



**CS1320**  
***Creating Modern Web and  
Mobile Applications***

Lecture 1:

# Course Introduction

# Welcome To CSCI1320 (CS132)

- Professor ([spr@cs.brown.edu](mailto:spr@cs.brown.edu))
- HEAD TAs ([csl32headtas@cs.brown.edu](mailto:csl32headtas@cs.brown.edu))
- TAs and ETAs ([csl32tas@cs.brown.edu](mailto:csl32tas@cs.brown.edu))

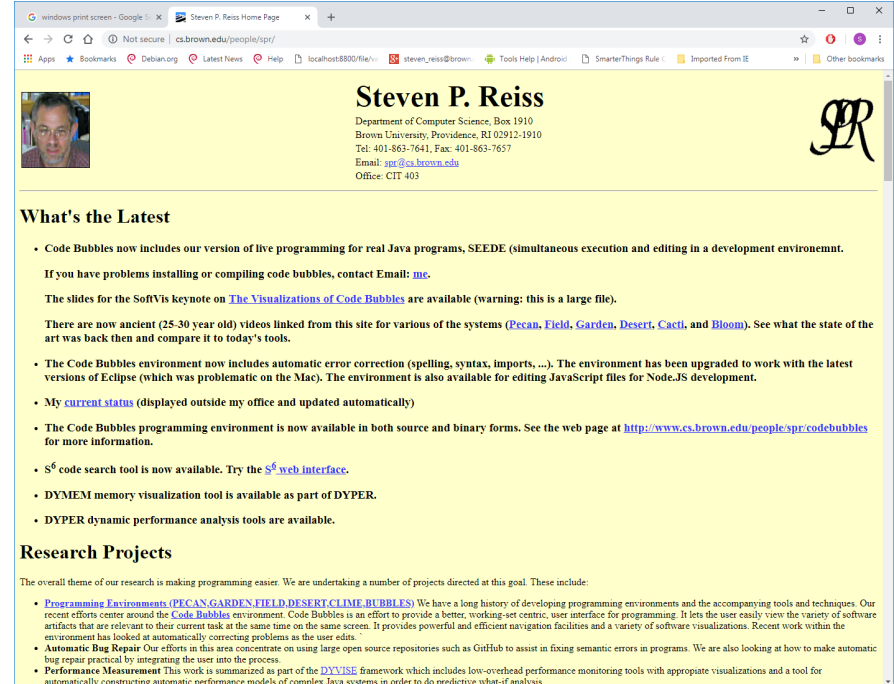


# Everyone today uses the Internet

- On-line shopping from
- Browse social media
- Read email
- Do web searches
- How many do on-line research (in place of using the library)?
- Do you use an app or a web browser?

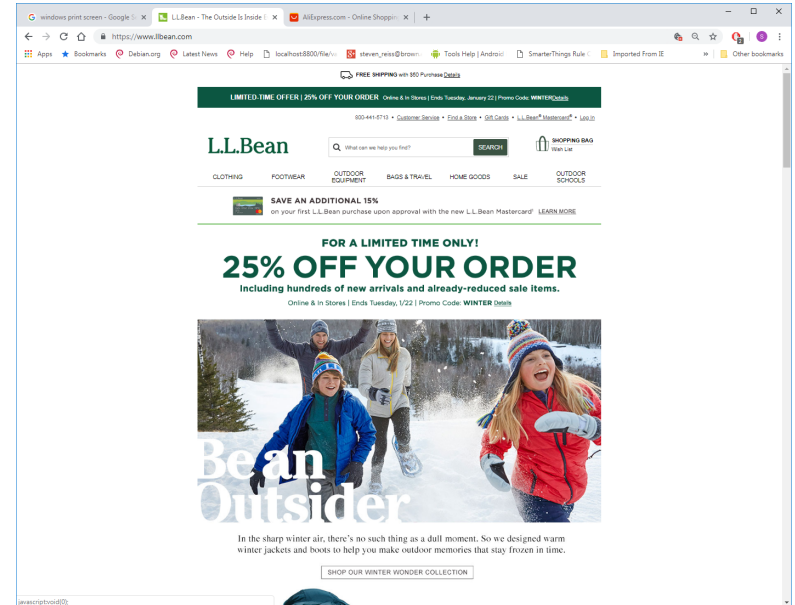
# Who Has Created a Web Site?

