

CIS 311: Interactive Web Development

California State Polytechnic University, Pomona

Fall 2017

Instructor: Stefan Bund, MS
Office: Building 164 Room 3027
E-mail: [slbund at cpp.edu](mailto:slbund@cpp.edu) (replace at with @)
Phone: 949 433 7487 (no texts accepted)
Office Hours: Tuesday, Thursday 4 - 5:50 pm, office above

Guide to the Syllabus

CIS 311: Interactive Web Development	1
Fall 2017	1
Class Meetings and Office Hours, Holidays	2
Course Description	2
Learning Objectives	3
Prerequisites	3
Textbook and Software	3
Exams, Projects and Assignments	3
Grading	4
Class Communication and Getting Help [IMPORTANT]	5
Course Policies	6
Student responsibilities:	7
University Policies	7
Tentative Course Schedule and Graded Items	8
Term Grading	8
Term Final Project	9
Micro-Projects and Quizzes	10
Microproject Schedule	13
MP 1: Inputs	13
MP 2: Objects	13
MP 3: JSON	14
MP 4: Data	14
MP 5: Adding Responsiveness to Data	14

MP 6: CRUD	14
MP 7: Persistent Clients	14
MP 8: Integration (Extra Credit)	15
Syllabus Version History	15

Class Meetings and Office Hours, Holidays

Term Dates

September 21 until December 8.

Final exam dates for our 2 sections:

1-3 pm class final exam is on	Thursday December 7th from 11:30 am - 1:30 pm
6 - 8 pm class final exam is on	Thursday December 7th from 6pm - 10 pm

Class Meeting Times and Locations

class session , Bronco Code	meets in room...	between
311-02	98C 4-035	1-3 pm
311-03	98C 4-029	6 -8 pm

Holidays this term affecting meeting times

Thursday 23rd, Thanksgiving

Course Description

From our catalog:

The Design and development of web applications for business. Principles and applications of modern website design. Use of client-side scripting for website dynamics and interactivity. Development of server-side scripts for three-tier web applications. 4 units lectures/problem solving.

Instruction will focus upon core protocols supporting browser markup and web-based applications: html 5, CSS, javascript, server side scripting, cloud infrastructures and internet-facing data bases. The course entails a hands-on basis for learning, with deliverables in the form of projects and programming exercises. Students should use the course as a means to discover a potential major (application development) while gathering deep understanding about the security and infrastructure demands of employee and public-facing web applications. Additional discussion explores the corporate strategic impact of web development, and organizational/cultural change.

Learning Objectives

Students successfully completing this course should have acquired the ability to:

- Understand different types of web applications and how they work.
- Analyze and translate user needs and requirements into a software architectural model.
- Create wireframes and prototypes of user-centered and SEO-friendly web sites
- Create structure and content for web sites using standards-compliant HTML and HTML 5.
- Create styles for web sites using Cascading Style Sheets (CSS).
- Interact with users using client-side Javascript.
- Understand how ASP.NET works to enable dynamic contents and web applications.
- Control the flow of scripts, use conditional statements, loops, strings and arrays, write reusable modules in Visual Basic.
- Collect, validate and process information entered by users via web forms.
- Build dynamic web site that connect to a text file and database, insert, present and modify data in sophisticated ways.
- Understand the basics of web analytics and optimization, including the role of content, link building, social media marketing, usability, accessibility, other search engine and conversion optimization tactics.

Prerequisites

A minimum grade of C (2.0) in CIS 304, 305

Textbook and Software

Optional Textbooks:

Responsive Design text at our bookstore

Required Web Development Software for assignments, one of the following:

- Notepad (plain mode, default on all Windows)
- notepad ++ (freeware)
- Dreamweaver (student discount available at bookstore)
- Eclipse is also excellent, especially with the JSDT installed
- Net Beans or other full featured editor used in CIS 234, capable of making HTML pages

For the final project please acquire via DreamSpark:

- Microsoft Visual Studio 2017 (free and available on CIS dept website -> MSDNAA Download) with built-in MS SQL server.

Please acquire this tool by the final two weeks of the class, when we will produce a product using ASP.NET.

Exams, Projects and Assignments

Exams

One final exam during exam week. No midterm.

