Client Side Scripting for the Web - Syllabus
Upper Level IST & CS Elective, Instructor: Sally Kyvernitis

IST Elective: **CIS 3344 Advanced Web Application Design and Scripting**, pre-reqs: CIS 2229 and CIS 3309
CS Elective: **CIS 4350 Client Side Scripting for the Web**, pre-reqs: CIS 2107 and CIS 2168

Today's web applications are relying more on JavaScript (client side) code and less on server side code (e.g., php, jsp, dotNet). These web pages use AJAX, a technique that involves making JavaScript calls to server side web APIs that can access a database. AJAX is asynchronous which means it works in the background, without locking up the UI and this can create a more responsive experience for users. Although we will write our own server side Web APIs (using java/JSP), this course focuses on client side scripting, including AngularJS which is an open-source, “single-page-application” web application framework that was written and is maintained by Google and others. The Angular framework provides a client side Model-View-Controller that makes it easy to develop and test “single-page-applications”.

A “single page-application” accomplishes its work by (a) calling web APIs (using JavaScript) and (b) modifying the HTML Document Object Model (DOM). Angular code can be written by simply including a JavaScript reference in the web page, just as you would reference jQuery. Here is a very simple example to demonstrate how Angular works. This example shows JSON data (as might be returned by a Web API), a sample HTML page with Angular/javascript code (only 30 lines long), and a demonstration of how the page runs:

```
http://curran.github.io/screencasts/introToAngular/exampleViewer/#/26
```

According to Libscore (a JavaScript analytics service), Angular is used by the websites of NBC, Walgreens, Intel, Sprint, ABC News, (and many others) - and its popularity is increasing. Client side frameworks are the future of web application development. We will be using Angular 1, but we will investigate Angular 2 (which is in beta testing) and ReAct a framework which appears to have greatly influenced the Angular 2 redesign.

Since rapid change is the "name of the game" in web development, students need to be able to keep themselves updated by learning from the internet. In addition to writing (client side) Angular web pages and (server side) Web APIs, each student will learn about then write a tutorial on some aspect that is relevant to the topics of this course.

**Course Topics**

In this course, we will “work our way up” to AngularJS by first learning the fundamentals upon which Angular is built:

- HTML and CSS (including CSS positioning)
- Responsive Web Design (makes a web page usable on various screen sizes, e.g., cell phone, computer screen)
- JavaScript (code that runs in the browser, can access the HTML Document Object Model, and can invoke Ajax calls to server side Web APIs)
- jQuery, a library of JavaScript functions that manages inconsistencies that exist between various browsers, making JavaScript more robust and reliable for client side functionality (and less lines of code for you to write)
- Ajax, a technique whereby JavaScript makes asynchronous calls to Web APIs
  - XML and JSON, two formats for data that is sent over the internet.
- Setting up a Database: using a data modeling tool to create and populate a (mySql) database, basic SQL statements (Select, Insert, Update, Delete)
- Creation of Server-Side Web APIs to respond to client side requests to get and/or update data
- Angular Introduction, followed by an Angular project (several weeks allocated)
Course Format

- **Weekly Lab Assignments.** Weekly homework will be highly related to (reinforcing) lecture topics
- **Use of Lab Time.** Every week in lab there will either be a graded **lab quiz** or **lab activity**.
  - Short quizzes provide a quick feedback mechanism for you (to see how you are doing) and me (to learn which concepts I need to reinforce). Quizzes are usually closed book, closed computer (paper).
  - Lab activities are short assignments that must be started and completed in lab, then shown to the Lab Assistant for grading.
  - No Make Up Quizzes. If a student misses a lab quiz (or arrives after another student has submitted their quiz and left the room), there is no make-up (for any reason including illness and/or other unavoidable problems). However, every student is allowed to drop their lowest quiz grade.
  - No Make Up Lab Activities. If a student misses a lab activity (or arrives too late to complete the work) for any reason, they get a grade based on what they were able to complete. Every student is allowed to drop their lowest lab activity grade.
  - Exceptions: if a student has a long term documented illness, we might be able to make an accommodation. Ask your instructor if you encounter such a problem.

- **Project.** The culmination of all the labs will be the student’s project. While the weekly lab grading may just focus on functional testing, the project will include a code review (done by me) of all the source code, so it's best to keep your code clean and well designed as you go along. The project requires that students have all of their web code functioning and incorporated into a single NetBeans project, with each lab working properly (regression tested).
- **Tutorial.** Each student will write and publish a tutorial on an approved topic. Selected students will present their tutorials to the class towards the end of the semester.
- **Comprehensive Test.** There will be a comprehensive test at the end of the semester (no midterm, no final).

If a student already has extensive background in any topic that is covered, they may substitute other work. These students should discuss their background with the instructor at the beginning of the course.

