Course Overview
CS132, 'Creating Modern Web Applications', is a spring semester course within the Brown CS department. The course takes a holistic look at the process of developing web and mobile applications and aims to bring students to a point of mastery of many of the most used web technologies and development practices.

The course has three intertwined parts. The first involves learning the fundamentals of web applications. In this part, we will look at the fundamentals of web applications. There are a number of different technologies available today. We don’t expect students to be an expert in all of these, but we do expect that students should be aware of them, what they are good for, and how they work so they can reasonably determine what should be used for a new application and can talk intelligently about web applications. There are also a number of factors to consider in building web applications including human factors, universal access and security which will be covered in this part of the course. These will be covered mainly in lectures, assignments, and labs.

The second part of the course involves learning a specific set of technologies for building a web application. Here we will concentrate on HTML, CSS and JavaScript for the front end and Node.JS for the backend. We will also cover databases, both SQL and No-SQL, and various forms of client-server interaction. These will be covered in four programming or design assignments where the students get to apply the technologies.

The third part of the course involves a semester long group final project in which the students will be working in teams with external companies, non-profits, and other organizations to create or augment a real modern web or mobile interface. It is also possible for students to propose and direct their own project if it is well defined at the start of the semester.

The course has two tracks, one intended for CS concentrators, and one intended for non-Concentrators with previous design experience. The concentrator track is designed for students with prior programming experience (CS15/16 or CS17/18 at a minimum, CS32 or CS33 is recommended). Concentrators will learn how to build and maintain all parts of a modern web application. The specific technology assignments and labs for concentrators will involve programming, generally in JavaScript. The design track does not assume any programming background, but does assume a strong design background. Designers will learn how to apply their skills to web and mobile applications and learn how to interact with programmers to create good-looking and effective applications. The technology assignments and labs for designers will involve a minimum amount of programming and a significant amount of design.
The course can be used as a Computer Science capstone. If a student is taking it as a capstone, they must be in their last two semesters, and must either propose and supervise a team project or act as the team leader on their group project.

Students interested in entrepreneurship in Computer Science will find that the course provides several opportunities. Students can propose a project that could eventually be the basis of a start-up, and have a team of work on it. Project development will include ample opportunities to get experience in selling the underlying ideas including elevator talks, a poster presentation, and in-class presentations. The prototype system that is produced as the project can be used as a prototype for a potential start-up.

**Course Mechanics**

The class will meet Monday, Wednesday and Friday from 10:00 to 10:50 in Metcalf Auditorium. Students are encouraged to attend each class. Information is posted on the course web site ([http://www.cs.brown.edu/courses/csci1320](http://www.cs.brown.edu/courses/csci1320)) and students are expected to keep up to date with the announcements there.

There is no text book per se for the course. The course web site provides a set of reference links that can be used to get information about specific technologies. Assignments will include web-based tutorials providing additional information.

The course will involve a team project with various milestones during the semester. It will involve four programming/design assignments. There will be in-class labs on specific technologies. The labs will be preceded by a prelab assignment that should introduce students to the material in a tutorial manner. The labs themselves will let students demonstrate the associated technology in a creative way.

Dr. Reiss will have office hours Monday and Thursday from 1-3 pm. TA hours will announce during the first week of classes.

**Project**

Much of the course will involve a team project proposed by either outside sponsors or students. Project teams should ideally be four people. The project schedule is tentatively:

<table>
<thead>
<tr>
<th>DATE</th>
<th>PROJECT EVENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/27</td>
<td>Student Project Proposals due</td>
</tr>
<tr>
<td>1/30</td>
<td>Project proposals available</td>
</tr>
<tr>
<td>2/3</td>
<td>Project preferences due</td>
</tr>
<tr>
<td>2/6</td>
<td>Project teams announced</td>
</tr>
<tr>
<td>2/17</td>
<td>Initial client report</td>
</tr>
<tr>
<td>2/27</td>
<td>Project specifications due</td>
</tr>
<tr>
<td>3/3</td>
<td>User feedback report</td>
</tr>
<tr>
<td>3/6,8</td>
<td>Project elevator talks</td>
</tr>
<tr>
<td>3/13</td>
<td>Initial project design</td>
</tr>
</tbody>
</table>
### Grading

Your grade will be determined from programming assignments, labs, class participation, final project, and final exam:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Assignments</td>
<td>8% each (32% total)</td>
</tr>
<tr>
<td>Labs, participation (iClicker)</td>
<td>23%</td>
</tr>
<tr>
<td>Group Project</td>
<td>35%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>10%</td>
</tr>
</tbody>
</table>

A rough estimate of the required time would be 120 hours over the semester for the final project, 10 hours each for the four programming assignments.

Each student can use three late days throughout the whole semester. Late days can be used only for assignments, not for the final project. Students can use late days as they want; they can use all three late days for one assignment or three different assignments. Without late days, late assignments are penalized 15% per day. Your passes will be applied optimally at the end of the semester on your behalf. All assignments must be turned in, even if they get 0 credit. Again, the final project will not be accepted late.

Students are also required to do in-class labs. They will get checked-off at the end of the class time, if they participate in the labs in class. They are not expected to finish the labs to get checked off. However, if students miss the class, they should finish their labs and show the results to TAs during TA hours.

The portion of iClicker grade will be determined when the total number of labs that the class will have is finalized. The correctness of iClicker answers does not affect the grade.

### Collaboration

In general, this course encourages collaborative efforts. The final projects are collaborative. The in-class labs are collaborative. Especially for their final projects, students are encouraged to use publically available libraries and materials as long as they are appropriately cited. However, the programming
assignments and final exam are not collaborative exercises. They should be your own work. Details of the policy can be found on the web site and the collaboration policy that must be signed by all students.

**Tentative Class Schedule**

Lecture: Course Introduction  
Lecture: The Browser  
Lecture: Accessibility  
Lab: HTML/CSS  
Lecture: JavaScript  
Lecture: HTML5, dynamic web pages, ...  
Lab: JavaScript/JQuery  
Lecture: Requirements/Specifications  
Lecture/Lab: Front end frameworks (angular, react ...)  
Lecture: The Web Server  
Lecture: Web App Architectures (ajax. sockets. cookies, sessions)  
--------- Holiday  
Lecture: Node.JS  
Lab: Note.JS  
Lecture: SQL Databases  
Lecture: NoSQL Databases  
Lab: Databases  
Project: Elevator Talks  
Project: Elevator Talks  
Lecture: Backend Technologies, MVC, PHP, CMS, Templating  
Lecture: Ruby/Django/... (ORM, MVC)  
Lecture: HCI Navigation  
Lecture: HCI Design  
Project: Poster fair  
Project: Poster fair  
Lab: HCI Design  
--------- Spring Break  
Lab: AWS  
Lecture: Security  
Lecture: Security  
Lecture: Privacy and Security  
Lab: Security Challenge  
Lab: Security Challenge  
Project: Presentations  
Project: Presentations  
Lecture: Testing  
Lecture: Testing  
Lab: Testing  
Lecture: Nobbles  
Lecture: NativeScript or Meteor  
Lecture: Future of Web Apps  
Lecture: Course Review  
Project: Final Presentations  
Final Exam