Simple website development

Sklyarov M.S., *st. gr. ESm-31*

The history of web is not so long. It's only 20 years old. By January 1993 there were fifty Web servers across the world, by October 1993 there were over five hundred. Beginning from 2002, new ideas for sharing and exchanging content such as blogs and RSS (Rich Site Summary), rapidly gained its popularity on the Web. This new model for information exchange is called Web 2.0. Instead of using static HTML (Hyper Text Markup Language) pages, nowadays there are a lot of the websites where you can actually change the content by yourself without editing the HTML, for example facebook.com, twitter.com, vk.com and much more others.

What do you have to know to begin your journey in a web development?

Actually nothing. You have to grab a book about basics of HTML, CSS (Cascading Style Sheets) and go through it. Or you can watch the series of videos about this technologies at lynda.com or any other website. That will give you basic concept how it works together. It is really easy and everybody can do this. There are only 89 HTML tags at all. And if we talk about cascade stylesheets - it's a styling for your page, written almost in plain English. That will be enough to build any look of the webpage you want. If you'll need some interactions inside your page without reloading it, you'll need to learn JavaScript. Don't be scary it's not hard at all. This programming language is lightweight and is executing in your browser, so you don't have to know how to work with threads, control memory usage, etc.

You don't need to learn all HTML tags, all parameters from CSS. If you get the basics and see how it works together - it's enough to go further. You will grab missing knowledge in the process of producing websites.

Backend development.

The last, but not least part of developing a website is how you do your backend development. In the beginning there was only PHP (Hypertext Preprocessor), which was embedded into HTML code of your page. Nowadays, there is a lot of alternatives with better syntax, object-oriented way of development, easier database management and generators with clear, PHP-free HTML output. You can use Ruby, Python or even PHP languages with entire web-frameworks, such as Rails, Django or Yii.

What is a framework?

Framework is a collection of tools, such as classes, objects, methods, generators, which help you to do your work faster and easily. If we are talking about Rails - it has a lot of tools to help you:

1. It can be connected to any existing database without any effort.

2. It has an implementation of Active Record software design pattern, which means, that you will never be able to make a mistake in your SQL (Structured Query Language) statements, because you will never write those. It really makes your life much more easier and your code much more cleaner, because you don't have to learn one more language. Also when you switch from one RDBMS (Relational Database Management System) to another, you don't have to rewrite all your SQL statements.

3. It has a lot of generators, which help you to create models, views, controllers, javascripts, CSS files, etc. With generators you can automate almost everything you are doing by hands in every project.

4. It gives you a very easy way to use the work of other people. I'm talking about the huge amount of Ruby Gems (libraries, which are developed for different purposes by developers all over the world). If you need authorization in your website - use CanCan gem, for authentication you can use Devise, if you need a rich text editor - choose the one you like the most and add it as a gem into your app.

Is it still too hard to learn?

Don't worry! If you are going to build blogs, shops, simple websites, there are a lot of website constructors for you, such as WordPress, Joomla, PrestaShop. To use them you don't have to know all that stuff, I mentioned before. They are pretty much like word processing applications, but besides text, they have buttons, links, etc. Work process with this constructors is 90% drag and drop, you just look for plugins you like the most and add them to your app. Some of them have options, which you can change to customize plugin appearance. Also it is much more faster, then manual building of the website, using HTML, CSS, Javascript and backend development, but the result is not scalable, not flexible in adjustment.

So, if you want to build web applications (websites) for small businesses and regular people - you can use a constructor, but if you have an intention to build next twitter or Facebook, then you should dive in learning web development much more serious. This technologies are not hard to learn, they were created to be simple, fast and powerful.

EL Adviser Pochatko T.V.