

Experiences of Software Development Practicals

Practice Project

Jeroen Keppens

Software Engineering Group Project (2017/18)

Date: 3–24/10/2017, Place: BH(S) 7.01/2/3

Introduction

Computer Science is a discipline that spawns new technologies, paradigms and approaches continuously. After you graduate, you will have to continue to learn new technologies. If you become a Software Engineer, you should expect to have to learn at least some new knowledge and skills for many of the projects you undertake. Starting a project fully versed in all of the programming languages, tools, technologies and processes that you could possibly need in your project is a rare luxury rather than the rule, especially when you are starting your career. For that reason, you should not expect to be fully trained in everything you need for your group projects before they start. You will need to so some training yourself. Don't worry about this though: the more experienced you are with learning new Computing technologies, the easier it becomes as you will start to see how the general concepts and principles you have learned apply to many seemingly disparate technologies. This Practice Project is a first opportunity to learn something new as you develop a system.

This practice project will take place during the first 4 weeks of the Software Engineering Group Project Practicals on Tuesdays 9:00-11:00, Tuesdays 11:00-13:00 or Thursdays 16:00-18:00, in October 2017. Your task involves the development of a small webpage using HTML, CSS and Javascript. You will form a team of 3 or 4 people at the start of the first lab, set objectives for your project based on the assignment, organise your training in between lab sessions and develop your webpage during the lab sessions. TAs will be on hand to help you organise your work and point you in the right direction (though they won't solve your technical problems for you!). This project is not assessed but we will keep track of who completed the work.

Getting Started

Start by forming a team of 3–4 people in the lab session you have been allocated to attend. As this project is not assessed, you should not overthink who you work with. You could just work with the people sitting next to you in the first lab. If you're struggling to form a team of 3–4 people, it is ok to form a 2 or 5 person team although I recommend against that. You must not work independently on the project and you must not work in a team with more than 5 people. Once you have formed your team, **register your team using Team Feedback**. Then, read the remainder of this Practice Project assignment.

Assignment

The objective of this project is to develop a small website that helps third year (level 6) Computer Science and Computer Science with Management students select their module diet. The website must show students the available modules, with a clear indication of how the module fits in the programme (core, compulsory or optional), the assessment structure of module, the module's credit volume, term(s) and lecturer(s), and a short description of the content. The website should enable students to check whether a module diet is valid for their programme. Ideally, it enables students to generate a module diet that is presented in a concise table, which identifies any errors in the module diet. The website could also suggest modules that fit a particular stream or area of expertise students might wish to focus on. Ideally, such suggestions are provided even if the student does not explicitly select a stream, but has selected one or more modules belonging to a particular stream. The website must **not** rely on a backend and run entirely within a web browser.

Technologies and Training

The assignment stipulates explicitly that you must develop a web application using only front-end languages. There are three core front-end languages used in web applications:

- HTML (hypertext markup language) is a language that marks up documents with tags indicating the roles of elements in a document (e.g. title, paragraph, table, etc.). Although it is possible to use WYSIWYG editors to generate HTML, this project is going to require a little more finesse. If you have never written any HTML before, you will need to learn HTML. Don't worry, the language is easy.
- CSS (cascading style sheets) is a language that specifies how elements in an HTML page are displayed. Normally, you will be producing CSS files alongside your HTML pages so that layout and content are separated. It is a good idea to do that right from the start. You can use CSS files produced by others. However, it is very likely that you will want to tweak the layout of your pages a little so having some knowledge of CSS will be helpful.
- Javascript (or ECMAScript) is a programming language that enables you to make your pages more dynamic. For example, Javascript allows you to get your pages to respond to events that take place within the page. Once you try to implement the "should have" requirements of the assignment, you will need Javascript. Beware, although Javascript looks very similar to Java and has the word "Java" in its name, Javascript is not Java and there are some very significant differences between Java and Javascript.

There are some very good component libraries that simplify the creation of a slick webpage, such as Bootstrap, Foundation and Skeleton. You are very welcome to use these if you wish. That said, the starting point for your work is to make sure you understand the basics of HTML, CSS and Javascript. Links to training materials are provided on the KEATS page under the Practice Project topic.

