**OSU Extended Campus Web Developer Application Response**

I am writing about developing a web interface for an instructor to display digital images and 3D models. There are many different methods, tools, libraries and frameworks to solve this type of problem and this is just one potential way to do so. Normally, I would also do my best to leverage existing code from the Extended Campus and OSU. Drupal is used heavily at OSU but we will assume that Drupal will not be used for this example.

**Step 1: Initial Design and Analysis**

1. **Information Gathering**
   1. **Backend:** The first step I would take would be to speak with the Instructor and gather information related to his expectations for this project. Right away I would be seeking to understand his goals and to get a general feel for what he is hoping for.
   2. **Front End:** I would also seek to gain an understanding of the look and feel they are going for. I would bring samples of similar sites and OSU sites to show and make sure the Professor has an idea of where the project is heading in regards to visual aspects and user interface. I would also sketch up a very broad responsive site layout with pencil and paper containing sections, articles, menus and headers.
2. **Initial Application Design** 
   1. After gathering all the preliminary information on the backend and front end I would begin to layout the logic of the site from a very high level. For instance, what classes and objects might be integrated. One initial class might be a file class that can handle and process file upload and storage to the server.
   2. The design stage would also look at site design for the client and server side logic. This would include looking for coding patterns that can be bundled into functions.
   3. Initial design would also begin to focus on code reusability which is important if the site grows, or is adopted by other faculty in the Department or Campus Wide.
   4. This design would most likely utilize the MVC design pattern which is crucial to keeping the view level not tied to the underlying logic of the site. This would allow changes and additions with the minimum level of hassle.
3. **Discuss building a web application interface for future projects**
   1. I would also look meet with staff and discuss the need for meeting the initial deadline but developing a web application so that future versions could support native (or non-native) mobile devices and tablets. This would be different then responsive web design which would be incorporated for the first version of the site. I would most likely support this through developing a RESTful API utilizing JSON.

**Step 2: Create Project Timeline**

1. Three months could be a potentially quick timeframe for a project depending on the project team size therefore the timeline would be rather brisk. I would draw on aspects of Agile Development as we stay in communication with project leaders including the Professor, Fisheries and Wildlife Department and the Extended Campus.
2. The suggested timeline would be the following:

|  |  |
| --- | --- |
| **Task** | **Time Allotted** |
| Application design | 1 Week |
| Development | 7 Weeks |
| Testing and Finalizing | 3 Weeks |
| Deployment | 1 Week |

**Step 3: Design and Application Development**

1. Frameworks, Libraries and Tools
   1. Version Control
      1. Most likely would use Git and either GitHub or BitBucket as a repository
   2. CSS
      1. Implement existing OSU CSS style
      2. Most likely utilize SASS
   3. Server
      1. Dependency Manager
         1. Composer for PHP
      2. Framework
         1. This would be something I would discuss, but with PHP I would lean towards the Laravel framework. It has a large community and provides a nice structure for developing a RESTful API. Another nice feature is its database migrations which would be useful if the project and team size grows.
   4. Client
      1. Dependency Manager
         1. NPM for JavaScript
      2. Framework
         1. I am open to any but if I were tasked with choosing I would go with React JS with a Flux architecture or Angular with a MVC architecture.
         2. React really excels at creating fast single page web applications. Given the short time frame I would lean towards Angular due to the fact that it provides the MVC and not just the View Component like React does.
2. Database Design
   1. For handling the files, I would build the MySQL tables to handle links to the file storage system. Security would be important and issues like SQL injection would all be considered.
   2. The actual files would be stored in a separate place from the site code.

**Step 4: Testing and Launch**

1. The final stage would be a launch where the system would go live and be tested. With personal projects I have found utilizing a number of testers with a number of devices will really flush out small problems and issues. In my timeline this phase has one fourth of the development time because this is the stage when the project can really be tightened and last minute issues ironed out.