Week 13: Web Security & Final Considerations

Agenda

- Overview
  - Final Project Review
  - What's Due?
- Web Security
- Final Considerations
  - Future of Web Development
  - Next Steps
- Final Project Progress
- Cybersecurity Discussion (two posts)

Overview

Last week, we learned more about SQL and the kinds of statements we can use to manipulate databases or the data inside of them. This week, we will briefly cover some basic security concepts to help make our pages/forms more secure. We will also talk about the future of web development and some steps you can take to expand your knowledge in the general field.

Final Project Review

The Review portion of the final project is essentially a quick presentation on your work done so far. You can use a video, screen shots, a word document, etc. to talk about your website. Why did you choose this topic? How much progress have you made? What were some obstacles? What features are you proud of? Why did you choose that color scheme? These are some questions you can answer in your review.

Look over the Final Project Details inside of the Final Project folder for more information on the requirements. It doesn't have to be a long presentation, but you should show enough of your website that anyone could understand what it was about and how it works.

I will allow you to present your work during office hours that week as an alternative. You must email me first. The Final Project Review is **due on Thursday, May 4h at midnight.**

What's Due

- Final Project Progress
- Cybersecurity Discussion (two posts)
- Final Project Review

Web Security

As a rule, with your websites, you should not trust any user. The assumption must be that any user will use your form to try and manipulate your database in unexpected ways. Of course, this isn't always the case, but you can never be too safe. Websites which use forms to communicate with databases are particularly vulnerable to many types of attacks. Think about all the ways you use your social media platforms. You post links, upload files, share content, search for items, etc. and each of these poses as a potential threat for the company hosting the content.

Building your forms correctly can go a long way in ensuring your databases remain protected. However, it is important to note that there are many aspects to security. I won't cover them all, but I'll mention some that are commonly implemented.

File Protection

This is simply keeping your important files somewhere outside of the web root folder. During this semester, I've told you all to put your pwd.php file in the same folder as your other file. Since we are only creating basic websites with no real data, that is fine. However, in normal instances, you would place it with the other server files users that couldn't access. Ideally, it would be outside of the public html folder on 000webhost. That's not a 100% solution though and other protections would be necessary.

If you store your database credentials incorrectly, and a server can't parse the file as PHP for some reason, the page will be served as plain text. This will expose your password to anyone that knows where the file is located. Always protect your files and don't share password information with anyone.

Form Validation/Sanitizing

Validation means verifying that the data being input is what you expect. For example, if there is an input for email, you might have some code to check if an "@" or "." character is present. You can use methods such as regular expressions to check input against some criteria.

Sanitizing user input means cleaning the user's input. It may involve stripping any angled brackets, SQL keywords, HTML characters, etc. that could impact your website. Without those characters, it's harder for hackers to insert malicious code or alter your databases.

Any user can disable JS in their browser so if all your validation/sanitization is on the client-side, someone could bypass your measures entirely. PHP and JS combined is the best bet if you are going to try and ensure the input data is correct.

In the class example provided this week, I use a "prepared statement" to insert data into a database. Prepared statements help prevent SQL injection, which is when a user adds SQL code to their input to alter your database. It's one method for sanitizing user input.

Selecting Quality Programs and Updating Them

There will be times where you might find it necessary to utilize third-party applications for your website. You must be sure that there are no vulnerabilities that could compromise your system inside those applications. Also, staying on top of updates is important as they may include patches for security bugs. Often, hackers will exploit older versions of programs that people still use to steal data.

Open-source programs are great because they are free and often maintained by a good community of developers. However, exploits in these programs are more well-known and easier to find for hackers. Understanding how these programs are vulnerable is important if you want to use them.

Configuring System Permissions Correctly

This one is not as important for our class but setting permissions correctly in your database or administrative areas is important. Some people make the mistake or providing guests or other users with "super user" or administrative rights. A person who shouldn't have permission to access the system could wreak all sorts of havoc if you don't correctly establish who has access to what and for how long.

Implement Login, Session, etc. Features Correctly

If a user visits your website and you make use of cookies or a session, you should ensure that those sessions are closed or reset upon their leaving. When connections to databases are left open, it increases the chances of a middle-man attack where someone interferes and intercepts data. This is why I mention closing the connection in all of my PHP/SQL examples.

Use Encryption (SSL, HTTPS, etc.)
This one is very important as several data breaches have exposed passwords which should have been disguised. Some companies might store passwords as plaintext and that is a very bad practice. If a hacker had access to a table with that data, they would have the username/password combinations and could sell them or get into user accounts. However, with encryption, these password/username combos would be harder to decipher, and your users would be protected.

