CSC309 Final Lecture

Final Exam

- No aids allowed
- Some programming - vanilla JavaScript
- Short answer type questions
- Some code reading
- Comprehensive (except Chrome extensions)

PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

St. George Campus

DECEMBER 2016 EXAMINATIONS

CSC 309H1F
Instructor:
Karen Reid
Duration: 3 hours

Examination Aids: None

Student Number:	
Last (Family) Name(s):	
First (Given) Name(s):	

---

*Do **not** turn this page until you have received the signal to start.*
(In the meantime, please fill out the identification section above,
and read the instructions below *carefully*.)

Marking Guide

# 1:	______/10
# 2:	______/ 8
# 3:	______/10
# 4:	______/ 6
# 5:	______/ 7
# 6:	______/ 8
# 7:	______/10
# 8:	______/14
TOTAL:	______/73

This final examination consists of 8 questions on 15 pages. A mark of at least 29 out of 73 on this exam is required to pass this course. *When you receive the signal to start, please make sure that your copy of the examination is complete.*

Answers that contain a mixture of correct and incorrect or irrelevant statements will not receive full marks.

Good Luck!

Note: You may detach this page for easier reference.

Basic JavaScript

```
JSON.parse()
JSON.stringify()

document.getElementById(string)
document.getElementsByTagName(string)
document.createElement(string)
element.innerHTML
element.apend(element)
element.empty()
alert(value)

array.length()
array.push()
array.splice()
object.toJSON()

event.preventDefault()
```

JQuery

```
$(selector).append()
$(selector).html()
$(selector).empty()
$(selector).parent()
$(selector).insertAfter()
$(selector).on()
```

Express route handling

```
req.send()
req.get()
req.params()
req.query()
req.body()
req.route()
```

Topics

Browser

- HTML5
- CSS
- DOM
- Forms
- Validation
- JQuery
- React
- Templates

Server

- Node
- Express
- MongoDB/Mongoose
- Validation
- JSON
- REST

Topics

- Communication
  - HTTP
  - GET, POST, PUT, DELETE
  - Sessions, Cookies
  - AJAX
  - JSON
  - Security
  - Promises (basic idea)
  - Web Sockets (basic idea)

Not on Final Exam

- Kate Hudson’s talk on Data-driven decision making
- Chrome extensions
- Today’s material

What haven’t we talked about?

- Testing!
- Large Scale Web Apps
- Search Engines
- Microservices
- …

Testing

- Functionality Tests
  - Unit tests: E.g. Mocha
  - Integration tests
  - End-to-end - E.g. Selenium
  - HTML CSS validation
  - forms and form validation
  - cookies - test for correct operation and deletion

Testing

- Usability testing
  - need real people to test your application
  - navigation
  - consistency
  - content

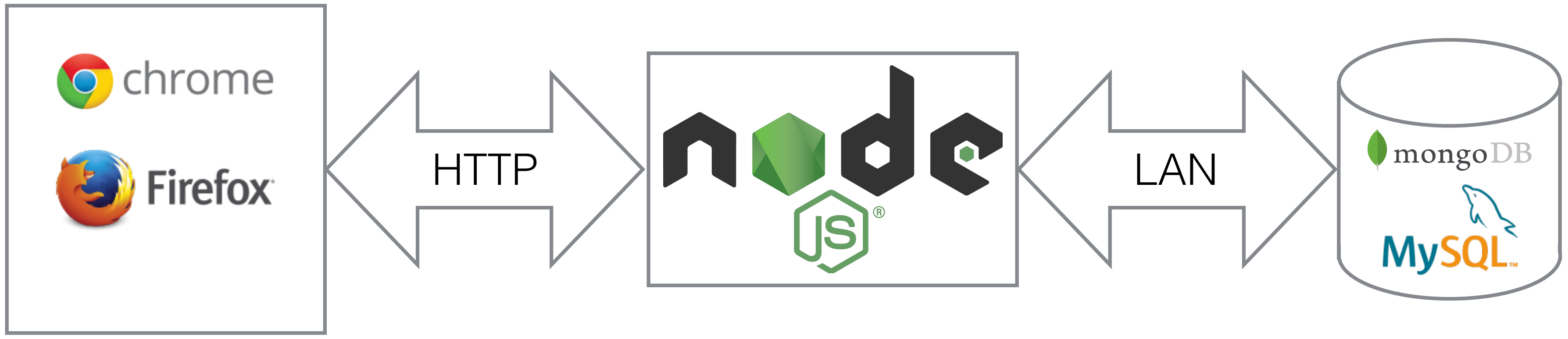
Testing

- Interface testing
  - application - requests sent to server and results displayed
  - web server - handles all requests correctly
  - database server - queried give expected results
  - when connection between layers fails, appropriate error messages
- Database testing
  - integrity
  - response time
  - data retrieved is shown accurately

