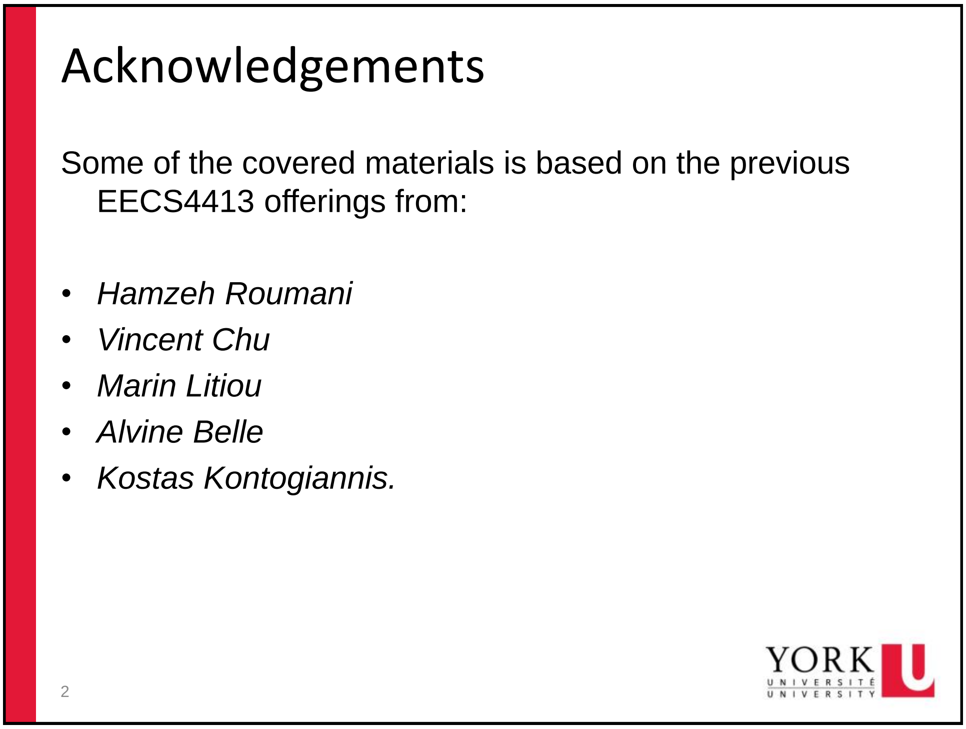
LE/EECS4413 B

Building E-commerce Systems

W 2024

Jan 08, 2024 Lecture 1.

1



Acknowledgements

Some of the covered materials is based on the previous EECS4413 offerings from:

- *Hamzeh Roumani*
- *Vincent Chu*
- *Marin Litiou*
- *Alvine Belle*
- *Kostas Kontogiannis.*

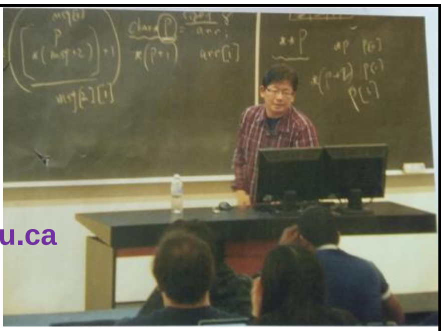
2

overview

- course outline (bird’s-eye view)
  - what this course is about
- logistics
  - course organization
  - Labs, tests, SMQ, projects..
  - grading scheme, etc.
- introduction to e-commerce system
  - High level view
  - layering principle, internet/web,
  - Client side (review):HTML/CSS/JS

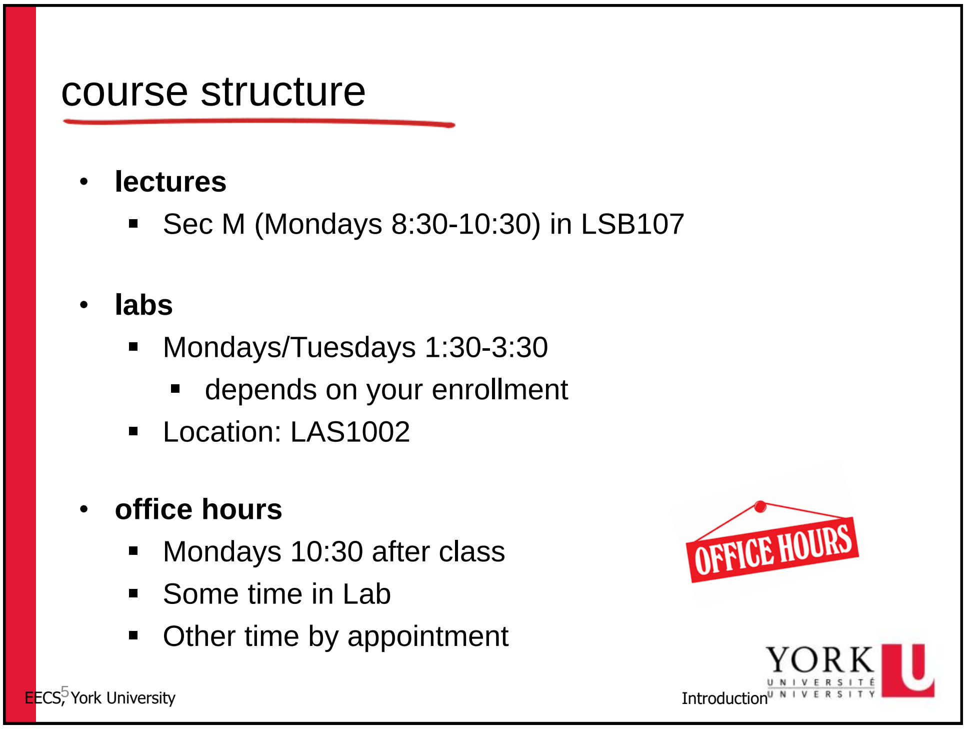
The instructor

- Dr. Hui Wang
  - Email: **huiwang@eecs.yorku.ca**
    hui.wang@yorku.ca



Don’t leave eClass messages

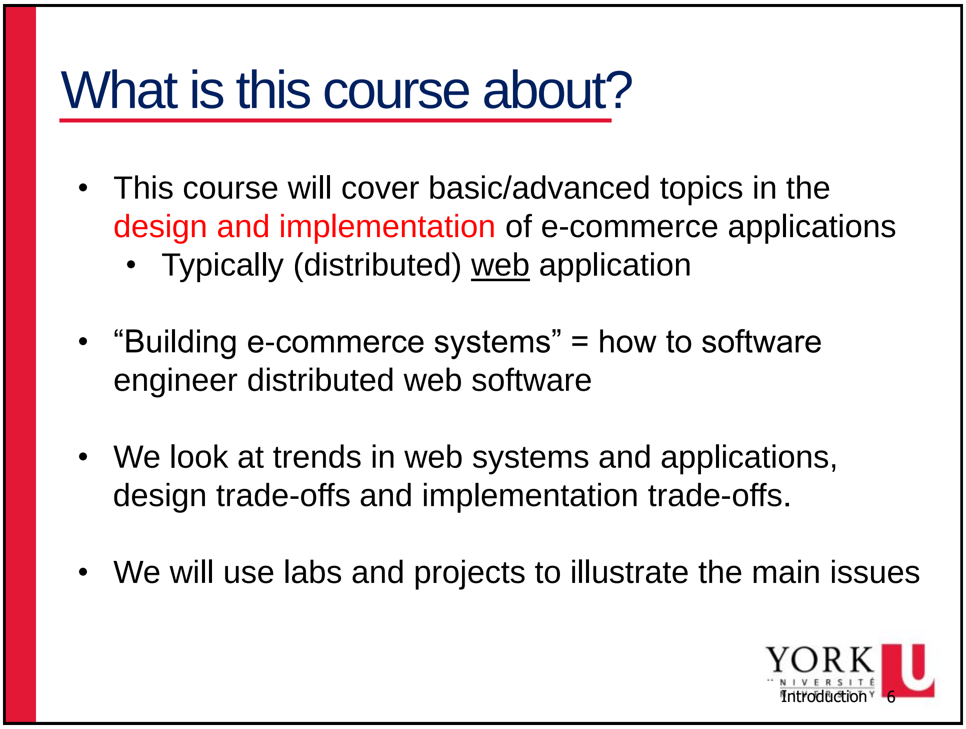
- Usually teach EECS1012, 2030, 2031, 4413
- Worked as software developer in web applications (ERP)



course structure

- **lectures**
  - Sec M (Mondays 8:30-10:30) in LSB107
- **labs**
  - Mondays/Tuesdays 1:30-3:30
    - depends on your enrollment
  - Location: LAS1002
- **office hours**
  - Mondays 10:30 after class
  - Some time in Lab
  - Other time by appointment

OFFICE HOURS



What is this course about?

- This course will cover basic/advanced topics in the design and implementation of e-commerce applications
  - Typically (distributed) web application
- "Building e-commerce systems" = how to software engineer distributed web software
- We look at trends in web systems and applications, design trade-offs and implementation trade-offs.
- We will use labs and projects to illustrate the main issues

Course learning outcomes

Upon completion of the course, students are expected to be able to

1. Become familiar with the various components that make up the web app and how these components interact with each other

2. Develop the expertise required to design and implement a whole web application consisting of a session management, database access, and analytics on the server side, and page formatting and interactivity on the client side

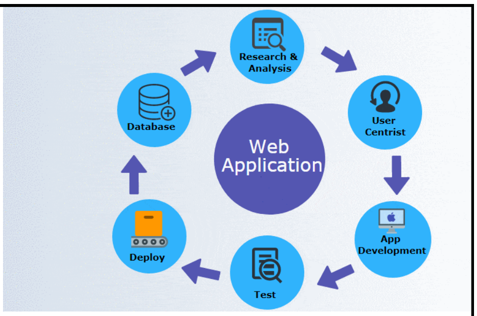
3. Develop the skills required to build restful web services that interact with Ajax powered client apps using a variety of transport protocols for data transfer

4. Learn and be able to comply with the best design practices and design patterns to implement a code that is maintainable, scalable, fosters interoperability, and minimizes exploitable vulnerabilities

5. Develop the expertise needed to implement complex applications collaboratively by leveraging abstractions and APIs, naming conventions, documentation, and organization

6. Compare and contrast existing frameworks and approaches and develop an insight into the various forces that are driving the current web trends

Course scope



- Software engineering process
  - Requirements
  - **Architecture and Design**
  - **Implementation**
  - Testing
  - Deployment
  - Maintenance and Operations

- We focus on
  - **Architecture and Design**
  - **Implementation**
    - “lower and medium level”: Socket, LAMP, Servlet, JSP

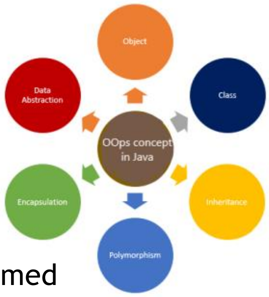
- We also touch (but is not the main focus)
  - deployment
  - advanced tools: e.g., Angular, React, Spring boot