Quizzes

Each week one quiz is given on Thursday at the end of the class period. In cases where the student does not score above 50% on the quiz, the instructor will accept a programming project, posted as a one minute youtube video on a blog owned by the student. **This programming assignment is called a microproject.** For each microproject, the following rules apply:

- if you wish to submit a microproject, your quiz must be 50% or below.
- submit an email to Stefan, at the above email. In your email, include the URL of the blog post where you have posted your video for the microproject.
- the grade on your microproject replaces the quiz grade.
- the email with the URL to your blog is due on the **Tuesday of the following week by 6 pm.**
- the late policy below will apply to microprojects.

Quiz grades appear as "microproject" in your grade book on Blackboard. Seven of them will be given.

Note Well

The best way to prepare for a quiz is to do the microproject. Questions on the quiz will deal directly with coding, and the only way to get experience with coding is to do a project. If you have a difficult quiz (50% and below), you may work the following five days to deliver working code, and potentially improve your score. Additionally, by doing each microproject, you will conveniently add code which you may submit for credit on microproject 8, the extra credit project at the course end, thus further improving your score. *To underscore our point, if you code, we will ensure you are rewarded.*

Extra Credit

In cases where students wish to earn extra credit, they may do a microproject at the end of the term. The details of this eighth microproject are at the end of the Syllabus, under Microproject 8.

Make-up policy

There will be no make-up exams except for serious and compelling reasons that are substantiated with formal documents. For example, medical cases have to be substantiated with valid doctor or hospital note stating that the student is too ill to attend the exam. Athletic events are acceptable also, and will require documentation from the team, per our university norms.

Late assignments or projects

Late assignments or projects submitted within one week after due date and time will be penalized 50%. Late assignments or projects submitted more than one week late will not be accepted. If the student submits an assignment or project late, the student is responsible to send an email to notify the instructor upon submission; otherwise the assignment or project will not be graded.

Tutoring

For free tutoring on campus, contact the CIS department in the CBA Administration Building.

Grading

Grade	Percentage
A	93.00-100.00

A-	90.00-92.99
B+	87.00-89.99
B	83.00-86.99
B-	80.00-82.99
C+	77.00-79.99
C	73.00-76.99
C-	70.00-72.99
D+	67.00-69.99
D	63.00-66.99
D-	60.00-62.99
F	0-59.99

For a detailed breakdown of assignments, weight and their details, please see section 11, 'Tentative Course Schedule and Graded Items.'

Class Communication and Getting Help [IMPORTANT]

E-mail

All emails must be sent to the instructor with a Cal Poly email account, must be signed with the student's first and last name, and must have "CIS 311" in the subject line, or it may not be read or responded. Please consult the syllabus before sending emails, especially in the area of microprojects. **Messages sent through Blackboard will not be read.**

Microproject submission

For all microprojects, a youtube video is to be produced, detailing your work, lasting no more than one minute, with code displayed at all times during the video, using the Inspect Element feature of your browser. A link to the video is to be posted to a blog. In order to have your microproject graded, please submit a link to your blog via CPP email by Tuesday of the following week.

Coding Questions

In cases where you have a technical question, please post this to our Github at

- <https://github.com/stefanbund/311/issues>

You may also send the professor an email at `slbund at cpp dot edu`.

How to Win at Emails to Your Professor

Expect that your email is read on a mobile device, to return a reply to you very quickly and build value in your experience. Thus, compose your email carefully:

1. include screenshots of errors in the Javascript console, in your Browser
2. run Inspect Element on the running page, screen shot the code you are running
3. error codes given by your IDE, browser are really helpful!
4. include only a short number of sentences, and try to limit the email to one or two precise questions. This guarantees a quick reply.
5. The closer you ask questions to a deadline, the harder it is to get a fast reply. Schedule your work so you get help efficiently.

General Course Expectations on Help and Technical Assistance

- Your process should include reading the syllabus, reading the assignment, reviewing our learning materials, trying code, collecting results, **then** asking about running code. This will dramatically speed-up your experience. I would prefer not to answer your email with `please read syllabus page 19 subsection 18, paragraph 3, clause 18`.
- Please understand that very general questions are harder to answer, so emails with the subject line `PLEASE HELP (multiple emoji)`, or `I AM CLUELESS (emoji 1 and 2)` are harder to manage, and are likely to be delayed.
- Emails with precise subject lines like `DynamoDB Error in AWS, or Authentication Failure on Facebook` will be answered quickly throughout the day, 7 days a week.

Course materials

Lecture presentations, assignments, projects documents, classroom exercises and solutions, will be posted on Github. All graded assignments and projects will be visible in Blackboard Gradebook. Our github repository is:

- <https://github.com/stefanbund/311>

Refer to the **Syllabus** for all homework, lesson plans and other details critical for your success in 311. Refer to **Github repository** for source code and technical resources.

