Course Concepts

- LAMP Architecture
- Web Architecture
- Web Standards
Course Content

- LAMP: Linux, Apache, MySQL, PHP
- GitHub
- jQuery
- Docker
Why LAMP?

- Standard, well documented stack
- Teaches web design fundamentals
- Open Source

Why not Node.js/METEOR/MEAN/whatever?
- Fewer “standard” tutorials and examples
- Recently popular technologies, but do not exemplify standard web design principles
- LAMP principles translate natively to these technologies
Why GitHub?

- Industry standard
- Public!
- Accountability
- Branching/rollback/repository/tracing
Why jQuery?

- Adding client-side development to LAMP
- Standard library
- Widely adopted
Why Docker?

- Emerging industry practice
- Principles translate to other services
- Re-usability
- Consistency
Administrative Notes

- Room (Dragas 1117)
- Meeting times (4:20-7:00)
- No Exams
- Demos and projects
- Simulates long-term, professional development environment
Course Resources

- http://www.cs.odu.edu/~jbrunelle/cs518
- Syllabus
  - You are responsible for knowing all policies in the syllabus
- Readings
  - Listed under the day they are expected to be completed.
- Lecture Notes and Assignments
  - Posted on schedule page before class
Additional Resources

• W3C
• Stack Overflow
• Waterloo Course Website:
  https://opencs.uwaterloo.ca/web-basics/
  https://opencs.uwaterloo.ca/web-programming/
Optional: Textbooks

- Beginning PHP5, Apache and MySQL Web
- PHP, MySQL, & JavaScript
Development

- Course prerequisite: CS330 – OOP and Design
- Assuming basic HTML and CSS experience
- Course development done in LAMP
- Git for source control (https://try.github.io)
- Docker for deployment & grading
Course Projects

- Single project
  - Long-term development
  - F16: Stack Overflow
  - F17: Slack
  - F18: Social Media
- 4 “releases”
- Intra-group or individual only
  - Collaboration via class mailing list and slack only
Grading

• *Dockerized* development
  - No magic laptops!

• Submissions via GitHub
  - Used for demos and grading
  - Creates a public portfolio

• Feedback/grading sent via email
Rough Grading Outline

- 4 releases, 25 points each
  - 15 points – Functional requirements
  - 3 points  – project write-up
  - 3 points  – website usability
  - 2 points  – aesthetics
  - 2 points  – status report

- Extra credit
  - Additional features
  - 0-10 extra points pending quality
Class Policies

- Follow academic integrity policies
- Attendance is not required
  - But you are responsible for material covered in class
  - Attendance required on demo days
- Seeking help
  - Mailing list *only*
  - *All unapproved collaboration considered an honor code violation!!*
- Office hours: Friday afternoons
  - Please email me to make an appointment
How to cheat

- Cheating results in a score of 0/25 for your milestone
- Inter-group collaboration
- Sharing code
- Passing off open source code as your own
- Failing to cite your code “inspiration”
Dr. Justin F. Brunelle

- BS, MS, PhD in CS from ODU
- Advisor: Dr. Michael L. Nelson
  - s/VT/ODU/
  - s/Electronic Music/.*/
  - Ford muscle cars
- Digital preservation research @ ODU
  - Web Crawling, JavaScript, Web Architecture
- Principal Researcher @ MITRE
  - Helping government adopt emerging tech
  - Specialize in data & cloud computing