- Web systems are distributed software
  - We study many heterogenous components, in different languages (focus on Java) but also PHP, JS/jQuery
  - We study communication patterns among components

Main topics (tentative)

- Web App Architecture. Preliminary knowledge/Review
  - Client side: HTML, CSS, JavaScript
  - Java (cmd, thread, serialization), UML, design patterns
- Client-Server, low level: socket programming
- Web applications (server side)
  - LAMP/CGI
  - Java Servlets
  - JSP, JavaBean, MVC pattern
  - SQL, Database access: JDBC.
  - More: listeners, filters, Ajax, JSON
- Web (RESTful) services, micro services
- Advanced topics: More design patterns, performance, security
- Other Advanced topics (tutorials):
  Deployments: Docker container, Node JS, React, Angular, Springboot, JPA

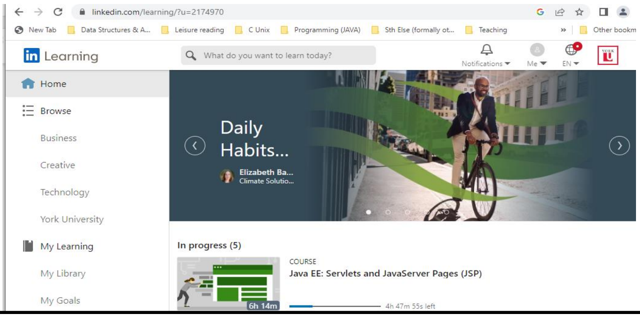
Expected Background knowledge



- Official pre-requisite (only): LE/EECS2030
- Good knowledge of OOP, Core Java is assumed
  - Inheritance, polymorphism, interfaces, abstract classes
  - Collections…
- Familiarity with basic upper networking protocols, socket programming, HTML, CSS,JavaScript, and SQL are helpful/expected but not required (can catch up easily)
- Now, let me know your backgrounds…

course resources

- eClass page *****
  - primary means of all communication: course lectures, lab instructions, online quizzes & uploading assignments, announcements & discussion forum, deadlines and evaluation, etc.
- web resources
  - no specific textbooks
  - we will use many web resources
  - Tons of them.. E.g. w3schools, LinkedIn learning



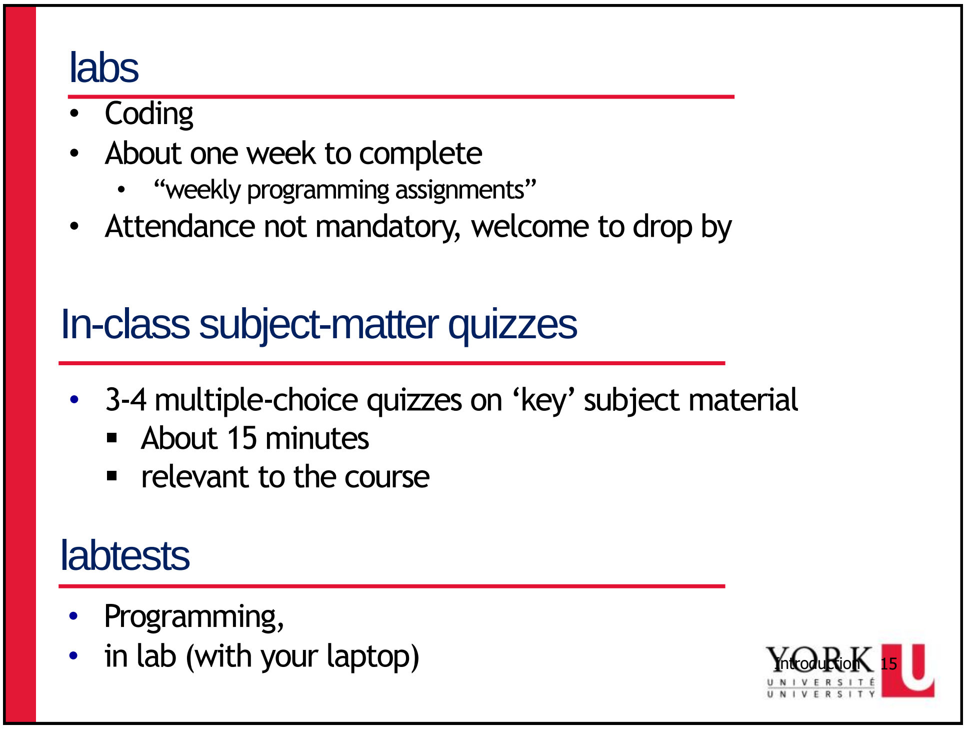
Evaluation tentative

- in this journey, you have
  - 7 labs some are graded   12%
  - 3-4 In-class subject-matter quizzes   10%
  - 2 lab tests   20-25%
  - 1 project (in teams of 3 or 4)   25~30%
  - final exam   30%

  - *the dates and further details will be available in the near future*

- letter grade computed using normal YorkU mapping

≥ 90	≥ 80	≥ 75	≥ 70	≥ 65	≥ 60	≥ 55	≥ 50	≥ 40	< 40
A+	A	B+	B	C+	C	D+	D	E	F



labs

- Coding
- About one week to complete
  - “weekly programming assignments”
- Attendance not mandatory, welcome to drop by

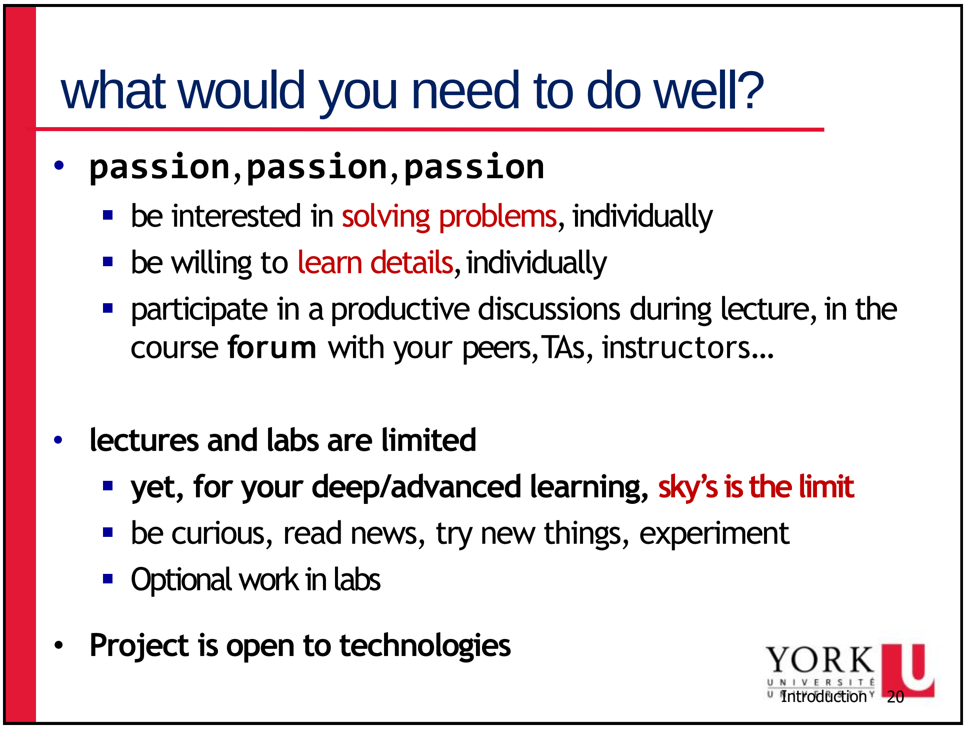
In-class subject-matter quizzes

- 3-4 multiple-choice quizzes on ‘key’ subject material
  - About 15 minutes
  - relevant to the course

labtests

- Programming,
- in lab (with your laptop)

15

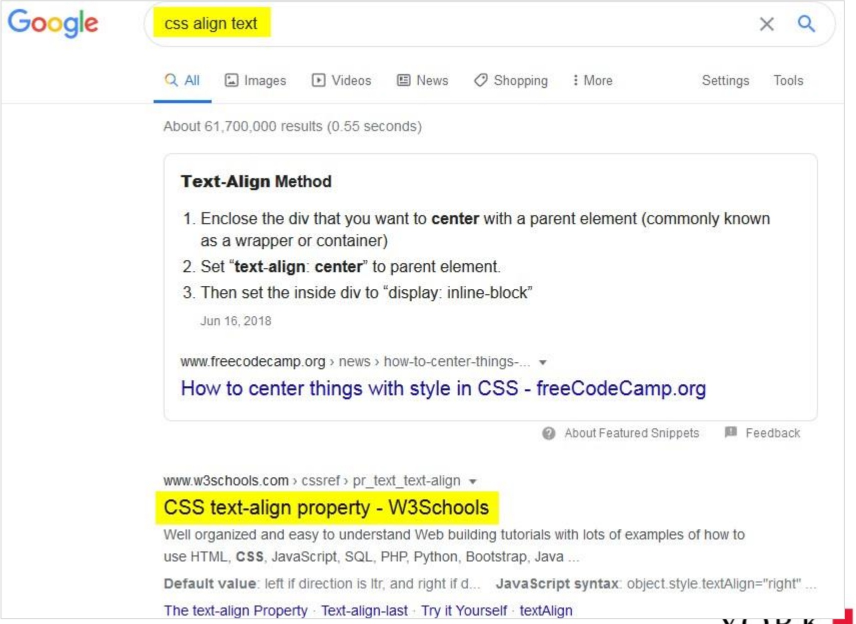


what would you need to do well?

- **passion,passion,passion**
  - be interested in solving problems, individually
  - be willing to learn details, individually
  - participate in a productive discussions during lecture, in the course **forum** with your peers, TAs, instructors…
- **lectures and labs are limited**
  - **yet, for your deep/advanced learning, sky’s is the limit**
  - be curious, read news, try new things, experiment
  - Optional work in labs
- **Project is open to technologies**

20

also, look it up



Google

css align text

About 61,700,000 results (0.55 seconds)

Text-Align Method

1. Enclose the div that you want to **center** with a parent element (commonly known as a wrapper or container)
2. Set "**text-align**: **center**" to parent element.
3. Then set the inside div to "display: inline-block"

Jun 16, 2018

www.freecodecamp.org › news › how-to-center-things-...

How to center things with style in CSS - freeCodeCamp.org

www.w3schools.com › cssref › pr_text_text-align

CSS text-align property - W3Schools

Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, **CSS**, JavaScript, SQL, PHP, Python, Bootstrap, Java ...

Default value: left if direction is ltr, and right if d... **JavaScript syntax**: object.style.textAlign="right" ...

