Web and Mobile Application Development

Welcome to CS275

Introduction

Material created by:
David Augenblick, Bill Mongan, Dan Ziegler, Samantha Bewley, and Matt Burlick

Introduction

This course will teach a cross-platform approach to web application development

- We can split this course into two parts:
  1. Client side features
  2. Server side features

Introduction

Client Side Technologies

- HTML4/HTML5
- CSS
- JavaScript
- JSON
- JQuery
- Ajax

Server Side Technologies

- Node.js
- MySQL

Introduction

- Ultimate goal – to be able to build an integrated (client + server sides) web application
  - Web page accepts input parameters and passes a request to the server
  - Server interacts with a database to obtain and then return requested information
  - Web page then posts the information

Introduction

- The course will NOT focus on the following:
  - Android or iOS mobile application development
    - Although we may have a brief talk about this at the end of the course.
  - Advanced features of web page design
    - Although some of our demos will include some of these features

Introduction

- Students will be expected to “fill in the blanks”
  - Basic concepts with be taught and demonstrated, while more advanced features will need to be researched
- Students may develop in the environment of their choice (after all everything we do should be cross-platform) but we recommend developing locally on your machine.

Administrative Stuff..

- Faculty
  - Matt Burlick: mjburlick@drexel.edu
    - UC, Room 137
    - Office Hours:
      - Tuesdays 11:00am-12:00pm
      - Wednesdays 4:00pm-6:00pm
      - And by appointment
- TAs/Graders
  - Denisa Qori: dq38@Drexel.edu
    - Office hours in CLC, UC152, Wednesdays 10:00am – 12:00pm

Administrative Stuff..

- Assessment
  - Assignments 35%
  - Exams 20%
  - Practicum 20%
  - Final Project 25%

Administrative Stuff..

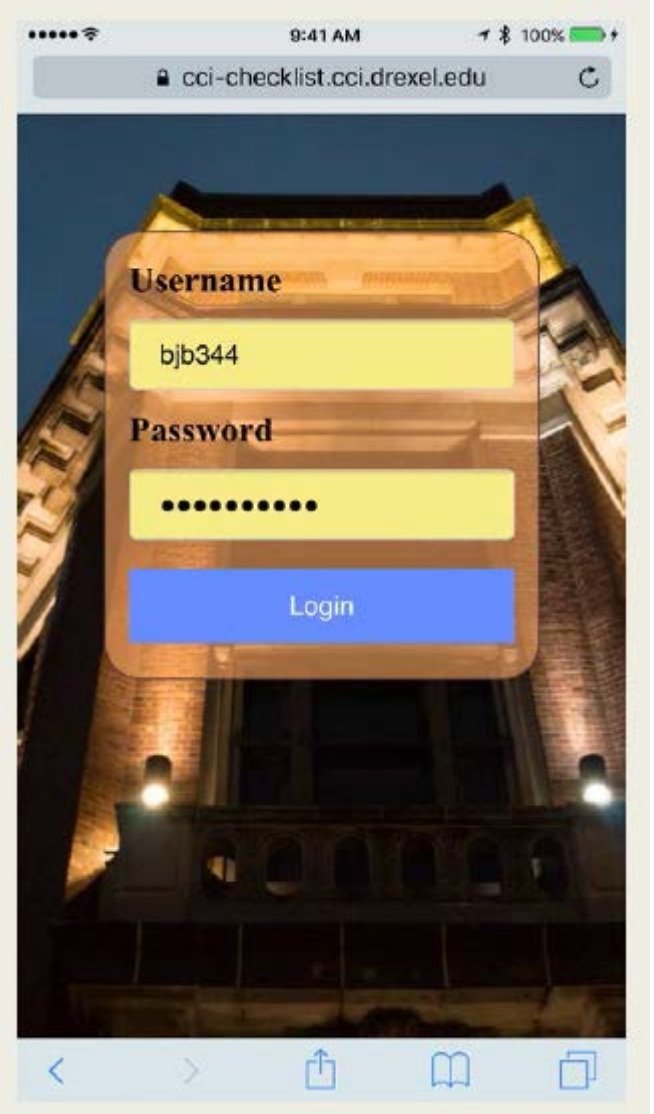
- Assignments
  - Approximately five
  - Done individually
  - Started during a lecture
  - Completed by Sunday evening
  - Screencast submission to Blackboard
- Exams
  - Approximately two.
  - Administered at the beginning of two of the lectures.
  - Purpose is to test your understanding of the theory

