EECS 4413: BUILDING E-COMMERCE SYSTEMS

Fall 2021, Section A

About this Course

Copyright © 2021 Vincent Chu. Course materials based on and used with permission from Professor Hamzeh Roumani.

VERSION

v1.0 14 September 2021

Copyright © 2021 Vincent Chu. Course materials based on and used with permission from Professor Hamzeh Roumani.

••••••

ACKNOWLEDGMENTS

THANKS TO:

- Hamzeh Roumani, who has shaped EECS-4413 into a leading hands-on CS course at EECS and who generously shared all of his course materials and, more importantly, his teaching philosophy with me;
- Parke Godfrey, my long-suffering Master's supervisor and mentor; and
- Suprakash Datta for giving me this opportunity to teach this course.

Copyright © 2021 Vincent Chu. Course materials based on and used with permission from Professor Hamzeh Rouman

PRINTABLE VERSION OF THE TALK

Download PDF

Copyright © 2021 Vincent Chu. Course materials based on and used with permission from Professor Hamzeh Roumani.

WHAT'S AN E-COMMERCE SYSTEM?

When you think about ecommerces and e-commerce systems, what comes to mind?



WEB TECHNOLOGIES IN VARIOUS APPLICATION DOMAINS:

Examples
E-commerce (e.g.: Amazon and eBay, PayPal, online transactions and trac
Online banking and investments
Web information and search engines, ebooks, Wikipedia; social networkir
Online gaming, music, and film in the home, user-generated content (e.g.
Health informatics, on online patient records, monitoring patients.
E-learning, virtual learning environments; distance learning.
GPS in route finding systems, map services: Google Maps, Google Earth.
The Grid, as an enabling technology for collaboration between scientists.
Sensor technology, Internet-of-things (IoT), to monitor earthquakes, flood
Seamless integration of physical systems and computing: self-driving care

ling)

ng: Facebook and MySpace. : YouTube, Flickr).

ls, or tsunamis.

s, autonomous drones, etc.

E-COMMERCE SYSTEMS

Business applications designed to be used for transactions over the web.

- Include electronic shops and auctions
- Examples: Amazon.com, ebay.com, Shopify, photo.net, Uber

Common functionality (client view):

- User management
- On-line transactions
- Payment
- Delivery of a product, service
- Multi-media
- Mobile support
- Social network support (forums, discussion groups)
- Analytics (recommender systems)

DON'T MISS THE FOREST

It's easy to get lost in the specifics and particularities of one technology or framework and forget the broader concepts that apply to all these tools and connect them all together.

Our focus should be on the broad concepts and not the technology or framework specifics.



Copyright © 2021 Vincent Chu. Course materials based on and used with permission from Professor Hamzeh Roumani.

FOR THE TREES

It's easy to get lost in the specifics and particularities of one technology or framework and forget the broader concepts that apply to all these tools and connect them all together.

Our focus should be on the broad concepts and not the technology or framework specifics.



13 September 2021













kubernetes



FOR THE TREES

It's easy to get lost in the specifics and particularities of one technology or framework and forget the broader concepts that apply to all these tools and connect them all together.

Our focus should be on the broad concepts and not the technology or framework specifics.











HIGHLIGHTS

MICROSERVICE ARCHITECTURE

- Move away from monolithic server-side applications.
- Scalable and platform-agnostic.

THE CLOUD STACK

- Amazon Web Services, Google Cloud, Microsoft Azure, etc.
 - Infrastructure-as-a-Service (IaaS),
 - Platform-as-a-Service (PaaS),
 - Software-as-a-Service (SaaS), and
 - Function-as-a-Service (FaaS).

RICH MVC CLIENT-SIDE FRAMEWORKS

- Full view migration from the server-side to the client-side.
- Single-page application and page modularity: Angular, React, VueJS, etc.

CROSS-CUTTING THEMES

- Scalability
- Security
- Telemetry

side to the client-side. odularity: Angular, React,

THIS COURSE IS ABOUT THE FOREST

The key pedagogy of this course is to emphasize the concepts, not the technologies, and learning by doing.