Outcome

You will be deemed to have completed this project successfully if (i) you have produced a webpage using HTML, CSS and Javascript that meets the “must-have” and some of the other requirements, (ii) the webpage was developed by a team of about 3–4 people, with every member of the team making a significant technical (i.e. HTML, CSS and/or Javascript code) contribution to the work, and (iii) you have attended at least three out of four of lab practical sessions for the full 2 hours and worked on the project throughout these lab sessions. Note that it is crucial that you contribute to the code and allow others to contribute to the code as well. A website that meets the requirements but was developed by only one member of the team with others merely watching but not contributing will not be treated as a successful outcome (not even for the person who did all the coding).

If you complete this project successfully, you will be awarded a badge on KEATS. The award of this badge has *no* impact on your module mark or the team you will be assigned to in future projects.



Etiquette

In this practice project, you are required to work in a team. Therefore, your behaviour will affect the learning experience of others in the project. You should be considerate to others and try ensure everyone has a chance to get the most out of this project. There are a few behaviours that really frustrate teams and yet remain quite common in the years I have taught this module. It is very important that avoid these:

- *Don't be late. Don't leave early.* You cannot participate fully with the project if you're not there to contribute. Worse still, if you miss important discussions, you are creating work for others as they will have to inform you of what was discussed. If you are going to be late or have to leave early due to unavoidable circumstances beyond your control, inform your team as soon as possible, explain your circumstances and make an extra effort to compensate for what you missed.
- *Make an effort.* It's ok to fail at completing tasks and to find some tasks very difficult or impossible. After all, you are here to learn. But if you don't put in sufficient effort, then you are setting yourself up for failure. If you are struggling with work, do not suffer in silence. Tell/show your team what you are doing to try and succeed. Ask for help if you need it. If others help you out, watch closely how they solve problems and learn from them so that you become a better computer scientist.
- *Show some enthusiasm and be friendly.* Good teams tend to create a pleasant, friendly atmosphere where everyone is enjoying what they are doing. Of course, although it is good to have fun, don't exaggerate. Be sensitive that there is work to do and incessant joking around can get in the way of work.
- *Don't be shy, express your views.* Your team will want to know you are on board with the team's decisions and whether you have any questions or concerns. Moreover, you may have interesting ideas and important issues to raise.

- *Listen to others.* People will be more productive if their opinions are valued and the concerns are heard.

Suggested Approach

Practical 1: 3 or 5 October 2017

After you have formed your team, start the project by specifying what you would like to achieve by the end of the project in three weeks time. All you need for this are some pens and paper. This session will involve brainstorming and exploring ideas. Try to keep your conversation focussed and maintain progress.

Projects require clear objectives so that you can aim your efforts towards a clear goal. If you fail to define your project objective clearly, you risk wasting effort on work that does not contribute meaningfully to what the project should achieve. That does not mean your objectives cannot change throughout the project. If your progress on the project or obstacles you encounter indicate that the project objectives are no longer achievable, then you must adjust your project objectives accordingly. But even though project objectives can change, you still need them to guide your work.

In this particular project, which comes with a rather specific and narrow set of requirements, it would be a good idea to set your objectives **analysing** the project requirements with a view to specifying **what** website you want to build. The first step in your analysis might be to produce:

- A list of the information items your web page will need to display.
- A set of scenario you envision your users will be in when they access your system. Consider what information your user will have and be prepared to enter when they access your website and what information they will expect to find in your system. A good starting point for identifying scenarios might be what you would expect to find and do when selecting your third module diet.
- Based on the scenarios, a list of features you expect your webpage to have. Add some notion of priority to the features of your webpage, e.g. distinguishing between “must-have”, “should-have” and “could-have” features.

As this assignment is rather simple, you should be able to complete this analysis within 30–40 minutes.

