

## CLASS PROJECT

The first challenge for this assignment is to find a group. If you have not already done so, please take the poll here: <https://goo.gl/R8VUEL>

Each student in the course will be enrolled into groups by the staff based on their responses to this poll.

### MINIMUM PROJECT REQUIREMENTS

At minimum, you must create a **CRUD** (**C**reate, **R**ead, **U**ppdate, **D**eleate) Web application that supports Web 2.0-style Ajax interactions. In other words, your Web app must:

- allow users to *create* accounts and add new persistent data
- *read* data from the server and present it on a webpage
- allow users to *update* their existing data
- allow users to *delete* their own data and accounts
- do all of the above without reloading the webpage (*Ajax*)

Piazza, Gmail, Twitter, SnapChat, and Facebook are all examples of such apps, though yours obviously won't need to be so sophisticated.

In contrast, a static personal website or even a dynamic [weather-lookup app](#) would not meet our requirements, since those are read-only. There is no way for a user to create an account or to update their personal data.

In the spirit of creative freedom, **you are free to use any programming languages, libraries, or frameworks that you like**. However, if you would like to get help from your TA, please ask them what tools they are most familiar with, so that you can get the best help from them. You will also need to agree with your group members about which technology stack is appropriate for you and your team. If none of the members of your group are very experienced in any particular web technologies, you are encouraged to use the tools and languages covered in the course lectures. You **will** be able to build an application that meets the requirements of the project using only these resources.

## CLASS PROJECT

Here are a few high-level rules:

- Your Web app must work properly when viewed in a recent version of Google Chrome without installing any plug-ins or extensions, and without logging into any third-party services such as Facebook, Twitter, or Google+. This means no Flash, no Silverlight, no Java, etc. It also means that your client-side code must be some form of HTML5/CSS/JavaScript. (But you can write code in other languages that compile into HTML5/CSS/JavaScript.)
- All of your project's code must be open-source and hosted in a public Github Repository.
- Your project must not contain offensive content or otherwise be in bad taste. We realize that “taste” is inherently subjective and context-dependent, but you must trust the staff's judgment. Our goal is not to stifle your creativity, but rather to ensure a safe, inclusive environment for your classmates. Think something that would be featured in a PG-rated movie, and that you would feel comfortable showing off at a big family reunion.

## GROUPS

You need to form a group of 3 or 4 students. Each group will submit a **single** submission for each project milestone and all members of the group will receive the same grade for that milestone. It's your responsibility to divide up the work fairly amongst members. The staff will not intervene in intra-group conflicts. Also, we will expect 4-person groups to attempt more ambitious projects than 3-person groups. Each group will be assigned a TA who advises and grades their project.

## MILESTONES

To provide some structure to this semester-long endeavor, we have provided five graded milestones. For each milestone, you must submit a PDF report documenting your progress. For legibility, use an 11-point or larger standard font with at least 1-inch margins. Also, when you include URL links to your app and code, *make sure they are clickable* so that the staff does not need to copy-and-paste URLs to visit them. Designate one group member to submit all project reports on Blackboard, and include all of your group members' names and UR email addresses on it. Also, pick a (tasteful) team or project name so that we can easily identify your group.

**CLASS PROJECT**

***When you submit your PDF file in Blackboard, put all of your group members' names in the filename so that the staff can easily identify the creator of each PDF file by glancing at the filename. e.g., if your group is John Smith and Jane Doe, then you can turn in a file named "Milestone 1 - John Smith and Jane Doe.pdf"***

(If your designated group member can't make the deadline for whatever reason, it's OK for someone else in the group to submit. But remember that deadlines are final; no extensions.)

---

***Project Milestone 1*** due at 11:59pm on Sep 30

By this milestone, you need to turn in a *one-page* project proposal. Your project's details will likely change throughout the semester, but it's important to have some starting point. Your proposal must address these questions:

- Who is the target audience for your Web application? [Stretch goal: Identify a target audience that none of your group members belong to.]
- What problem is it intended to solve?
- How will it meet the minimum project requirements?
- Why is your proposed Web application unique or creative beyond simply meeting the minimum requirements?

We understand that this milestone comes early in the semester before you've learned much in this course. But in many real-world projects, you must often start designing and prototyping before you're fully aware of the constraints.

The staff will be available in lab and office hours to answer your questions about project proposals and to work with you to refine your ideas.

Don't forget to put your team or project name on your proposal!

And if we think your proposal needs to be amended, then we will give you feedback by Oct. 7.

---

**CLASS PROJECT*****Project Milestone 2*** due at 11:59pm on Oct 14

By this milestone, you need to have implemented the “C” part of your CRUD app: ***allow users to create accounts*** and, optionally, to add some persistent data of their own such as user profile information or preferences.

You need to enable your TA to verify that they have successfully created an account. The best way to do so is to create a login page so that they can log in with their newly-created account. You will need a login page eventually. However, if you don't want to create a login page, you can also implement functionality such as making sure that ***duplicate accounts*** cannot be created with the same user name. Regardless of which way you choose, you need to make it easy for your TA to verify that you've actually created an account and stored information on the server. Otherwise, for all they know, your Web app could have done nothing and just said that it created an account!

