The Development of an Academic Library’s Mobile Website

Junior R. Tidal
CUNY New York City College of Technology

How does access to this work benefit you? Let us know!
Follow this and additional works at: https://academicworks.cuny.edu/ny_pubs
Part of the Library and Information Science Commons

Recommended Citation
THE DEVELOPMENT OF AN ACADEMIC LIBRARY’S MOBILE WEBSITE

CHAPTER 2

Junior Tidal

Introduction

Mobile devices have become more ubiquitous among academic library users. It’s now common to see through analytics that smartphones, tablet computers, e-readers, and even portable gaming consoles are connecting to online library services. This chapter is a case study of how a small academic library supported its users through the creation of a mobile-optimized library website. It documents the chronological changes from the website’s humble beginnings on a shared Windows IIS server to its current configuration on a Linux-based cloud server. Throughout its existence, the website was developed with adaptability in mind, and flexibility, in order to respond to unpredictable changes in information technology. This adaptability includes not only the changing landscape with regard to standards for mobile website development, but also changes in user preferences over time with regard to devices and website architecture.

Literature Review

Content management systems (CMSs) have changed the way that libraries have crafted their online identities. These systems centralize webpage content creation

* This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License, CC BY-NC-ND (https://creativecommons.org/licenses/by-nc-nd/4.0/).
and allow for sites to be better organized. Libraries have employed CMSs to facilitate the presentation of complex websites in a user-friendly fashion (Black, 2011). A CMS can also put content production in the hands of all librarians and not just a single “gatekeeper,” flattening technological hierarchies of control. This distributed system of creation was important for the Darian Public Library’s website, where Drupal was selected to support its active internal blogging community (Sheehan, 2009). Another library with goals similar to our own, the library at the University of California, Santa Cruz, adopted Drupal because it could bring consistency to older webpages and be used to remove defunct and redundant webpages (Hubble, Murphy, & Perry, 2011).

Libraries have crafted mobile library websites to reach their users. Librarians and library developers who have studied users so that they can meet their organization’s specific needs explain that understanding your user base will help in designing a mobile site that will be “heavily trafficked by your users” (Bridges, Rempel & Griggs, 2010, p. 318). Needs assessments have also been conducted through surveys to gather users’ preferences (Dresselhauls & Shrode, 2012; Cummings et al, 2010). Preliminary research on device usage has been conducted to justify a mobile library site (Wilson & McCarthy, 2010).

In discussing mobile web development for libraries, it is difficult not to touch upon the adoption of responsive design. Glassman & Shen said it best, stating that the “mobile web has been replaced by the responsive web—one site fits all” (2014, p. 89). Responsive design is a practice where websites are coded to conform to the screen size of the device being used to view it. The page adapts (or responds) to a layout specific to these dimensions. The flexibility of responsive design makes updating a library site easier and makes the site functional regardless of the device it is viewed on (Kim, 2013).

Background

The Ursula C. Schwerin Library serves the research needs of the New York City College of Technology, City University of New York (CUNY), located in downtown Brooklyn, New York. Locally known as City Tech, the college supports over 17,000 students in associate and baccalaureate programs across a wide variety of programs. Students have access to over twenty CUNY campus libraries throughout the five boroughs of New York City, in an arrangement similar to that of a consortium resource-sharing library system. The library’s integrated library system, monographs, and some electronic resources are centralized.

The student population of City Tech is quite diverse. Almost half of the student population was born outside of the United States. The majority of the student population identify as black, Hispanic, or Asian. Over 60 percent of students speak a language other than English at home. Most freshmen receive need-based
financial aid. The college can be characterized as a commuter school with many students working full- or part-time jobs, with a significant number of students being enrolled in continuing education programs.