Based on this initial analysis, you can **design** a web page. This design is likely to include:

- Some sketches of what the website will look like, including the various widgets on the page and the layout.
- A list of behaviours (or event handlers) your page will require. Identify which events the page needs to respond to and what the effect of the event should be.
- If time permits, you may draw your scenarios as story boards.

Again, you should be able to complete this discussion within an hour.

Finally, in the time that remains, produce a **plan** for completing this website. You should not seek to implement this webpage in between practicals. Instead, you should complete your training in HTML, CSS, Javascript and anything else you plan to use in between practicals,

and do all of your coding during the practicals. That means, your total development time equals up to 6 hours of development time per person multiplied by the number of members of your team. Coordination and group coding sessions will reduce that time significantly. Develop your plan within those constraints as follows:

- Identify the work (parts of the design) that you plan to implement in each practical. If possible, aim to complete the work early into the fourth practical as that will allow you to cope with unforeseen difficulties.
- Identify the training you will need to do in between each of the practicals. I would strongly recommend that all members of the team complete the same training in between each session.

Write down your plan in a Team Feedback meeting. **You should have a plan and training allocation by the end of the training session.**

Practical 2: 10 or 12 October 2017

Start the practical with a short meeting. Each member of the team should state in 1–2 minutes what they did (learned) since the last practical and any difficulties/problems they experienced. Then, someone should review the tasks you had agreed to complete in this meeting. If anyone has any concerns about the team's ability to do the tasks allocated for this practical, now is a good time to raise these. Decide how you will complete today's task and who is doing which subtasks (if working as individuals or in sub-teams).

It would be a good idea to strive to complete the static page content (HTML and CSS) in this practical, so that you can start work on the page's behaviour in the next practical. As you are working on the static pages, produce the most important content first. You may discover that your plans were a little overambitious (the opposite problem of an underambitious plan is extremely rare). Be prepared to scale the work down if needed.

15–20 minutes before the end of the practical, you should meet again with the whole team. Create a Team Feedback meeting for this purpose and take attendance. Each individual or subteam should discuss progress made. Revise your project objectives, plans and training requirements as necessary, and record these commitments in the team Feedback meeting minutes.

Practical 3: 17 or 19 October 2017

Start the practical with a short meeting. Each member of the team should state in 1–2 minutes what they did (learned) since the last practical and any difficulties/problems they experienced. Then, someone should review the tasks you had agreed to complete in this meeting. If anyone has any concerns about the team's ability to do the tasks allocated for this practical, now is a good time to raise these. Decide how you will complete today's task and who is doing which subtasks (if working as individuals or in sub-teams).

In this session, you should aim to incorporate behaviour into your webpage using Javascript. Focus on easiest and/or most important behaviours. If you're new to Javascript coding, you may find pair programming very helpful.

15–20 minutes before the end of the practical, you should meet again with the whole team. Create a Team Feedback meeting for this purpose and take attendance. Each individual or subteam should discuss progress made. Revise your project objectives, plans and training

requirements as necessary, and record these commitments in the team Feedback meeting minutes.

Practical 4: 24 or 26 October 2017

Start the practical with a short meeting. Each member of the team should state in 1–2 minutes what they did (learned) since the last practical and any difficulties/problems they experienced. In this practical, you need to finish your website as you will move on to another project in the next practical. Therefore, it is extremely important that you focus on finishing tasks. Therefore, anything that is important and unfinished needs to be finished first. Avoid starting too many new tasks as it is pointless to work on something that you will not be able to finish.

Aim to complete your website at least 30 minutes before the end of the practical. Once you are done, ask a TA to review your website and assess whether you have completed the project successfully. Please note that the TAs may be very busy in this session and you will need to give them enough time to look at your site. If you are finished with the website in this practical, but insufficient time remains for the TA to look at your site, they will review it in the next lab session.

In the final 15 minutes of the practical, create your fourth and final Team Feedback meeting. Take attendance. As a team, list your most important achievements and identify a number of lessons learned (e.g. things you would do differently if you were to do this project again).

Peer Assessment

After the final practical, take some time to write some constructive feedback on your experience of working with each of your peers.