McFat Web Application Documentation

Web technology

Winter semester 2018-2019

Jens Mellberg - Matteo Miloš - Bastien Pery

Table of contents

1 Application description

Mac group was composed by four students and four different nationalities. Croatian, French, Swedish and Finnish. Unfortunately, one of us drop the course just in the middle. But, As part of our Erasmus exchange program, we wanted to develop a web application which could be used by a lot of people around the world and especially by Erasmus people.

Our application reached this goal because, thanks to it, every european student is now able to find important information about McDonald in Europe.

Our McFat Web Application is divided into four different parts, each including different functionalities:

- MacHome
- MacMap
- MacStats
- MacMenu

2 Functionalities

2.1 MacHome

The MacHome page is basically a simple home page which includes 3 different things. A McDonald logo, a brief description of the application and a navigation bar which helps you to reah any other part of our application.

2.2 MacMap

The MacMap page is a little bit more complex. Indeed, like all other pages, it includes the navigation bar but also a map and one button.

The map is rendered by an external map API provided by `https://www.esri.com`. Thanks to this API, you can easily navigate on the map, zoom-in or zoom-out on it. Also, you can see that almost all european McDonald locations are displayed on it. (We linked some data with the API). If you click on one location, it will display the latitude, the longitude and the city of this location.

The second thing on this page is the button "Display your coordinates and find the nearest McDonald!". If you click on this button our application will give you your location (if you accept it) and give you how far is the nearest McDonald and in which city!

2.3 MacStats

On the MacStats page you can find some new different elements. Indeed, on this page, thanks to 2 different forms, users can display some statistics about European McDonalds.

For example, you can select as many countries as you wish and choose which data you want to know such as number of restaurants in these countries, number of restaurants per 100k inhabitants... When you click on the button "Draw Statistic", some charts will appear and show you the data you asked for.

2.4 MacMenu

The MacMenu page is the page where user can create and see his order. Meals are divided into 5 categories: main meals, side meals, salads, drinks and deserts. Each user can add as much meals of each category as he wants. Meals are added using drag-and-drop functionality. As user is adding his meals, concurrently our web application is calculating how much calories will users order have. At anytime, user can remove any of his meals from the order by just clicking on the picture of the meal. As the meal is removed, also the sum of calories will be downgraded for appropriate amount. When user is finished with his order, if he wants, he can write his name and email, and order will be sent to his email address.

3 Web technologies used

In our application we used a lot of different web technologies.

First of all AJAX. Our application is an asynchronous web application. To perform this task, we used the jQuery API.

We also used some HTML5 technologies which were required:

- Canvas as graphic technology for the statistic charts on MacStats
- WebStorage as database to store all the McDonald locations and do some analysis on it
- WebWorkers on the MacMenu page which calculate how long we need to create an order
- The Geolocation API to find the user's location on MacMap
- Some HTML5 forms on MacStats
- The drag and drop HTML5 techonlogy on MacMenu.

4 Separation of concerns

Separation of Concerns(SoC) refers to the idea that computer programs should be separated into distinct sections so that each section addresses a specific concern. By distinguishing parts of our code within well-defined interfaces, we can create programs that have distinct subroutines mapped out into more readable terms. In doing so, we can maintain order within our program while allowing for more concise logical conclusions on a step-by-step basis.

In our web application, we used Web Storage as model part of our app. We were using local storage functionality, which stores data with no expiration date. In our database we were saving two types of objects, one which had latitude, longitude and city of each McDonald's restaurant, and one which was containing type, name, and amount of calories for each of our meal. That data was accessed from controllers we implemented in JavaScript. After we have loaded that data, we were using a lot of different computations based on it in our controllers. All that was necessary because of view part of our app, which we implemented as html files. There we have rendered all the parts of our app so it is visible to users. Also, there were some dynamic computations we needed based on user activity, such as calculation of total calories, so after every meal that was added to the order, our view and controller part communicated because of calculating and rendering new sum of calories.

4.1 MVC

As we said before, we used a lot of different technologies with different roles in our application. To make it maintainable and easier to understand for developers, we divided our all application following the MVC pattern.

We have different controllers which are controlling views update, making computations and be the link between the view and the database.