History of the City Tech Library Mobile Site

To understand the development of the library’s mobile site, it is best to first examine the creation of the main library website. The library’s homepage (https://library.citytech.cuny.edu) was first hosted on a Windows IIS server in 2002. The site consisted of several static HTML pages that were manually edited using Microsoft FrontPage. The maintenance of the library website reflected its numerous curators, as the work was shared between the technical services librarian and the multimedia librarian. There was a lack of uniformity not only in the aesthetics of the site, but in its information architecture as well. Even though the site had its own URL domain, pages were not properly grouped in appropriate directories. Numerous pages were orphaned, contained broken links, or were simply outdated.

Workflows for creating and amending content were cumbersome in this early iteration of the library website. Librarians had no direct access to the site through a browser. Instead, content was e-mailed to the web services librarian, marked up in either FrontPage or Adobe Dreamweaver, and then uploaded via FTP to the college’s server. As the amount of content grew on the library website and services such as electronic resources began to proliferate, it was apparent that the web server was in dire need of upgrading. There were more demands for space and for processing power to serve webpages.

The library web server, hosted by the campus Computing and Information Systems (CIS) department, was shared with several other City Tech academic departments. Due to this configuration, a number of security restrictions were implemented by CIS. Many types of software were prohibited from being installed on the server, as database software and web scripting languages were seen as possibly disruptive to other departmental websites. This prevented the installation of various programs, such as CMSs, analytics tools, and web form processing. Shell access outside of the college’s IP range was unavailable, making updating content or repairing server problems from off campus impossible.

Acquiring a Server for the Library

In 2007, after a written proposal from the Web Services and Multimedia Librarian, and the Chief Librarian, the provost allowed the library to procure its own web server. This Dell server ran on a Red Hat Enterprise Linux (RHEL), Apache,
MySQL database, and PHP configuration also known as a LAMP setup. This is a popular web server configuration, utilizing open-source technologies. Red Hat is a specific distribution of the Linux operating system. Apache is widely used open-source software that delivers websites. MySQL (server query language) is a relational database language, and PHP (Hypertext Preprocessor, a recursive “backronym”), a web processing language.

This new server configuration lifted the restrictions that had impeded the previous shared server. With access to PHP and MySQL, the library website could now implement blogs, CMSs, web analytics, and server log reports. A PHP script was written to display uniform headers and footers across the site. Navigation menus and library information were consistently displayed on every page of the site; lack of consistency was a problem that had plagued previous iterations of the library website. More importantly, the implementation of this new server had opened an opportunity to create the library’s first mobile website.

**Mobile Site 1.0**

Based on information gathered from analytics data, we found that mobile device connections to the library website were increasing. It’s important to note that these visits were occurring prior to the release of the first generation iPhone in 2007. Some of the devices connecting to the site were cellular phones with web capabilities. Surprisingly, some of these devices were portable gaming consoles that had Wi-Fi and web browser capabilities. Concurrent with this activity, electronic resource vendors were beginning to release light mobile versions of their products. These combined factors spurred the library’s first attempt at developing a mobile webpage.

The first mobile page for the City Tech Library consisted of static HTML pages. It was developed in a text editor and contained neither CSS declarations nor images. This first page was very basic and contained only three links: to the library’s shared CUNY-wide catalog, to a page containing electronic resources, and to a page containing library contact information and hours. The page displaying electronic resources was limited to those resources that were mobile friendly. This first mobile site was hosted at the URL http://library.citytech.cuny.edu/mobile. A PHP script placed on the homepage of the library website redirected users to this page based on the user’s browser agent type. User testing was not conducted in the creation of this first mobile site. This first mobile page on the library website was short-lived for a number of reasons. First, the PHP script to redirect users to the mobile site didn’t always work. The script functioned by detecting what kind of browser a user was using to visit the site. Unfortunately, some users logging in with a cell phone browser were not redirected to the mobile site. Second, this agent detection script required constant maintenance. New cell phones that were web-enabled were being released, as well as a plethora of other various mobile
devices and browsers, all of which needed to be added to the redirection script manually. At the time of this first mobile site’s existence, no statistics were taken as to how often the page was visited. This was problematic as there was no data to support the continuing existence of the page.

