1. Describe one project in detail that you built or worked on. Please include information about the HTML, CSS, JavaScript and at least 1 server side language that was involved in the project.
   1. ShareShare
      1. Front End
         1. Responsive Fully Functioning
            1. Desktop does not move around
            2. Goes into mobile version
            3. Uses CSS Queries and JavaScript to load appropriate files
         2. Login and Register
            1. Live JavaScript Feedback
      2. Server
         1. PHP
      3. Programming Style
         1. Fully MVC
         2. Built on PHP classes
   2. Other
      1. Work
         1. Live Sync
            1. HTML, CSS, JavaScript, Cold Fusion
      2. Computer Science Capstone
         1. Native React
   3. Servers
      1. Configured and used Apache and TomCat with Java utiling Gradle for build
2. Tell us about a time that you had to explain something technical to someone who was non-technical.
   1. Instructor
      1. Radiation Physics and Intro to Entrepreneurship Tech Day
   2. Goals
      1. Engaging
      2. Clear
      3. Structured
      4. Hands on
   3. First steps was building clearly from the ground up
      1. Stop frequently for examples
   4. Utilize hands on activity
   5. Example
      1. Developing a Mobile and Non-mobile App
         1. How it works benefits and negatives
3. Will you please give us a very general walkthrough of the key elements that your CMS/CRM/LMS of choice contains?
   1. Drupal
      1. Easy to use and set up
      2. Structure
         1. Data (Nodes)
            1. Related sets of information
         2. Modules
            1. Functional plugins that shipped with Drupal or were created by the community
         3. Blocks and Menus
            1. Blocks- Provide output to user
            2. Menus- Provice Navigation
         4. User Permissions
         5. Template
            1. The site Skin (CSS)
      3. The structure provides a directional flow of data
      4. Provides Themes
      5. Easy to Collaborate and Promote unity on large sites at OSU
4. What type of database system are you familiar with? How was this system used
   1. MySQL
      1. Relational Tables
      2. Utilize Keys
   2. Two Examples
      1. Work
      2. ShareShare
         1. Join for Friends to make sure they are still active (double check)
5. Why would you use an API? What are the pros and cons?
   1. Pros
      1. Ability to allow multiple devices to talk to each other
      2. With JSON an elegant way to transfer data
   2. Cons
      1. Security
      2. For ours we are using OAuth2
6. Is there a particular framework/system that you enjoy working with the most? If so, what is it and why do you enjoy it? What are your criticisms of it, if any?
   1. NPM
   2. UI Kit for front end
   3. Like
      1. Prebuilt and tested code
   4. Dislike
      1. Any Framework will often conflict
   5. React
      1. Beautiful
      2. Elegant
      3. Eliminates complexity
7. What is the benefit to using a code management system. What is one you've used in the past? Are there any pitfalls to using this system and, if so, what?
   1. Pros
      1. Allows teams to work easily together
      2. Can Create Branches for new features
      3. Easy rollback especially with error prone development
      4. Combine with Slack
   2. Used
      1. Git
   3. Pitfalls
      1. Can be complex to set up
8. Are you familiar with mobile first design? If so, what are your thoughts on this technique?
   1. Feel strongly at OSU majority of traffic was mobile
   2. Poorly designed mobile sites are everywhere
   3. Really Like Youtube
      1. Both Desktop and Mobile
   4. Important not to go to mobile (in my opinion)

**Questions**

1. Day to Day
2. Work Life
3. Current Projects