Common Types of Threats (Not required reading but good to know)

- https://www.acunetix.com/websitesecurity/sql-injection/ (SQL Injection)
- https://www.us-cert.gov/ncas/tips/ST04-015 (Denial of Service Attacks)
- https://owasp.org/www-community/attacks/xss/ (Cross Scripting Attacks)
- https://www.sitepoint.com/8-practices-to-secure-your-web-app/ (Good Practices)

Final Considerations

This semester, we covered several web development concepts. I know it was stressful at times, but a lot of the things learned are applicable in other languages and environments. If you eventually become a programmer or web developer, the knowledge you obtain from this course will be relevant. Here is a list of things we covered that you could potentially brag about on a resume:

- HTML
- CSS
- Responsive Web Design
- JavaScript (Cookies & AJAX)
- PHP
- SQL/MySQL
- Cybersecurity in Web Development
- Usability/Accessibility

With these skills, you have the foundation for what it takes to be a full-stack web developer. Front-end web developers typically work on HTML/CSS/JS to create the visual look/feel of the website. Back-end developers usually work with a programming language/database system to create the server-side technologies which feed into the front-end. In this course, you got a taste of both worlds, but there are many more things to consider.

Future of Web Development
Like most fields in the information technology sector, web development is a rapidly evolving area. New tools and methodologies are constantly being introduced and implemented across websites daily. I will briefly discuss a few of the features or trends that are becoming more popular in websites.

Responsive Web Design (RWD)
This is already an essential component to any website. Without RWD, your website will not rank highly on most search engines. As smart phone usage increases, the amount of traffic to websites

through mobile devices will also increase. It is important that a website works well on any screen size, but phone sizes are particularly important.

Mobile-first approaches to website construction may become more common as web development agencies/teams prioritize smart phone layouts over desktops. I personally expect properties such as grid and flexbox to become the standard for websites. Floats will still exist in websites for years to come, but the other properties are more useful for precise layouts.

Progressive Web Apps (PWA)

These are web applications which behave similarly to native applications on your device. When you open apps such as calendar or calculator on your phone, the app loads regardless of network connection and is always usable. Progressive Web Applications have the same feel as a website/native application, but they are installed on your local device. Rather than appearing in a browser tab, they display on your device's screen and can connect to an outside source. If the connection isn't good, the application will still have the core functions enabled.

You might already have one of these installed on your phone. One example is Starbucks; their website functions very similarly to their app, but you can use it offline. It won't allow you to look up locations or place an order, but you can still add items to your cart and search their menu. The addition of these types of applications can increase revenue for a website by allowing users to access content without installing a large app. They are much lighter with code and can do things like push notifications to a user using less network resources.

- https://web.dev/what-are-pwas/

Frameworks

There are several frameworks out there which build upon a language to increase productivity and enable developers to create robust features. Some examples of CSS frameworks are Bootstrap and Foundation. JavaScript has many frameworks which include but aren't limited to JQuery, Angular, React, Node, and Backbone. Frameworks will keep being created and developers will have to learn them as project needs change. Rather than focusing on learning any one framework, you should learn how to work with code in general. If you understand how code works in a programming language, frameworks will be much easier to understand.

Artificial Intelligence/Machine Learning

Websites used to be incredibly costly to create and businesses would (and still do) pay thousands of dollars for their website to be generated. However, it is now easier than ever to build a website with a handful of clicks. Services such as Wix and WordPress exist to enable the creation of appealing websites on the fly. Any person with some time and some technical knowledge can create a website without the need of a developer. There are some problems with these websites though:

- They can introduce code bloat (extra code that isn't necessary)
- It can be difficult to alter styles
- The website may look generic

Websites with more complicated features and unique designs generally require talented developers to build them. These developers often conduct research on client needs and competition so they can craft a suitable and focused product. What if there was a system that was able to conduct that research and produce a website based on it? This is where artificial intelligence and machine learning become critical.

A program that can teach itself to build relevant content based on client needs would expedite the web development process and cost much less than a team of web developers. Instead of hiring an agency, a business could use a piece of software which behaves similarly to a Content Management System like WordPress. This software would potentially offer more customized content dynamically and quickly.

- https://medium.com/@mindfiresolutions.usa/the-effect-of-machine-learning-on-web-application-development-c88a9e5f9553
- https://www.honeybadger.io/blog/machine-learning-for-web-developers/

Next Steps

If you want to pursue a career in web development, there are some things you can do to better prepare yourself for the field. I will provide some general tips and resources that are entirely optional to review. However, reviewing them might be useful if you'd like a job as a web developer or something related.

