CS 290 Final Project

Code and demo due by 5:00pm on Friday, 12/8/2017

In this course, a final programming project will take the place of formal exams to test your understanding of the material. The final project will involve working with a team of 3-4 people to implement a substantial web app that utilizes all of the major components of the web application stack covered in this class. Specifically, you and your teammates will write a web app that satisfies all of these requirements:

* The app uses HTML and CSS to implement a well-designed client interface.
* The app uses client-side JS to enable relevant user interactions with the client interface.
* The app is served using a Node.js-based (or other approved) serving stack.
* The app dynamically generates pages based on data stored in a back end data store, such as MongoDB.
* The app’s client interface communicates with the app’s server to create, read, update, and/or delete content in the back end data store.

Within those boundaries, your team may write any kind of web app you want. There are infinite possibilities, and I encourage you to be creative and to implement an application you find interesting or that solves a real problem for some set of potential users. Below are some guidelines and logistics for the final project.

# Be realistic about your project

Importantly, you should also be realistic about what you can actually accomplish with a small team in a few weeks. In particular, feel free to treat the final project as an opportunity to deliver a prototype, not a polished product. If you’re having trouble thinking of an idea for an app to implement, please reach out to me, and I can help you figure something out.

One thing in particular you should be cautious of is implementing an app that requires user authentication and logged-in sessions. Getting the details of these things right can be tricky, and I don’t want you to get bogged down in these details. Implementing logins is not needed for the kind of prototype you’ll implement for this project. If you do want to implement authentication and logged-in sessions, you *must* discuss with me to get approval.

# GitHub repositories

The code for your final project must be in a GitHub repository set up via GitHub Classroom. You can use this link to form your team and create your final project repository:

<https://classroom.github.com/g/aBN2E3Ly>

The repository created for your team will be public by default, and I encourage you to keep it public. These final projects should be nice demonstrations of your web development abilities and will be a good item to have in your CS portfolio. It will be great to have the code in a public GitHub repo so you can share it easily when you want to. However, you will have full administrative control over the repository that’s created for your project, which means you’ll be able to make it private if you wish.

If you’ve already started a GitHub repo for your project, don’t worry. The repository created via the GitHub classroom link above will be completely empty, so you can simply use [git remotes](https://git-scm.com/book/en/Git-Basics-Working-with-Remotes) to work with both repositories. I can help you set that up if needed.

## Working with a team on a shared GitHub repo

When working with a team on a shared GitHub repo, it’s a good idea to use a workflow that uses branches and pull requests. This has a few advantages:

* By not working within the same branch, you can better avoid causing conflicts, which can occur when you and another member of your team edit the same parts of the code at the same time.
* It helps you to be more familiar with the entire code base, even the parts that other team members are working on, because you’ll see all of the changes to the code as you review pull requests. This can help you develop more rapidly because you won’t have to spend as much time understanding code that others have written.
* It helps to ensure high quality code. Code in pull requests is not incorporated into the master code branch until the code request is reviewed and approved. That means everyone has a chance to improve pull request code before it becomes permanent.

One simple but effective branch- and pull-request-based workflow you might consider is the GitHub flow: <https://guides.github.com/introduction/flow/>.

# Canvas groups

Your team must also put your team members into one of the pre-created final project groups on Canvas. To do this, you can go to the “People” section of our Canvas course and navigate to the “Groups” tab. You should see several groups starting with "Final Project Group". You can choose one of these groups to use for your team. If you're the first person to sign up for a group, you should be able to manage the membership of that group and add your teammates.

# Grading demonstrations

The grade for your project will include a brief (10-15 minute) demonstration to me (Hess) of your project’s functionality. To get a grade for your project, your team ***must*** do a demo. Demonstrations will be scheduled for finals week. I’ll send more details on scheduling when we get closer to that time.

# Code submission

All code for your final project must be pushed to the master branch of the repo created for your team using the GitHub Classroom link above before your grading demo.

# Grading criteria

Your team’s grade (out of 100 points) for the final project will be based on successfully implementing a complete web app that satisfies these criteria:

* The app uses HTML and CSS to implement a well-designed client interface.
* The app uses client-side JS to enable relevant user interactions with the client interface.
* The app is served using a Node.js-based (or other approved) serving stack.
* The app dynamically generates pages based on data stored in a back end data store.
* The app’s client interface communicates with the app’s server to create, read, update, and/or delete content in the back end data store.

Remember, if your team does not do a demo for your project, you will receive a zero for it.

## Individual grades

Your individual grade for the project will be based on your team’s grade and also on evidence of your meaningful participation in your team’s work on the project, including from these sources:

* The commit log of your GitHub repository.
* Your presence at and participation in your team’s project demo.

In particular, if your GitHub commit log shows that you did not make meaningful contributions to your team’s implementation of your app and/or you do not participate in your team’s demonstration of your app (without explicit prior approval by me), you will receive a lower grade on the project than your teammates. I may use other sources as evidence of your participation, as well.