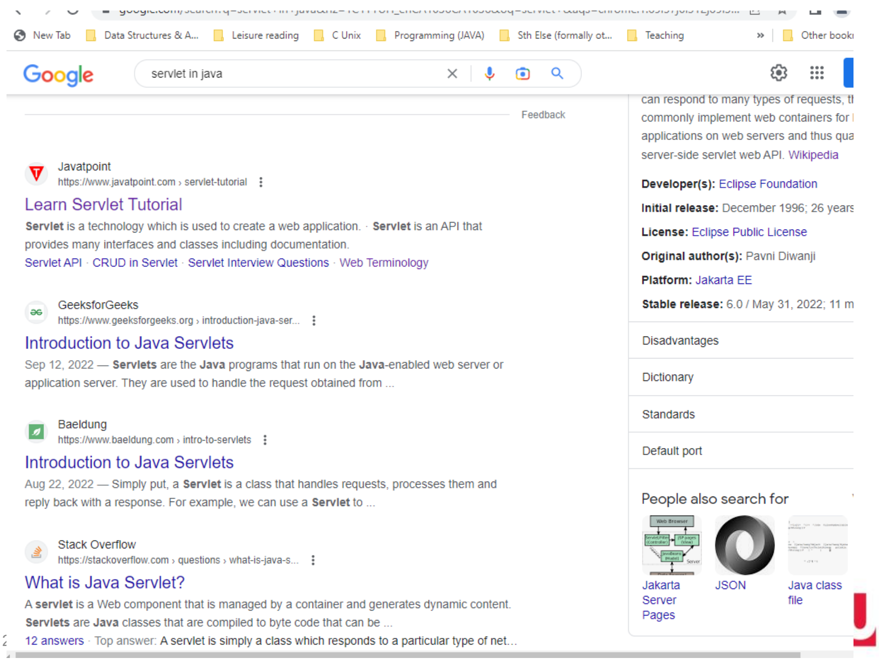
The text-align Property · Text-align-last · Try it Yourself · textAlign

YORK UNIVERSITÉ UNIVERSITY

23

23

also, look it up



Google

servlet in java

Javatpoint
https://www.javatpoint.com › servlet-tutorial

Learn Servlet Tutorial

Servlet is a technology which is used to create a web application. · **Servlet** is an API that provides many interfaces and classes including documentation.

Servlet API · CRUD in Servlet · Servlet Interview Questions · Web Terminology

GeeksforGeeks
https://www.geeksforgeeks.org › introduction-java-ser...

Introduction to Java Servlets

Sep 12, 2022 — **Servlets** are the **Java** programs that run on the **Java**-enabled web server or application server. They are used to handle the request obtained from ...

Baeldung
https://www.baeldung.com › intro-to-servlets

Introduction to Java Servlets

Aug 22, 2022 — Simply put, a **Servlet** is a class that handles requests, processes them and reply back with a response. For example, we can use a **Servlet** to ...

Stack Overflow
https://stackoverflow.com › questions › what-is-java-s...

What is Java Servlet?

A **servlet** is a Web component that is managed by a container and generates dynamic content. **Servlets** are **Java** classes that are compiled to byte code that can be ...

12 answers · Top answer: A servlet is simply a class which responds to a particular type of net...

can respond to many types of requests, tl commonly implement web containers for l applications on web servers and thus qua server-side servlet web API. Wikipedia

Developer(s): Eclipse Foundation

Initial release: December 1996; 26 years

License: Eclipse Public License

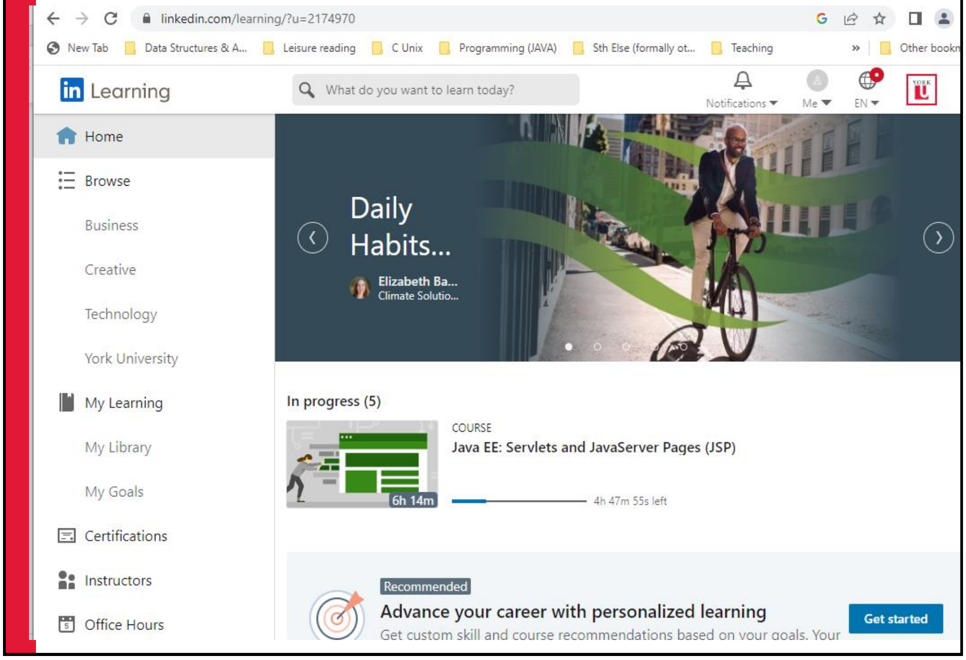
Original author(s): Pavni Diwanji

Platform: Jakarta EE

Stable release: 6.0 / May 31, 2022; 11 m

People also search for

24



linkedin.com/learning/?u=2174970

New Tab | Data Structures & A... | Leisure reading | C Unix | Programming (JAVA) | Sth Else (formally ot... | Teaching | » | Other bookm

Learning

What do you want to learn today?

Notifications | Me | EN

- Home
- Browse
  - Business
  - Creative
  - Technology
  - York University
- My Learning
  - My Library
  - My Goals
- Certifications
- Instructors
- Office Hours

Daily Habits...

Elizabeth Ba...
Climate Solutio...

In progress (5)

6h 14m

COURSE
Java EE: Servlets and JavaServer Pages (JSP)

4h 47m 55s left

Recommended

Advance your career with personalized learning

Get custom skill and course recommendations based on your goals. Your

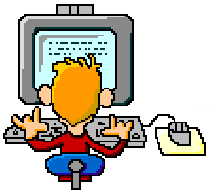
Get started

25

Useful suggestions

- Come to the lectures
- Watch videos, read the lecture notes!
  - Videos, Notes will be finalized shortly after class
- Do the labs on your own!
  - Discussion allowed only for labs. Not for others
  - Discussion != collaboration != sharing solutions

- Don’t be shy to ask for help
  - come to the lab session, office hour
  - eClass forum
  - email me (specify “EECS4413”)
- Practice, practice, and practice!



26

26

Academic Integrity

- Honesty, originality and academic integrity matters to us.
- Plagiarism and cheating are not tolerated!
- Read https://lassonde.yorku.ca/academic-integrity for the consequences. Read slides on eClass

- Weekly labs: discussion ✓
    - Discussion != sharing solutions
- SMQ, tests, exam:
    - Discussion not allowed ✗

- Any questions so far?

now let’s move on to concepts related to e-commerce web application design

Agenda

- What is E-commerce?
- What are the typical features of E-commerce systems?
- What are the challenges facing E-commerce system?
- What is the typical architecture of an E-commerce system?
- Which technologies can be used to develop E-commerce systems?

What is e-commerce?

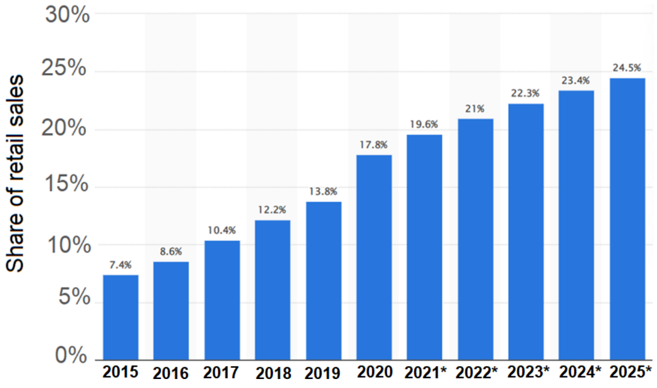
- E-Commerce started in the 1960s
- It has quickly evolved especially with the emergence of smartphones
- It is also called e-Business or electronic business
- Watch this video to find out more.
- Read this article

What is E-commerce? B2B and B2C

Source: https://www.techtarget.com/searchcio/definition/e-commerce

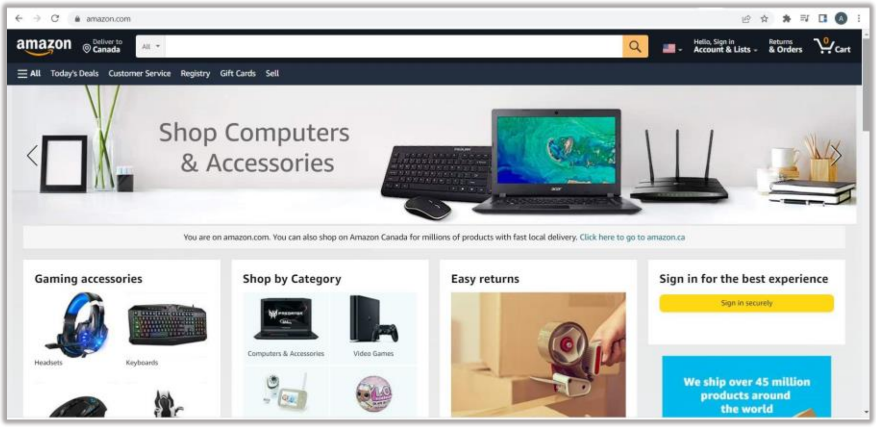
E-commerce worldwide sales/projections: 2015 to 2025

In 2022, E-commerce sales are expected to represent 21% of the total retail sales worldwide.



Source: https://www.statista.com/statistics/534123/e-commerce-share-of-retail-sales-worldwide/

Are you familiar with Amazon.com?

