CSCI 1320
Creating Modern Web Applications

Lecture 1: Course Introduction
Welcome To CSCI1320 (CS132)

- Professor (spr@cs.brown.edu)
- HEAD TA (cs132headtas@cs.brown.edu)
- TAs (cs132tas@cs.brown.edu)
What is a Web Application

- A program that the user interacts with through the Internet.
  - Interact via a browser
    - Or a mobile front end
  - Using standard (HTTP) protocols
  - Where the program runs on a server
  - (Where the program uses a database)
Sample Web Applications

- Name some web applications
- Which you like to use
- Which you don’t like to use

Lecture 1: Course Introduction

5/1/17
What Makes a Web App Different

- Why do we have a course like CS132?
  - Why isn’t CS32 sufficient
- Can anyone write a successful web application?
- What is a “web designer”?
- Architecture and Expectations
  - Web architectures are different
  - Users and clients have different expectations
Elements of Web Applications

- The importance of Human-Centric Computing
- Distributed Programming
- Security and Privacy
- Scalability
- Evolution
- Software Engineering
Human-Centric Computing

- User interface design
- Ease of use
- Looking good
- Accessibility and internationalization
- These make or break a web application
Distributed Computing

- Web applications are inherently distributed
- They use facilities outside of programmer’s control
- They are written in a multitude of languages
- Communication is asynchronous
- Frameworks try to make this simpler
- Nothing is standard
Security and Privacy

- **Are major concerns**
  - In the press daily
  - Your application is exposed to the world
  - All types of attacks are possible

- Same interface used by multiple users at once
- Multiple applications might run on same server

- **Private data needs to be secure**
  - Especially sensitive data (e.g. credit cards, health data (HIPAA))

- **Applications often have real-world implications ($$$)**
- **Liability issues arise**
Scalability

- How many users do you expect to have
  - After you’ve been slash-dotted
  - On Cyber Monday
- Handling 1,000 users at once is hard
  - Handling 10,000 requires a different approach
  - Handling 1,000,000 requires rethinking the application
Evolution

- **Web apps need to change**
  - The look gets stale after a year (more or less)
  - New functionality desired
  - Users expect new features, new look and feel
  - Need to keep up with competition

- **Different form factors and capabilities**
  - Different browsers
  - Tablets, phones, watches and other devices
Software Engineering

- Different programming languages and models
- How to work in teams
- How to work with clients
- How to work with deadlines
- How to organize a large project
- How to plan for evolution
- How to plan for problems

Lecture 1: Course Introduction
What’s Involved in Web Apps

- Requires **understanding**
  - The application, the users, and the needs of the users

- Requires **design expertise**
  - User interface design, usability, scalability, maintainability

- Requires **sophisticated programming skill**
  - Handling 10,000 users; 3-5 9’s of up time; updatable

- Requires **programming expertise** in several areas
  - Interactive, Large-scale server technology, distributed programming

- Typically requires a **development team**
  - Designers, programmers, testers, users
In This Course

- You are going to build **a real web application**
  - For real users
  - In teams with mixed skills
- You are going to learn the **basics of web applications**
  - Won’t become an expert in all of them
  - Will learn the alternatives, terminologies, etc.
  - Will learn enough to build your own application if desired
  - Will become an expert in some aspect for your project
- You are going to learn **to work in teams**
CSCI1320 has Two+ Tracks

- **Concentrators Track** (for concentration credit)
  - For CS students with programming background (CS32/CS33)
    - How necessary is CS32/CS33
  - Emphasis on programming skills
  - Responsible for programming aspects of projects

- **Designers Track**
  - For students with design skills
  - Emphasis on web design and learning how it can be used
    - Assignments are design- not programming-oriented
  - Responsible for human-centric aspects of projects
  - Please email cs132headtas@cs.brown.edu
CS132 Has Two+ Tracks

- Others Track
  - For students without significant programming experience
  - Who don’t know HTML/JavaScript
  - Who have little design experience
CS132 Has Two+ Tracks

- **Capstone Track**
  - Students taking the course as a capstone
  - Expected to either
    - Propose and supervise a project
    - Serve as the team leader on their project
    - Both
CS132 Has Two+ Tracks

- Entrepreneurship Track
  - Student(s) propose a project that is basis for a startup
  - Initial proposal is for MVP
  - Elevator talk, poster, presentations
  - Build prototype in the course, develop product over summer
Course Mechanics