- Static or dynamic?
- What technology did you use?
- How often do you update it?



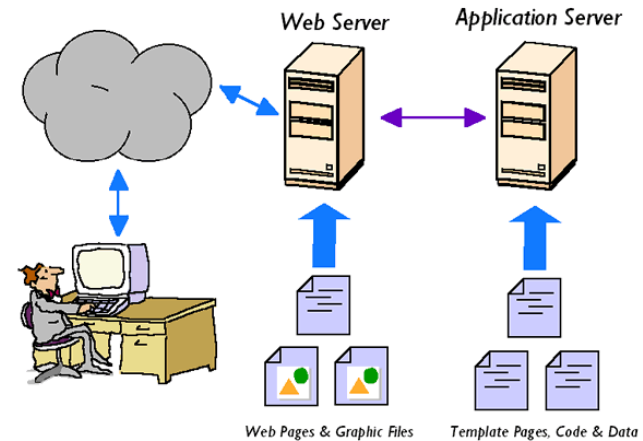
# Creating a Modern Web Application

- Think of simple shopping site
- Would you feel confident creating it
  - What do you see as the difficulties
  - How much work would it be
- You should be able to do this
  - By the end of the course



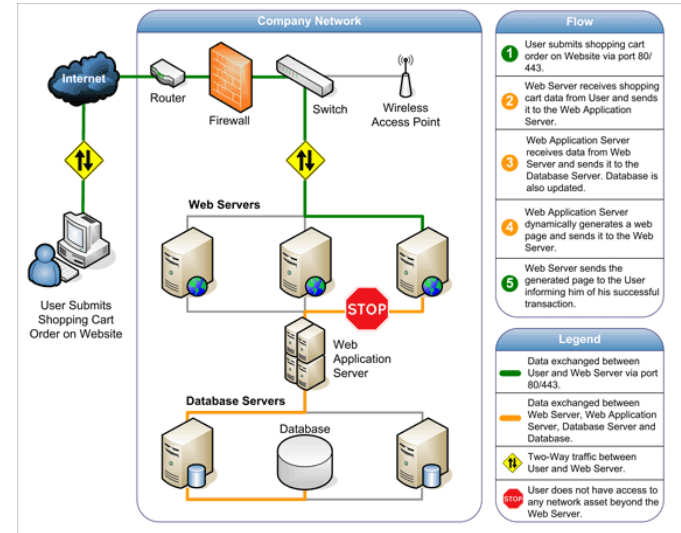
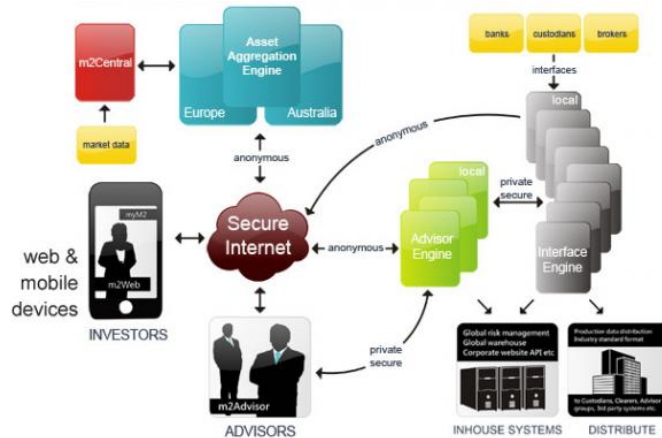
# What is a Web or Mobile Application