Subject to Change: This syllabus and class schedule are subject to change. If the student is absent from class, it is the student's responsibility to check on announcements made and make up the work while absent.

Course Policies

Classroom environment: The classroom is a special environment in which students and faculty come together to promote learning and growth. It is essential to this learning environment that respect for the rights of others seeking to learn, respect for the professionalism of the instructor, and the general goals of academic freedom are maintained. Student conduct which disrupts the learning process shall not be tolerated and may lead to disciplinary action and/or removal from class.

Using laptops, cellphones and other electronic devices:

- Using laptops during the class for anything other than this class, personal conversations, talking or texting on cell phones or other distracting behavior are prohibited.
- As a courtesy to all, please turn off all cell phones and pagers during class. If the student needs to be reached for family medical or significant work-related issues, the student must present evidence to the instructor before the class starts.
- Absolutely no cellphones or other electronic devices may be used during an exam or quiz.

Attendance:

- Arrive on time.
- Always whisper when the instructor is lecturing.
- If the student needs to leave early, the student must let the instructor know before the class starts, and choose a seat that minimizes disruption to the class when leaving.
- If the student has to miss the class, the student must send an email to let instructor know before class and explain the reason.
- If the student is sick and contagious, the student should not come to the class and risk getting others sick. Email the instructor before class to inform him of your decision.

Student responsibilities:

Each student is responsible for the successful completion and submission of all assignments and projects.

The instructor will not review your assignments or projects before grading for the entire class to ensure fairness. The instructor will, however, help you understand the expectations and clarify the requirements. Spot assessments will help you to outline questions and receive pre-due date feedback. **Whenever you have questions related to your studies**, please send an email to the faculty member, or attend his office hours, posted on page one, above.

The instructor will not debug assignments or projects for an individual student. The instructor will, however, help you gain knowledge and skills in analysis and design, problem solving, coding, testing and debugging, and answer specific questions about course topics. **It is always appropriate to discuss bugs and other complications with the instructor, however, the process of fixing problems rests with each student, exclusively.** Make sure you have spent significant time and effort in researching and working on your own before asking help. **To help you in this effort, many code examples are posted on our course GitHub, and to support your learning, each lecture is posted on our youtube channel. Be sure to watch each youtube instructional item at least once!**

Team Work

No team work will take place during 311 this term.

University Policies

Students with Disabilities: Upon identifying themselves to the instructor and the university, students with disabilities will receive reasonable accommodation for learning and evaluation. For more information, contact Services to Students with Disabilities at <http://dsa.csupomona.edu/drc/>.

Academic Integrity: Students should understand or seek clarification about expectations for academic integrity in this course (including no cheating, plagiarism, or inappropriate collaboration); neither give nor receive unauthorized aid on examinations or other course work that is used by the instructor as the basis of grading; take responsibility to monitor academic dishonesty in any form and to report it to the instructor or other appropriate official for action.

Cheating and Plagiarism: Cheating is the actual or attempted practice of fraudulent or deceptive acts for the purpose of improving one's grade or obtaining course credit; such acts also include assisting another student to do so. Typically, such acts occur in relation to examinations. However, it is the intent of this definition that the term 'cheating' not be limited to examination situations only, but that it include any and all actions by a student that are intended to gain an unearned academic advantage by fraudulent or deceptive means. Plagiarism is a specific form of cheating which consists of the misuse of the published and/or unpublished works of others by misrepresenting the material (i.e., their intellectual property) so used as one's own work. Penalties for cheating and plagiarism range from a 0 or F on a particular assignment, through an F for the course, to expulsion from the university.

In assignments where code is required, students who allow their work to be copied will receive a zero on the first episode of cheating. Forewith, meetings with the university will follow, to escalate, potentially, to expulsion. All code you submit must be coded by you without copying from another source.

Computing Resources: At Cal Poly Pomona, computers and communications links to remote resources are recognized as being integral to the education and research experience. Every student is required to have his/her own laptop/computer or have other access to a computer with all the recommended software for this course. Find out more about how to access to the university's information resources from Information Technology Services.

