MINIMUM PROJECT REQUIREMENTS

At minimum, you must create a CRUD (C​reate, R​ead, U​pdate, D​elete) 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.

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.)

● 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?

​ 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.

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 and a scheduled

​ meeting with your TA ​of up to five pages in length. (You don't need to fill up all five

pages.)

● 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

UNIVERSITY OF ROCHESTER PRINCIPLES OF WEB APPLICATION

DEVELOPMENT (CSC 210)

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.