Course Description

Web applications used to rely almost entirely on server side processing, where each button click resulted in a whole new web page being sent from the web server to the browser. Today’s web applications focus more on JavaScript (client side code) to asynchronously invoke (server side) Web APIs and just update part of the user interface. This technique, called AJAX, creates a better and more interactive user experience. Although we will write some Web APIs (using Java/JSP code) to provide database access, this course focuses on the browser (JavaScript) code that initiates then processes the Web API data.

Since web development is constantly changing, this course focuses on web development fundamentals and encourages students to use the internet to keep themselves updated. Each student will learn about then write a tutorial on some aspect of client side web development.

There are always Web development frameworks designed to help you write less code. The problem is that these frameworks come and go. Rather than spend a lot of time on any one framework, we will (at the end of the course) get some exposure to React, a popular JavaScript library that helps you develop Single Page Applications (a one page application that uses JavaScript and AJAX extensively, to update the content area).

Course Topics

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

- HTML (web page content and structure)
- CSS (look and feel of web page content)
- JavaScript (code that runs in the browser, can access elements on the page, and can invoke Ajax calls to server side Web APIs)
- 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.
- Exposure to a React, a Single Page Application framework.

Prerequisites

- Grade of C- or better in CIS 2109 Database Management Systems
- Grade of C- or better in CIS 2168 Data Structures

The only assumption is that students have knowledge and skills obtained in the pre-requisite courses listed just above. All other topics will be introduced as new material, even though some students may already have had some exposure. Students with more experience can add extra functionality to their weekly assignments (if they wish), as long as they meet all homework requirements and submit on time.
Textbook
There is no text book. Web references and other materials will be posted online.

Course Format

- **Homeworks.** Almost every week, there will be a programming assignment that is highly related to (reinforcing) lecture topics. To get a grade for your homework you have to
  - Complete the homework assignment and test it locally.
  - Publish it and test what you published.
  - Upload a zip file of your WHOLE web application (NetBeans project) into Canvas.
  - Demonstrate homework functionality (from what you published) to the TA during lab period.
  - If you have not completed your homework by the due date, you can still complete it by the following week (with a -20% penalty). After that, homeworks are not accepted, but you still have to complete the work to avoid further deductions on your project grade. (The project is just the culmination of all your labs, regression tested so that everything all works together.)
  - If there is ever any question about a Homework grade, we will go by the code that you have uploaded into Canvas.

- **Project.** Your project is the culmination of all your labs (all combined into a SINGLE, regression tested, web application). Your project grade is based on functional testing plus a code review of all the source code. So, each week, be sure to keep your code well designed/organized, bug free, using self documenting names and adequate comments. EVERY WEEK, your NetBeans Project shall grow – don’t create a new project each week.

- **Lab Activities.** Almost every week, during your lab period, there will be a graded lab activity.
  - Lab activities are short exercises that are started in lab. Before the end of lab, you must demo your work to the Lab Assistant for grading and also upload the code to Canvas. (Any questions about Lab Activity grade will be based on the code submitted into Canvas.)
  - The activities are typically designed to get you started on your upcoming homework programming assignment for the week. So it is not uncommon that a Lab Activity asks you to do a subset of the homework assignment. Even if you do not completely understand everything that you are doing in the lab activity, try to keep a reasonable pace so that you can finish as many milestones as possible. You can always repeat your lab activity (and experiment, digging deeper) after the completion of the lab period.
  - If you miss a lab activity, there is no make-up (but if a student has a long term documented illness, let the instructor know – perhaps an accommodation can be reached). If you miss a lab activity, it is likely that you will have trouble getting started on your homework assignment.
  - To account for any unexpected problem, your lowest Lab Activity grade will be dropped.

- **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. Students will present their tutorials to the class towards the end of the semester.

- **Tests.** There will be two tests. **The best way to do well on the tests is to experiment a lot with sample code and homework code.** There will be no final exam.
### Tentative (and approximate) Grade Weights

<table>
<thead>
<tr>
<th></th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab Activities, must be completed in lab</td>
<td>10%</td>
</tr>
<tr>
<td>Weekly Programming Homeworks</td>
<td>15%</td>
</tr>
<tr>
<td>Project (culmination of all Homeworks with code review)</td>
<td>15%</td>
</tr>
<tr>
<td>Tutorial (including presentation to class)</td>
<td>10%</td>
</tr>
<tr>
<td>Two Tests</td>
<td>50%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

**GRADE EXCEPTION:** if the average of your two tests is less than a C-, then your course grade will also be less than C- (which means you will have to repeat the course).

- Why? Homework assignments are an opportunity for learning. If a student gets too much help with their homework, they can get a good grade on the homework without learning anything. If you want to do well in this course, play with the homework sample code, break it, then fix it — this is the only way to learn how web development works and the only way to prepare yourself for doing well on the tests.

### Grade Scale

<table>
<thead>
<tr>
<th>Grade Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>93-100</td>
<td>A</td>
</tr>
<tr>
<td>90-92</td>
<td>A-</td>
</tr>
<tr>
<td>87-89</td>
<td>B+</td>
</tr>
<tr>
<td>83-86</td>
<td>B</td>
</tr>
<tr>
<td>73-76</td>
<td>C</td>
</tr>
<tr>
<td>70-72</td>
<td>C-</td>
</tr>
<tr>
<td>63-66</td>
<td>D</td>
</tr>
<tr>
<td>60-62</td>
<td>D-</td>
</tr>
<tr>
<td>0-60</td>
<td>F</td>
</tr>
</tbody>
</table>

### 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.
Tentative Schedule

For detailed requirements of homeworks, lab activities, and the tutorial option, see the class website (google “sallyk temple 3344”).

Tentative Course Schedule (Lists Homeworks and Tests)

- Homework 1: Home Page
- Homework 2: Routing
- Homework 3: JS Intro (Create your own JS object with visual representation)
- Homework 4: JS Advanced (slide show)
- First Test
- Homework 5: JS Framework (click sortable HTML table)
- Homework 6: Tutorial Proposal
- Homework 7: Database Setup (easy, work on Tutorial)
- Homework 8: Web API (first time writing server side code)
- Tutorial Presentations
- Homework 9: React Intro
- Homework 10: React Advanced
- Second/Last Test (no final exam)

Spring 2020 IMPORTANT DATES:

- Monday 1/27: Last Day to Drop (you don’t pay for the course and it will not appear on your transcript).
- Monday 3/9: Mid Term Ratings end.
- Wednesday 3/18: Last Day to Withdraw. If you are doing poorly and do not want to have to later retake this CS elective to improve your GPA, withdrawing is a good option for you. Grade “W” remains on your transcript but does not affect your GPA.