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

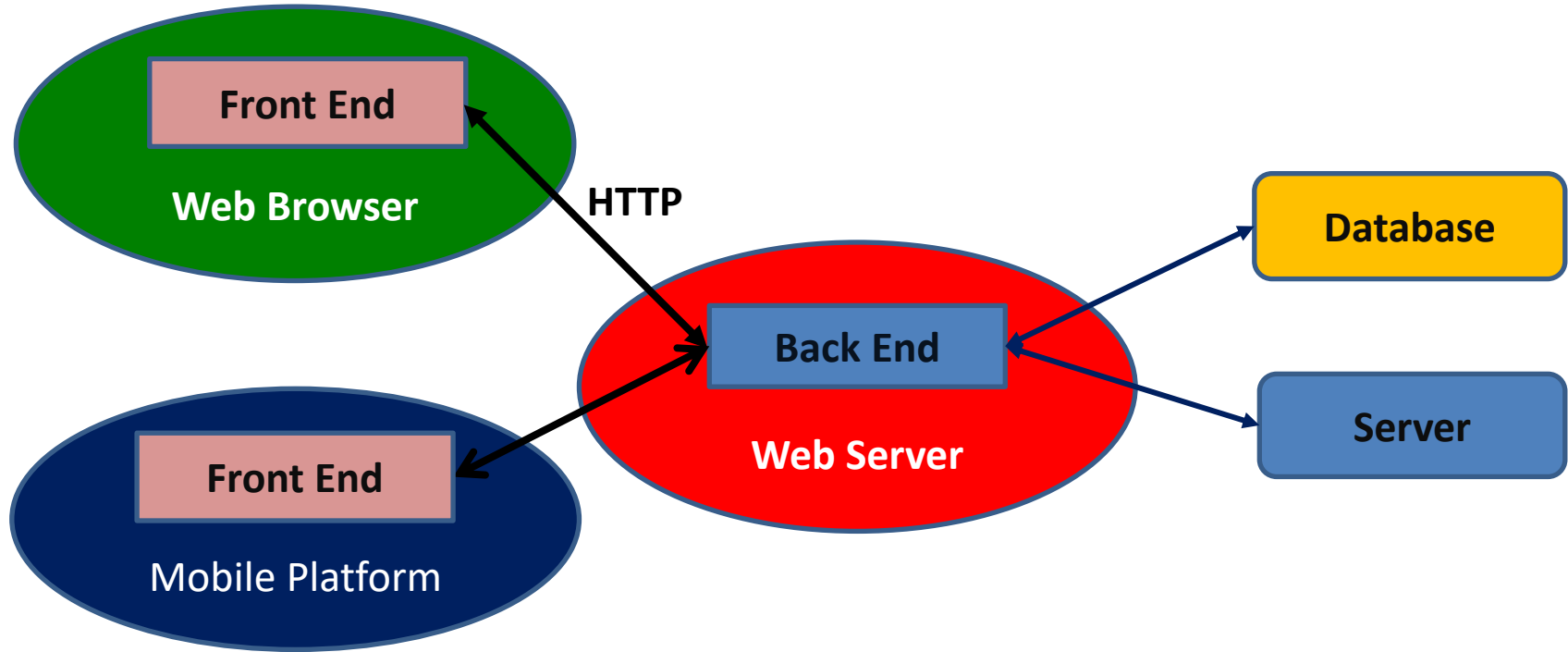


# Sample Web and Mobile Applications

- Name some web and mobile applications
  - Which you like to use
  - Which you don't like to use



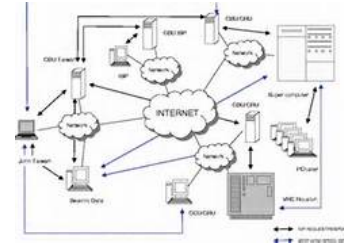
# Web Application Architecture





# Elements of Creating Web/Mobile Applications

- The importance of Human-Centric Computing
- Distributed Programming
- Security and Privacy
- Scalability
- Evolution
- Software Engineering (specs, design, testing, ...)



