Lappeenrannan teknillinen yliopisto

School of Business and Management

Sofware Development Skills

Maslov Ilia, x083784

LEARNING DIARY, BACKEND MODULE

**LEARNING DIARY**

**Introductory Chapter**

I checked the general information and read about all the four modules that I have enrolled into. In the past, I have been participating in the high school extracurricular activities to study front-end development, with a full cycle of such a development, with a few topics concerning the graphical design and the back-end programming. Ever since, I have been rarely employed directly as a software developer (even though, I did some websites for my friends for some minor remuneration). To be frank, I haven’t really wanted to become a front-end developer, but rather a programmer, but I was told that “I should take a path to front-end dev, and then may be, if you want, you would get into programming branch of the extracurricular course.” As such, I have spent 3 years of such 3 hours/week in a classroom (a predecessor of bootcamps, but for high schoolers). In the process somehow, I wasn’t entirely sure that I “really needed this,” but since I started, I have a habit of finishing something, sort of on the auto-pilot. Quite recently, I have found that the industry has become significantly larger, e-commerce and digital marketing expanded, and if before webdevelopment was for hobbysts, now it is quite fashionable even. I have combined my previous studies of webdevelopment with occasional MOOC courses. I have got a business degree, and IT management degree on my Master’s. Now, since I am slowly finding myself in the IT industry where I have a sort of talent, I (even if I don’t get to be a programmer), still might benefit as a manager with the knowledge from these courses. Hence, here I am, studying it and refining it further.

**Learning how to do the course tutorial:**

In this part, I would aim at describing how I did the learning by doing the project.

In general, I would like to keep it at a short summary, as there wasn’t much new for me to report, other than SASS pre-processor/pre-compilor whatever it is called, which turns out to be only a shorter CSS. HTML – I knew the tags and how to structure them. DIV and page breakdown by markup – it wasn’t very hard, I practiced it for a lot, and I believe it isn’t something that is very distinctive in terms of skill. Javascript – there wasn’t that much of it either in here, and besides I have learned on how to do it before.

Something that was somewhat neutrally entertaining/novel for me was the practice doing the GitHub thing of posting/committing, which is in a way just a more advanced technology of file hosting, than is Google Drive. Also, this console commanding of npm run sass was somewhat cool in terms of making me feel like a PC wizard. Otherwise, these weren’t that much significant I would say – quite easy, and quite repetitive on learning just to how post/save the things and how to setup the working environment. Undoubtedly these are transferable skills for the software development, yet these didn’t feel as if “I know how to program now”. These were somehow “miscellaneous”-feeling to me, as if something supportive and ephemeral, like the SCRUM or agile methodologies, which don’t really demand learning, and everyone could do it.

What I did kind of like was to understand the SASS. The knowledge was concerned primarily with the use of standard CSS tricks and neats, which was somewhat clear to me back before. However, there were also some tricks as to more programming-oriented approach that exists in SASS – such as the use of @easeOut thing, which I understand as the object that is programmed separately, so that one doesn’t have to mention it several times. I would have appreciated if I got the knowledge whether this is enough for going into front-end developing (at least, some relatively unqualified jobs, which don’t demand front-end framework). Because if not, I believe I could qualify myself as a relatively qualified worker in the area of web-development, at least one which is qualified enough for certain projects. After all, I am in the “LTD. Co. business” of branding myself for the future employers and partners on the projects.

Ok, so that, and then as a final note on the whole process of doing this course tutorial as a sort of summary. I believe that for the most part it still wasn’t a very well tutorial series to study, as it mostly still included copy-pasting after him. However, given that I knew the subject in advance, I kind of extracted the value from it, at least by having a template based on which I then created my version of the website, which I have synergetically integrated with my side objectives of creating a personal website for promoting my personal brand. Anyway, that’s that, and then I got to do the final project. There, I have kind of tweaked the things that I needed to tweak and added the content that I needed to add in order to make it a finished personal website.

**Doing the final project:**

Link to the hosted video explaining the project <https://youtu.be/FRIOBUsWS_8>

(**Please note**: the website is also responsive, which I forgot to mention, including on mobile versions, which didn’t work without a certain HTML code that wasn’t mentioned in the tutorial, but obviously the project is still going to be refined, it’s just that I might do all the 15-20 features that I want to add over longer time than what is required for a 3 Cr. project course to pass, so I do it before the 31st of July).