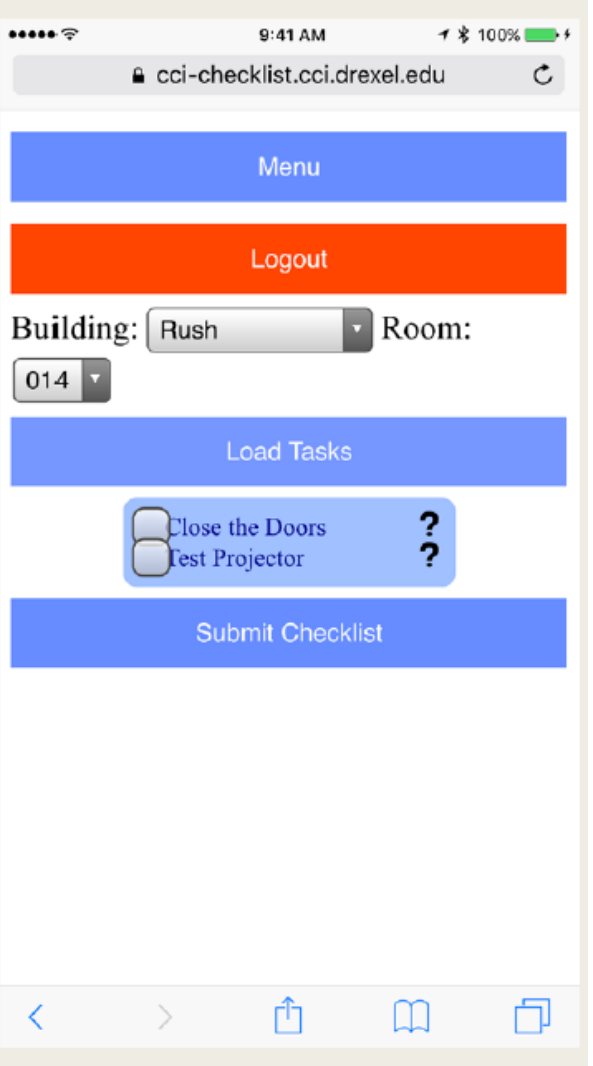
Administrative Stuff..

- Practicum
  - To test your ability to re-produce a site given limited time and resources there will be an individually done in-class practicum.
  - We will provide you with a desired web program to complete and you will have the entire lecture period to attempt to replicate it.
  - Done individually.
- Final Project
  - May work in groups of 3-4.
  - Design and implement a full web program demonstrating all of the ideas/technologies of the course.
  - Must do a project pitch around the middle of the term as well as requirement and design documents

Sample Final Projects

- CCI Checker
  - CCI staff can check status of rooms
  - Mark things done
  - Mark issues