HTML/CSS/JavaScript

These are the primary web technologies you will encounter on almost every website out there. HTML provides the foundation for a web page while CSS is used to style it. JavaScript adds interactivity to your page by allowing you to manipulate elements in the HTML document.

You don't need to memorize every HTML tag, but knowing the common ones such as headings, paragraphs, lists, img, div, etc. is helpful for understanding how you can structure a page. Code validation is also necessary so tools such as the W3C validator should be in your arsenal. Aria roles are great for accessibility and are an attribute which is used in HTML elements. You should be prepared to learn some of the different aria roles and how they work to improve accessibility on a page.

With CSS, it's important to understand specificity and how to position elements on a page. Specificity will help you structure your styles, so they don't conflict with each other. It is also useful for when you need to work with a framework. Sometimes you'll have to overwrite a style and you'll need to be more specific with your own CSS to do this. As for positioning, you should understand floats and clear fixes. Float is an older property, but it can be found in millions of websites. Properties such as Flexbox and Grid should be used for the creation of all new websites but knowing float will allow you to work with more websites.

It's not a bad idea to learn a framework such as Bootstrap or Foundation. Understanding how different components work or themes in different content management systems can be good for obtaining new ideas for your own styles. Also, you don't necessarily want to create every web page from scratch as that would be time-consuming. Having prepared templates or bits of code will help speed up your workflow.

Server-side Technologies

We covered PHP in this class because it's open-source and free to use. It has a lower learning curve than some other languages and is easy to set up as well. You don't have to learn any specific language or framework for back-end/server-side technologies. PHP, .NET, Python, BackBone.js, Node.js, etc. can all be used to communicate with external data sources. What you should understand is how databases and servers are generally accessed. How is data processed when it is requested by a page? How can you optimize these data requests? How can we make it so that these requests are secure?

- https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction
- https://learntocodewith.me/posts/backend-development/

Version Control

This allows a user to keep track of changes in a document in a centralized area. If you made a change to a page that didn't pan out so well, you could use a version control tool to revert your work back to older rendition of that code. Github and Subversion are example of tools which use version control. If there are multiple developers working on a program, version control will keep track of any changes and the person that made them. There might be one master document with several branches and different developers can add their changes to the main document when they're ready.

You might have done something like save a word document as assignment-v1.docx or assignment-v2.docx to keep a backup version of your old work. Rather than do that, you could have all your work in one area with the history of changes stored. Version control is a powerful tool and learning it will make you a more efficient developer. It's also great for working in teams.

- https://github.com/
- https://betterexplained.com/articles/a-visual-guide-to-version-control/

Browser Developer Tools

A web browser is your best friend when it comes to testing your code. Google's Web Developer Tools are among the best tools out there for checking your website's code and performance. Becoming familiar with web tools in general across all web browsers is a must if you want to be a good developer. You can manipulate HTML, CSS, and JavaScript from the browser tools, see network information, and debug your code as needed. Audits can also be performed which check for essential components and standards on your website.

- https://developer.chrome.com/docs/devtools/
- https://www.shopify.com/partners/blog/dev-tools

Databases

We've already learned that the amount of data being generated yearly is increasing rapidly. Raw data is meaningless without structure and databases provide some sort of structure to that data. You should understand the different components of a table (rows, columns, data types, etc.) and how tables can be related to each other. You don't have to memorize every SQL command but understanding the basic CRUD statements such as select, insert, update, and delete are useful since they are used very often. If you want to go further, database encryption is also very important for protecting user data from outsiders. This can technically be seen as a server-side technology, but databases are important to understand on their own.

Programming/Problem Solving

Programming can seem overwhelming at times because there are numerous terms, and they can sound complicated. However, ii you break down programming to its most basic concepts, it's actually not so bad. In most languages, there are objects which can be manipulated. These objects all have properties and functions/methods they can perform. We can relate this to human beings as an object. Humans have properties such as height, weight, age, name, etc. which help define them. There are also some things/functions humans can do such as walk, talk, and eat.

The same goes for most things in a programming language. Strings, numbers, arrays, etc. have properties or methods which can be used with them. In almost every language, there is some sort of method to get the length of a string. There is also an operator to concatenate strings. The major differences are going to be in the syntax.