**Drupal**

In 2010, the library migrated away from its custom PHP-scripted setup and to the Drupal 6 (D6) CMS. Drupal (http://www.drupal.org) is a modular, open-source CMS that utilizes MySQL and PHP to manage web content. It is modular in the sense that there are numerous modules, similar to WordPress plugins, that can enhance the functionality of D6. A large community of developers is creating these modules, including a few who have ties to academic libraries. Since these components are open-source, a library site can be fully customized to meet specific user needs. Blog and calendar feeds can be aggregated from other sources and displayed on the library website. The presentation of content is customized through CSS scripts and PHP code. Code of an existing module can be modified to render an appropriate display of that aggregated content. For example, a series of library events from a Google calendar can be aggregated, parsed, and reproduced in a table form on the site. Finally, one of the more powerful aspects of D6 was the ability to modify content within the browser. This is probably the most practical reason to adopt a CMS. It may seem ubiquitous among content management servers to provide this today, but at the time, this feature was very useful for librarians with different levels of technology experience. Users could simply log into the library website via the browser and update content as needed.

Using a CMS for editing was very different from updating HTML pages or PHP scripts, which previously required a connection into the web server through a shell and then editing the scripts through a command-line text editor. The library’s electronic resources had been kept in a flat-file database. This flat-file database was simply a text file that contained information about a single electronic resource per line, including the URL, the name of the resource, the date added, and a description. The electronic resources librarian would have to manually enter each resource line by line to update it. This process was prone to numerous errors. Existing resources required a search and find command to make any changes to the resource’s URL, name, or description. Through D6, the workflow to update these databases was significantly improved. Now, librarians could log into the site through their mobile device or workstation and make changes faster and more efficiently through the web browser. Electronic resources were stored within Drupal’s custom Content Creation Kit module, which was much more user-friendly than editing flat files.

At the time of this Drupal migration, analytics tools were used to gain insight into our users and their behavior. Log server files, Google Analytics, and
its open-source alternative Piwik were all used to allow us to better understand visitors to the library's site. These tools were implemented using Drupal modules designed for this purpose. This analytic data has been used to drive the development of the mobile website (Tidal, 2015). For instance, analytics can show what devices are connecting to the site, how fast the connection is, the network provider, and a masked IP address that can provide geographic locations. The site can then be tailored to these factors by approaching site content from a “mobile first” perspective. “Mobile first” is a practice where content is developed from the perspective of a small screen. The design, layout, and content are constrained and, as screens get larger, are amended with more features. This also impacts bandwidth, as constraining file size to improve website performance on slower cellular broadband networks can be factored into the creation of the site.

Although mobile device visits to the library website make up a small percentage of overall traffic, their number has increased steadily since 2011. From 2011 to 2016, mobile traffic increased from 1 percent to 7 percent of overall traffic. This is reflective of overall trends of cell phone ownership. The Pew Internet Research and Life Project reports an increased number of cell phone owners, surpassing that of desktop and laptop workstation owners (Smith, 2015). Many of these owners also use cell phones as their primary access point to get online (Smith, 2015). This trend required the City Tech library to respond by creating another iteration of the library mobile site.

**Mobile Site 2.0**

The second iteration of the library mobile site was developed in 2012. Hosted on the same server as the main library website, a virtual server was configured in Apache serving this separate mobile site. The mobile and desktop sites’ file structure were both contained within Apache’s default /var/www/html directory. Yet the mobile site was contained in a subdirectory aptly labeled `mobile`, whereas the desktop site was contained in a subdirectory called `libSite`. This set the foundation for two separate Drupal installations.

