Project proposal

CSCI 110 Section 1

This is due **Thursday Oct 6**, 2016 at 11:59 pm on Canvas. Your project proposal will either be accepted, or rejected with revisions. If rejected, you will have 48 hours to make the revisions and re-submit.

Consider the feedback I've given you and pick one of your ideas to do for a project. Please write a project proposal with the following sections. The quality of your writing doesn't need to be perfect but make sure it's complete.

- Overview: a one sentence description of your project
- Detailed design: describe all of the parts of your project and how they interact with each other. Here are some things you might need to include:
    - Will your project need a frontend (a website)?
        - If it's a website, what are all of the different pages that will be on the website? What will each page do? Which pages will be interactive? How will the user navigate to pages?
        - A browser-based game like battleship will need to display the graphics using a website
    - Will your project need a backend (to store data about users)? In general, I would avoid choosing projects that need backends.
        - A project that sends emails at a later date will need a backend to store the emails and send them out when the user specifies.
        - A social network will need to store information about each user. Avoid having one of these if possible.
        - A game like battleship will NOT need a backend - the two players should take turns on the same computer
    - If you're writing a text-based game, how do you plan to parse user input? What are the possible commands the user can input? What will the plot of your game be? What are the possible endings? What will the different areas that the user can explore?
- Risks and challenges: identify at least three difficulties you will encounter and how you plan to address them
- Timeline: a week-by-week timeline of milestones to hit in your project before we do presentations at the end of November

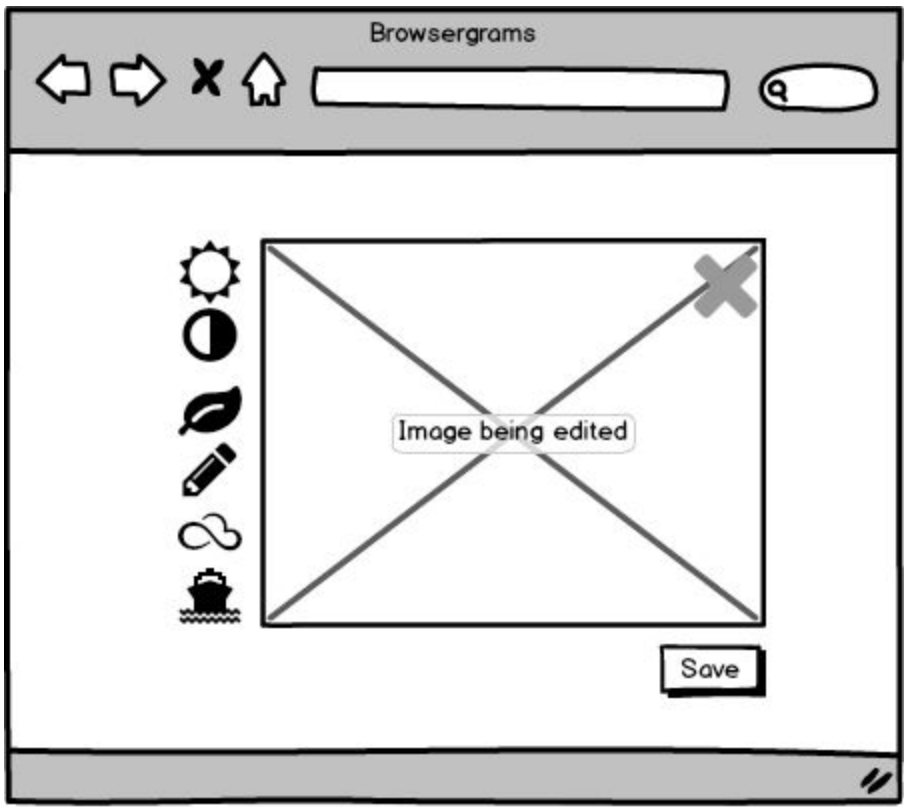
Here's an example proposal for a photo editing app.

Browsergrams

By Sheng Wu

Overview
Browsergrams is a browser-based photo editing tool that lets users select a photo, apply filters, and make simple tweaks like brightness and contrast.

Detailed design
Browsergrams will be a frontend-only website containing JavaScript-based photo editing tools. The user interface will consist of a photo editing area. Without an active photo, there will be a "select photo" button to let the user chose a photo to edit. When this button is clicked, a dialog will open that lets the user choose an image file.



After choosing the image file, the "select photo" button will disappear and the user will be able to use buttons to the left to tweak the photo. The following edits will be available:

- Increase or decrease brightness
- Increase or decrease contrast
- Apply one of six filters

During this process, if the user wants to discard his or her edits, there will be a translucent X in the upper right hand corner of the editing area. If this is clicked, a dialog box will appear to ask "are you sure?" If the user clicks yes, then the photo editing area is reset to how it was at the beginning with the "select photo" button visible. If no, then the dialog box will disappear and the user will be able to continue editing the photo.

There will be a button to let the user save the photo with his or her edits. When he or she clicks save, a dialog will appear that lets the user choose a location and filename for the saved photo.

Risks and challenges

- To have a good user experience, we don't want the user to accidentally overwrite an existing photo with our tool. So when the user tries to save a file with the same name as an existing file, our tool should warn him or her and ask for confirmation.
- I'm not very familiar with the HTML, CSS, and JavaScript needed to create an interactive browser tool. So I've started using Codeacademy's HTML and CSS course to learn more about these languages.
- It's difficult to understand some of the algorithms needed to edit photos e.g. to increase clarity (local contrast). To address this, I've already started reviewing papers and other tutorials to give myself a head start on learning how to do local contrast enhancement.

Timeline

- By Friday Oct 14: user interface mockups done in balsamiq
- By Friday Oct 21: user interface done in HTML and CSS; does not yet function
- By Friday Oct 28: JavaScript done to let user select a photo to edit; tool displays photo
- By Friday Nov 4: brightness and contrast adjustments implemented
- By Friday Nov 11: 6 filters implemented
- By Friday Nov 18: saving feature implemented
- By Friday Nov 25: presentation done and rehearsed