1. The web application must use HTML 5 and CSS for page content and layout. Pages must be properly formatted using the HTML 5 specification. Tables are not to be used for general content layout. Tables can and should only be used for tabular information. CSS must be used for styling the content and doing the visual layout. **An HTML5 DOCTYPE is required with the 5 required tags.**
2. The pages/sections that make up the web application must have a consistent design/interface. There should be elements of each page/section that they share in common such as a header, menu, footer, etc. You have flexibility in how you implement your design, but it should not just be a random set of page/section designs. The user should have a consistent and understandable experience when moving from page to page, section to section, or application function to application function.
3. The web application must be well-structured and logically organized. Changes to common elements in a page design must be easy to implement. A common element is an element that is repeated on multiple (or all) pages/sections such as a header, menu, and/or footer. **If a common element has to be changed by editing it in multiple places in the code, then your implementation is not correct.**
4. The web application must have content or functions that are publicly available and content or functions that can only be accessed if authenticated (logged-in), public and private content. When a user is logged-in they must have some visual cue that indicates they are logged-in. The ability to logout must be available. After the user logs out or if they never log in, they must not be able to access the protected pages or functions.
5. For the purpose of testing the login and accessing the protected content **the following user ID and password must work for general access to protected content**:
   * **User ID: test**
   * **Password: pass**
   * If you need to implement a separate login for administrative features or a different category of user, then supply the login credentials that are necessary to perform the login.
6. The web application must utilize PHP and proper PHP techniques shown in class.
7. **You must properly use GET andPOST**. Using GET for private information is not acceptable. GET should not be used to take an action such as deleting information or submitting a password. Remember that GET places the information in the URL!
8. **The web application must use form elements beyond what is needed for a login form.**
9. Any place where users can provide input you must supply appropriate and informative feedback if the information entered is not complete or correct. For example, if the user provides incorrect login information they should receive feedback that the login failed...not just be re-presented with an empty login form with no message.
10. The web application must contain a page where there are multiple photos presented on the page.
11. The web application must contain a page that contains a YouTube or another video embedded in the page.
12. The web application must utilize JavaScript and proper JavaScript techniques as shown in class.
13. The web application must utilize jQuery and proper jQuery techniques as shown in class.
14. The web application must utilize jQuery UI or Bootstrap interface elements.
15. The web application must utilize AJAX. AJAX can be implemented using jQuery or the capabilities provided by JavaScript.
16. The web application you build is not to be trivial in simply meeting the technical requirements set forth in this document. Yes, you are to meet the requirements but you are also to build a web application that has a purpose and delivers functionality or capability. The requirements are parameters to be used in design and implementation of the application; they are not intended to be the end product. You should build a web application that you would be happy to show to a prospective employer or client. You should also make sure that you can complete the development by the due date.
17. The five tags which are DOCTYPE, HTML, Header, Title, and Body are in my index.php
18. I use a menu bar throughout the webpage to have a consistent design interface specifically lines 79-93 in index.php
19. I use multiple templates such as Template.php
20. For public content users can learn about all the different houses of the Game of Thrones universe(Stark.php), but they must be logged in in order to vote on their favorite house(Success.php lines 90-98).
21. Lines 104-125 in index.php show how to properly log in and take you to success.php on a successful login
22. Examples of PHP are all over my final project such as lines 32-85 in martell.php
23. For POST line 116 is used for login information and GET is used in line 139 in order to get the youtube video to show up
24. In lines 90-99 of success.php I created a form in which a user can choose which house is their favorite
25. User feedback is giving a lot such as line 90 in success.php in which a message is shown of “ thank you for submitting” after a user submits an answer
26. In Stark.php more than one photo are shown on the page at the same time
27. In index.php lines 135-152 a youtube video is shown
28. A little JavaScript is shown in line 85 of Stark.php by switching two images on a mouseover/mouseout
29. Some Jquery is used in lines 8-20 of index.php
30. I use bootstrap in lines 4-6 of index.php
31. Some AJAX is used in the getJSON function of index.php
32. This web application is not trivial because this webpage helped me show my passion for Game of Thrones#WinterIsHere