

PTX2 Timeline

March 16:

Created Github project and made everyone involved contributors. Distributed documentation for learning different systems being used. Added timeline to repo in form of project page

March 30:

Created mockups of site in Keynote.

Drafted up database structure in Django.

Using the python package BeautifulSoup, we scraped the registrar page consisting of tables of all course listings. Got title, department and numbers, and a link to the Blackboard book page using the package BeautifulSoup, which takes urls and makes the page source into "soup" - searchable and processable by HTML tags. Wrote scraper function for Amazon pages because we couldn't make progress in getting to the Blackboard page for each course because we couldn't log into CAS through terminal.

April 1:

Continued work on scraper. The Blackboard book page link in the registrar page table lead to a redirect page - we "soup"-ified that page and then got the link to the next page, which was the CAS login page. Still unsure how to get around CAS login.

Needed to contact Help Desk again about getting booklist data without scraping from Blackboard. Began putting together website using Bootstrap. Ran into issue where tutorial wasn't matching up with actual page, but overall looks correct. Add listing page essentially completed. Realized some more learning about Github required. (Merge? Oh wait, everything is gone...) Got a page running on AppEngine (ptonptx2.appspots.com), but may have some issues with SQL (we have to pay?). Looking into possibly switching to Heroku.

April 3:

Realized existence of discrepancy between version of Bootstrap in tutorial and version being used. Have to use different classes, but things are looking more correct. Refer to Bootstrap's own documentation for future. Figure out how to have site wide elements (header, sidebar); thinking of using SSI, which seems straightforward. Got contact for registrar (Vicky Glossen, but she won't pick up...)

April 4:

Finally got past the CAS login in terms of Blackboard scraping ourselves (Vicky Glossen said there was no comprehensive list of courses and books anywhere). We used the package CookieJar to store a cookies and establish a session in order to log in to CAS as a guest remotely via terminal. The next page was finally the Blackboard page. From the Blackboard page, we then scraped the isbn and labyrinth price. We caught non-books like pequods and a TI 81 calculator. We also caught Labyrinth book pages with links to other book pages, ones that only said "this course not available", and duplicated book pages. We converted the ISBN (which was given as ISBN 13) to ISBN 10. We got appended the ISBN 10 to a formulaic Amazon url for book products to get the Amazon page, from which we got the image and the price. The information for all the courses was stored in a list of dictionaries. Each course's dictionary consisted of a department and number dictionary, a course title, and a list of dictionaries for each book in the course, in which each book was a dictionary consisting of title, isbn10, isbn13, Amazon price, labyrinth price, author, etc. We ran the scraper one night for four hours, and it finally returned a text file that we converted to JSON.

April 9:

Integrated template pages into the Django environment, allowing them to be displayed with CSS locally and theoretically on Heroku (though we didn't commit there yet). Realized that we need to standardize how we use Git to prevent confusion/losing work.

April 10:

Restructured file hierarchy. Templates and static files now in the app folder, where Django wants them to be.

April 11:

Added templates and about page. Altered template structure by making use of Django's base.html templating system. Created blocks for sidebar.html, nav.html, and the actual contents of each page.

April 14:

Began experimenting with URLs. Figured out how to get URLs to load content from different views by associating bookshelf.html with bookshelf function in views.py.

April 15:

Added more URLs and associated them with more functions, for example, the course pages and book pages. Updated the scraper to get required/recommended status (addendum: we did not ultimately end up using this because of the conflict it would have with models, and because we didn't feel it was all that necessary).

In the page for each book, figured out how to nicely display all of the courses that a book is used in with the power of the Django forloop tag. Now courses are displayed as a comma-spliced list, with an "and" before the last one, sans Oxford comma.

Trying to figure out how to get all the books on the bookshelf to be the same height, because currently, some are being pushed to the side by the variations in height of the books in the previous row. Temporarily creating a new row div every 4 books, but not happy with the arbitrariness of this fix and want to explore different possibilities.

April 16:

For bookshelf problem, found a bit of javascript called equalizeThumbs that makes every thumbnail the same height, so the different books will line up perfectly. Still not a perfect fix, because have to manually set the height (for some reason, the default height, is like 100px, so even if we set a minimum and maximum height, it will always be set to the specified minimum), and this ends up making the books in owned and selling much larger than they need to be, but still slightly less hacky than the previous one.

April 17:

Moved scraper into the app directory, added a simple webpage accessible by admins to initiate scraping. Continued creating/updating templates for webpages, in particular began mass introducing modal windows into the interface.