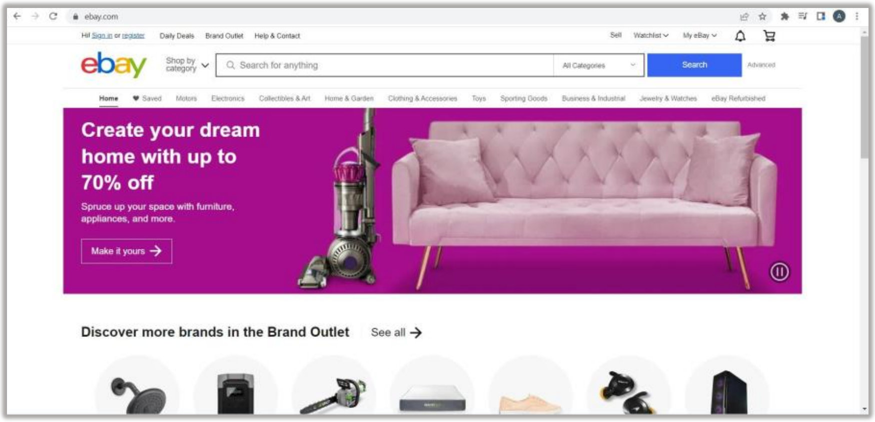


You are on amazon.com. You can also shop on Amazon Canada for millions of products with fast local delivery. Click here to go to amazon.ca

Source: amazon.com

35

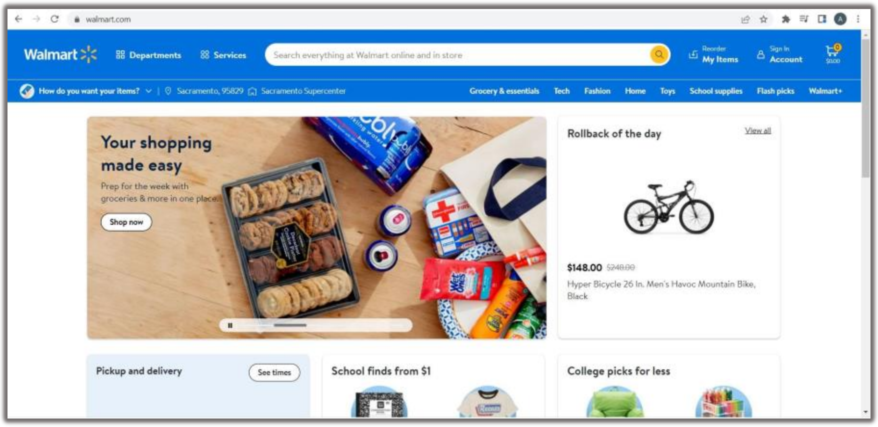
Are you familiar with ebay.com?



Source: ebay.com

36

Are you familiar with walmart.com?



Your shopping made easy

Prep for the week with groceries & more in one place.

$148.00 $248.00

Hyper Bicycle 26 In. Men's Havoc Mountain Bike, Black

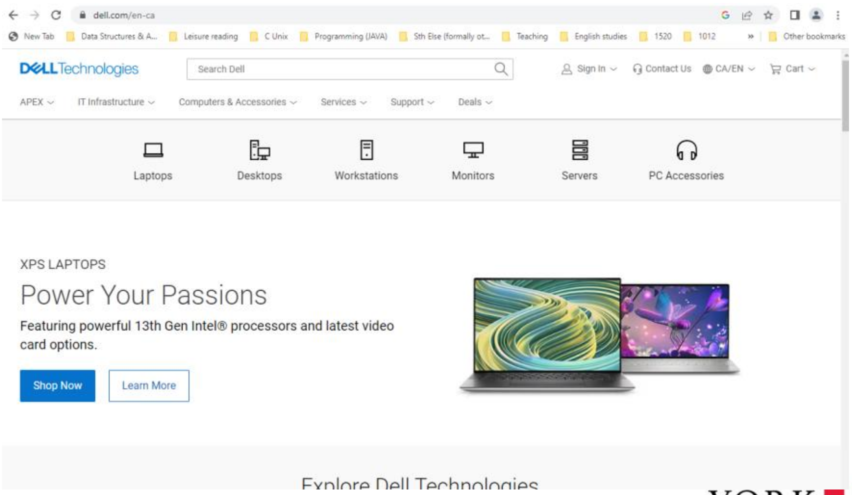
Source: Walmart.com

8

YORK UNIVERSITÉ UNIVERSITY

37

Are you familiar with dell.ca?



XPS LAPTOPS

Power Your Passions

Featuring powerful 13th Gen Intel® processors and latest video card options.

Explore Dell Technologies

YORK UNIVERSITÉ UNIVERSITY

38

Exercise (3-5 mins)

Browse the following e-commerce sites: **amazon.com**, **walmart.ca**, and **ebay.com** and **dell.ca.**

- What are the five common functions (or use cases) of these sites?
- Are there any features that seem to miss from these e-commerce sites?
- Can you name three common technologies (e.g., languages, protocols) used on these sites?
- How do these sites proceed to accommodate several users simultaneously?
- How do these sites proceed to achieve security and availability?
- Which of them supports the most effective search strategy?

9

39

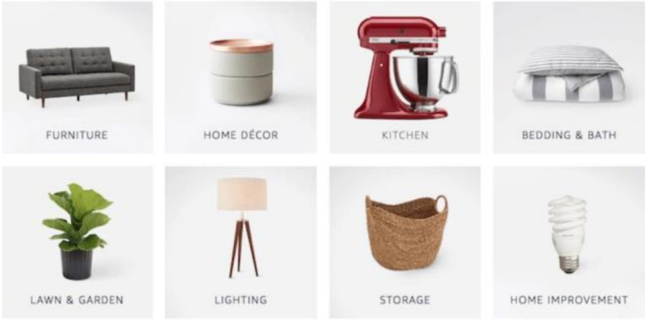
E-commerce systems

- Also called web systems.
- An E-commerce system is a website that allows accessing an online store to browse, purchase or sell tangible goods, digital products or services.
- By involving the electronical transfer of data and funds between two or more parties, E-commerce systems support online shopping.
  - No need to go anywhere physically: with just a few clicks, you can shop anything from anywhere (e.g., living room), anytime.
  - You just need a device connected to the Internet.

10

40

E-commerce systems: the word closer to your door…



You can find a variety of products on e-commerce websites: clothes, cars, gardening materials, food, etc.

41

Examples of e-commerce systems

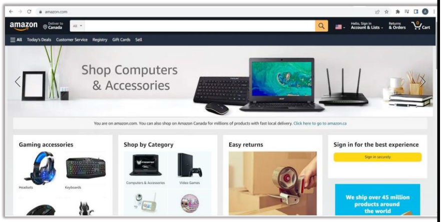
- Best Buy
- Amazon
- Dell
- Adidas
- Chewy
- Alibaba
- eBay
- Etsy
- Overstock

- Home Depot
- Bricks
- Costco
- Macy's
- Newegg
- Rakuten
- Walmart Marketplace
- Wayfair
- Craiglist
- ….
- ….

42

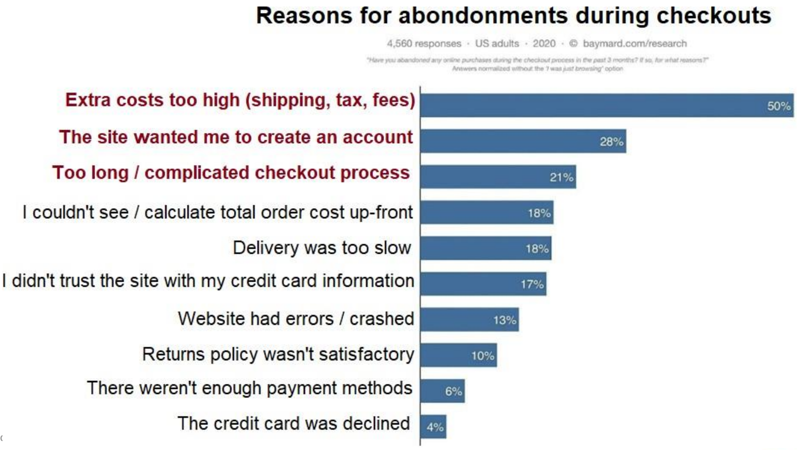
Main functionality of e-commerce systems

Their core functionality (client view) usually include:


amazon.com
amazon
Deliver to Canada
All
Hello, Sign in Account & Lists
Returns & Orders
0 Cart
All
Today's Deals
Customer Service
Registry
Gift Cards
Sell
Shop Computers & Accessories
You are on amazon.com. You can also shop on Amazon Canada for millions of products with fast local delivery. Click here to go to amazon.ca
Gaming accessories
Headsets
Keyboards
Shop by Category
Computers & Accessories
Video Games
Easy returns
Sign in for the best experience
Sign in securely
We ship over 45 million products around the world

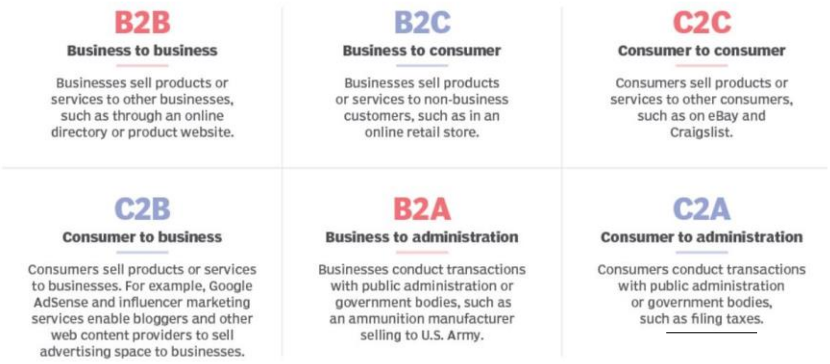
- User management
- Catalog of products, information
- Online transactions
- Checkout/payment (e.g., Credit card, PayPal)
- Shipping product/delivery of service
- Mobile-friendliness
- Multi-media
- Live chat
- Social network support (e.g., forums, discussion groups)
- Analytics/promotion (e.g., recommender systems).

Common functionality: watch out!



Sc

Types of e-commerce



B2B
Business to business
Businesses sell products or services to other businesses, such as through an online directory or product website.

B2C
Business to consumer
Businesses sell products or services to non-business customers, such as in an online retail store.

C2C
Consumer to consumer
Consumers sell products or services to other consumers, such as on eBay and Craigslist.

C2B
Consumer to business
Consumers sell products or services to businesses. For example, Google AdSense and influencer marketing services enable bloggers and other web content providers to sell advertising space to businesses.

B2A
Business to administration
Businesses conduct transactions with public administration or government bodies, such as an ammunition manufacturer selling to U.S. Army.

C2A
Consumer to administration
Consumers conduct transactions with public administration or government bodies, such as filing taxes.

Source: https://www.techtarget.com/searchcio/definition/e-commerce

YORK U UNIVERSITÉ UNIVERSITY

45

E-commerce: the good, the bad and …the shopping cart

- + 24/7 Availability
- + Access to a larger (international) market
- + Speed (no waiting line)
- + Less expensive
- + Product customization/recommendations
- + Access to detailed product information
- + Ease the retargeting/remarketing of customers
- + Easy access (you don't get lost in the wrong street when trying to find the online store)
- + Etc.

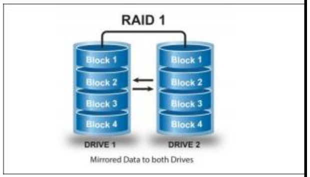
- -Relative lack of customer service
- -Need to wait days to receive the ordered product(s)
- -Security issues (e.g., hacking)
- -Websites crashes
- -Need to have an internet connection
- **-Not possible to try before buying**
- -Complexity in taxation, and regulation.