This version of the mobile site also used a PHP redirection script provided through a D6 module. It was similar to the script used in the first iteration of a site: if a user visited the library website using a mobile-enabled device, it would redirect the user from http://library.citytech.cuny.edu to http://m.library.citytech.cuny.edu. The design of the site was also user-centered from the ground up. For instance, because we simply added the letter `m` to the virtual domain of the mobile site, users would need to type less within the browser’s address bar to get to the mobile site. This made the site more usable compared to a site that simply amends the mobile site’s location as a subdirectory, such as http://library.citytech.cuny.edu/mobile. It also reinforced the fact that the mobile site was separate from the desktop one.
Even though the main library website was utilizing D6, Drupal 7 (D7) was chosen to manage the mobile site. D7 was chosen over D6 because it supported a number of modules and themes that specifically optimize webpages for mobile devices. Coincidentally, a number of D6 modules were unfortunately unavailable for D7. This had little bearing, however, on the mobile site since it was intended to be minimalist in design. It was preferred that the mobile site have a small footprint to accommodate mobile users connecting through cell networks or spotty Wi-Fi connections.

One D7 module that was used to accommodate this smaller footprint is the popular jQuery Update (https://www.drupal.org/project/jquery_update). jQuery is a JavaScript-based library used to easily implement JavaScript in a webpage. It is useful in navigating the components of a webpage as well as enhancing a page’s functionality. It is cross-platform-compatible so that any device can process its scripts. These scripts allow a device to display animation, process event handling, and navigate a webpage’s source document object model (DOM). The scripts are especially useful in providing feedback to the end user viewing the page. Unfortunately, there is no support for later versions of jQuery for sites using D6. This is not the case for D7 installations. This was very important because of the intended use of jQuery Mobile, a separate JavaScript library derived from jQuery. jQuery Mobile, as its name implies, supports mobile devices. It gives developers the tool kit to design touch-based interfaces, responsive websites, and applications.

A D6 module was installed on the desktop website to detect mobile devices. This module redirected users to the mobile site if they were detected using a mobile device. It was much more efficient than the in-house-created PHP script in the first mobile site iteration. A link to the library’s desktop site was also present in the footer of each mobile page. Members of the library’s website committee felt that it was necessary to have this link in case there were online transactions or tasks that could not be accomplished through the mobile instance. Although the D6 desktop site was not mobile-optimized, users were still able to access content not found on the mobile site.

The content of the D7 site (figure 2.1) was based on the most popular pages visited on the main library website. This data was collected through web analytics tools Piwik and Google Analytics. The popularity of these pages dictated the order in which links appeared on the page. The webpage’s header contained the library’s logo with the current day’s operating hours underneath. Since the “Hours” was the most visited page according to analytics of mobile devices, it was given the greatest priority in the visual hierarchy of the page. However, analytics tells only part of the story. In order to see how effective changes were based on analytics data, we later conducted a usability study to confirm whether or not our changes aligned with user preferences. Log files and analytics data alone don’t necessarily justify design decisions.
Links on the new mobile site were separated into groups of large buttons. The first version of this D7 mobile site contained a link to a page with library directions. A second set of buttons immediately below the first included links to a catalog search, a page containing mobile-optimized electronic resources, and a linked button leading to electronic books. The last button group contained links for reference desk contacts and a contact button for the library’s circulation services.

A PHP script was developed to manage electronic resources between the mobile and desktop versions of the site. To alleviate the workload of managing two sites, this automated script would replicate the electronic resources found on the desktop site, parse them, and display the results on the mobile version of the page. Only resources that contained a mobile version would be replicated on the site. This was especially useful at the time when e-books were increasing in popularity (Tidal, 2012).

The layout of the mobile site was designed to be responsive to screen size, since jQuery Mobile is inherently responsive. Large buttons that were easy to tap were employed for smaller screen sizes and resolutions. This approach also supported mobile device users using touch interface because if links are too small, users could possibly have problems accessing the links with their fingertips. There was also a lack of images to create a lightweight site that focused on loading in a speedy and efficient manner.

**Evaluating the Mobile Site**

After the mobile site was launched, a usability test was conducted during the 2013–2014 academic school year (see Tidal, 2014). The objective was to ascertain
The Development of an Academic Library’s Mobile Website

if the mobile site could support students’ research needs. In order to entice volunteers, users were given campus bookstore gift cards as incentives to participate in the study.

