Advanced PHP

WDV 441

Monday 6:00pm - 9:00pm

Week 1

Introduction

- Welcome :)
- Who is this yahoo?
  - Garritt Grandberg, VP of Tech & Engineering and Senior Software Engineer for Visionary Services
    - ~30 person IT consulting firm, in business for over 25 years
      - http://local.fedex.com
      - http://www.desmoinesperformingarts.org
    - Went to school to be a pilot, came out a programmer
    - 20+ years developing business applications both desktop and web
    - Working with PHP for 15+ years (10+ professionally) starting with PHP3
    - Oversee all development and developers
    - Final say on all development hires
    - Actively develop on several large PHP-based business applications
    - Hobby game programmer, currently using Unity3D
      - https://g2-games.com
      - https://legendsofthebrawl.com
- Slow down man! You talk to dang fast.

What is this class going to cover?

- Organization of a project
- Web Application Security
- Web Applications and their components
- Source Control
- Troubleshooting PHP code
- Object-Oriented Programming
- MVC Software Architecture
- SQL/Database Design/Abstraction Layers
- User Login/User Rights
- File uploads/data importing
- Reporting/data downloads
- RESTful web services
- JSON

What should I know about PHP ?

- I expect you to be knowledgeable and comfortable with the following
  - Variables & Arrays (declaring/utilizing)
  - Assignment operators (=, +=, -+, ++, --)
  - Loops (for/while)
  - Branching (if/switch statements)
  - Comparison operations (==, <, >, <=, >=, !=, ===)
  - User-defined Functions
  - Basic understanding of Classes and OOP
  - Database connection, a simple query of data, and display to a page
  - How PHP works (ie what happens when a PHP page is requested)
- Ah man is he serious! I forgot half this stuff should I drop?
  - No! but please talk with me after class or email me

What do I need for this class?

- A pen or pencil (maybe for a few notes?)
- You are free to use any PHP-capable IDE you are comfortable with. I personally use NetBeans.
- Web server account with a MySQL account and access to phpMyAdmin
  - All-in-one servers will work too such as Laragon, XAMPP, WAMP, and Uniform Server

How to excel in this class

- Questions! There is no silly question. Don't be afraid to ask. When you have a question please ask at that time or write it down and ask/email me later. If you are not in class, be sure to write it down immediately and either email it to me or bring it up next class.
- Attend class. Participation is important and to avoid distraction please turn off/silence your phones. Preferably, place them where you can't see them.
- Find a study safe zone. Get out of your natural environment to eliminate distractions. Many public libraries have study rooms or areas that can provide some privacy.
- Start your homework early. Do ***not*** procrastinate. If you do, I can guarantee you will be losing easy points over the course of the semester.
- Each class builds on previous classes so it is very important to get your assignment done each week.
- While in class, collaborate with your lab partner (if assigned) but don't copy. Take advantage of lab time to work on your homework.
- **I want you to succeed**. Talk to me early if you are struggling.

Class Format

- In general, 2 hours of lecture and 1 hour of lab but this can vary
- Every class will begin with a quiz on the previous weeks topics. This quiz will be 5 questions.
- 10 min break each hour
- Again, phone silent or off. Prefer to put them away.
- Please keep your PCs viewing class-related content only (except during breaks)
- Ask questions any time
- Don't be afraid to tell me when you do not understand or if I am going too fast

Syllabus

- Zzzzzzzzzzzzzzzzzz..................

Web Applications

- What is a web application?
    - Simply put, software that runs on the web
    - Also known as SaaS (software as a service)
    - Natural progression as the capabilities of web platforms grew closer to that of desktop applications
- What are some examples?
    - Blackboard
    - Office365
    - Google Docs
    - SalesForce
- Is a web application a website?
    - Traditionally a website is a means of marketing/communication (ie providing information) but no more.
    - Most of today's websites are a blend of a traditional website and a web application that controls it (ie a CMS).
    - amazon.com is considered a website but it is also a sophisticated piece of sales software

Web Application Security

- What does this mean?
  - Protecting an application and its data from external threats by use of hardware, software, and secure code design
- What are two of the most common vulnerabilities?
  - SQL injection
  - XSS (cross-site scripting)
- Hardware
  - Device that sits between the web server and the outside world
  - Uses software to analyze traffic to detect attacks and stops them before they reach the server
- Software
  - Similar to hardware but sits on the web server
    - apache mod_security
    - SSL encryption