YORK U UNIVERSITÉ UNIVERSITY

46

Challenges in developing e-commerce systems

- **Speed and performance**: load time should be significantly low
- **Scalability:** need to develop a highly scalable website that remains efficient when the number of users and resources increases significantly
- **Security**: need to protect sensitive information
  - Confidentiality (against disclosure),
  - Integrity, data persistence (against alteration)
  - availability (against interference, such as DOS)



- **Failure tolerance**: if a part of the system fails, the system should still function
  - Recovery from failures: the software should be able to "roll-back" to a stable and know state
  - Redundancy: services can tolerate failures by suing redundant components; most web applications run on clusters

47

Challenges in developing e-commerce systems

- **Distribution**: need to handle distributed software that is heterogeneous
  - software made of multiple components running on different computers in different geographic areas
  - Distributed software uses many architecture patterns, not encountered in monolithic applications
    - Client server, multi-tier, distributed MVC, P2P ..
  - Distributed software is heterogeneous: computers, networks, protocols, data, programming languages, operating systems
    - Middlewares hide the heterogeneity of the underlying software and hardware: web and http, web services
    - Communication protocols are as important as components, hence the importance of HTTP/REST/JSON/XML

48

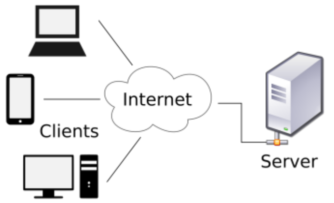
Agenda

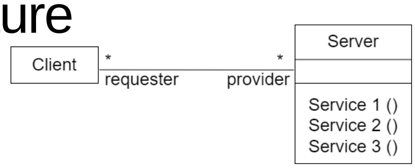
- What is E-commerce?
- What are the typical features of E-commerce systems?
- What are the challenges facing E-commerce system?
- **What is the typical architecture of an E-commerce system?**
- **Which technologies can be used to develop E-commerce systems?**

3

49

Client/Server architecture





A Client/Server architecture follows a Client/Server model:

- Also known as host, a **Client** is usually a device (e.g., computer, laptop, workstation), that needs to access a service or some information.
- A **Server** can be a physical device (e.g., a rack server, a virtual server) that can provide access to the data and services that the client needs.

A Client/Server architecture consists of a network application:

- It divides tasks and workloads between clients and servers that reside on the same system or are linked by a computer network.

17

Users only interact with the client. Client calls on the Server, which performs some service and returns the result -- usually immediately

50

Traditional architecture of a web/E-commerce system

The traditional architecture of E-commerce system is an ***n-tier*** architecture and usually a tree-tier architecture.

- **The presentation (client) tier**: is the part that is presented to the customer. It consists of the user interface and communication layer of the architecture.
  - The customer uses it to interact with the website on the frontend, and the application collects data and processes requests from the backend.
- **The business tier**: it is the central part of the application. It uses business logic, a specific set of business rules, to collect and process information.
  - It is also able to add/delete/change information in the data tier.
- **The data tier**: third tier allowing to persist data and process requests. That data may be stored using a <u>relational database</u>.

18

YORK U
UNIVERSITÉ
UNIVERSITY

51

principle of layering

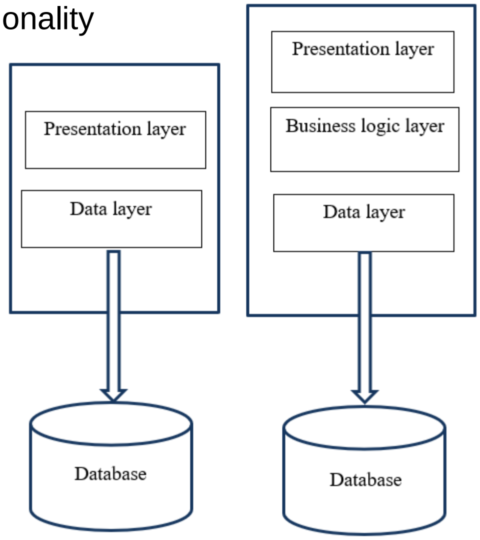
- dividing the application to 2+ groups/tiers/classes
  - that are functionally or logically related
  - conquering separately
- such that
  - each layer demonstrates (max) cohesion
  - dependency among classes is minimized
- **advantages:**
  - modularity, maintainability, reusability
- **disadvantages:**
  - reduced performance (some aspects)

Online Check-in
CHECK IN
Check-in
Baggage Check-in
Security
Immigration
GATE
Boarding Gate

52

2-3 layer architecture

- simple application functionality



Presentation layer

Data layer

Database

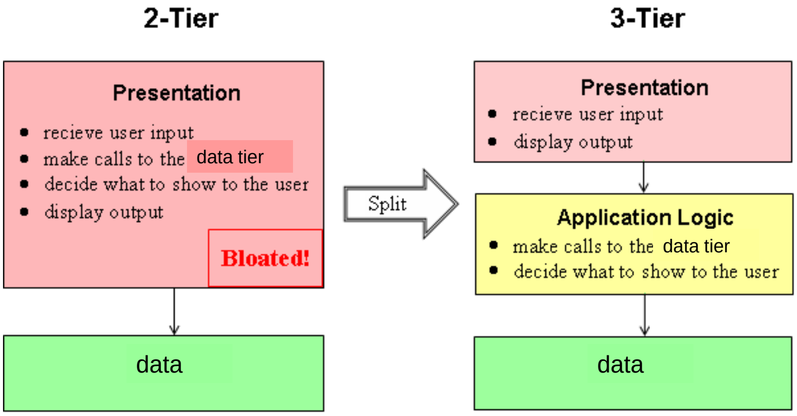
Presentation layer

Business logic layer

Data layer

Database

53



2-Tier

Presentation

- recieve user input
- make calls to the data tier
- decide what to show to the user
- display output

Bloated!

data

Split

3-Tier

Presentation

- recieve user input
- display output

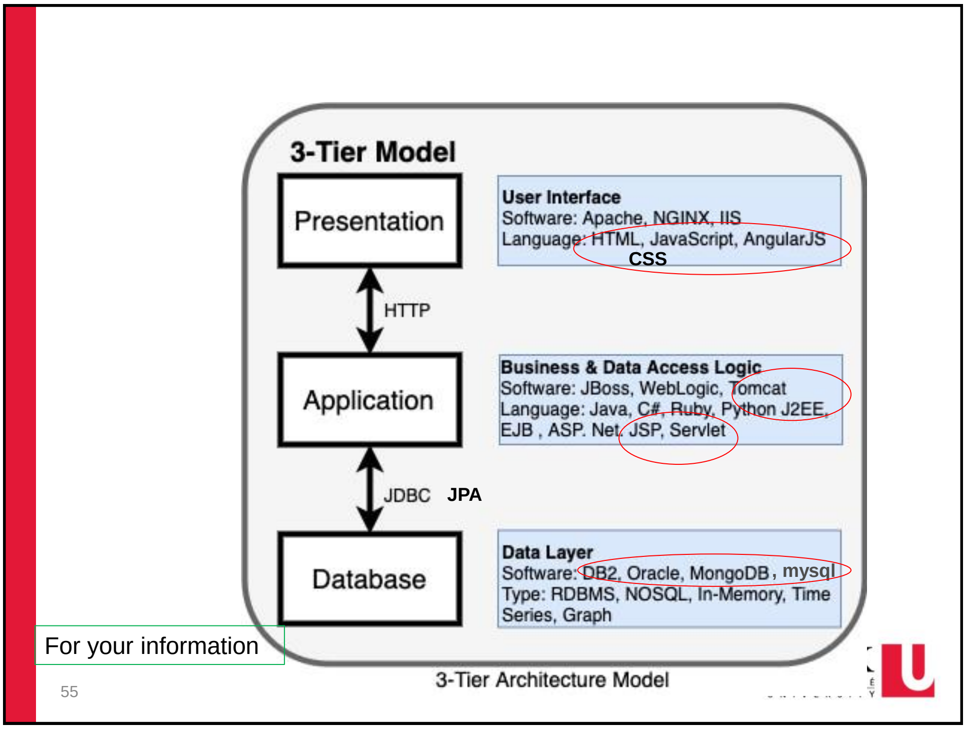
Application Logic

- make calls to the data tier
- decide what to show to the user

data

54

54



3-Tier Model

Presentation

HTTP

Application

JDBC **JPA**

Database

User Interface
Software: Apache, NGINX, IIS
Language: HTML, JavaScript, AngularJS
CSS

Business & Data Access Logic
Software: JBoss, WebLogic, Tomcat
Language: Java, C#, Ruby, Python J2EE, EJB , ASP. Net, JSP, Servlet

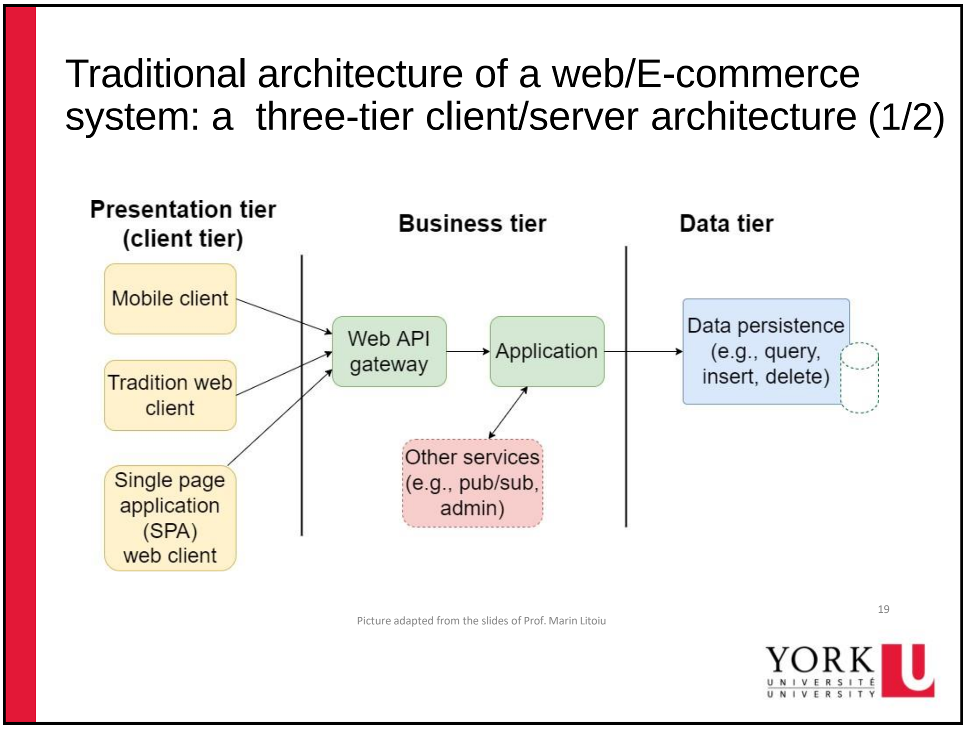
Data Layer
Software: DB2, Oracle, MongoDB , mysql
Type: RDBMS, NOSQL, In-Memory, Time Series, Graph