A cognitive walkthrough was used as the evaluation instrument for the mobile site. This was moderated by a testing proctor and digitally recorded. Participants were asked to complete a series of task scenarios derived from usability goals. For instance, one goal could be users being able to find a book on the library website. A task scenario would be written to place the user within a realistic context to achieve this goal. For example, a task could be “Imagine you are in an English class and you need to complete a research paper on the book *The Hitchhiker’s Guide to the Galaxy*. Using the library website, find the call number of this book.” Here, the scenario gives the user some context for completing the usability goal and is a call to action to complete the task using the site. Users are asked to think aloud as they attempt to complete the goal while their responses are recorded. This process provides proctors a glimpse into the thinking of the users as they interact with the website.

Participants in the usability test consisted of a variety of students: ten students, five for the first round of usability testing and five for the second round of testing. This approach was modeled after the common practice of using a limited number of participants (Nielsen, 2000). The basic idea is that having more than five users test the site won’t reveal any new usability problems, but will simply confirm the existence of issues that have already been identified by previous test participants. However, this assumption has been contested (Faulkner, 2003; Bevan et al., 2003). The sample included traditional two-year and four-year students, as well as nontraditional, or continuing education, students. Prior to the study, a screening survey was given to participants, asking which devices they have used. This allowed them to be paired with tablets with which they were familiar so that students using devices with which they were unfamiliar would not taint the test.

After the first round of testing, the mobile site was restructured based on the data collected from participants. Notable changes included the addition of a search box that takes students to the catalog instead of to a separate page accessible through a hyperlink. Numerous participants found it more useful to have this search box, rather than a link, on the mobile site homepage directly. Participants also revealed that students preferred having a button to take them directly to their library account to view loaned items and renewals. This option was not available on the site, requiring participants to log in through the catalog’s search screen. Electronic resources were also organized by device type so that users of Apple or Android products could select which page suited them. Links on the homepage were also rearranged. The Directions page link was moved to the bottom of the page, as it was deemed the least important page by participants.

Even though the site was restructured, the overall usability of the mobile site was well received during usability testing. Participants in the first and sec-
ond round completed tasks at high rates of success. The thinking aloud protocol (TAP) metric gave further insight into users’ reactions of the site. TAP, also known as the think aloud method, is simply where users speak their thoughts aloud while performing various tasks. Lewis and Rieman, who were the first to employ the method, stated that when testing a design, users’ “comments are a rich lode of information (1993, p. 83).” Users express what they are doing, thinking, and feeling, which provides feedback to observers. This process not only helps identify usability issues, but helps to pinpoint potential obstacles as well.

The TAP protocol for this study provided librarians with information on how to improve the site (figure 2.2). Users responded that the site was easy to use and that the site was pleasing to the eye. Conversely, they also noted that the site failed in their expectation of its having spelling correction when using the site’s online catalog.

**FIGURE 2.2**
Screenshot of the D7 mobile website after usability testing. Note the changes in comparison to figure 2.1.
The Mobile Site Gets Attacked

During the latter half of 2014, the mobile website went down following a malicious attack on the library’s Drupal server. A hacker vandalized the mobile site, and the homepage was replaced with online taunts. The library’s mobile site was not the only one affected. This security vulnerability compromised an estimated 12 million sites running D7 (“Millions”, 2014). As per university protocols, the server hosting the site was disconnected and the site was taken offline. This resulted in the whole library domain going dark as the library’s multiple sites were hosted on a single machine. However, disabling the mobile site rectified this problem. This attack ended the last iteration of the mobile site before we moved to a responsive design solution.

Transitioning to a Responsive-Design Website

Even though mobile users make up a small portion of overall visitors, it was important to support that minority group of users. In response to the recent hack of our mobile site and the demand for mobile support, the possibility of implementing a responsive-design website was explored. At the time, numerous other academic libraries were adopting responsive-design websites as a way to cater to both mobile and desktop users. This notion of providing a more customized experience for users on different devices was extremely appealing to the City Tech Library team. Despite homebrewed PHP scripts that parsed content, it was very difficult to maintain and update content for both the desktop and mobile sites. This amalgamation helps to prevent posting inconsistent content. Through the consolidation of the desktop and mobile sites into a single responsive site, librarians can actively add more content without switching between the two.