You must implement your own account creation and persistent data storage layer rather than using a third-party login service such as Facebook, Twitter, or Google+ authentication. You can use libraries to help you implement this functionality, but you cannot simply use an off-the-shelf user login service.

Don't worry about security for this project ... just do the best job you reasonably can. For instance, you don't need to activate HTTPS since many low-cost Web hosting services don't support it.

Also, don't worry if your app looks bare or ugly at this point. Just focus on making the user account functionality work well.

The deliverable for this milestone is a ***PDF progress report*** of up to five pages in length. (You don't need to fill up all five pages.) Your progress report must address these issues:

- Include the public URL of your Web application that the staff can play with online
- Include the public URL of your project's source code (We recommend putting your code on a site such as [GitHub](#).)
- How did you implement your login system? What programming languages, libraries, and technologies did you use?
- Draw a diagram of the steps that a user takes to log in, and what happens to their data as it flows through your Web app across different server-side scripts, into and

**CLASS PROJECT**

out of databases, to the client's Web browser, etc. Then explain that diagram in detail in your report.

- What do you feel especially proud of so far about your project's implementation?
- What challenges or setbacks did you encounter so far when implementing this milestone? What was especially hard or frustrating?

In addition to grading your progress report, your TA will try out your Web app to make sure it works so far. They must be able to open up several different Web browsers and create independent user accounts in your app. In other words, your app cannot simply use IP address as a proxy for user identity, since someone on a single computer using multiple Web browsers should be able to simultaneously log in with several different accounts.

---

**CLASS PROJECT*****Project Milestone 3*** due at 11:59pm on Oct 28

By this milestone, you should implement enough functionality for a user to:

- Create an account (from Milestone 2)
- Log in and see an on-screen indicator that they are logged in
- Remain logged in even after closing their browser and re-opening it, assuming cookies are enabled (Hint: use [cookies](#) with `expires` and/or `max-age` tags)
- Not be allowed to log in again until they have logged out
- Log out
- Do *something* relevant to your app when they are logged in. This can be pretty much anything; we just want you to make some forward progress on the functionality of your project.

Again, the deliverable for this milestone is a ***PDF progress report*** of up to five pages in length. (You don't need to fill up all five pages.) Your progress report must address these issues:

- Include the public URL of your Web application that the staff can play with online
- Include the public URL of your project's source code (We recommend putting your code on a site such as [GitHub](#).)
- What new functionality in your app did you implement beyond just the login system?
- What challenges or setbacks did you encounter so far when implementing this milestone? What was especially hard or frustrating?

In addition to grading your progress report, your TA will try out your Web app to make sure it works so far. They must be able to open up several different Web browsers on a single computer, create independent user accounts, log in, remain logged in even after closing and re-opening their browser, do something in your app, and then log out. Your app should behave as a reasonable user would expect when they are logged in and logged out.

---

**CLASS PROJECT*****Project Milestone 4*** due at 11:59pm on Nov 18

By this milestone, you should implement some non-trivial Ajax functionality, which is the final part of your minimum project requirements. You should also make non-trivial progress toward implementing your actual project idea. There's about a month left until your final project is due, so this milestone is here to help pace your group.

Again, the deliverable for this milestone is a ***PDF progress report*** of up to five pages in length. (You don't need to fill up all five pages.) Your progress report must address these issues:

- Include the public URL of your Web application that the staff can play with online
- Include the public URL of your project's source code
- What Ajax functionality did you implement? What steps does your TA need to take to test your Ajax functionality?
- What non-trivial progress have you made toward your project's goals?
- What challenges or setbacks did you encounter so far when implementing this milestone? What was especially hard or frustrating?

In addition to grading your progress report, your TA will try out your Web app to make sure that the Ajax part works as advertised. The clearer you describe your Ajax functionality in your progress report, the easier it will be for them to test it.

---

**CLASS PROJECT*****Project Milestone 5*** due at 11:59pm on Dec 2

By this milestone, you should have completed most of the basic functionality of your project to the point where it's a working prototype that you would be happy to show off to friends and family. There's only two weeks left before your final project presentations, so you can spend that time making it work better.

Again, the deliverable for this milestone is a ***PDF progress report*** of up to five pages in length. (You don't need to fill up all five pages.) Your progress report must address these issues:

- Include the public URL of your project's source code
- Create a mini-tutorial that your TA can follow to use your app. Where should they start, and what are they supposed to do with it? What are the common use cases?
- What challenges or setbacks did you encounter so far when implementing this milestone? What was especially hard or frustrating?

In addition to grading your progress report, your TA will try out your Web app to make sure it works so far as described in your mini-tutorial. The better you make your tutorial, the easier it will be for them to test your app.