For your information

3-Tier Architecture Model

55

55



Traditional architecture of a web/E-commerce system: a three-tier client/server architecture (1/2)

Presentation tier (client tier)

Business tier

Data tier

Mobile client

Tradition web client

Single page application (SPA) web client

Web API gateway

Application

Other services (e.g., pub/sub, admin)

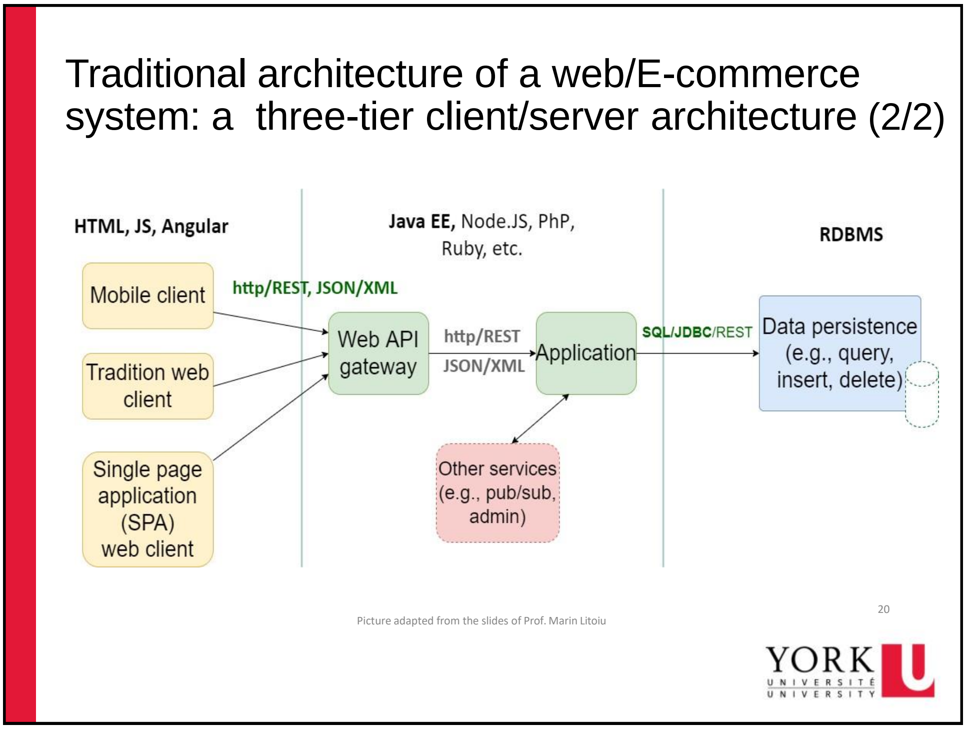
Data persistence (e.g., query, insert, delete)

19

Picture adapted from the slides of Prof. Marin Litoiu

YORK UNIVERSITÉ UNIVERSITY

58



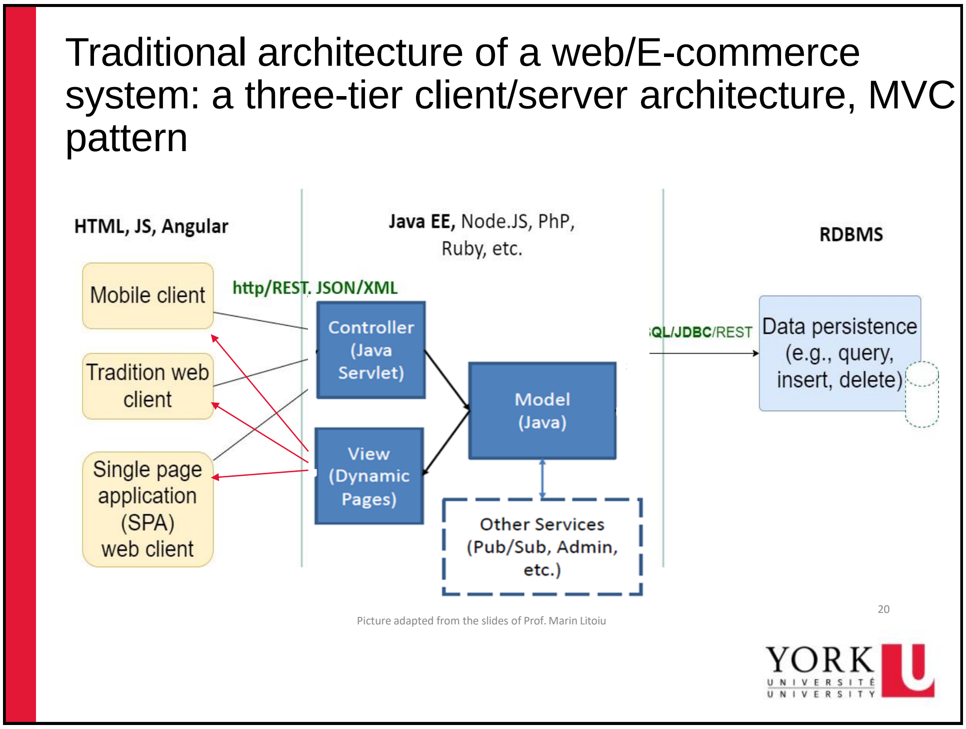
Traditional architecture of a web/E-commerce system: a three-tier client/server architecture (2/2)

HTML, JS, Angular
Java EE, Node.JS, PhP, Ruby, etc.
RDBMS
Mobile client
Tradition web client
Single page application (SPA) web client
http/REST, JSON/XML
Web API gateway
http/REST
JSON/XML
Application
SQL/JDBC/REST
Data persistence (e.g., query, insert, delete)
Other services (e.g., pub/sub, admin)

Picture adapted from the slides of Prof. Marin Litoiu

20

59



Traditional architecture of a web/E-commerce system: a three-tier client/server architecture, MVC pattern

HTML, JS, Angular
Java EE, Node.JS, PhP, Ruby, etc.
RDBMS
Mobile client
Tradition web client
Single page application (SPA) web client
http/REST. JSON/XML
Controller (Java Servlet)
View (Dynamic Pages)
Model (Java)
Other Services (Pub/Sub, Admin, etc.)
:QL/JDBC/REST
Data persistence (e.g., query, insert, delete)

Picture adapted from the slides of Prof. Marin Litoiu

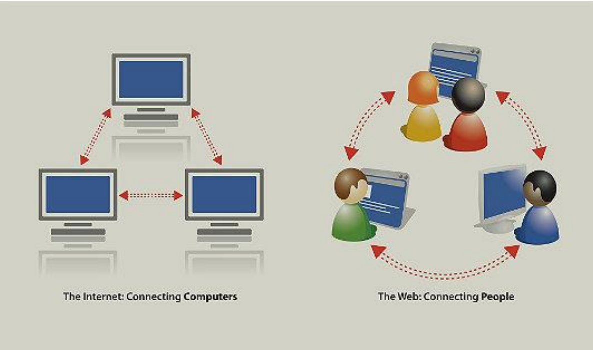
20

60

internet & services

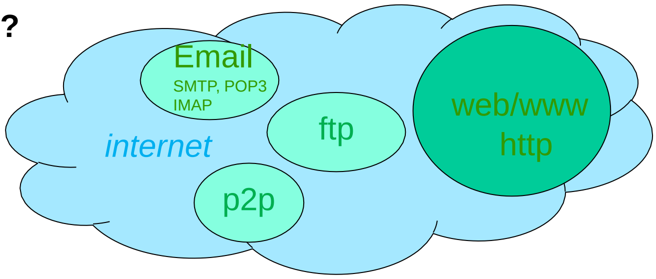
internet = www?

www ≠ internet
Internet: infrastructure
www: one of the services on internet

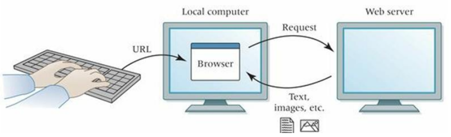

The Internet: Connecting Computers
The Web: Connecting People

internet & services

internet = www?

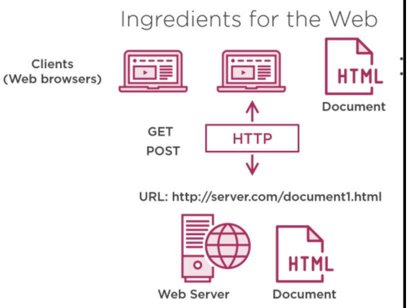


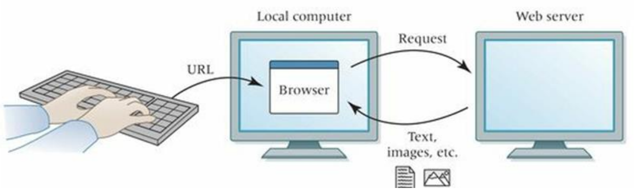
www = web

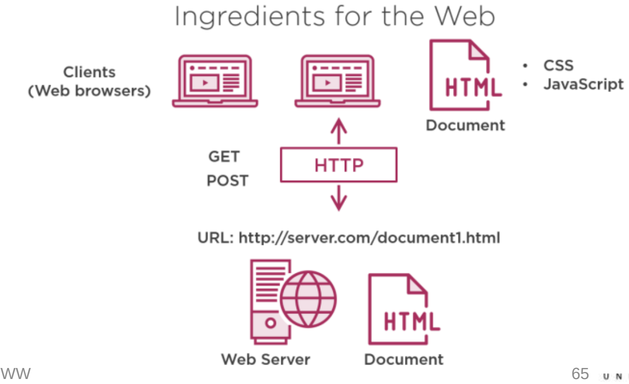


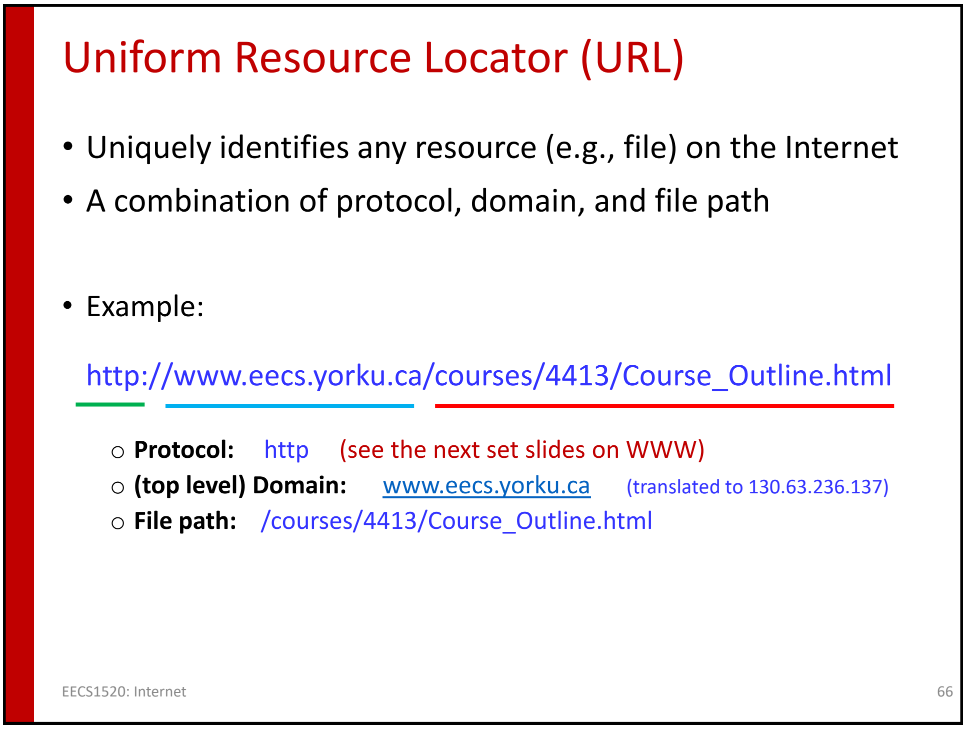
www: it's an information space system—based on request & response— with these features:

- **HTML**: to describe (hypertext) documents/pages
- **URL**: to uniquely locate a resource
- **HTTP**: to describe how requests & responses operate
- **web server**: computer to respond to HTTP requests
- **web browser**:to make HTTP requests frc and render/display the HTML document received









Uniform Resource Locator (URL)

- Uniquely identifies any resource (e.g., file) on the Internet
- A combination of protocol, domain, and file path

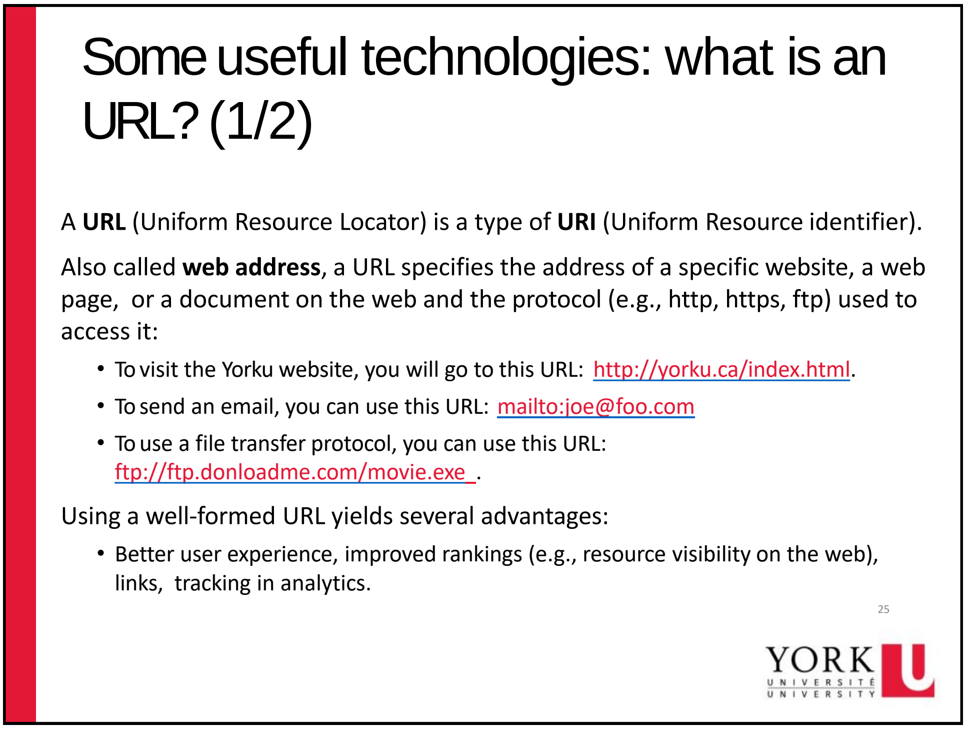
- Example:

http://www.eecs.yorku.ca/courses/4413/Course_Outline.html

  - **Protocol:** http (see the next set slides on WWW)
  - **(top level) Domain:** www.eecs.yorku.ca (translated to 130.63.236.137)
  - **File path:** /courses/4413/Course_Outline.html

EECS1520: Internet 66

66



Some useful technologies: what is an URL? (1/2)

A **URL** (Uniform Resource Locator) is a type of **URI** (Uniform Resource identifier).

Also called **web address**, a URL specifies the address of a specific website, a web page, or a document on the web and the protocol (e.g., http, https, ftp) used to access it:

- To visit the Yorku website, you will go to this URL: http://yorku.ca/index.html.
- To send an email, you can use this URL: mailto:joe@foo.com
- To use a file transfer protocol, you can use this URL: ftp://ftp.donloadme.com/movie.exe .

Using a well-formed URL yields several advantages:

- Better user experience, improved rankings (e.g., resource visibility on the web), links, tracking in analytics.

25

67

Some useful technologies: what is an URL? (2/2)

URL syntax:

scheme://subdomain.domain-name.domain-extension/path-to-resource?parameters

- **Scheme**: it specifies the email provider (e.g., **mailto**), standard protocol for transferring computer files (e.g., **ftp**), a protocol (e.g., **http** or **https**) used to access the resource
- **The colon** (:) **and two forward slashes** (//): used to separate the scheme from the rest of the URL
- **Sub-domain** (optional): consists of any words or phrases that come before a URL's first dot (e.g., **www**)
- **Domain name (hostname)**: it usually consists of a name specifying the location where a resource (e.g., a website) is located
- **Domain extension**: it is the bit following a website name (e.g., **.com**, **.org**, **.ca**, **.net**)
- **Path-to-resource** : reference to the folder structure of the website.
- **Parameters**: they are query strings or URL variables. They're the portion of a URL following a question mark (?). They can be used for several purposes (e.g., searching, filtering, translating).

https://webapp.eecs.yorku.ca/ltcloud?id=123

68

Some useful technologies: web browsers

Also known as browser, a **web browser** is a program that provides an interface that allows finding, accessing, displaying, and viewing websites.

It retrieves information from various resources on the web and allows displaying that information to the user.

That information is transferred thanks to **HTTP**, which specifies how text, images and videos are transmitted on the web.

Double clicking a browser (e.g., Chrome) icon installed on a computer launches that browser and allows searching that browser or typing a URL into its address bar.

- A browser also allows e-mailing, transferring files, using social media sites, and participating in online discussion groups, etc.

27

69

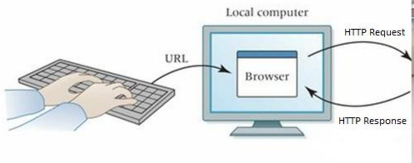
Some useful technologies: web servers

The web server stores web content (e.g., websites or information).

A user can request some of that content by using a browser, which is installed on the user's device (e.g., computer). That browser allows the user to visualize the requested web content.

Main technologies allowing to transfer information on the web, from servers to clients (computers of users):

- **HTML (Hypertext Markup Language)**
- **HTTP (Hypertext Transfer Protocol) / HTTPS (Hypertext Transfer Protocol Secure)**
- Web browsers.

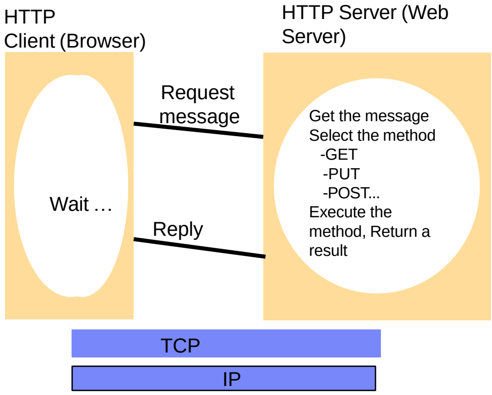


28

70

Some useful technologies: HTTP (Hyper Text Transfer Protocol)

- **Client sends a message (request); the server responds with another message (reply)**
- **The message includes the method name, the arguments, the results of the method and how data is represented**
- **http/1.0 was connectionless**
  - ○ The server closes its connection, release its resources after first request
- **http/1.1 connection-oriented**
  - ○ The server keeps the connection open; subsequent requests from the same client are served faster
- **http/2.0**: partially adopted
- **http/3**-in progress
- **The server listens (usually) on port 80**
- **The messages are delivered through the underlying TCP/IP protocol.**



HTTP Client (Browser)

HTTP Server (Web Server)

Wait …

Request message

Reply

Get the message
Select the method
-GET
-PUT
-POST...
Execute the method, Return a result

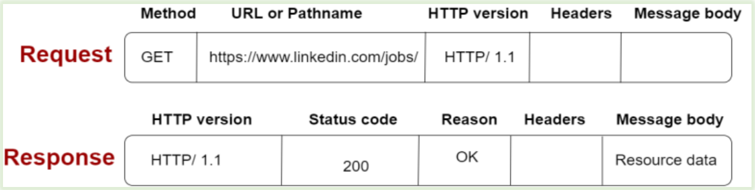
TCP

IP

29

71

Some useful technologies: HTTPrequest and reply messages



	Method	URL or Pathname	HTTP version	Headers	Message body
Request	GET	https://www.linkedin.com/jobs/	HTTP/ 1.1		

	HTTP version	Status code	Reason	Headers	Message body
Response	HTTP/ 1.1	200	OK		Resource data

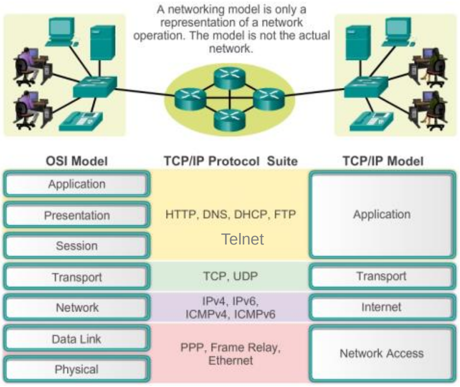
Methods

GET: requests a resource whose URL (locator) follows
POST: specify the URL of a resource that can process the data (used for forms, when you send user data to the server
HEAD: Identical to GET, does not return data, but info about the data (size, type, etc..)
PUT: store the data to the URL specified
DELETE: deletes the resource specified
TRACE: the server sends back the request message; used for diagnostic
OPTIONS: the server sends back the methods it supports and its special requirements
CONNECT: establishes a connection client server
PATCH: update the state of the server based on the request data

revisit

72

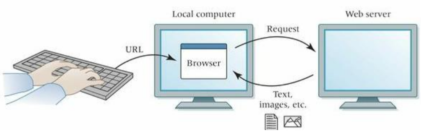
Under the hood – network layers



A networking model is only a representation of a network operation. The model is not the actual network.