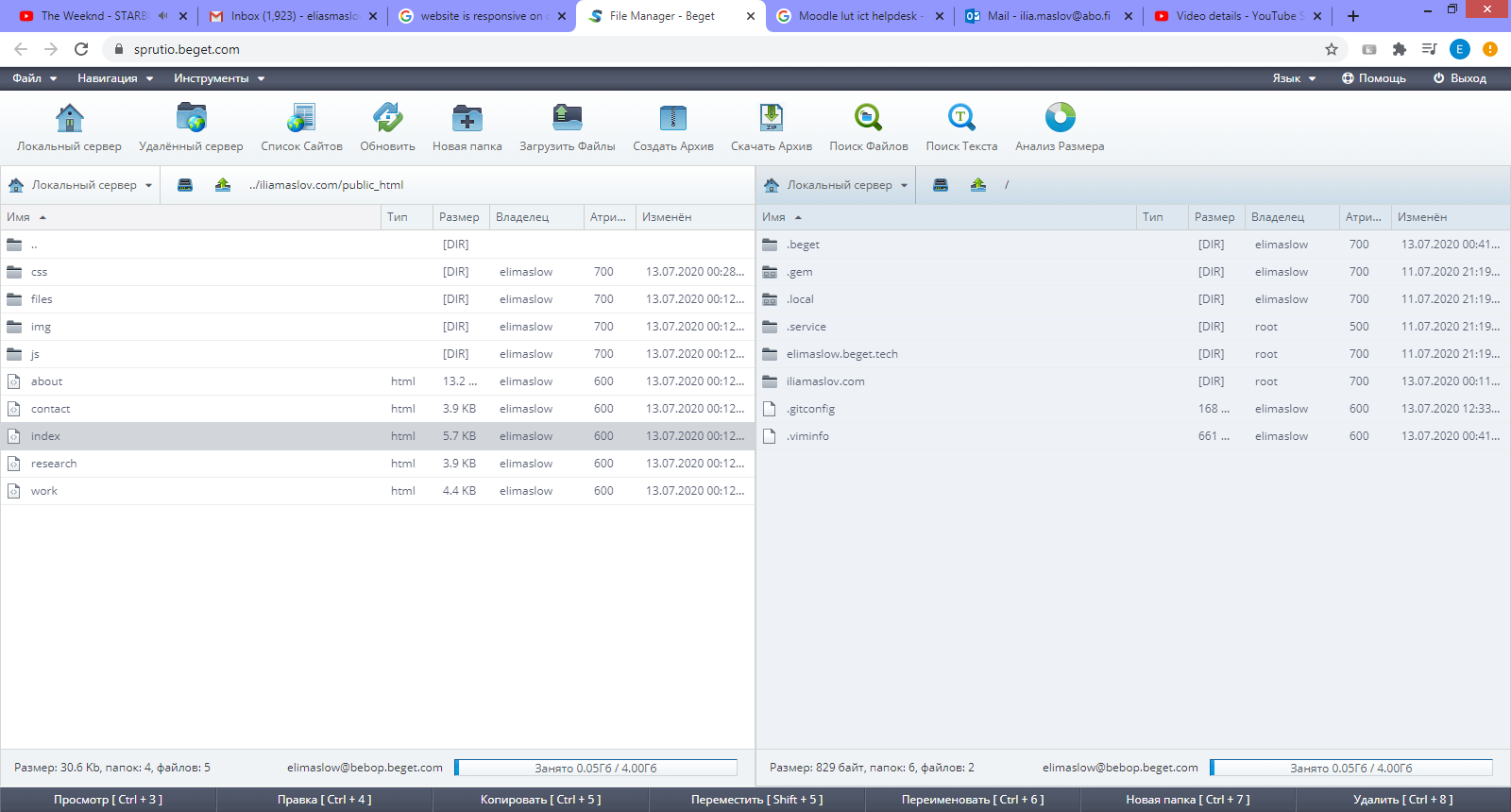
The first part of this project was really simple: create the “skeleton” of the task, such as creating folders, files that will be placed and so on. Over time, the skeleton grew in size, and so there were a multiple of tabs through which I have had to go skipping. For example, at this stage I’ve decided to add a favicon, and so I have had to copy-paste the same code to all of the existent pages (4 pages at this time, but still quite a lot). This made me think the reiterative nature of webdesign, where you are going through the code several times, so it is much better to build it all at once, but it isn’t always that easy as the creative process hits when you are doing something.

