Term Project Assignment

Project Guide

Unlike other assignments, in the term project you may be richly collaborative in terms of helping each other, including viewing other students' code to help them debug, openly discussing designs and algorithms, etc. Using resources wisely is an important aspect of this term project. However, you still must conceive and understand your entire design and code. Within your team, the work must be shared equally.

You may use any materials, including source code, designs, images, text, sounds, or anything else, from any sources. You just need to clearly cite each use, so it is very clear what is yours and what is not, and in the latter case where the materials came from.

Your deliverable must run on Athena or a docker container. You need to deliver a single zip file with all your code and a link to the "live" version. If you dockerize your project, you need to submit your docker file (Dockerfile) and your docker compose file (docker-compose.yml).

Term Project

Your task is to design and implement a program of your choosing. You must program in JavaScript - you can talk to the teaching staff for exceptions. The project may be graphical, text-based, or even file-based.

We encourage to your choose a project based on machine learning, data science, or a IOT device prototype. Talk to the teaching staff for exceptions.

You may use any materials, including any JavaScript libraries (e.g. NPM). You should understand that using such materials requires that you train yourself on their use (presumably through online materials). TA's and instructors will try to assist you, but you should expect less direct support if you use libraries that we did not cover in class.

You may use any code, any graphics, any sounds, anything at all, as long as you clearly cite precisely where it came from and, if possible, who created it.

You have an extended period to finish this assignment, with no other required assignments during the last 3 weeks. You are expected to invest at least 30 total hours into this project in this time, and it is further expected that your project will clearly reflect the quantity and quality of design and craftsmanship that goes along

with that much time on this task. Most of you will exceed the 30-hour bar, but this is a minimum expectation for passing the term project.

What to submit

You will have 4 weekly deliverables for this term project, each graded separately, determining the entire term project grade.

We do not expect you to be experts at writing proposals, or storyboards, but be sure to give this serious consideration. We expect a large amount of effort to go into the design of your project, as well as into the actual code and presentation.

Deliverable #0

Due Date:
Thursday, Nov 3rd, 2016

Project Proposal:
Define the problem you intend to solve and a description of how you intend to solve it. In particular, list all modules/technologies you plan to use.

Team Name:
Submit the name of your team.

Team Members:
Submit the name of your team members.

Deliverable #1

Due Date:
Thursday, Nov 10th, 2016

Project Proposal:
Include a complete and thorough project proposal, clearly defining the problem you intend to solve and a fairly detailed description of how you intend to solve it. In particular, your proposal must list all modules/technologies you plan to use.

Technology Demonstrations:
For each module/technology you listed in your proposal, you must include a demonstration of your suitable competency with that technology. This demonstration will be done in person by the due date, but you also need to include in your submission the code artifacts and any other materials required in this regard. Note: anyone who cannot demonstrate mastery of their required

technologies by this date will have to modify their term project to no longer use those technologies.

Competitive Analysis

This is a written document summarizing your competitive analysis, where you search the web and find several existing products that are similar to what you propose. You need to use those products and carefully think about them, then list a few features in those products that are especially compelling, that you plan to include in your program. Also, list some anti-features, which are features you do not like in those products and plan to change in your program. Do not just exhaustively list all the features of those apps, since they may have lots of features that are not really relevant to your program. Focus to those few features that are directly relevant to what you are building.

Storyboard

Here you should include hand-drawn pictures (do not worry about making them pretty) showing how your app will work from the user's perspective. Using your storyboard, you will walk through the user experience of your program, and you should do this prior to writing lots of code.

Code Artifacts

You must have some early work up and running, some compelling demonstrations of various technologies (not just the default demos that most modules provide, but your own code, even if based on those demos). In addition, you need to have some code that clearly is attempting to solve some of the core problems you are addressing. This does not require a working demo, but it does require that you must show some very clear progress in that direction.

A timesheet

In your top-level project directory, include a file named "timesheet.txt". In this file, keep track of the time you spend on the project. There is no specific time requirement, and more time will not necessarily equate to a higher grade. It is expected that you should invest at least 10 hours per week and 30 total hours to obtain a satisfactory grade.

Deliverable #2

Due Date:
Thursday, Nov 17th, 2016

Previous Files and Updates
Include all the files from the previous deliverable, and also include the file updates1.txt that explicitly calls out any updates you may have made to the documents you submitted in the first deliverable.

