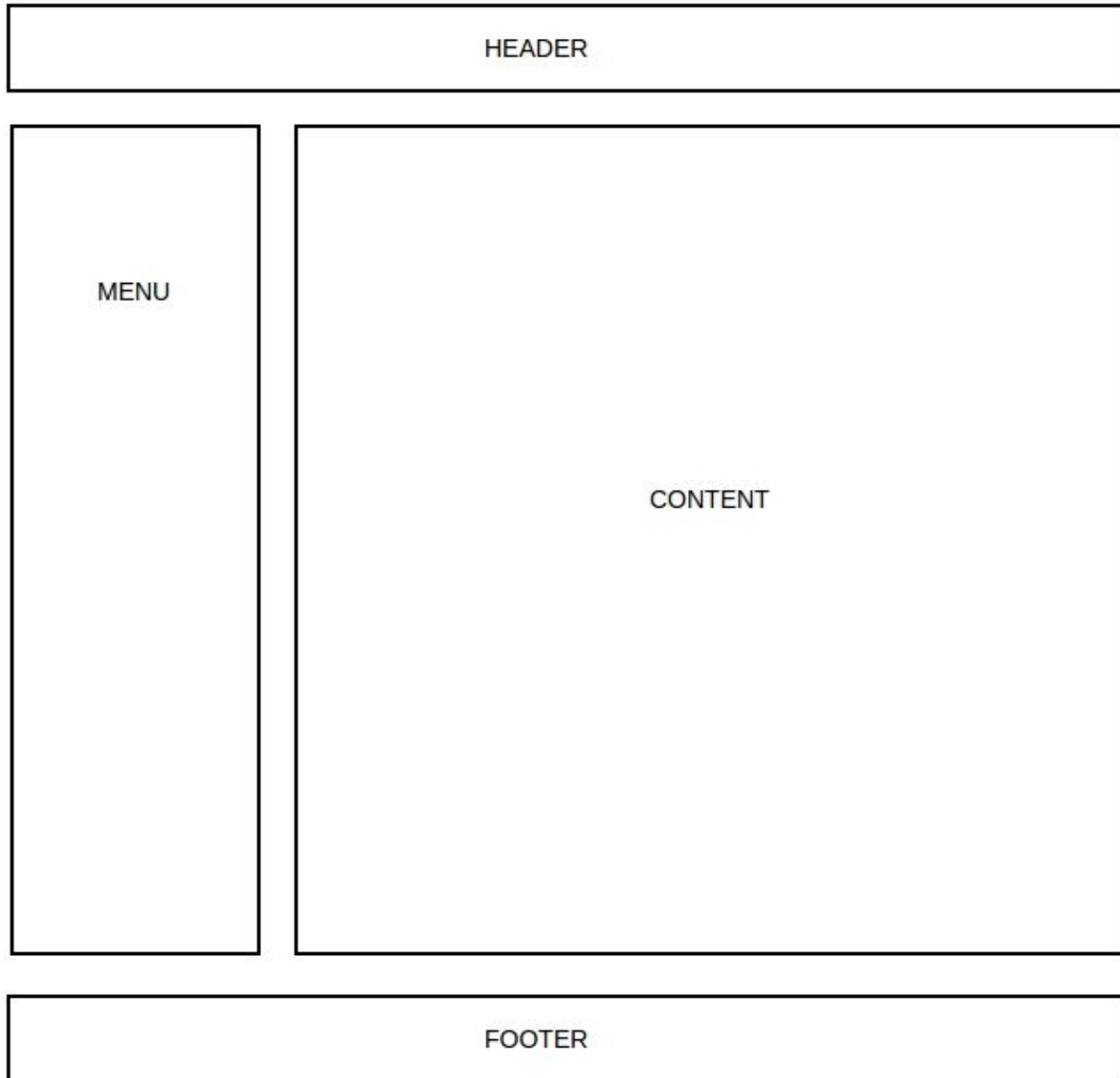And even more for small portable devices:

We'll provide you the required assets.

Assignment 2

Using a client-side JS library/framework of your choice let's build a simple single-page application.

The layout structure should be the following:

```
+------------------------------------------+
|                 HEADER                   |
+------------------------------------------+

+----------+    +-------------------------+
|          |    |                         |
|          |    |                         |
|          |    |                         |
|          |    |                         |
|  MENU    |    |        CONTENT          |
|          |    |                         |
|          |    |                         |
|          |    |                         |
|          |    |                         |
+----------+    +-------------------------+

+------------------------------------------+
|                 FOOTER                   |
+------------------------------------------+
```

The header and footer content is up to you, any "lorem ipsum" kind of text will make the trick.

The app is like a back-office application which should allow the user to manage a catalog of gifts. Gifts are grouped into categories. A gift has a name and a price, and a category has just a name.

Through the menu and the content sub-views, we need to offer the access to the following features:
- Create/Read/Update/Delete category
- Create/Read/Update/Delete gift

When creating or updating a gift, it can be included in one or more categories.

To focus on the front-end code, there's **no need to store the data in a persistent manner**. If the page is refreshed is ok if everything is lost. On the other hand, since a real application would be making requests to the server side, you should simulate those calls (for instance, it's ok just by making actions which would require interaction with the server to wait for some seconds before completing). The important thing is that **only the dynamic data** should take some seconds to be rendered, **not the whole page or section**.