April 18:

Added CAS login to every single view by adding in a user.is_authenticated clause before doing anything. Maybe a little bit hacky because same line of code copied and pasted for each view, but at least it works now. All urls redirect to login and then to bookshelf if not already logged in. Decided to use the built-in Django User model, which means we need to overhaul database again. Using this model so the site plays more nicely with the CAS login, which creates a new user upon log in. Extending User model with the renamed "Profile" model, which contains all of

the fields of the old Users model. Still need to figure out how to automatically create and link a profile, but for now just manually setting a connection using the admin page. Ran into problems because you couldn't access the site without a profile (due to templates referencing fields of the profile), so we had to get on the creating of a profile immediately. Actually pretty easy to do, just make a new profile if trying to access one fails and set the link in views.

April 20:

Fixed course badges in sidebar. Accurately determines how many books from a course a user has.

Added ability to add and remove courses from the sidebar. Done in the form of a modal window with a search bar exclusively for courses. Supposed to return you to a search result page consisting of a list of courses with add buttons next to them. All courses link to the course page regardless.

Currently having trouble getting the button to be inline with the course title, so ugly minus glyphicon floating over each course name.

Templates created for search result page, modal window added, just have to link them together with a view function.

April 22:

Couldn't figure out how to make the buttons inline just yet, but need to have some kind of prettier implementation for beta test, so made each course in the sidebar a drop down menu. Still not perfect because need to access drop down to go to course page, which should probably be accessible with just one click as opposed to the two currently.

April 23:

Linked searching for courses for the add course modal. Works in a similar way to the JSON assignment, and filters the database objects based on whether or not they match the query. Created search error page that is returned when a search returns empty. The search result page is a list of courses with add buttons. Have the add buttons in the course result page set so the button changes to a disabled "Already added" button and the course gets a strikethrough if the course in the search result page is already added.

Reordered course search results so that they are sorted by department and then secondarily by number using python's built in list sorting.

Created the sell page, which is linked by each listing in each book page. Currently in two parts (one for comment and one for price), because don't know how to properly deal with two different forms in one page. We think we need two forms because the comment and price belong to two different models. Not optimal, but at least its sort of working. Looks kind of ugly though, because forms are being automatically generated by Django, which does include pretty bootstrap classes.

Later prettied up both parts of the sell page by manually coding the form and including bootstrap. Still not sure how to consolidate into one page, which would be optimal, but at least it looks nice now.

April 24:

Minus glyphicon changed to an X glyphicon next to courses in sidebar, and put inline with course name.

Changed course search results page again and swapped button out for a glyphicon on the left of course title to better match the sidebar. Icon is a check if course already added and a plus if not.

Implemented book search by title and have rudimentary search result page with the titles of the books returned. Very similar to the finished course search. Added in a minimum query length to searches to prevent massive results pages like the current PTX has. Decided to make the minimum length 3 as opposed to the usual 4 required by forum searches because 3 is the length of a department code and seems natural to support. We also don't enough courses/books for length 3 queries to wreak THAT much havoc (and we have already scraped all of the courses/books). Added this error to the search error page.

Fleshed out book search by creating bookshelf-esque rows of returned books. Also fixed ordering of the results (made it alphabetical).

Edited title of page to reflect that it is showing the results of the search for "query" and also added a header to the search results page to show the same thing. Scrapped sell pages for a sell modal, which seems to be a running trend with our site (the use of modals). Also figured out how to include both forms in the same modal. Ran into a strange error where we couldn't dismiss the modal and the entire screen was greyed out. Ended up being an issue of when the modal was included in the wrong place in the template and was in a nav div.

Put listings in each book page and a buy button, which leads to another page that confirms purchase. Also created a pending page to test these transactions, and created a link to it in the sidebar. Books bought properly appear in the pending transactions page, but no email is given, and there is not a confirmation page to complete the transaction yet.

May 1:

Realized that people were searching for "COS" in the top book search bar and got "Cosmology" books. Made the top bar search return list of courses that matched three letter department designations followed by any title matches. Also returned list of courses first for word searches in which the word matched a course title. Still kept the feature where if someone searched for "COS 333" it would return only the books in that course (as well as the course page). Also implemented search by matching queries and book titles or course titles or course designations without spaces. So "The Practice of Programming" and "thepracticeofprogramming" would return the same thing, as would "COS333" and "cos333" for people who don't like space bars. Reapplied pretty fix to sell modal, including input form add ons to the price field to indicate only whole numbers will be accepted. Also added in the prices that the book is currently going for according to Labyrinth, Amazon, and student sellers to make it easier for new sellers to decide what they want their price to be. Added autofocus to various modals. E.g. price field is automatically selected in the sell modal. Added in bit of validation to the price field; made sure that all prices are bigger than 0.