Working Demo

At this point, you should have a working demo of your project, approximately at what would be considered a C-level final project. It may be missing features, it may have a less-than-polished user interface, it may have bugs, it may even crash more than desired, but it has to basically work for some reasonable definition of "work". Again, your project must clearly show results consistent with at least 10 hours per week of productive effort.

Video: Initial Assembly - Rough Cut

Intro 1 min: put yourself on camera introduce your idea and project.

Initial Assembly: The selected shots are moved from the order they are filmed in into the approximate order they will appear in the final cut.

Rough-cut: More shot selection, approximate trimming. The sound is untreated, unfinished, and will require sound editing. Often dialogue and sound effects will be incomplete. The titles, graphics, special effects, and composites are represented by by crude placeholders.

A timesheet

See above for details.

Deliverable #3

Due Date:

Thursday, Dec 1st, 2016

The Project Zip File

Submit a single zip file named termProject.zip containing all the files described below except your term project video, which will be handed in separately (as described above).

Video: Final Cut

Final cut: The final sequence of images and sound are selected and put in order.

In the file project-video.txt, include the URL of a web-hosted (say, in YouTube) short video (3-to-5 minutes long) demonstrating your project. Do not show every last feature. Instead, focus on the most important features, the highlights, and the parts that are distinctive.

 You can use a screen capture utility with a voiceover. Or you may use a webcam or other video camera (even a decent cell phone camera) and directly video your screen. You do not need expert production value, and in particular you do not have to extensive editing of your video (and so you can have pauses, imperfect scene changes, and occasional verbal slips). However, the video should be of sufficient

quality, and should clearly demonstrate your project, such that we can use it as part of our grading process (though we will also run your code, and in fact you will also run it for us). We may also place (excerpts from) some videos in a student term project gallery on the course website. Also, critically, your video should meet your own standards so that you could reasonably place it in an online portfolio of your academic work, and submit it to a prospective employer as part of your CV.

Project Source Files and Support Files

Include all your project's files that are required to build and run your project.

Project Readme File

Include a file named readme.txt that explains, at a high level, what your project is and how it should be installed and run. If you use any 3rd party libraries, be sure to either include the library in your submission (preferred), or if this is impossible, to include very clear instructions on where and how to download and install the library.

Design Documents

In your top-level project directory, include a directory named "design", and in that directory place all your design documents. At a minimum, this must include a paragraph or two describing the problem you are solving and the general approach your code takes to solve it. After reading this, another programmer should be able to make sense out of the module or modules you designed and the most important functions they contain. Also, you should discuss your user interface (assuming you have one, as most of you will), explaining not just what your program looks like but why it does so. For example, many of you started your projects by studying similar programs on the web -- you should list those here and explain what was good and bad about those programs and how they influenced your design. You should also include any other supporting design documents you might have (particularly from earlier deliverables), such as storyboards or any other materials you made in the course of designing your project.

A timesheet

In your top-level project directory, include a file named "timesheet.txt". In this file, keep track of the time you spend on the project. There is no specific time requirement, and more time will not necessarily equate to a higher grade. Again, though, it is expected that you should invest at least 30 hours to obtain a passing grade.

Grading

Complexity and Sophistication [15 pts]
Projects should show a level of complexity and sophistication in their design, coding, and user experience, all in keeping with your hard-won programming prowess. They should be neither too simplistic nor too complex. They should also be interesting, though we will take a broad view of this.

Robust Operational Program and Hardware (hardware is optional) [15 pts]
Your program should work. It should load and run with no exceptions. It should not crash or hang.

User Interface [15 pts]
To the extent that it is appropriate for your particular project, you should have a polished user interface. For projects without an end user experience, these points will be distributed across the other grading criteria.

Effort [15 pts]
Once again: you are expected to invest at least 30 hours into this project, and it is further expected that your project will clearly reflect the quantity and quality of design and craftsmanship that goes along with that much time on this task.

Design [15 pts]
Your design should include well-chosen functions, data structures, and algorithms, explained in a clear design document as noted above.

Style [5 pts]
Your code should abide by all the style guidelines we have discussed in this course.

Video & Presentation [20 pts]
Both your video and your oral presentations should clearly highlight the most interesting and important aspects of your project. And while we do not require expert video production value, your video still should be reasonably well done. For example, the voiceover should be very clear, concise, and easy to understand.

Bonus [0 to 10+ pts]
For extraordinary work, we may award bonus points. Please discuss more ambitious bonus attempts with the instructor prior to investing too much time into them!