ED11D-0916

Developing Free and Open Source Interactive Teaching Tools

Ethan L. Nelson, University of Wisconsin–Madison

ethan.nelson@aos.wisc.edu

Motivations

- Universities often adopt course content management systems (CMSs).
- These CMSs provide an avenue for instructors to disseminate course materials, proctor exams, and add discussion aspects to courses.
- While they are useful for enabling distance education courses or supplementing traditional teaching styles, they are usually "black boxes."
- This stems not necessarily from being closed-source but rather by offering very limited customization for instructors.
- Self-developing online teaching tools provides the flexibility necessary to create novel supplementary teaching tools, and it can be accomplished on a limited to null budget with use of open source technologies.

Site Construction

- The figure on the right provides an overview of the infrastructure components necessary to host a teaching tool with example implementations.
- Generally, you need a server, a web framework, and a user interface.
- Web pages are dynamic if content depends on user input, elsewise static.

Static Sites

- Static sites are a collection of pre-generated web pages and files; everything must be rendered ahead of time, but this is only done once.
- Generation of static sites can be completed on a computer separate from the server (for examples, figures from a climate model).
- The key limitation to static websites is limited customization from the server side. Rather, everything must be manipulated from the pre-generated state on the user side.
- Finding a host for static sites is much easier. Universities often offer free web space for files, but sites like GitHub will also host static sites for free.

Dynamic Sites

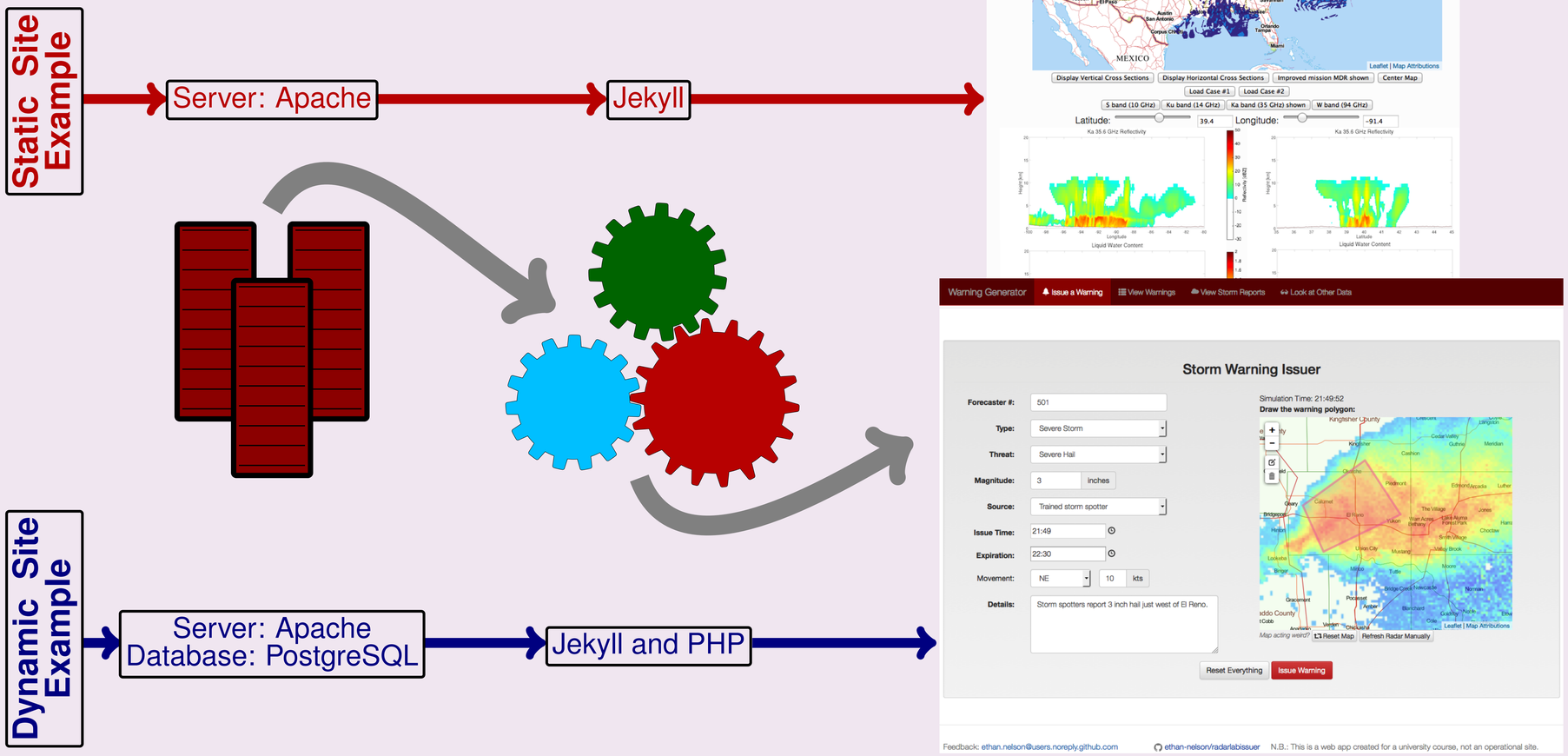
- Dynamic sites require a server that takes some user input, possibly queries a database, assembles a custom page, and serves the result.
- An understanding of web development and relational databases is usually necessary for dynamic sites, but this can be overcome by consulting online guides or collaborating with others.
- Alternatively, web app systems like Python Django abstract away many of these components so you can focus on the web pages.
- Dynamic sites require dedicated computing power to render the queries.

Concluding Remarks

- While a custom solution requires some time investment, a teaching tool built from the ground up provides for complete flexibility.
- Distributing the underlying code base for a teaching tool online through services like GitHub provides the opportunity to collaborate with others in development and improvement.

Acknowledgments

Thanks to NASA HQ under the Earth and Space Science Fellowship Program Grant #NNX14AL35H for providing educational funding. Also, thanks to Tristan L'Ecuyer for allowing me to implement these tools in his radar and satellite meteorology course. Finally, thanks to the Delta program at UW-Madison for focusing my attention on the teaching-as-research process.



		Server	Web Framework	User Interface
Static	Description	Computer to serve pages to internet	Templates page layout	Page interactivity
	Technologies	Apache, Nginx	Jekyll, Markdown	CSS, Javascript
	Solutions e.g.	GitHub Pages, university web hosting, self-hosted		Custom, Bootstrap, Foundation
Dynamic	Description	Computer to serve pages and host database	Templates page layout and compiles page	Page interactivity and communication with server
	Technologies	Apache, Nginx, or high level language like Python; "backend" database: PostgreSQL, MySQL	Wordpress, Drupal, Jekyll, Flask, PHP, Pyramid, Django, Ruby on Rails	CSS, Ajax, Javascript, HTML forms
	Solutions e.g.	Heroku free plan, AWS education, self-hosted		Custom, Bootstrap, jQuery, AngularJS

Code for these examples is available at `http://github.com/ethan-nelson`.
This poster available at `http://ethan-nelson.me/AGU`.