Copyright Policy: Copyright laws and fair use policies protect the rights of those who have produced the material. The copy in this course has been provided for private study, scholarship, or research. Other uses may require permission from the copyright holder. The user of this work is responsible for adhering to copyright law of the U.S. (Title 17, U.S. Code). A full description of Cal Poly Pomona's copyright policy is included in the University's Intellectual Property policy. The course web site contains material protected by copyrights held by the instructor, other individuals or institutions. Such material is used for educational purposes in accord with copyright law and/or with permission given by the owners of the original material. Students may download one copy of the materials on any single computer for non-commercial, personal, or educational purposes only, provided that (1) do not modify it, (2) use it only for the duration of this course, and (3) include both this notice and any copyright notice originally included with the material. Beyond this use, no material from the course web site may be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way without the permission of the original copyright holder. The instructor assumes no responsibility for individuals who improperly use copyrighted material placed on the web site.

Tentative Course Schedule and Graded Items

Term Grading

The Term will contain 7 micro-projects, each worth 6% or 5%, and due at the close of each unit in the Lesson Plan. A Final term project will take place worth 20%.

Each week a quiz is given, based upon the content of the prior week. A total of 9 days of programming are allotted for each quiz. Each quiz is based upon the coding work assigned for each microproject, affording students the ability to account for their skills in either coding or during an administered exam. Code will be covered on most if not each question, each week. Therefore, performing the micro project is the best preparation for each quiz.

In cases where students receive a 50% or below on a quiz, the Instructor will view their blog, and apply the grading rubric, below to their micro project grade.

Weekly Quizzes	Unit	Weight (%)	Due
quizzes occur on Thursday each full week at the end of class	1	6	10/5
	2	6	10/12
micorprojects are due the following Tuesday of each week via email, midnight	3	6	10/19
	4	6	10/26
	5	6	11/2
	6	6	11/9
	7	6	11/16
extra credit microproject	8 *	6	12/7 by email
total microproject points		42	sum of mp / quiz 1- 7
total Final Project points		8	
Term value of final project		16%	
Total points		50	

* MP 8 is optional, and can be submitted to replace up to six points on prior MPs.

Term Final Project

Interactive Web Product. The final project compiles each micro-project into a cohesive interactive web product. Compile a product video on a promotional page of the product's website. ASP.NET is to be used. Students will integrate resources from Microsoft in creating their individual project. Resources for the project are located at this address,

- <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/index>

This link will also be posted to the course GitHub page. Students should plan to devote 15 hours to the project, minimum. Grading is based on the percentage of the tutorial which you have finished and documented.

How to Submit Your Project for Grading

1. **Public blog.** Students will demonstrate their project on their term blog, where their micro projects are posted.
2. **Submission deliverable.**

1. Students will copy/paste the code they used to produce the site in **a single .pdf document**.
2. In this document, screen shots of each working UI are to be inserted, to accompany their code. The document should show the working product with the code they used to produce it.
3. A dropbox will be sent to students on the due date, via CPP email, to assist with their submission.
4. Only one submission per student may take place, and must be submitted by the closing time of the final exam, as posted on the school's final exam schedule (and on our GitHub, for convenience).
5. Multiple documents, unorganized images, or departure from rule 1 results in a zero.
6. Your submission's filename must contain your full name, and the words, '311 ASP project'.

Grading Rubric, for ASP.NET projects:

- A. Project completes an ASP.NET project with several key elements. Elements include
 1. integration of SQL Server
 2. deployment of a web container (IIS) on a personal device (laptop / PC)
 3. integration of SQL queries producing HTML markup
 4. integration of a responsive CSS in a uniform fashion on all pages,
 5. evidence of input / update and delete on database contents
 6. a basic security mechanism.
 7. use of URL parameters
 8. use of the ASP.NET model / view paradigm
- B. Project is missing 10% of the A criteria.
- C. Project is missing 20% of the A criteria.
- D. Project is missing 30% of the A criteria.
- E. Project is missing 40% of the A criteria.

Micro-Projects and Quizzes

Each week, a quiz is given, and is valued the same as a microproject. Students should prepare the micro project for the above due dates, then take the weekly quiz. You receive credit by submitting a blog post of your work, where a one-minute youtube video is present.

If a student receives a 50% or below on the weekly quiz, the Instructor will review their blog, which should contain a video of the microproject, due for that week. This way, if a student codes well but does not test well, their code will be reviewed. Also, in cases where students do not complete the microproject, their quiz grade will suffice.

For each unit in the schedule, below, a small project is made by the student to practice and implement the technologies taught in class. Helpful source code will be presented and made available online on our course GitHub. Students will be expected to show competency implementing the code demonstrated for the unit in a working micro-project.