During the 2014–2015 academic year, a prototype of a redesigned website was developed. This version employed responsive-design techniques. Responsive design is a technique where a website presents a customized experience for the user, tailored to the device used to visit the site. A website will be displayed with the same content and a similar layout on a desktop, smartphone, or tablet computer. This is accomplished by CSS style sheets, rules, and declarations, which are triggered by specific device widths. For instance, a device width of 480 pixels, common for smartphones, activates CSS rules that cause the site to “respond” to that width. Based on the detected dimensions of the screen, the site will shift its layout and conform to the display of the device. Altering the browser window size on a desktop workstation will also cause the site to conform to the new window width.

Responsive design alleviates the problem of device detection. Detecting what type of device is being used to access the site through agent requests is not perfect. The previous site’s detection methods, as previously stated, were problematic. Not
only will responsive design overcome such shortcomings, but it will also cause less of a dependence on updating the ever-growing list of devices that can be used to access the mobile Web. Responsive design enforces consistency between the mobile and desktop site, and this encouraged the library’s adoption of D7. At this point modules used in the D6 version of the library website were available for D7. This hindered the early adoption of D7. Alternate modules to accommodate the D6 installation were developed and implemented in this new D7 site. Different modules and themes were tested to use responsive design for the prototype site.

The most successful theme is a port of the popular web framework known as Bootstrap (https://getbootstrap.com and https://www.drupal.org/project/bootstrap), which was developed by two Twitter developers, Mark Otto and Jacob Thornton. This free, open-source framework contains several useful features. One feature of note is the framework’s contribution to the rapid deployment of a website’s front end. Numerous built-in features commonly found on websites, such as form controls, buttons, and navigation components, were readily available out of the box, and so additional code wasn’t necessary to create them. Bootstrap is also cross-platform-compatible for a wide range of devices. This is important, because different devices use a wide variety of web browsers. An Android tablet may render a webpage much differently from an iPhone. An Apple MacBook running the Firefox browser may display web content differently from a Linux workstation running the Opera browser. A cross-platform web framework displays webpages as similarly as possible across this wide range of devices. Bootstrap is built with responsive design, so it runs on a wide range of devices of various screen widths.

Adopting responsive design didn’t stop at the redesigned website prototype. At the time, the library website housed three WordPress installations. This included the library’s blog, an orientation site for incoming freshmen, and a newsletter site updated each semester. All of these sites under the library’s domain were updated with responsive-enabled themes. The library staff intranet was initially a MediaWiki installation but was also converted to a mobile-friendly WordPress instance.

Although D7 and its modules made the site more accessible for mobile and desktop users alike, the transition process was not without obstacles. Some features within Bootstrap are incompatible with the Bootstrap theme developed for D7. Numerous related open bug issues can be found on the Drupal Bootstrap project site (https://www.drupal.org/project/bootstrap). As a result, extensive theme modifications were required on the prototype site. To better customize Drupal Bootstrap, a “child” Bootstrap theme was created. A child theme is a subtheme of an existing Drupal theme, known as the “parent” theme, that inherits many of its characteristics and functions. This child theme contains updates to the PHP-based header files that controls the elements first loaded into the browser. These modifications include details such as specific div CSS classes to the header’s navigation section, menu modifications, and header spacing. CSS modifications were
implemented to change the positioning of default elements, change theme colors, and tweak element margins and paddings.

In addition to these customizations, the library website also migrated from the City Tech CIS-hosted servers to an Amazon Web Services (AWS) instance in the cloud. There were a number of factors influencing our decision to use AWS. First, the server housed by the CIS department needed an upgrade in any case. Other factors included the increased demand for online library services and the library’s vision of creating a flexible infrastructure that is not only scalable for future technologies, but reliable as well. However, the main reason to use a cloud-based service was cost. Taking into account the amount of traffic the library website had experienced, along with the growth of content, a subscription to AWS was a fraction of the cost of upgrading the library’s web server. There were nevertheless concerns about adopting AWS. CIS wanted to make sure that the library web server was not storing any information that would identify a student, such as Social Security or student identification numbers. The library website collects none of this, so it was given the green light to adopt the AWS platform.