Created the review and confirm page for sales, so that when both parties review the transaction, the book is transferred from the selling of the sell party to the owned list of the buy party.

May 3:

Added button to course page to let user add course directly from course page. Also added in condition to change into disabled "Already added" button if such. Made badge next to pending section in sidebar red so easy to notice. Made the badge to only be red when the value is not 0. As per friend's suggestion, made the navbar fixed so it is visible at all stages of scroll. Finally figured out how to get the remove course glyphicon to be right next to the course title! Actually pretty easy to do; just add in a badge that isn't pull-right before the text...

Added a “Sell book” option to the sidebar so people don’t need to navigate to a book page before they see the option to sell a book they don’t currently own. Was easy enough to do; just borrowed the modal functionality of the add course and stuck the search bar inside the modal.

May 4:

Fixed issue in buying where a book was not removed from needed even after buying it. The problem ended up being a problem with the models and trying to follow an incorrect field.

Sent emails to both parties in a pending transaction. Whenever a book is bought and named pending, both parties are sent emails alerting them about the other person and how to contact them. History page created to list completed transactions in which both parties review and confirm the transaction. Now all the major functionalities work - adding and removing courses, adding and removing books to needed, buying and selling.

Added on to the previous equalizeThumbs javascript and made it so that all of the images are centered along the same y-value, so the titles all start at the same place. Still having the same problem that the blanket height isn't working that well for all books, but still slight aesthetic improvement.

May 5:

For each book page, added in a condition so that the add to need button is disabled and changed to “already needed” when it is such.

Figured out how to populate the listings table of the book page and changed the font sizes of the table to be more in sync with rest of site’s design. Also decided to change table from hover to regular, because the hovering looked unnecessary. Finally got around to ordering the student listings by price. Were initially going to offer other sorting options, but because we only have two fields, and comment would just be ordered alphabetically and not provide any additional meaningful information, scrapping that and just having a static table. Added author and ISBNs to the book page.

Added in Help and FAQ sections to the sidebar to make navigation and use of the site easier for people. Also wrote the Help and FAQ sections. Need to alter the about modal to reduce overlap of information. Got rid of My Profile from the sidebar and moved the link up to the navbar where it tells you who you are logged in as. Still need to fix the profile page...Fixed course badges again. Realized that numbers for books selling were still being calculated according to an old definition of which books you were selling, so were getting the wrong numbers.

May 6:

Cleaning up bookshelf now that major functionalities are taken care of. Added in buttons to each type of book to allow a bit more control over what the bookshelf contains. Added to Needed a mark as owned and remove button; to selling an edit and remove listing button; and to owned, a sell and remove book button. Fixed display of bookshelf buttons. Made into XS and also made full-width so they looked more uniform.

Added in link to book page when clicking on lowest student price.

Changed bookshelf so that all three sections show up even when they are empty. Added in help text for empty sections. Implemented this kind of empty help text to all aspects of site (history, pending transactions, and listings).

Realized there was a bug in removing books when removing courses, because of COS217 and COS333 both needing the Practice of Programming. Made it so other courses’ book lists are checked before removing a book.

Further edited equalizeThumbs so it calculates heights individually for each section of the bookshelf. Now things look much better.

May 8:

Make profile modal to edit name and preferred meeting place. Moved it from a link in the sidebar to a link from the username in the top search bar, because testers said it was more intuitive.

May 10:

Added more information to help and faq pages, based on user feedback we got from people. Added ability to cancel transactions, which is linked in the pending transaction page. Next to the confirm button is a cancel button, which puts the listing on sale again and emails both parties about the cancellation.

Edit bug where error page comes up because people put nothing under selling price when selling a book.

Turned off debug mode

May 11-12:

Writing of documentation. Divvied up portions to different team members and then compiled it all together into a cohesive report.

May 13:

Last minute fixes before submission. Fixed a bug with being able to edit a listing that is already bought. Fixed a bug with the ordering of the search results and made the order consistent between the two search options. Added in a slightly more strict search query requirement to prevent user from getting too many results. Added missing page title.