Tentative Lab / Tutorial Schedule

3. JavaScript Intro
4. Advanced JavaScript
5. Your Tutorial
6. Database Set Up
7. Web API (write a back end Web API using java/JSP, provides data from your database table)
8. Angular Client Side Manipulation.
9. Angular Triple Lab (users can log on, log off; they can insert, edit, and delete from your db table).
Labs, Tutorial, and Project

- Labs are introduced during lecture. Although there may be some additional instruction at the beginning of lab, the labs are mainly self-directed with help available.

- The most current list of labs (with updated due dates) is listed at the top of blackboard. Please check this single place to learn what lab we are working on and when it is due.
  - Late penalties are based on time of submission of your zip file into blackboard. You are allowed to submit a lab up to 24 hours late with no penalty (as a courtesy, to allow you time to check your coding style and publish before submitting your zip file).
  - If there is ever any question about a lab grade, we go by the code that was submitted into blackboard.
  - If you miss the lab deadline, you can still submit a lab up to 1 week late (also with the 24 hour courtesy) – with a 20% late penalty. Lab homework is not accepted after that, but you still have to complete the work to avoid additional penalties when you submit your project.

- The lab requirements can be found in blackboard and in the 3344 labs page, but DO NOT WORK AHEAD since labs may be changed right up to the lecture when they are assigned. I typically have a "We Are Here" entry in blackboard which I keep moving down as elements are updated for the current semester.
  - If you start working on your lab homework the day before it is due, you risk missing deadlines, doing poorly in the weekly quizzes, and falling farther and farther behind. It is REALLY IMPORTANT that you start early on your labs and ask for help if whenever you are unable to make effective progress.
  - It is also so important to know how to debug, in each of the languages that we use (HTML, CSS, javaScript, jQuery, JSP, java, and SQL). I try to include important information about how to debug for each lab.
  - At the end of each lab, there are instructions for lab submission which usually have you publish your code then attach a zip file of all your source code right back into the blackboard assignment (where the instructions were).

- Tutorial. One great way to learn is to write a tutorial for others. Each student will select an appropriate topic (that is related to client side web development), get that topic approved, and then publish a tutorial on that topic. Towards the end of the semester, selected students will present their tutorials to the class.

- Your project is your entire web application, the culmination of all your labs (even those labs that you may have missed or "messed up"). Before submitting your project, remove all unused code, perform regression testing, and check that you have met the functional requirements and design specifications of the lab assignments.

- The CIS department computer labs are NOT open 24/7. Learn the lab hours and adjust your schedule accordingly. If you want to work from home, set up your own development environment. I have tutorials and instructions that can help you with this. Those who promptly set up their own web development environment do much better in this course than those who do not.
Tentative Grade Weights

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labs (Homework)</td>
<td>22%</td>
</tr>
<tr>
<td>Lab Activities (about 5-6, must complete in lab, no make ups, lowest grade dropped)</td>
<td>8%</td>
</tr>
<tr>
<td>Lab Quizzes (about 5-6, in lab, no make ups, lowest quiz grade dropped)</td>
<td>20%</td>
</tr>
<tr>
<td>Tutorial (that you write and publish and possibly present)</td>
<td>15%</td>
</tr>
<tr>
<td>Final Web Site (Project, culmination of all Labs)</td>
<td>15%</td>
</tr>
<tr>
<td>Comprehensive Test (given last day of class)</td>
<td>20%</td>
</tr>
</tbody>
</table>

Total: 100%

Grade Scale

- 93-100: A
- 83-86: B
- 73-76: C
- 63-66: D
- 90-92: A-
- 80-82: B-
- 70-72: C-
- 60-62: D-
- 87-89: B+
- 77-79: C+
- 67-69: D+
- 0-60: F

Early grade estimates:
- For tutorial grade (15%), the lab grade is the best estimate.
- For project grade (15%), the lab grade is the best estimate.
- So, for early grade estimates, the lab grade (actually 22%) is weighted as 52% (22+15+15) until the tutorial and/or project grades become available.
- Lab activities are weighted as they will be in the final grade: 8%
- For comprehensive test (20%), the quiz grade (20%) is the best estimate
- So, for early grade estimates, the quiz grade (actually 20%) is weighted as 40% (20+20) until the comprehensive test grade comes in.
- Also, I do not drop the lowest quiz grade or lab activity grade until later in the semester.

Miscellaneous

- **Attendance:** If you must miss lecture or lab, check Blackboard to see what material was presented and ask your classmates about anything else that may have been discussed.
- **Communication:** Please contact me as soon as possible if you think you are running into difficulties. Ask me, or your lab instructor, or another student for help AS SOON AS POSSIBLE.
- **Disability Disclosure:** Any student who has a need for accommodation based on the impact of a disability should contact me privately to discuss the specific situation as soon as possible. Student must provide me with a note from the office of Disability Resources and Services (100 Ritter Annex, 215-204-1280).
- **Academic Honesty and Ethics:** Temple University and I expect you to observe the highest ethical standards. When working in the lab or on your project, you may consult others, but the work you submit must be your own. Never share your answers with others. Never accept answers from others. Unless otherwise directed, all quizzes are closed book, closed computer. All violations of academic honesty will be handled according to university policy.