Technical requirements for the final project:

1. you must work on your project on your own (no group projects)
2. it must be a full web app, with html, css, javascript, php. Focusing only on the html and css would not suffice.
3. it must work with files, preferably json-formatted ones. You may NOT use databases for your project.
4. it must have a coherent theme, not be just a collection of technically competent pages
5. it must have users (with Register and Login menu options, and an Admin site available only to the system administrator).
6. it must store data for each user (could be files the user uploaded, comments the user has made, personal information about the user, etc) and each user must be able to manage/edit their data in some way.
7. it must be dynamic: pages must change via interactivity with the user or when changes happen on the server.
8. it must have parts that are constructed dynamically without hardwired limitations, such as any number of rows in a table (not a hardwired value), any number of images (not just a hardwired number) etc.
9. it must include a feature (e.g from W3schools) or optional topic in Web programming that you researched on your own, understood it and integrated it.
10. it must include some original code in javascript and php, or code that was adapted constructively. You must acknowledge all code that you borrowed or adapted from other sources.
11. it must have some kind of Javascript graphics (besides included images). Optionally, if this is what attracts you the most, it can primarily focus on dynamic javascript graphics, e.g. implementing some demo code in javascript (similar to the "moving points" assignment and the Game of Life assignment; or, for example, an adaptation of a demo "game" available on W3schools demo site, etc.).
12. it must have some interactive features (buttons, forms, etc.) which implement some local communication (on the client, in javascript, e.g. for changing the way something is visualized in the browser) and others that implement communication with the server (php, e.g. those that send to or request some info from the server).
13. clean and well documented code, both in terms of organization of files (separate folders for css, js, php, organized in possibly several files with functions that are logically grouped together) and of code (separate functions, working on parameters that are passed to them and using local variables; well-motivated use of global variables; suggestive names for variables and functions and reasonable amount of comments when the good names are insufficient to describe what a specific function or piece of code is doing, etc..)
14. In the remaining weeks of the semester, you must submit the current status of your project by the usual homework deadline. During every class you must be prepared to present what you have done and where you are.

15. Your in-class presentations during Weeks 12-14 (for the current status of your project), and your final in-class presentation on Week 15 are part of your final project grade. You should be able to answer any questions about decisions you made and about how you implemented the different parts of your project. Missing one of these presentations or failing to submit the current version of your project for any of these weeks automatically lowers your grade for the final by 10%.
16. You have until the end of the final exam period to clean up and debug your code. You will submit the final version in your code at the final project submission site.
17. Your final version of your final project must work. If, for instance, your php login does not work and that makes the rest of your site inaccessible, we will not be able to grade the rest of your work and you risk receiving a failing grade. If you know your login does not work, submit a version of your project without it so we can test the rest of your code. However, you will lose points for not having a functioning login.

Creative features (examples):

- a well crafted "story" that your web app is telling, i.e. a coherent theme or application domain.
- a creative integration of the required technical requirements
- an aesthetically appealing look. But: do not spent an excessive amount of time on this aspect; you are absolutely allowed and encouraged to research through the W3schools HowTo demos and use or adapt their css styles.
- a user-friendly user interface, with features that are either self explanatory or include hints or directions for how to be used.