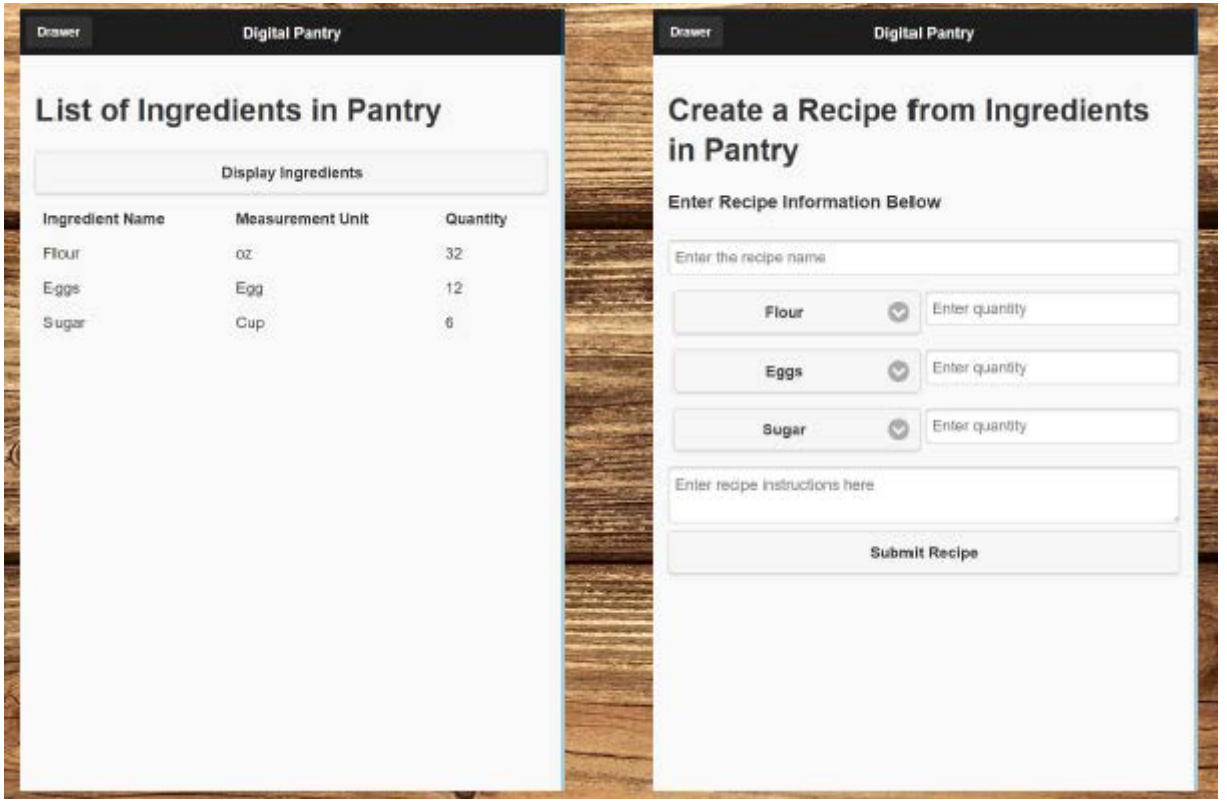


Sample Final Projects

- GEO-Tag
  - Have your friends use your keyword to post their location
  - “Scoreboard” on who has most friends in most places

Sample Final Projects

- Digital Pantry
  - Allow users to store and manage ingredients and recipies.



Sample Final Projects

- Truck Stop – Food truck review app
  - Where food trucks are currently located.
  - How long is wait
  - Costs
  - Ratings

Administrative Stuff..

- Policies
  - Work to be done individually unless otherwise stated.
  - Any dispute of a grade must be made within 5 days or receiving your grade.
  - Missed Assignments – If you miss a lecture that contains a “lab” (starting of an assignment) you may still submit your screencast on time.
    - However if this occurs more than once, then you will receive a zero for any subsequently late assignments.
  - Missed Exams – If you miss a lecture that contains an exam, you may only make this up if you have a note declaring this as an excused absence. In this scenario, you must make arrangements with a TA or instructor to take a “make up” exam prior to the following lecture.
    - Again this can only occur once.
  - Missed Practicum – Only excused missed practicums can be made up and must be arranged with a TA or instructor prior to the following lecture.

