Client Side Scripting for the Web Syllabus (Instructor: Sally Kyvernitis)

IST Required Course: **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 client side code (JavaScript) and less on server side code. Using a technique called AJAX, web pages can invoke HTTP requests to server side web APIs that can access and modify data stores. Because AJAX is asynchronous, it works in the background, without locking up the UI, thus creating a more responsive experience for users. In this course, we will write our own server side Web APIs (using java/JSP), but the focus of this course is on client side scripting, including **AngularJS**, an open-source, Model-View-Controller, “single-page-application” web application framework written by Google.

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://cis-linux2.temple.edu/~sallyk/tutorials_AngularJS/angular_curan_kelleher/26_search_features.html](http://cis-linux2.temple.edu/~sallyk/tutorials_AngularJS/angular_curan_kelleher/26_search_features.html)

Since web development continues to undergo rapid change, this course teaches students how to keep themselves updated by learning the newest developments from the internet. Each student will learn about then write a tutorial on some aspect of client side web development.

Course Topics

Since this course assumes no web development (or design) experience, we cover all of the following topics:

- 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)
- 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
- Ajax, a technique whereby JavaScript makes asynchronous calls to Web APIs
  - XML and JSON, two formats for sending data over the internet.
- Angular Introduction, followed by an Angular project (several weeks allocated)

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.
Course Format

- **Tests.** There will be 3 tests with no final exam. The tests could be administered in lecture or lab (hands on). There are no make-up tests. If you miss a test, discuss your situation with the instructor - an accommodation might be possible, depending on the circumstances.

- **Weekly Programming Assignments (homework).**
  - Weekly programming assignments are highly related to lecture topics. In order to receive a grade for the homework, you have to publish the required functionality and submit the source code (a zip file of your whole web site) into the Canvas assignment.
  - It is very important to know **how to debug** in each of the languages that we use (HTML, CSS, JavaScript, jQuery, JSP, java, and SQL). Be sure to pay close attention to important information about how to debug for each lab.
  - Homework due dates are defined (in one spot only) in the schedule at the top of Canvas. You can submit programming homework up to 24 hours late with no penalty (as a courtesy, to allow you time to publish, test, and submit your code). If you miss the deadline, you can still submit up to 1 week late (also with the 24 hour courtesy) – with a **20% late penalty**. Programming homework is not accepted after that, but you still have to complete the work by the project due date (or your project grade will suffer).
  - Programming homework grades are typically based on functional testing that occurs within 1-2 weeks of the lab’s due date. Remember to regression test each week so that you do not introduce errors into previously published code (or else you will get points deducted). If there is ever any question about a lab grade, we re-evaluate based on the source code that was submitted into Canvas (but you can still get deductions if a future homework causes the previous homework to stop working). Late penalties are based on the date of the Canvas submission.

- **Use of Lab Time.** Every week in lab there should either be a graded **test** or **lab activity**.
  - Lab activities are short exercises that must be started and completed in lab, then shown to the Lab Assistant for grading. If you miss a lab activity, there is no make-up. These exercises are not heavily weighted, but they are designed to help you understand how to start your homework programming assignment for that week. If you miss lab activities, it is likely that you will not know what to do for your homework programming assignment.

- **Project.** Your project is the culmination of all your labs. Your project grade is based on functional testing plus a code review of all the source code. So be sure to keep your code (each week) bug free, well organized, and with self-documenting names. EVERY WEEK, your NetBeans Project shall grow – you do not create a new project each week.

- **Tutorial.** Because it is imperative that students are able to learn web development concepts independently from the internet, 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.

### Tentative Programming Homework / Tutorial Schedule

3. JavaScript Intro
4. Advanced JavaScript
5. Your Tutorial
6. Database (creation, data population)
7. Web APIs (server side Web APIs are written using java/JSP, providing access to your database)
8. Angular Client Side Manipulation.
9. Angular Triple Lab (3 submissions: insert, edit, and delete from one of your db tables, logon/logoff).
Tentative Grade Weights

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly Lab Activities</td>
<td>10%</td>
</tr>
<tr>
<td>Tutorial (that you write)</td>
<td>15%</td>
</tr>
<tr>
<td>Three Comprehensive Tests (no final exam)</td>
<td>75%</td>
</tr>
<tr>
<td>Weekly Programming Homework Assignments</td>
<td>-25%</td>
</tr>
<tr>
<td>Project (culmination of all labs: regression tested, code review)</td>
<td>-25%</td>
</tr>
</tbody>
</table>

Your grade is based on your lab activities, tutorial (written by you), and your test scores. However, you lose points for NOT doing homework assignments/project.

Grade Scale

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93-100:</td>
</tr>
<tr>
<td>A-</td>
<td>90-92:</td>
</tr>
<tr>
<td>B</td>
<td>83-86:</td>
</tr>
<tr>
<td>B-</td>
<td>80-82:</td>
</tr>
<tr>
<td>C</td>
<td>73-76:</td>
</tr>
<tr>
<td>C-</td>
<td>70-72:</td>
</tr>
<tr>
<td>D</td>
<td>63-66:</td>
</tr>
<tr>
<td>D-</td>
<td>60-62:</td>
</tr>
<tr>
<td>D+</td>
<td>57-59:</td>
</tr>
<tr>
<td>D</td>
<td>50-52:</td>
</tr>
<tr>
<td>F</td>
<td>0-50:</td>
</tr>
</tbody>
</table>

Early grade estimates:
- For Tutorial grade and Project grade, the Weekly Programming Assignment grade is the best estimate (assuming that you are doing your own work).

Miscellaneous

- 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 as explained in my lab documentation. Those who promptly set up their own web development environment do much better in this course than those who do not.
- **Attendance:** If you must miss lecture or lab, check Canvas 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 on your homework assignments, your tutorial, or your project, you may consult with others for help, but the work you submit must be your own. Never share your code with others. Never accept code from others. Unless otherwise directed, all tests are closed book, closed computer. All violations of academic honesty will be handled according to university policy.