Deliverable. The coding practices taught in the unit should be present on each page in some functional, explainable way. The student should show that they have learned how to use the skills, then apply them in a very simple demonstration. For each project, an outside observer

should see them moving data within their page, then display it in HTML markup in a very simple business process.

Each student will use a public blogging service to describe their work in a brief video demonstration (youtube is adequate for this purpose). The assignment is a blog post, completed by the due date, which links to their functioning micro-project video. **Be sure, in each video, to show the code you used, using the Inspect Element feature of your browser, not in a separate IDE.**

Best Practices

For your micro-project, understand that A projects are small portfolio pieces, adequate to show employers a wide range of web development skills.

Due dates

Micro-projects must be submitted by end of the next unit. For example, class ends on a Thursday, then the micro-project will be at midnight. Your micro-project must be online and accessible for full credit.

Grading Rubric (for each weekly micro project)

A

Student accomplishes the week's micro project goals, per the table "Weekly micro project goals," on the following page.

B

Student accomplishes 85% of the coding goal

C

Student accomplishes 75% of the coding goal

D

Student accomplishes 65% of the coding goal

E

Student accomplishes 55% of the coding goal or below

Lesson Plan, Lecture Topics and Session Objectives

Months	Class sessions	Lecture Topic	Unit # on github
sept	21, 26	Basic Javascript. Basic user interface interactivity, HTML elements	1
	26, 28	Responsive project element demonstration and discussion. Navigation, URLs and basic UI for HTML 5, using bootstrap CSS.	quiz

Months	Class sessions	Lecture Topic	Unit # on github
oct	3	Javascript objects. Manipulating member variables and functions in javascript core language	2
	5	Responsive project element demonstration and discussion. Alerts, labeling, wells and other means of displaying data to users.	quiz
	10	Handling Data with Javascript Objects. Javascript Object Notation (JSON), java objects, arrays, array traversal, functions, forEach iteration, Document Object Model (DOM) and javascript, text nodes	3
	12	Responsive project element demonstration and discussion. List elements, tabular outputs, menus with dynamic content.	quiz
	17	Cloud Security and Querying NoSql Databases. OAUTH, Amazon Web Services SDK, Identity and Access Management (IAM), DynamoDB and DocumentClient, event-driven javascript, Web Identity Federation, Third Party Authentication Systems and Cloud Computing	4
	19	Responsive project element demonstration and discussion. The HTML div and CSS styles. CSS grids, page layout and row-based container elements in CSS	quiz
	24	Integrating services into client-side javascript. HTML Elements integrated with cloud computing APIs	
	26	Responsive project element demonstration and discussion. Continued work in developing means to display interactive output from data, in rows	quiz
	31	Basic Get, Put, Update, Delete on NoSql cloud environments. Transferring cloud JSON into HTML elements	5
nov	2	Responsive project element demonstration and discussion. Triggering navigational elements via javascript and the window object in the HTML DOM	quiz
	7	Persisting client-side data with cookies and adding interactivity to interactions on the web. URL parameters in the DOM	6
	9	Responsive project element demonstration and discussion. Integration of prior weeks' discussion into cookies. Leveraging data transfer via URL parameters	quiz
	14	HTML FILE protocols and uploading images to the cloud	7
	16	Responsive project element demonstration and discussion. The responsive image, img tag, and the row of divs. Building interactivity to links which involve data transferred over URL parameters, including images.	quiz

Months	Class sessions	Lecture Topic	Unit # on github
	21	Final Project work time. Get personal consultation, coaching and time to discuss project.	8
	23	Final Project work time. Get personal consultation, coaching and time to discuss project.	
	28		
	30		
dec	final exam	final exam, final project submission due date	

Microproject Schedule

MP 1: Inputs

Decide upon a general project theme, related to a real world subject, company, or concept. Create four pages then connect them with a Bootstrap menu. Create a **registration** page which accepts an email. Create a **username** page which asks for a username and password, as well as their full name. Validate that these fields contain inputs once they press a Submit button on both pages. Be sure you have coded each unit 1 example source on our github.