- There are three parts to the course
  - Learning **the fundamentals of web applications**
  - Learning the **basics of building web applications**
  - Creating a web application for a client
- Reflected in time commitment and grading
Fundamentals of Web Apps

- There are lots of different web technologies
  - More than we can cover in one course in any depth
  - But a web app expert should know of them all
    - What they are, what they are good for, how they work, ...
    - You need to know what to use
    - You need to be able to talk to clients and others

- There are lots of things to consider in designing and building a web application
  - Security, human factors, universal access, ...
  - You need to understand and deal with these
Fundamentals of Web Apps

- Covered in lectures, homeworks, tutorials & labs
  - I’ll try to make this accessible to both tracks
  - Questions and comments are encouraged
  - **PLEASE**
- Checked w/ homeworks, labs, participation (23%)
  - All should be relatively easy if you come to class
- Final exam (10%)
  - Possibly take-home, but don’t plan on it
Basics of Building Web Apps

- Understand a **specific set of technologies**
  - HTML5/CSS
  - JavaScript
  - Node.JS / SQL or NoSQL
  - Client-Server interaction (AJAX)
- These will be covered by **homework assignments**
  - Each one to two weeks long
  - Separate assignments for the two tracks
  - Count for 32% of your final grade
Web Application Project

- We have gathered a suite of projects from real clients
  - Mix of commercial, non-profit, local
- Based on your preferences we will assign teams
  - Four people, mixed backgrounds, apt for project
- Teams should meet weekly with sponsor & mentor TA
  - Keep them happy
- You will have opportunities to present your project
- Counts as 35% of your grade
  - Grading based on project itself, presentations, sponsor feedback
Student Projects

- We are going to allow a limited number of student projects
  - If you have a web applications you really want to create
  - Proposer will act as project mentor
- These need to be well-defined and scoped
- If anyone is interested in doing one of these
  - We need a detailed proposal for it today
  - Talk to the TAs for advice and suggestions
Project Schedule I

- 1/30: Project preferences out; due 2/03
- 2/06: Final project teams announced
- 2/17: Initial client report
- 2/27: Project specifications hand in
- 3/03: Potential user feedback reports hand in
- 3/06, 3/08: Project Elevator Talks
- 3/13: Initial project design presentation to TA
- 3/17: Project design hand in
- 3/20, 3/22: Project Poster Fair
Project Schedule II

- 4/12: Project web design due
- 4/17-4/19: Project presentations
- 4/19: Project design hand in if not presenting
- 4/21: Project prototype up and running
- 4/26: Prototype feedback from client
- 5/01: Project testing reports due
- 5/08: Final project presentations (whole day)
- 5/09: Final project hand-in
- 5/10-5/15: TA meeting with project team
Collaboration Policy

- Please pick one up, read and sign, hand in
  - You won’t be assigned to a project unless you do
- We expect you to do your own work on assignments
  - Not copy from others
  - Not copy from the web
- Much of the rest of the class is collaborative
- We **will** detect cheating
  - When in doubt about using something, ASK.
  - Always cite any external code, references, ideas, etc.
  - Several students got directed NCs for the course in the past
Intellectual Property (IP)

- **You own your code** (Brown’s policy)
- Since you are working with others, **they have rights**
  - Non-exclusive perpetual right to the code and its use
  - Complete rights to any images, etc. they provide
- You should **negotiate/agree** with sponsor on final rights
  - They might want code open sourced
  - They might want exclusive rights
  - Do this early in the process
- **Some projects are constrained**
  - Take this into account in choosing projects
Hours

- **TA Hours will be announced**
  - Based on homeworks, assignments, etc.
  - Each project will have an assigned mentor TA
    - You are responsible for setting up meetings

- **My office hours**
  - Monday, Thursday 1-3
  - Open office policy
  - I expect to see everyone at office hours at least once
Course Web Site

- http://www.cs.brown.edu/courses/csci1320
- Reference Materials
  - Links to that other material, cherry picked for you
- Calendar
- Keeping up to Date
Questions regarding the Course
Next Time

- The Browser, HTML, CSS, and the web front end
- Homework:
  - Collaboration Policy
  - Preliminary work for Lab 1 (due Wednesday)
  - Available on web site
You have been hired to work on creating a system for web-based course registration, let's call it Banter.

- What are the problems you would anticipate?
- How would you proceed?

What do you see as the potential problems?

What would you work on first?

How would you sell it?