Web Application Security (cont)

- Secure Code Design - build a secure foundation
  - Largely ignored in the past
  - It all starts here and with you
  - The single most important component is the developer
  - It is your job to develop with security in mind
  - Most vulnerable point of an application are its input points
    - GET/POST variables
  - Never, ever, ever trust what the user provided
  - Validate type of input content, not just that it was provided
    - ie if you are expecting an int value, make sure it is an int value
  - Only allow expected input, reject everything else
  - In this class you are expected to write your source securely
  - You must be meticulous. If you are telling yourself "it's good enough" that indicates internally you are compromising and that more work is needed.
  - I cannot overstress the importance of secure code design in todays world

Project Organization

- It is important to begin every web application with an organized structure
- Begin each week with a new folder named: weekN where N is the week number
- In this class we will be using the following directory structure for our web application work
    - UserDirectory
        - WeekNN
            - public (to store files publicly viewable such as images and our PHP pages that server up the HTML)
            - inc (to store files containing our business logic)
            - tpl (will be used with MVC to contain our templates)
- Beyond this base structure, you are free to organize subdirectories inside of the three primary folders as you please
- Let's take a look!

Coding Standards

- Any good shop will have a certain level of coding standards that you will be expected to adhere to
- These are in place for a reason and are important to follow
- May seem like a hassle at first but will be second nature in no time!
- We will be using some very simple standards for this class
    - Naming convention
        - variables and functions will be named using camel case syntax. The first letter will begin lower case and every new word within the variable name will begin with a capital letter (ie $firstName, $contactEmail, function saveContact).
    - Self documenting code
        - The name used for variables and functions should appropriately describe what the variable is storing or what the function does (ie $userCanAccess, function getUserName)
    - Curley braces
        - When utilizing curly braces, the beginning brace should be on the same line that starts the block

          ```
          if ($test) {
              echo "test";
          }
          ```
    - User proper indentation when nesting code blocks
- As the weeks go by I will be more and more critical that you follow these simple standards (ie you will lose points). If you ever have any questions please ask.

Page Organization

- Creating a well organized page is very important
- All business logic should be at the top of the page
  - Business logic is logic that performs things like login, rights, data lookup
- There should be no business logic below the opening <html> tag
- PHP that does come after the <html> tag should be display-only
  - Display only logic are things like "if" statements that control what is shown or code that displays values from a data lookup
  - Display logic uses the result of business logic so for example, the business logic would determine if a user is logged-in, the display logic would check that determination and show the user the appropriate interface
- Let's take a look!

Source Control

- Some Advantages
  - Provides ability to track and store changes to source over time
  - Changes that have a negative effect can be rolled back
  - Can be used as a means of moving source through different environments (ie from your development environment to production)
  - Helps prevent multiple developers on the same project from conflicting with each other
  - Allows multiple large developments on the same project to occur at the same time without conflicting with each other

Bitbucket and Git

- You all should have a Bitbucket account and Git
- Can I use Bitbucket for this class?
  - You can but it is not required and I will not be grading based on it
- What is a repository?
  - A Git repository is what stores your source and any changes made to your source
  - Repositories can be shared.
- Some other terminology
  - Checkout - creating a checkout of the source from the repository
  - Add - queue a file to be added to the repository. Nothing has actually been added yet.
  - Commit - schedule a source change to be placed into the repository
  - Push - push scheduled changes to the repository
  - Pull - pull any changes others have pushed to the repository into your checkout

Supplemental Reading

- PHP Basics
  - https://www.w3schools.com/php/
- Web Application Security
  - https://owasp.org/www-community/attacks/SQL_Injection
  - https://www.veracode.com/security/sql-injection
  - https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
- Bitbucket and Git
  - http://www.bohyunkim.net/blog/archives/2518

Assignment for Next Week

- Create a new directory on your server for week 1 using the presented guidelines and the syllabus
- Create an index.php page in the public folder that performs the following:
    - store a list of 10 names into an array
    - store a random number between 0 and 20 (see the rand() PHP function) into a variable
    - using the random number stored and PHP/HTML to display
        - show the text: Hello <name> if the number is between 0 and 9 where <name> is the value from the array at the index of the random number
        - if the random number is greater than the bounds of the array, show the text: Name List and then output all names in the array onto the page