This new web prototype was initially constructed on a Mac Pro workstation. Since OS X is based off of UNIX, the site was migrated to the AWS instance running Amazon’s distribution of the Linux operating system. Prior to this, the site was shared with other librarians in the department, who gave feedback on its design and function. The installation on the Mac Pro continued to be useful even after the site was migrated to AWS. It was possible to test modules or site modifications before they were implemented on the AWS production server. With the prototype hosted on AWS and its modifications complete, the next step was to begin usability testing of the responsive-design website. The first round of testing began in the fall of 2014, and the second round concluded at the end of the fall 2015 semester. In between the first and second rounds of usability testing, the prototype went live in the spring of 2015.

At the launch of this new redesign, there were additions to the library website through CUNY-wide initiatives. The discovery tool OneSearch was launched. This is the CUNY brand name of Ex LIBRIS’s discovery product Primo. Mobile-optimized, OneSearch uses a responsive design to support various devices. It has supplemented the CUNY Catalog, which is the web OPAC for the university’s Aleph system. The catalog itself has also been mobile-optimized by CUNY’s Office of Library Services. However, the library website committee decided that OneSearch would be the default search system for library materials since it is more usable than the CUNY Catalog.

The other change to the library website was the adoption of the popular LibGuides CMS. This off-site hosted solution provides the City Tech community with guides to assist them with their research. It replaced the previous MediaWiki installation that housed research guides on the library server. Echoing the redesigned library website, LibGuides 2.0 also utilizes the Bootstrap framework.
This allowed LibGuides to be customized, mirroring the appearance of the library website. It was as easy as copying and pasting Drupal’s CSS style sheets into LibGuides. (See figures 2.3, 2.4, and 2.5.)

**FIGURE 2.3**
Desktop version of the responsive-design D7 City Tech Library website.
FIGURE 2.4
Mobile version of the responsive-design D7 City Tech Library website.

FIGURE 2.5
City Tech Library's LibGuides 2.0 mirrors the main site.
There were a number of takeaways following the development history of the City Tech Library’s mobile site. These include the need to keep up with emerging technology through active library groups, the need to accept the technological and organizational challenges in developing a website, and the importance of user-centered design. These takeaways could be applicable to any type of library-based technology project. Adapting responsive design to our library website was no small task. A considerable amount of time went into researching responsive design, as well as the Drupal modules that support it. Current practices of other library websites and discussions through various venues assisted in selecting Bootstrap as the framework.

One community that has discussed Bootstrap extensively is code4Lib. code4Lib is a grassroots organization that focuses on coding and librarianship and has a very active journal and e-mail discussion list, as well as national and regional conferences. Some projects mentioned through code4Lib, such as Blacklight (http://projectblacklight.org), utilize Bootstrap. Both the Association of College and Research Libraries (ACRL) and the Library Information Technology Association (LITA) of the American Library Association (ALA) have also discussed the use of Bootstrap on their blogs and e-mail lists. Schofield has written a blog post about how some Bootstrap components are mobile-unfriendly (2014). Keeping up with these professional organizations has helped us to better understand how the library community at large utilizes web frameworks.

One of the biggest challenges of this development project was the attack on the site’s Drupal security exploit. The attack was very sudden and unexpected, but luckily did not occur at a busy time for the library. Not only did the attack help us to evaluate the library’s security and CMS, it was a catalyst to the redesign. Events such as these should be embraced as an opportunity to improve our services.