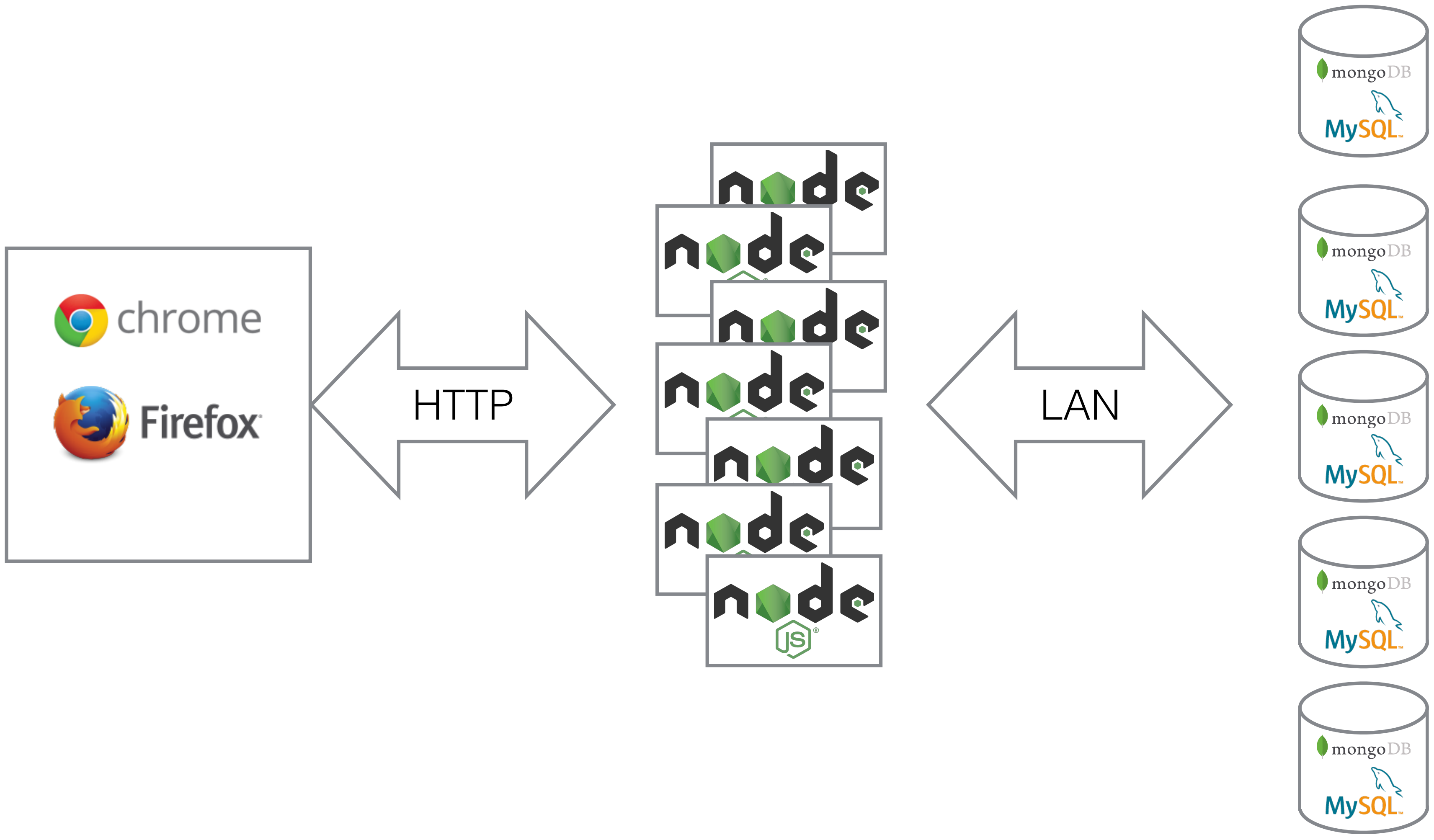
Testing

- Compatibility testing
  - works on all major browsers, including older versions
- Performance testing
  - load test - normal and peak
  - stress test - push beyond peak
  - crash recovery
- Security testing
  - regularly audit!

Standard Web App



Large Scale Web App



Scale-out

- Expand capacity by adding more instances
- Pros:
  - can scale to fill needs by adding and removing instances
  - fault tolerance
- Cons:
  - manage multiple instances and distribute work

Scale Out: Which server to send to?

- Browsers want to speak HTTP to a web server
  - Use load balancing to distribute incoming HTTP requests across many front- end web servers
- HTTP redirection:
  - Front-end machine accepts initial connections
  - Redirects them among an array of back-end machines
- DNS (Domain Name System) load balancing:
  - Specify multiple targets for a given name
  - Handles geographically distributed system
  - DNS servers rotate among those targets
  - How to handle sessions?

Load-balancing switch ("Layer 4-7 Switch")

- Special load balancer network switch
  - Incoming packets pass through load balancer switch between Internet and web servers
  - Load balancer directs TCP connection request to one of the many web servers
  - Load balancer will send all packets for that connection to the same server.
- In some cases the switches are smart enough to inspect session cookies, so that the same session always goes to the same server.
- Stateless servers make load balancing easier (different requests from the same user can be handled by different servers).
- Can select web server based on random or on load estimates

nginx

- Web server designed for scalability
- Load balancer
  - can handle SSL processing
  - application health checks (server fails)
  - session persistence
  - limits to mitigate DOS
  - bandwidth limiting

Scale-out assumptions

- Any server will do
    - Different requests from the same user can be handled by different servers
    - Requires database be shared across servers
- What about session state?
    - should be fast because it is accessed on every request
- Web sockets?
    - cannot load balance each request

Scale-out storage

- Data sharding - spread database across instances
  - each piece of the database is called a shard
  - Tolerate failures (and improve performance) by replication
  - Increases complexity - Applications must place data across multiple databases

Memcache

- Main memory caching system
- Key-value store (both keys and values are arbitrary blobs)
- Used to cache results of recent database queries
- Much faster than databases:
    - 500-microsecond access time, vs. 10's of milliseconds
    - Writes still go to database, so no performance improvement
    - Cache misses still hurt performance
    - Must manage consistency in software (flush memcache data when database is modified)

Scalability

- Building this architecture is hard!
- Need data centre expertise
- Figuring out the right number of components is hard

- Cloud computing:
  - allows for dynamic addition and removal of resources
  - outsources data centre management

The End

All the best on remaining assignments and exams.

Happy Holidays!