This course is not about becoming a proficient Tomcat developer, a Node developer, or a developer of some other specific server platform. It is not about proficiency in Java EE or some other programming language or specifically about using Angular or React or some other client-side platform. The idea is not to learn by memorizing concepts but by building actual services and actual client apps.



Copyright © 2021 Vincent Chu. Course materials based on and used with permission from Professor Hamzeh Roumani.

TOPICS we will cover 5 topics (APPROX. 2 WEEKS ON EACH)

1. MICROSERVICES

First, we will look at the microservice architecture. Define it and discuss the pros and cons versus the traditional monolithic architecture. Then, we will communicate to a different microservices and build our own in Java.

2. WEB SERVICES & TOMCAT

Web 1.0, circa 2000

Then, we will explore the server components for building a web application. The web server, session management, authentication, and server-side templating with the Tomcat server and the JSP templating language.

TOPICS we will cover 5 topics (APPROX. 2 WEEKS ON EACH)

3. SINGLE-PAGE APPLICATIONS & AJAX

Web 2.0, circa 2005

Asynchronous JavaScript and XML, or AJAX, allows us to uses XML or JSON to send and retrieve data from a server asynchronously (in the background) without interfering with the display and behaviour of the existing page.

4. NODEJS AND RESTFUL APIS

Web 2.0, circa 2010

JavaScript, everywhere! We will build lightweight NodeJS server-side web services. Decoupling presentation from the server and delegating layout and templating to the client while providing RESTful APIs to the client to callback to the server to request assets, app data and perform server-side actions.

TOPICS we will cover 5 topics (APPROX. 2 WEEKS ON EACH)

5. ANGULAR & MODERN SPA FRAMEWORKS

Web 2.0, circa 2015

Putting it all together!

Finally, we will upgrade our frontend client to use a modern (2nd-generation) single page application framework, Angular.

For more details about the topics covered and the expected learning outcomes, please refer to the Course Outline on the course website.

SCHEDULE

	Торіс	Dates	Labs
1.	Microservices	12-25 Sep	Project A
2.	Web Services & Tomcat	26 Sep - 09 Oct	Project B
	Reading Week	10 - 16 Oct	
	Test 1	21 Oct	
3.	SPAs & AJAX	24 Oct - 06 Nov	Project C
4.	NodeJS & RESTful APIs	07 - 20 Nov	Project D
	Test 2	25 Nov	
5.	Angular	28 Nov - 07 Dec	Project E
	Final Exam	09-23 Dec	

For more details, see: Schedule.

ASSESSMENTS

Component	Weight	When
Participation	5%	Throughou
Labs (5)	5% (1% each)	Before the
Test 1	25%	Thursday,
Test 2	25%	Thursday,
Final Exam	40%	In exam pe

For more details, see: Assessments & Grades.

Copyright © 2021 Vincent Chu. Course materials based on and used with permission from Professor Hamzeh Roumani.

Π

next lab or test

21 October

25 November

eriod

TEACHING APPROACH

PROF. ROUMANI'S **TWO** PRINCIPLES:

LEARNING BY DOING

- What Before HOW.
- I do, I understand.
- We learn by building stuff (rather than talking about building stuff), but with diminishing (or easing) scaffolding.

AS YOUR INSTRUCTOR

My commitment to you is to follow these principles as closely as best as I can. My lectures will be structured, as such:

- ~40 minutes of slides.
- ~50 minutes of coding examples.
- ~20 minutes of questions.

LEARNING BY ABSTRACTION

- Don't mix levels (by switching while teaching).
- We learn by choreographing the exposure. While some exposure to one level below the target is inducive to "curiosity" and conducive to "deep understanding".

ABOUT YOU

- Attendance (both lectures and labs)
- Take notes
- Do the Projects
- Ask questions (don't be shy)
- Use the forum (discuss with others)

This slide is intentionally left blank.

Return to Course Page.

Copyright © 2021 Vincent Chu. Course materials based on and used with permission from Professor Hamzeh Roumani.