Then, I have had to create CSS. Initially I have created only the basic definitions but for this website it quickly grew in size and quite soon I have had to deal with many styles and structures. It was about the same effort spent, but if in the first part I have had to create the CSS on my own, in the second part I have had to manage the CSS, rather than create it. Of course, I have created additional styles as the website’s demands grew, but there was less of it, as compared to the first one. In the process of creating CSS and the final look of the website later on the most satisfying thing for me was that even though I have used fairly basic stuff, the website still has had that modern outlook. I suppose, this comes from the different “ideas” of creating a website, rather than from the language in which it is written.

While going through the CSS, I have had to be concerned with a marketing perspective of the website quite a lot. In this particular example, I wasn’t very focused on anything other than having a good colour scheme, have things made in a proper manner and some features here and there (such as outlining key words to the customers). I wasn’t focused on the SEO, however, during the time of making the website I have kept it in mind and could probably understand SEO better now, after going through HTML “in hand” so to speak.

So, the final project wasn’t particularly hard to do, it was just a meticulous job of adding the proper CSS and HTML codes to make it look aggregable. I really don’t know what to say, because these things don’t look very novel to me – HTML and CSS. I mean, I have added the HTML tags when I saw the need to make the page has a content that is displayed in a certain way, and edited CSS when I wanted to make these tags look more aligned to each other and more responsive on the mobile version. In the end, I progressed towards the desired outlook of the website. This is kind of agile methodology that I followed – I churned a number of tasks to do for a website, then I did them. I still have on-going series of features to do, but these would be done later (e.g., a slide show on the front page showing the photos of my previous recommending partners/bosses).

As a sort of important note – it would take to note that GitHub hosting is bad (at least, how the tutorial proposed). I have uploaded my old version of the test website version (the one that I needed to do in the course project), I uploaded it, then the code didn’t work, I had to change the code, tried to reupload it, it showed me success, then the changes didn’t show, then I removed the whole project from GitHub, then I tried to upload the project again, then some random error popped up that prevented uploading it to host over GitHub methodique, then I spent time finding what’s the case, and then I just decided to host it on a normal website, and not that thing. So, I hosted it how I knew to do, and then below is an FTP folder of the website, where my personal website is hosted. I hope, this would clarify as a requirement for “hosting a project”.



**Explaining the project:**

The home page is index.html, and it is an introductory page. It has a show case, where the background image is present that shows a good background image in style with the theme of the website (personal website). I have decided to do this website in the professional and sharp style. The color scheme was chosen white-yellow-darkgrey. It is to compliment the website’s style: sharp, but stylish.

At this page one could already observe the navbar button in the right top that is to give more modern look, as well as increase the usability of the website (works also well on the mobile, so it’s responsive). It has “current” class that will change the color to have outlined title the same as will be the page (eg Home page will have Home outlined in the navbar).

Front page also has a short paragraph description of me (I have taken the one from LinkedIn). In general, it has a quite simple structure, but I think it turned out to be not too simplistic. With more content that I could generate over time (as in more “selling” paragraphs, more SEO oriented content), I would say that even this structure would seem quite modern.

The front page ends with footer, with my name, copyright and year. This footer (together with the navbar) will stay the same throughout all of the pages.

In another page there are projects. It has descriptive sections where the project titles are described as PROJECT title, but it will be added in the future, images, and GIT hub links underneath. The projects have onhover effect that makes it fade in a little. When hover is out, it is all returned back to default.

About Me page is the most descriptive, and it hosts my education, experience, etc. etc., with most sort of “intensive information-wise” feel to it. It has some features like onhover effect that makes the boxes darken, and buttons to download the content or links to go to other parts of the website fast. Here, I plan to extend it a little, by e.g., adding social media buttons under the name, but all of these could take forever to make, so I just post the project to get the credits now, and then I would finish it at my own good pace, rather than hurry like a squirrel.

The final page is contact. It has a descriptive section in the beginning, as on other pages, as well as it has the social media icons (downloaded the external library to get the icons, which were then styled in uniform style and color as other parts of the website) and the JS form for contact. JS form has 3 variables and input button, as well as validation for each of the input.