If we wanted to combine the strings “Hello” and “World”, we could do it like this in various languages:

JavaScript: var myString = “Hello” + “ World”;
VB: Dim myString As String = “Hello” + “ World”
PHP: $myString = “Hello” . “ World”;
WinBatch: myString = StrCat(“Hello”, “ World”);

All four of the lines above would produce the same string: “Hello World”. However, you can see that the way it’s done varies based on the language. Some lines have a special marker to indicate a variable is being created. Some use a plus sign while others use a period or comma for string concatenation.

The same goes for most major concepts such as functions, if/elseif/else statements, loops, arrays, objects, and comments. Instead of learning a particular language, you can learn the general concepts behind programming. Of course, if your job requires it, you should focus on learning a specific language. However, understanding those core concepts will enable you to work with a new language more quickly. The reason is because you’ll only have to adapt to the new syntax if you understand how programming works.

Programming is really an exercise in problem solving. If a client asks you for a feature such as an image slider, you will have to break that feature down into its smallest components and then build upward from there. It’s the same for any application. You’re combining all these small concepts into something bigger until you have a fully functional program.

Examples
This week’s class example will not be live. Instead, you’ll need to place the page on your 000webhost area for it to fully function. It will allow you to insert and update records in your table. You can use the pwd.php file from previous class examples (assuming you didn’t change your database information). If you don’t have the pwd.php file from previous class notes, you can fill out the one provided. Place the pwd.php and security-basics.php file in the same folder on your 000webhost account and the page should work for you

PowerPoint documents have also been released under Blackboard → Course Materials → Resources →PHP & SQL. One of them offers a brief introduction to PHP/SQL while the other will serve as a debugging guide.

You won’t be able to right-click the page and view the source this week to view my PHP code. Since PHP is processed by the server and rendered as HTML, the user can’t see the PHP code you’re using. The code for the example below is in the “examples” folder that was included with this week’s lecture notes zip folder. You can take all the contents of the examples folder and place them somewhere on your 000webhost account and the pages should work. There are comments in the code to explain what is happening. Feel free to look at the live link to see the page in action.

Here’s a link to **all** course examples:
https://cinf362.000webhostapp.com/examples/ (PHP/SQL Examples)
https://www.albany.edu/~cv762525/cinf362/examples/ (HTML, CSS, and JS examples)
https://www.albany.edu/~cv762525/cinf362/videos/ (videos)
https://drive.google.com/drive/folders/13sh0oaUeE9di4aZuTYczqNdpzdb4kzKE?usp=sharing (more videos)

Final Project Progress

Due Monday, May 2nd at midnight

For this week, you have two options for the assignment. The first option is to work on your Final Project and document your progress. The second option is to do a write-up on a web development trend that you think is interesting. Each option is described below.

Final Project Progress (Option One)

For this option, your task is to make progress on your Final Project. You should take screenshots of your work before you start anything on it. After you feel satisfied with the amount of work you have done, you can take more screenshots or submit live links to your pages. I will evaluate your work based on the primary screenshots provided and compare them to the latest version of your work. **If you've completed your project or do not want to do this, look at option two.**

Along with the screenshots, make sure to include a description of the work you did. In some instances, the work you've done won't necessarily show up on the screen. You can include screenshots of code to depict updates if necessary. The most important thing is that you clearly demonstrate that you've made progress on your website. If you've already done most of the work, that is fine. You can describe any remaining steps left, any obstacles you faced, or features you would have liked to add.

There isn't an exact threshold for what counts as "making progress" on your website. This means I won't be counting lines of code or anything like that. However, I should see that you've improved your website in some way. You don't have to document everything you did, but anything major should be discussed at the very least. Here are some ways to make progress on your page (just one isn't enough necessarily):

- Improved accessibility
- Added responsive styles
- Changed the color scheme
- Added new content
- Reorganized content
- Added terms for SEO
- Added a JavaScript or PHP based feature
- Cleaned up HTML, CSS, etc.

Acceptable Forms of Proof

- Video discussing progress
- Before/After screenshots of code or the web page
- PowerPoint with documentation
- Two zipped folders
    - First folder has your original code
    - Second folder has your updated code at the end of the week

A description must accompany your forms of proof so that you can explain anything I might not see when I evaluate your progress. The more details you provide, the better you'll submission will be. **You can use screenshots from this assignment for the Final Project Review as well.**

The work to prove you made progress on your final project is **due on Monday, May 2nd at midnight.** To submit the work for this exercise, visit Blackboard → Course Materials →

Lectures Notes for this week's class. You can also go into the "Assignments" folder and the submission area will be titled "Final Project Progress." You should be submitting proof of progress on your project (screenshots, explanation, etc.). The work submitted will be evaluated based on the rubric explained below.