# What's Involved in Web/Mobile 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 or mobile application
  - For real users
  - In teams with mixed skills
- You are going to learn the **basics of web and mobile 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

- **Concentrator's Track**

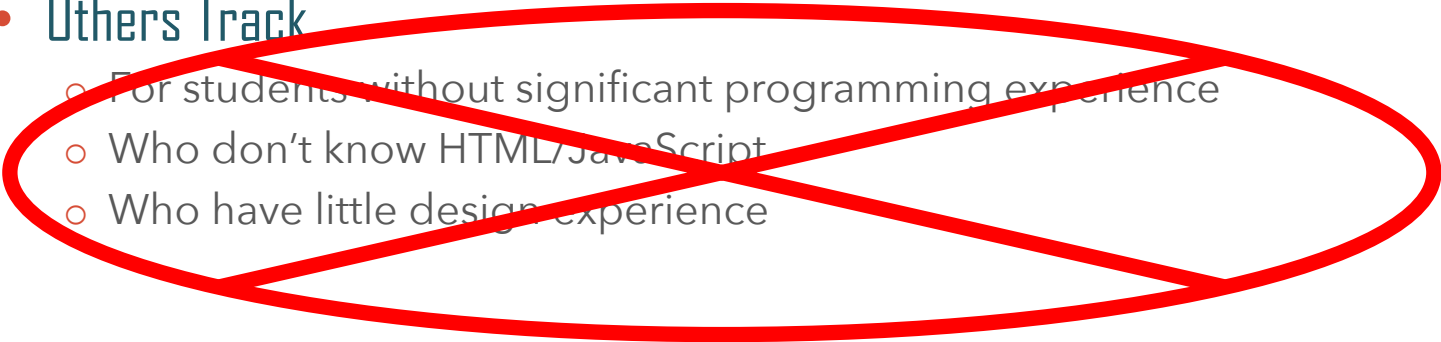
- For CS students with programming background (CS32/CS33)
  - How necessary is CS32/CS33
- Emphasis on programming skills
- Responsible for programming aspects of projects
- Different levels of programming in the projects

- **Designer's Track**

- For students with design skills
- Limited or no programming experience required
- Emphasis on web and mobile design and learning how it can be used
  - Assignments are design-oriented not programming-oriented
- Responsible for human-centric aspects of projects
- Please email [cs132headtas@cs.brown.edu](mailto: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
- 

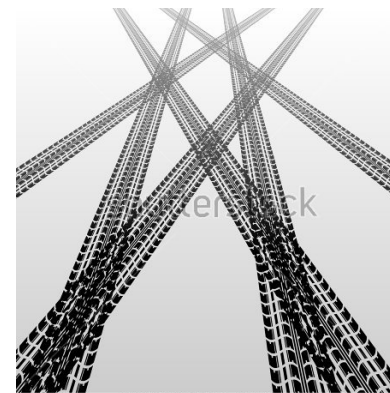
# CS132 Has Two+ Tracks

- **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
- **CS Design Track**
  - Mix concentrator and design assignments to maximize knowledge gained
  - For CS students with a strong design bent
  - Work on design aspects of final project



# 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 MVP (prototype) in the course, develop product over summer



www.shutterstock.com · 599905853

# Course Mechanics

- Laptops / Phones
  - Used in lab classes
  - Shouldn't be used in lectures
- SEAS, extensions, late days





# Course Contents



- There are three parts to the course
  1. Learning **the fundamentals of web & mobile applications**
  2. Learning the **basics of building web & mobile applications**
  3. **Creating a web or mobile application** for a client
- Reflected in time commitment and grading

# Fundamentals of Web Apps

- There are lots of different web and mobile technologies
  - More than we can cover in one course in any depth
  - But a web app or mobile 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 or mobile application
  - Security, human factors, universal access, testing, design, ...
  - You need to understand and deal with these