Resources

1. MDN, using javascript .test, https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp/test
2. Stack, input validation example, and links <https://stackoverflow.com/questions/13748770/email-validation-with-javascript?noredirect=1&lq=1>
3. our GitHub, unit 1 source code, <https://github.com/stefanbund/311/tree/master/1%20basic%20javascript%2C%20basic%20bootstrap%20navigation>
4. Bootstrap nav bars, <http://getbootstrap.com/getting-started/#examples-navbars>

MP 2: Objects

Construct a javascript object to contain the email, on the Registration page. Construct a javascript object to contain the username and password on the username page. Apply input validation to both username and registration pages, ensuring the inputs you desire are submitted, when the Submit is used. When bad inputs are put in, please throw a Bootstrap Alert. Assume that you must have MP 1, including a navigation menu system with Bootstrap. Add a home page which announces your project and contains several columns which describe what the site is about. Use the Bootstrap Jumbotron on the home page.

Resources

1. Bootstrap Jumbotron, <http://getbootstrap.com/docs/4.0/examples/jumbotron/>
2. unit 2 GH on objects, <https://github.com/stefanbund/311/tree/master/2%20javascript%20objects>
3. doing Bootstrap alerts, <https://github.com/stefanbund/311/blob/master/2%20javascript%20objects/doing%20alerts%20in%20bootstrap.png>

MP 3: JSON

Create a new page where the user gets access to data. Craft a JSON object which contains sample data and organize it on the page using Bootstrap columns. This page should link within your Bootstrap navigation system, available on each page. Make a .js file that allows you to update the menu in one file, then use it across all of your pages.

Resources

1. unit 3 code, <https://github.com/stefanbund/311/tree/master/3%20handling%20data%20with%20javascript%20objects>
2. <https://stackoverflow.com/questions/44502243/single-navigation-bar-across-website?noredirect=1&lq=1>
3. Bootstrap grids, <http://getbootstrap.com/examples/grid/>

MP 4: Data

For the page you made in MP 3, rewrite the page so that the JSON disappears. Replace the JSON object with data you get from your database in the cloud (AWS). Implement an authentication system using OAUTH which enables you to grab new data from the cloud.

MP 5: Adding Responsiveness to Data

Begin with the query results you got for each query in MP 4. For each div you create, with query results, add an HTML href, or link element. This element should contain results from your query, to construct a valid URL. Hence, you should create a column in your DynamoDB which contains a string sufficient to create a hyperlink. Then, when this column data loads within the page, a link is constructed, concatenating other URL strings. The link should be underlined. Create the link in a responsive way.

MP 6: CRUD

Use your new skills with CRUD. Building upon our samples, develop three pages which maintain your authenticated connection to AWS. On these three pages enable your first page to create two values of data in your database. On the second page, enable those values to be updated, after a get. On the third page, enable for your data to be gotten, then deleted. Assume that you hard-code the itemID to fetch the appropriate item from the table in your javascript on pages two and three of your mp.

MP 7: Persistent Clients

Return to micro project 6. In the page that does a 'put' on your database, store the information the user inputs to a cookie. Then, in the 'update' page, change the information in the cookie. Demonstrate the use of URL parameters by sending the user to a new page, using `window.location` after the update is completed by AWS. When you send the user to the new page, after the update, practice using URL parameters by loading the URL in `window.location` with the variables you just saved to the cookie. Have the console show the URL parameters once you arrive in the new page.

MP 8: Integration (Extra Credit)

Extra Credit Micro-project. Integrate each prior micro project to produce a four page site.

- On this site, you should offer a login page, directing the user to Facebook.
- After authenticating with Facebook, direct the user to a registration page, where they fill in information relating to their complete name, email, and uploading a picture.
- When this information is submitted to the database, redirect them to a home page.
- Enable a drop-down menu to contain a link to a profile page.
- The profile should contain the latest inputs to their profile including their photograph, name and email.
- A button will be placed on the profile page enabling them to edit their profile, which should update their profile, after getting its latest settings.
- They should also be able to upload a new profile image.
- An additional 'delete account' button will allow for the profile to be deleted in the database after an alert window asks them to confirm their intention.

Syllabus Version History

1.0 - 1.7. Added microprojects on a weekly basis, mp 1 through 7.

1.8. Added Extra credit microproject. Updated actual point values for assignments (adjusting each mp from 10 to 6 points except for ups 2 and 3, which stand at 5 points. Total course points adjusted from 90 to 48, maintaining the possibility for extra credit (6 points). Clarified grading for quizzes and microprojects.

1.9 Adjusted language to specify quiz, mp and grading rubric for summer '17 courses.

2.0 Added language on pages 2 and 3 to specify how to turn in microprojects via email, and an updated due date to do so