OSI Model	TCP/IP Protocol Suite	TCP/IP Model
Application	HTTP, DNS, DHCP, FTP Telnet	Application
Presentation		
Session		
Transport	TCP, UDP	Transport
Network	IPv4, IPv6, ICMPv4, ICMPv6	Internet
Data Link	PPP, Frame Relay, Ethernet	Network Access
Physical		

Web server A computer set up to respond to requests for web pages



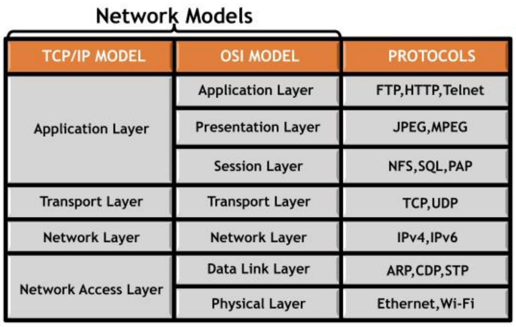
HTTP over TCP/IP

1. Reformat the URL entered as a valid HTTP request message.
   - If the server is specified using a host name (rather than an IP address), use DNS to convert this name to the appropriate IP address.
2. **Establish a TCP connection using the IP address of the specified web server.**
3. Send the HTTP request over the TCP connection and wait for the server’s response.
4. Display the document contained in the response. If the document is not a plain-text document but instead is written in a language such as HTML, this involves *rendering* the document

73

Under the hood – network layers

HTTP over TCP/IP

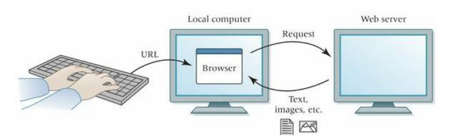


Network Models

TCP/IP MODEL	OSI MODEL	PROTOCOLS
Application Layer	Application Layer	FTP,HTTP,Telnet
	Presentation Layer	JPEG,MPEG
	Session Layer	NFS,SQL,PAP
Transport Layer	Transport Layer	TCP,UDP
Network Layer	Network Layer	IPv4,IPv6
Network Access Layer	Data Link Layer	ARP,CDP,STP
	Physical Layer	Ethernet,Wi-Fi

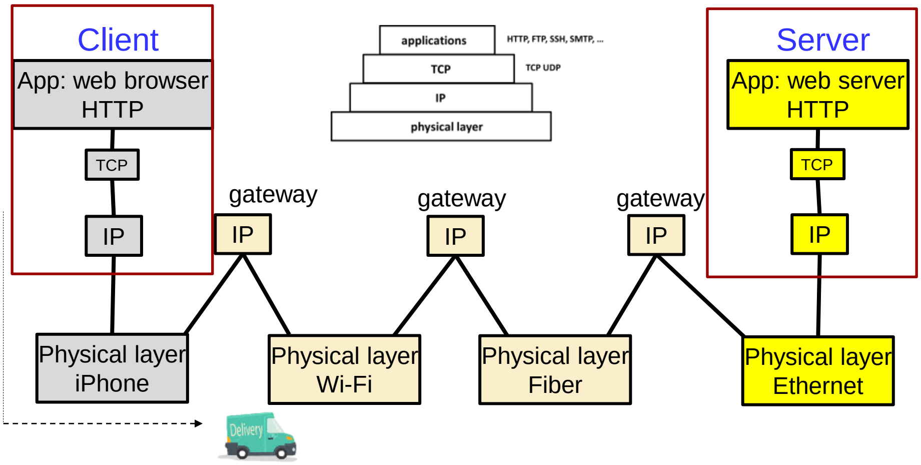
1. Reformat the URL entered as a valid HTTP request message.
   - If the server is specified using a host name (rather than an IP address), use DNS to convert this name to the appropriate IP address.
2. **Establish a TCP connection using the IP address of the specified web server.**
3. Send the HTTP request over the TCP connection and wait for the server's response.
4. Display the document contained in the response. If the document is not a plain-text document but instead is written in a language such as HTML, this involves *rendering* the document

Web server A computer set up to respond to requests for web pages

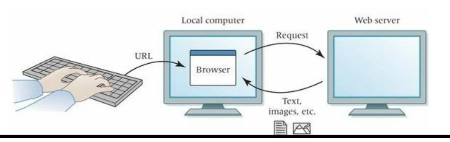


74

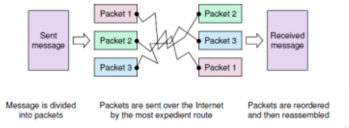
Information flow via TCP/IP



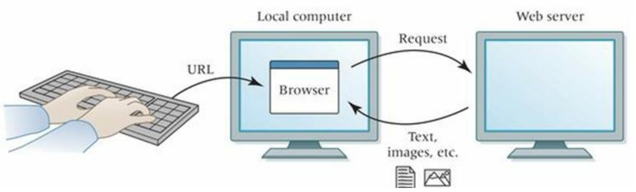
Web server A computer set up to respond to requests for web pages



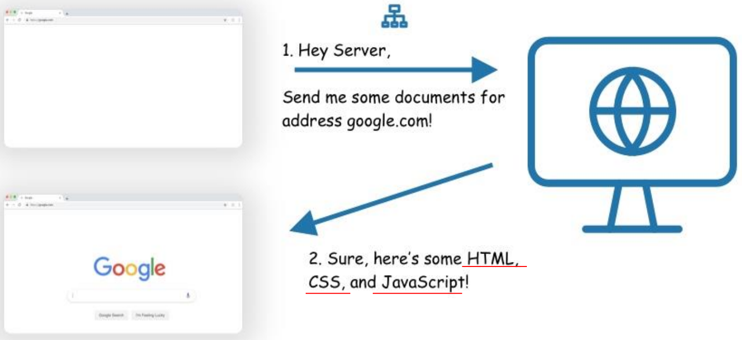
Internet


Sent message
Packet 1
Packet 2
Packet 3
Packet 2
Packet 3
Packet 1
Received message
Message is divided into packets
Packets are sent over the Internet by the most expedient route
Packets are reordered and then reassembled

75



The "Explain It Like I'm 5" version of how web browsers work

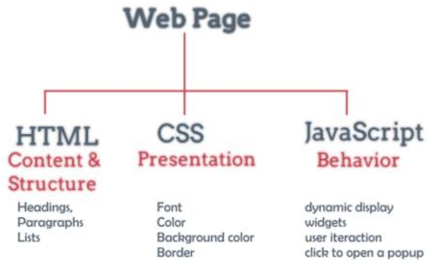


1. Hey Server,

Send me some documents for address google.com!

2. Sure, here's some HTML, CSS, and JavaScript!

3. Great! I know how to display those!



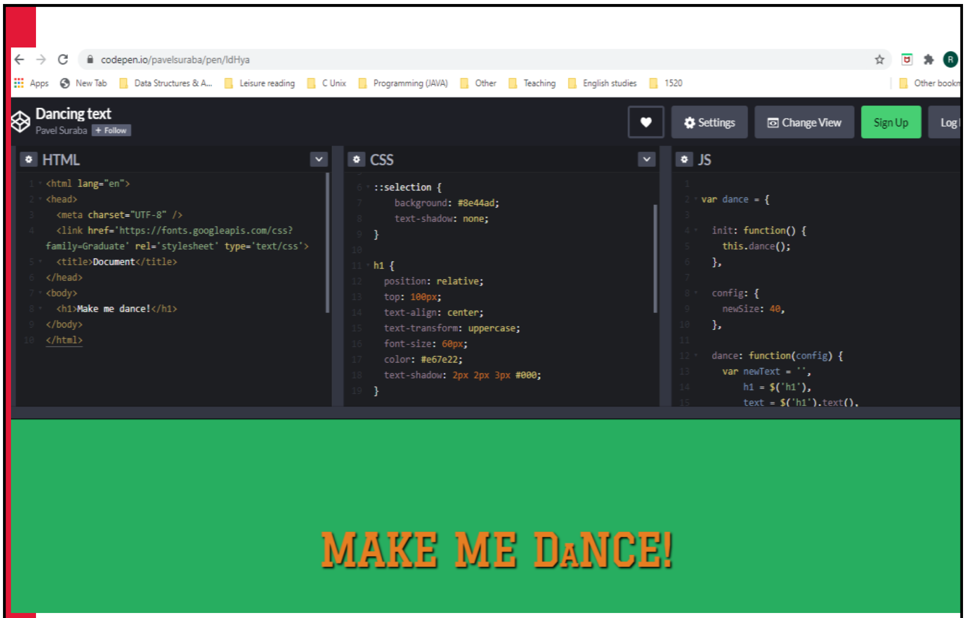
Web Page

HTML
Content & Structure
Headings,
Paragraphs
Lists

CSS
Presentation
Font
Color
Background color
Border

JavaScript
Behavior
dynamic display
widgets
user iteraction
click to open a popup

	HTML	CSS	JS
Language	HTML	CSS	Javascript
Purpose	Structure, Objects, Things	Looks, Style	Actions
Syntax	<p> <h1> 	P {color: red;}	var x = 5;
Grammar	nouns	adjectives	verbs
Building	Walls, structure	Paint, curtains	Electrical, Plumbing, AC

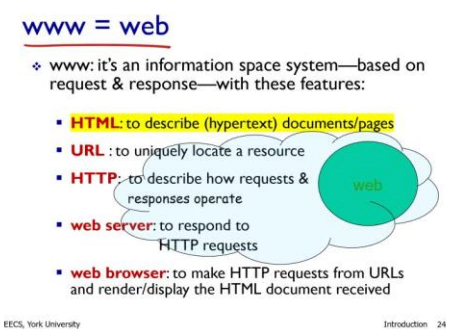


```html
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <link href='https://fonts.googleapis.com/css?
family=Graduate' rel='stylesheet' type='text/css'>
  <title>Document</title>
</head>
<body>
  <h1>Make me dance!</h1>
</body>
</html>
```

```css
::selection {
    background: #8e44ad;
    text-shadow: none;
}

h1 {
  position: relative;
  top: 100px;
  text-align: center;
  text-transform: uppercase;
  font-size: 60px;
  color: #e67e22;
  text-shadow: 2px 2px 3px #000;
}
```

```javascript

var dance = {

  init: function() {
    this.dance();
  },

  config: {
    newSize: 40,
  },

  dance: function(config) {
    var newText = '',
        h1 = $('h1'),
        text = $('h1').text(),
```

https://codepen.io/pavelsuraba/pen/ldHya



www = web

- www: it's an information space system—based on request & response—with these features:
  - **HTML**: to describe (hypertext) documents/pages
  - **URL** : to uniquely locate a resource
  - **HTTP**: to describe how requests & responses operate
  - **web server**: to respond to HTTP requests
  - **web browser**: to make HTTP requests from URLs and render/display the HTML document received

EECS, York University

Introduction 24

We start with (review) of client side

- HTML
- CSS
- JavaScript