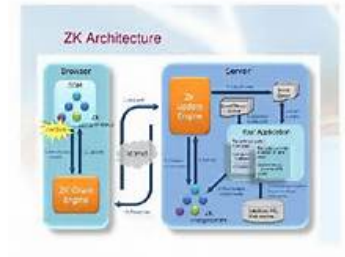
# Fundamentals of Web and Mobile Apps

- 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
- Tested in the Final exam (10%)
  - Take-home



# Basics of Building Web and Mobile Apps

- Understand a **specific set of technologies**
  - HTML5/CSS
  - JavaScript
  - Vue
  - NativeScript
  - Node.JS / SQL and NoSQL / AJAX
- These will be covered by **5 programming/design assignments**
  - Each one to two weeks
  - Separate assignments for the two tracks
  - Count for 35% of your final grade



# Web or Mobile 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
  - Teams should have a leader and a sponsor contact person
- You will have opportunities to **present** your project
- Counts as 32% of your grade
  - Grading based on project itself, presentations, milestones, sponsor feedback

# Student Projects

- We are going to allow a limited number of student projects
  - If you have a web or mobile application you really want to create
  - Proposer will act as project mentor
- These need to be well-defined and scoped
  - Should be something different (not another scheduling application)
- If anyone is interested in doing one of these
  - We need a detailed proposal for it soonest
    - **Today preferred, Saturday at the latest**
  - Talk to the TAs for advice and suggestions



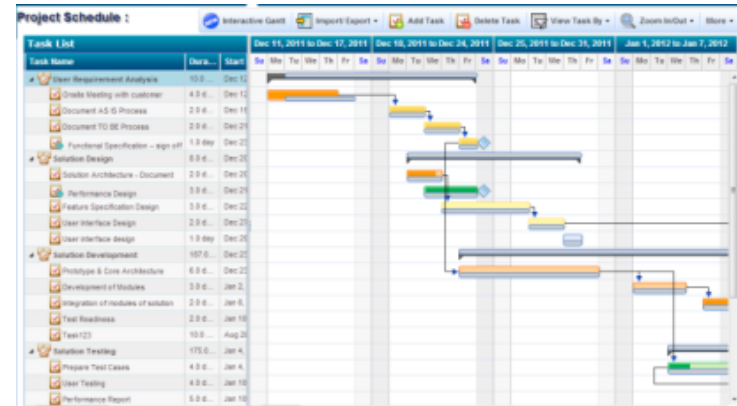
# Software Engineering in the Project

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



# Project Schedule I

- 1/28: Initial project preferences out; due 1/31
- 2/05: Final project teams announced
- 2/14: Initial client report
- 2/24: Project specifications hand in
- 3/02: Project Elevator Talks
- 3/06: Potential user feedback reports hand in
- 3/09: Initial project design presentation to TA
- 3/16: Project front end design hand in
- 3/16, 3/18: Project Poster Fair





# Project Schedule II

- 4/10: Project implementation design hand in
- 4/20: Project prototype up and running (target)
- 4/24: Prototype feedback from client
- 4/29: Project testing reports due
- 5/04: Final project presentations (whole day)
- 5/08: Final project hand-in
- 5/08-5/12: TA meeting with project team



# Collaboration Policy (Homework 0)

- Please download from the web site, 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 the 5 programming/design assignments
  - Not copy from others
  - Not copy from the web
- Much of the rest of the class is collaborative
  - Except for the final exam
- We **will** detect cheating
  - When in doubt about using something, ASK.
  - Always cite any external code, references, ideas, etc.
  - Always include external copyrights, etc.
  - Several students got directed NCs for the course in the past