Final Project Progress Rubric – 10pts

- Screenshots or live links provided – 5pts
- Explanation of changes is included – 5pts

Web Trends (Option Two)

Instead of working on your Final Project and documenting your progress, you can do a write-up about a web development trend or future concept that you think is interesting. It must be **at least three paragraphs long** and can be one of the ones mentioned above or something entirely new. A paragraph should be at least 5 sentences long.

What you should do is elaborate on this trend and provide **at least 2 cited sources** which contribute to your write up. The cited sources can just be links to the pages where you found the content. In your write up, try to at least include the following:

- Why is this trend interesting?
- Do you think this trend will last?
- How will it impact web development as a whole?
- Are there examples of this trend out there? Are they done well?

The submission area for option two is the same as option one. Instead of submitting proof of your progress, you'll submit a Word document with your write-up. The write-up will be evaluated based on the rubric explained below.

Web Trends Rubric – 10pts

- A relevant web development trend/future concept was mentioned – 3pts
- Explanation/reflection of that trend/concept was thorough enough – 3pts
- 2 sources were linked – 4pts (2 each source)

Cybersecurity Discussion (Two Posts)

Initial Post due Friday, April 29th at midnight

Write about the various types of security vulnerabilities you may be exposed to daily. Think critically about the activities or services you are involved with and how they could impact you. Which ones pose the greatest threats vs not? Is there anything you fear when it comes to online security issues? What are some of the biggest issues facing the world in terms of cybersecurity? To get started use some of the questions below to help you start thinking about all the various ways you are exposed to potential threats. You can also write about something in cybersecurity that interests you if you'd like.

- How many applications do you have installed?
    - Do you use the same password for any of them?
    - Have you given these applications permissions? How so?
    - What kinds of information do you think these apps store on you?
- What do you do to protect yourself from security threats?
- What are pieces of hardware/software that you personally use that can expose you? (smart tv, routers, modems, Bluetooth, smart watches, etc.)
- What are some potential security threats you may deal with?

    - Which are big vs small threats?
- Have you or anyone been a victim of hacking or attempted hacking? **Please note this is not a required question.** I have personally been a victim of fraud, but I understand if you don't wish to discuss it.
- How much spam do you receive?
    - Emails vs phone calls
- Do you think most governments actively spy on their citizens? Does it matter?
- What is the future of cybersecurity?

The below website allows you to check to see if your email has been involved in any sort of data breach. Feel free to visit it. You do not have to report your findings, but it will be worth your time.
https://haveibeenpwned.com/

Plug in any emails you may have used in the past to see if they have been exposed in a data breach. Personally speaking, all my email accounts have been involved in a data breach of some sort. You **do not** have to mention if you were a victim in your post, but it is something interesting to think about.

There are many areas to discuss with this so feel free to be as broad or narrow as you want. Computers, routers, modems, cellphones, credit cards, e-commerce systems, etc. are all potential sources of danger so you have lots of options to choose.

The initial post is **due Friday, April 29th at midnight**. To submit, go to Blackboard → Course Materials → Lecture Notes for this week's class. There will be a discussion area called "Cybersecurity Discussion" where you can post your initial post. You can also visit the Discussion Board area directly from the Course Materials folder.

Response Post due Monday, May 2nd at midnight
In your response post, discuss your experience as it relates to another student. Do you agree with what they've identified as threats? Do you find yourself exposed to similar threats? Are you as worried about those threats? Feel free to share any other things related to their experiences/thoughts that you think are relevant.

The response post is **due Monday, May 2nd at midnight**. To submit, go to Blackboard → Course Materials → Lecture Notes for this week's class. There will be a discussion area called "Cybersecurity Discussion" where you can post your response post. You can also visit the Discussion Board area directly from the Course Materials folder.

Cybersecurity Discussion Rubric – 2pts
The initial post is worth 1.5pts and the response post is worth .5pt for a total of 2 points. I will be evaluating your posts based on the following criteria:

- Was a specific topic related to cybersecurity discussed?
- Did you mention specific potential sources of threats?
- Did your response post contribute to the original post?
    - Avoid summarizing the other person's post or simply saying that it was a good post.

Next Week

This is the last week of official classes! We have no class next week, but the Final Project is due by May 11th. I will still host regularly scheduled office hours up until that day so feel free to stop by with any questions you might have.