Pic(k)it – My first project for Mashups

<a>Pic(k)it</a> is a website that prompts users to input a location from anywhere in the world and uses it in two ways. First, it gives a visual representation of the location in the form of a labeled map and second, it fetches and displays the most recent pictures trending at that location.

Initially, as you can see from my second blog post, I had planned to use the user’s input location in a different way. However, I switched ideas because I wanted to create something impactful. By styling the map and representing the individual pictures in the form of a polaroid, I have tried to build a beautiful user experience. Thus, far from a service effective application, pic(k)it is purely intended to be fun, interesting and cool.

I will now shed light on some of the tools that I have used to build the application, starting first with the data flow. Three APIs lie at the very core of pic(k)it, namely, Google’s Geocoding and Map APIs and Instagram’s insert name API. The Geocoding API takes an input address, queries this string with Google’s location database and returns its latitude and longitude coordinates. The more specific the input address, the greater the accuracy of the coordinates. The geocoding step is essentially what triggers the rest of the application and is thus, the first (and only, at the moment) form of user input.

For the second step, I pass the returned geocodes to instagram’s media search API that fetches the most pictures that have been posted in and around – within 1.5km of – the input address. I use geocodes instead of hashtags simply because the former yields more accurate results. When the instagram API successfully returns the data, I move on to make a third request, this time to the Google Maps API to initialize a map on the page. The map displays a (bouncy) marker of the user’s input address, the area from where I fetch the trending pictures, and flags that point to the exact locations of these pictures.

Requesting, parsing and outputting data were some of the straightforward parts in the process of building pic(k)it. The area that I struggled with the most was building the client side experience, particularly the designing, styling and positioning of elements on my page. Because I intended for a slick representation of the fetched data and since I was working with pure CSS styling for the first time, I put a large amount of effort in the design of the webpage. I also played around with jQuery to add animation to the webpage and make it a bit more interactive.

The final design of pic(k)it is the result of inspiration I found from some beautiful websites, such those of Apple, AirBnB and KissMetrics. I tried to imitate the design of the front pages of these and similar other websites in trying to grasp the user’s attention.

There are several ways in which this project can be improved or be implemented at a greater scale. One idea at the top of my head would be to implement a users’ account system through which users can take, upload and tag their pictures at different locations. Pic(k)it would create and maintain polaroid timelines of these pictures and thus, help create memories.

Please let me know what you think! Any suggestion, improvement, enhancement or critique will be appreciated.