The Internet

- What is “the internet”?
  - A collection of protocols, technologies and systems to enable communications and data transmissions between computers (basis for networks)
  - Communication protocols such as TCP/IP and HTTP allow computers to talk to each other (more details in next lecture)

The World Wide Web

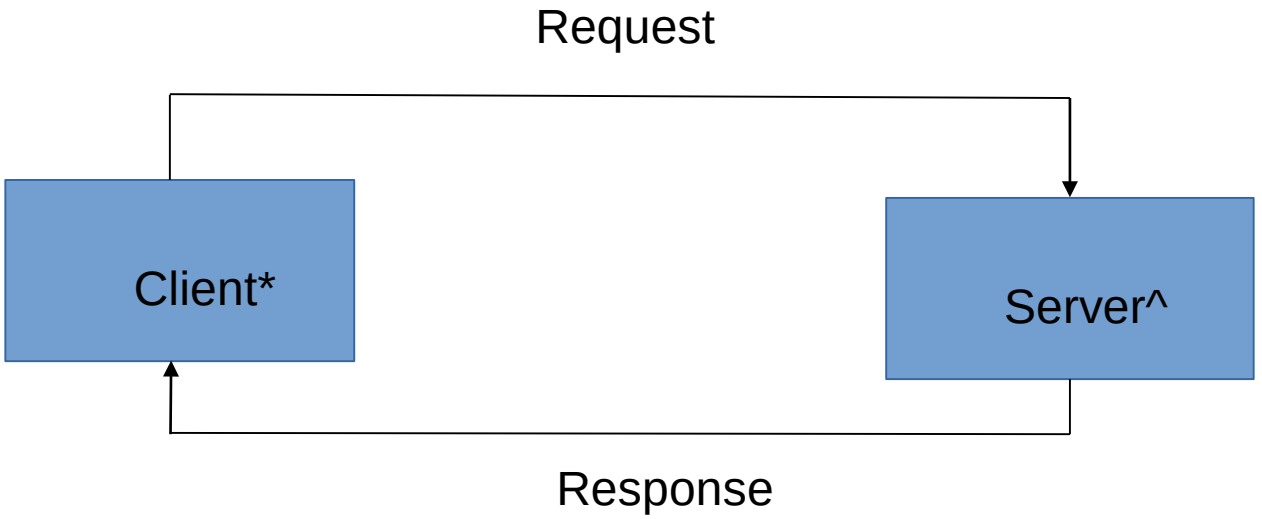
- What is the “world wide web” (WWW)?
  - A subset of the internet responsible for its quantum of use starting around 1990
- Consists of the following elements:
  - HTTP (HyperText Transfer protocol) – handles requests and responses
  - URL (Uniform Resource Locator) – to identify www resources
  - Web server software – programs to handle HTTP requests
  - HTML (Hypertext Markup Language) – for document publishing
  - Browser – makes URL based requests + displays the HTML results

Clients and Servers

- We typically think of web applications as having (at least) two components
  - Client
  - Server
- The job of the client is to make requests to the server and to render content to the screen
- The job of the server is to listen for requests, populate responses, and send back those responses to the requesting client.

Request-Response Loop

- The client-server model (the request-response loop)



* Includes a web browser
* Issues resource requests to server via URLs
* Waits for responses from server and then processes them

^ Listens for requests
^ Processes request
^ sends back response

Client-Server Basic Interaction

- Lets drill down a little into the details of the client-server interaction…

1. Browser sends request for a (HTML) file (or sometimes data) located at some address (for example http://www.somefilename.com)
   - We'll see how the URL address gets translated to the server's IP address shortly
2. Web server finds the file (or obtains the data), opens it and sends the content back to browser.