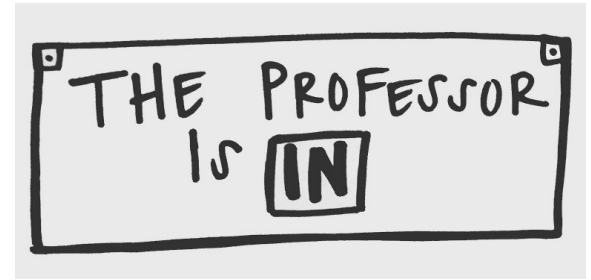
# Intellectual Property (IP)

- **You own your code** (Brown's policy)
  - In a group project, this is generally shared ownership
- **When you are working with others (sponsors), they have rights too**
  - 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 (must be in project definition)
  - They might want documentation on maintenance and use
  - They might want code non-programmers can easily modify
  - Do this early in the process
- **Some projects are constrained**
  - Take this into account in choosing projects



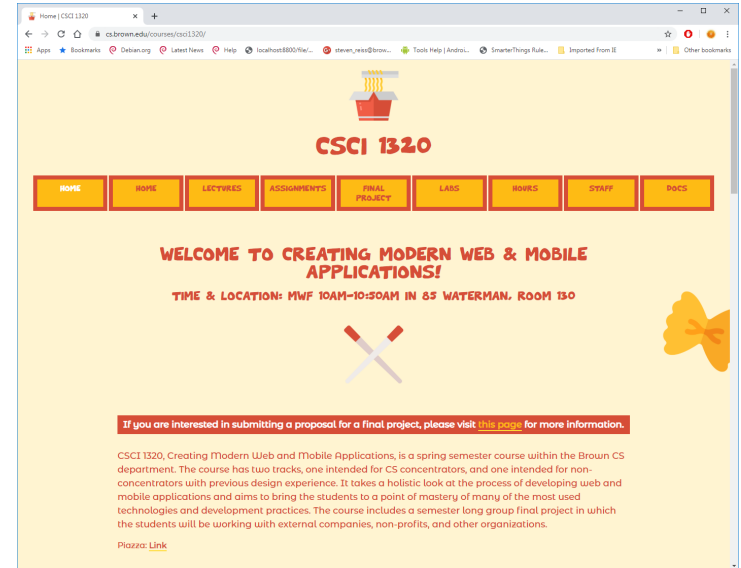
# Hours

- 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 (tentative)
  - Open office policy (8:30-3:30)
  - I'm hoping to see everyone at office hours at least once
  - I'm usually free even if the TA hours are very busy



# Course Web Site

- <http://www.cs.brown.edu/courses/cscil320>
- Reference Materials
  - Links to that other material, cherry-picked
- Calendar
- Keeping up to Date
- Piazza
  - Information for opting out



# Questions regarding the Course



# Next Time

- The Web Front End: The Browser, HTML, CSS
- Homework:
  - Assignment 0
    - Available on web page, due Friday 1/31
    - Collaboration policy
    - Account setup, etc.
  - Preliminary work for Lab 1 (due Wednesday)
    - Available on web site
    - **Get started now!!!**

# Problem

- You have been hired to work on creating a system for web-based course registration, lets 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





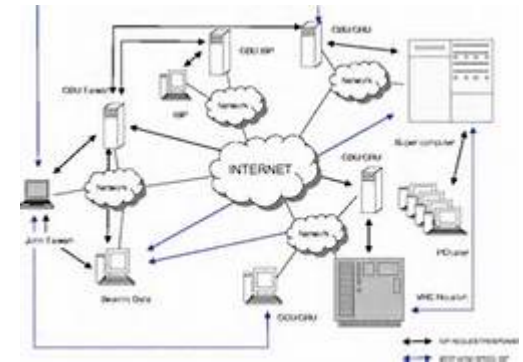
# Human-Centric Computing

- 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

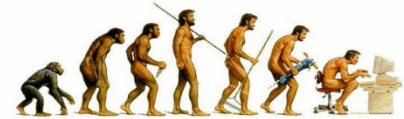


# Scalability

- How many users do you expect to have
  - After you've been slash-dotted
  - On Cyber Monday
- Handling 1000 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