---



**CLASS PROJECT*****Project Presentation and OPTIONAL final report***

Your group will do a 5-minute presentation during the final week of class. I will send out a Google Doc to coordinate times. Here's how it will work:

- Your whole group comes to the front of the class.
- I load up your Web app on my laptop, which is connected to the projector. (You can also bring your laptop if you want to demo multi-user functionality.)
- You give a brief introduction to the motivation behind your project. e.g., What problem are you trying to solve? Who is your target audience?
- You step me through using your Web app on my laptop to demonstrate all of its main functionality.
- Time expires after 5 minutes, and the next group comes up!

As stated in the [course information page](#), the quality of this presentation will count for 3% of your grade. But don't worry about whether your UI looks polished, since we didn't emphasize UI or graphics design in this course. Just make sure your app works well when I try to use it in front of the class.

***Also, by the time you make your presentation, make sure that the final copy of your code has been uploaded to GitHub, since we need to grade that as well.***

Also, you can write an ***optional*** final project report, due **48 hours before your final oral exam** (no later than Sunday December 11th at 11:59pm for groups meeting with the staff on December 13th). The purpose of this report is to help us when grading the quality of your final project, which counts for 15% of your grade. The staff will grade the quality of your project by playing with your live Web app. The purpose of your optional report is to provide some context for your project and screenshots to guide us through its features. A well-written report will help us when grading, but it's by no means necessary if you think that your Web app is self-explanatory. In terms of length, your report should take no more than 5 minutes to read, so it should be short.

In terms of logistics, if you choose to write a final report, please upload the PDF to Blackboard at least 48 hours before your final meeting with the staff (***between 11:59pm on Monday Dec. 5th and 11:39pm Sunday, Dec 11th depending on your scheduled meeting***).

---

**CLASS PROJECT****FREQUENTLY ASKED QUESTIONS**

Q: How ambitious should I make my project?

A: This is a hard question to answer. But here is a baseline: ***A project that merely meets the minimum requirements will not receive full credit.*** This intuitively makes sense, since we cannot justify giving 100% for any assignment where someone just does the bare minimum. Instead, think of the minimum requirements as what is required to earn a passing grade. In this case, a bare minimum project would probably deserve some grade in the B to B- range. Don't panic, though, because you can easily make your project more ambitious in a variety of dimensions, including (but not limited to) a great user experience, real-time updates, sophisticated use of external APIs, ingeniously creative concepts, non-trivial algorithms, etc. [Edit for clarification: You don't necessarily *need* to have advanced features to get a good grade on the project; but you do need to demonstrate genuine effort that results in a realistic working Web app. In contrast, a "bare minimum project" is one where a group blatantly tried to implement just the bare minimum functionality to satisfy the requirements with no thought into making it into a non-trivial, substantive project that they would actually be proud to show off. Anyone who is serious about this class won't make such a project. Come to my office hours to ask for more details.]

Q: Can I get help from people outside of this course?

A: Yes, as long as you and your groupmates do the majority of the programming for the project, and as long as you write down who helped you with what parts as comments in your source code. Treat people outside of this course like "online resources" for help, so the same academic honesty rules apply. e.g., it's OK to get help from an online discussion forum (e.g., StackOverflow) take a small snippet of code, paste it into your project, and write a comment attributing the origin. But it's not OK to simply copy and paste an entire Web application from somewhere else, either from online or from a friend coding it for you.

Q: Can I re-use or extend parts of a project that I created before this class began?

A: No, you must start a new Web application from scratch. If your old project exposes some sort of Web API, then your CSC 210 project can certainly connect to your old project to gather data, much like how it can connect to any Web API such as Twitter's API. But you must create a standalone project solely for this class.

**CLASS PROJECT**

Q: What happens to my code after the class ends?

A: It's yours to keep. Make sure you save a backup copy on your own computer, though, since we cannot rely on Web hosting services to preserve your data indefinitely. ***We encourage you to think creatively about how your CSC 210 project can extend into a larger, more ambitious project in the future.***

Q: How heavily are we graded on aesthetics and graphical design?

A: Not at all. The purpose of this course is not to focus on visual aesthetics. However, your project will probably be more impressive if it looks good. But good aesthetics alone is no substitute for meeting the actual requirements of the project.

Q: Will there be extra credit offered for impressive projects?

A: Nope, no extra credit. The class project counts for 40% of your total grade, so even if you do perfectly on it, you still need to do well on the exams to get an A in this course. However, if you're on the borderline between two grades, having an impressive project will definitely weigh into the staff's decisions in our final grading meeting.

Q: Can I change my project idea after submitting the initial proposal?

A: Definitely! But please go over changes with your TA to make sure they approve and are aware of your changes. And do not make any changes that would make your project either out of scope (i.e., not an Ajax CRUD Web app) or contain any inappropriate content.

Q: What languages, libraries, or frameworks should I use to get started?

A: We've purposely designed the project to be open-ended, to simulate a real-world software development scenario. However, we do recognize that some of you would like to see a canonical example set of tools to get started, so we will put together a sample "hello world" tutorial video for you soon using one recommended set of tools. Stay tuned!