Client-Server Basic Interaction

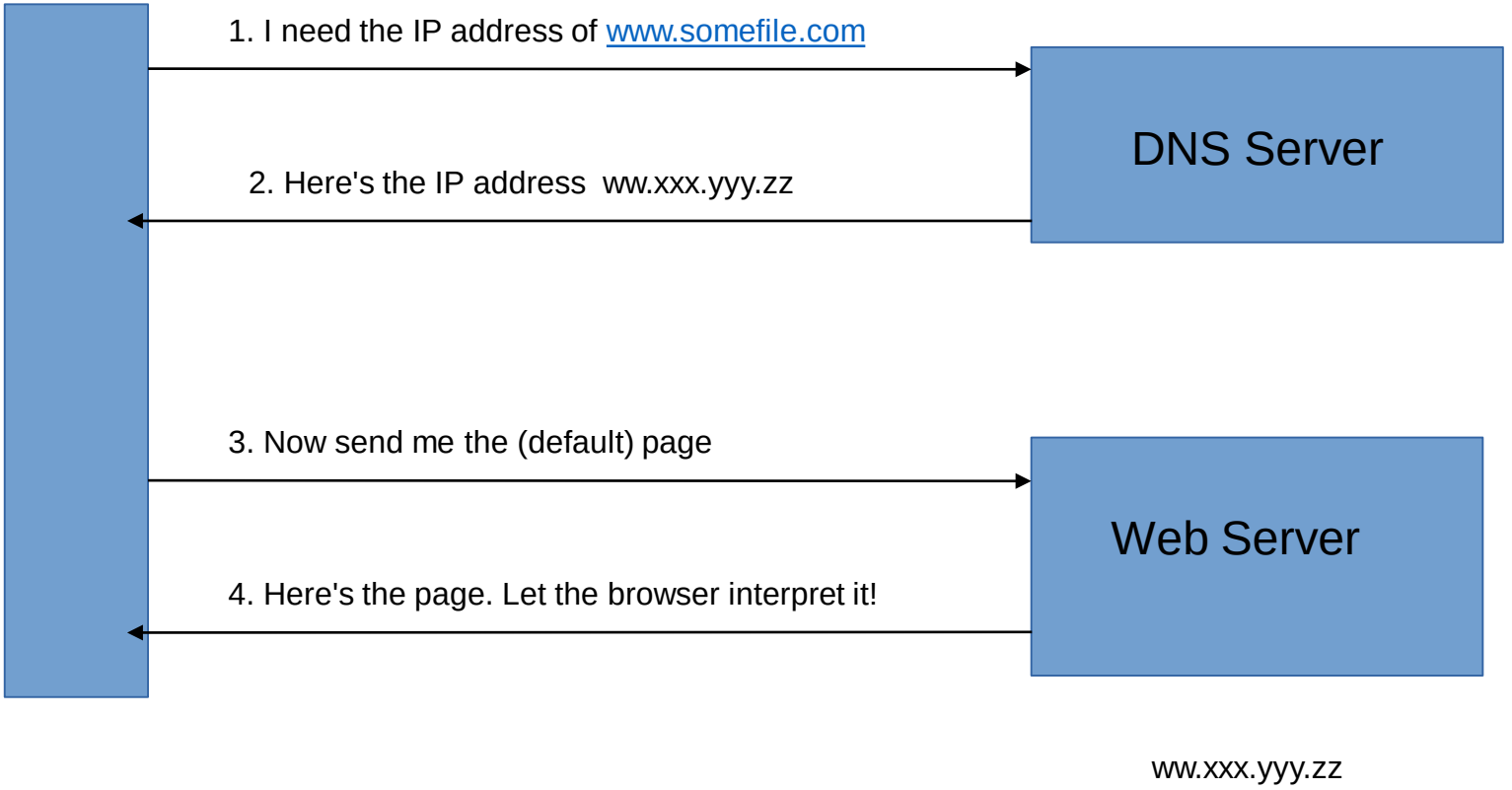
3. Browser interprets the HTML content (or data values) and renders content as specified
   - **If an image is specified within the HTML content:**
     - Browser requests and retrieves (from the web server) the image from the file source specified in the HTML code
     - Browser displays the image within the web page on the client's monitor
   - **If data is requested:**
     - Browser makes request (within HTML) to the server
     - Server obtains data and sends back to client
     - HTML / JavaScript parses and issues the data to the web page

Domain Name System (DNS)

- As described on the previous slide, the address (URL) submitted by the browser must map to the target server's IP address (every server has it's own)
  - Note – users deal with requests at the “user friendly” URL level, while the more complex IP address is mapped to behind the scenes
  - Kind of like using variable names in programming instead of memory addresses
- The DNS translates the URL to its IP address and passes it back to the browser
- The full communication is diagrammed on the next slide

The Browser – DNS Server Interaction

This is how the request URL is translated to it's IP address



Client - Browser