Lastly, one of the more important highlights of the mobile development project was incorporating users into the design. It is very easy for us as librarians to take on the role of building what we think is best for our users without actually communicating with them. Using the concepts of user-centered design, where users’ input is integrated into each stage of the design process, makes for a more usable website. User input can be the deciding factor in website steering committee discussions, design decisions, and the placement of forms and links. Although I’m not saying that the burden of a website’s design should fall squarely on the shoulders of the end users, it is imperative that they be able to use the library website effectively.

Future Improvements

Future improvements to the library’s website will not only run parallel with advances in mobile technology, but also respond to how mobile users interact with
sites. There are emerging expectations from the mobile Web, such as hamburger menus, long scrolling, and responsive design, that are now making enough of an impact to make them standards. The hamburger menu or sidebar menu is a popular trend in mobile websites. It is typically a button with three horizontal lines, replacing the navigation menu for devices with smaller widths. Since navigation bars can be long, menu items may not be properly displayed. The hamburger menu alleviates this by compressing the items into a single button. However, this menu system may not be feasible for all mobile projects since it takes another interaction to activate the menu. In addition, the icon itself may not be universally understood, one of the pitfalls of using icons on the Web. Still, the hamburger menu is found in many responsive frameworks and themes.

Mobile devices have changed the way that users interact with websites, specifically, how they react to scrolling long pages. The screen size of a mobile device is considerably smaller than that of a desktop. The notion of “the fold” on a website is no longer a factor. Due the constraints of a smaller viewport and the touch interface of the screen, scrolling is a more intuitive interaction with mobile sites. A long scrolling site also pairs well with simple navigation.

Responsive design is obviously one of the more important trends in mobile web development. Frameworks such as Bootstrap, HTML5 Boilerplate, and the recently developed WC3 CSS framework all employ responsive design. Frameworks make implementation of web applications rapidly deployable. OCLC has been utilizing a custom framework, known as CoreUI, to adjust its interface depending on the specific viewport size (Ganci & McCullough, 2015). Other current trends include content-delivery networks (CDNs), which reduce the file size of CSS and JavaScript. CDNs serve JavaScript and CSS files of frameworks so they do not need to be locally stored. This saves on maintenance and tightens version control. CDNs optimizes file size and bandwidth, a key component in making mobile sites less complex (Kim, 2013). CDNs will probably become more popular with distributed web frameworks. Responsive design in this sense will be more ubiquitous and will require even less coding knowledge to implement.

Libraries can currently adopt responsive design for databases and vendors that do not offer it natively. Reidsma describes customizing HTML and CSS for such cases and integrating them into the library’s website (2014). Responsive design may also affect wearable technologies, such as watches and virtual reality goggles. Viewports would scale down to 75 percent on watches, yet may expand to support virtual landscapes. This changes a number of factors, including user interactions, graphics, and content. However, it remains to be seen if libraries should develop for these emerging platforms, as they may not be the best way to deliver content to patrons.

The importance of crafting a satisfying user experience that places an emphasis on making human connections supersedes these newer trends. Mobile devices are becoming more ubiquitous, accounting for 30 percent of all Web traffic
globally (“We are Social”, 2016). Ten percent of Americans rely on smartphones for Internet access, especially minority and low-income users (Smith, 2015). This situation requires libraries to adopt new mobile technologies to serve these populations.

Analytics continues to play a significant role in the development of the library website, but it is not the only way to get insight into user behavior. That said, utilizing analytics information was useful for updating the mobile site’s various iterations. For instance, we could see what mobile devices our users were utilizing to visit the site, and the site could then be catered specifically for this audience. Unfortunately, this has the reverse effect of not optimizing the site for users who connect with uncommon mobile devices. Perhaps by adhering to the appropriate web standards and guidelines, the library’s responsive site can cater to these users as well. Yet, analytics is only one small piece in a larger puzzle. Usability tests and user feedback should be prioritized for future website improvements. Knowing how many users connect to the site with what device does not, on its own, indicate how users are interacting with the site.

References
Dresselhaus, A., & Shrode, F. (2012). Mobile technologies & academics: Do students use mobile technologies in their academic lives and are librarians ready to meet this challenge? Information Technology and Libraries (Online), 31(2), 82.


