SENG1010 TASK 5

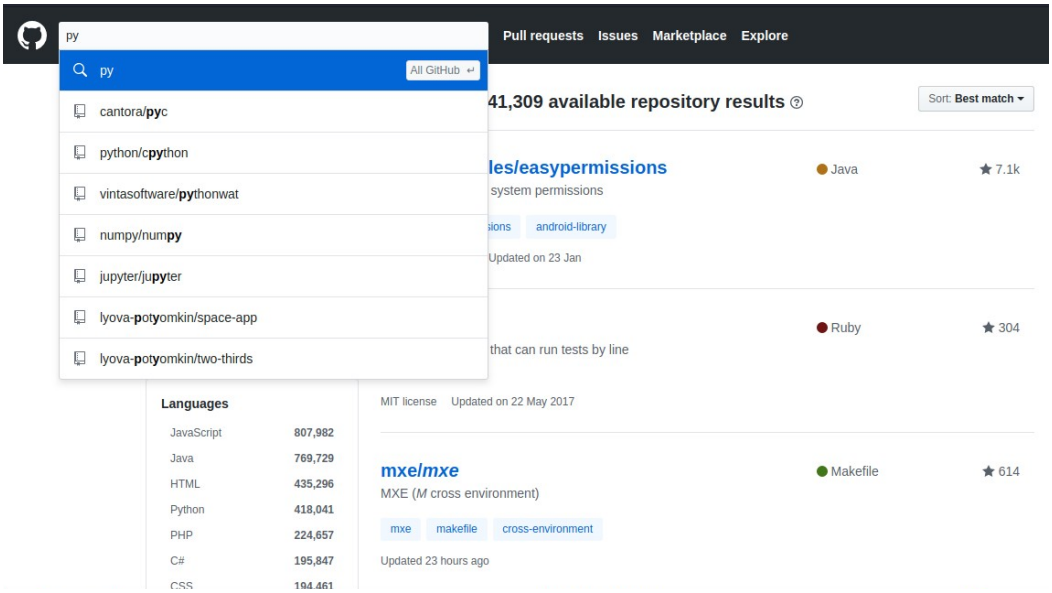
LEV POTOMKIN

Table of Contents

HELPFUL UI

GOOD EXAMPLE

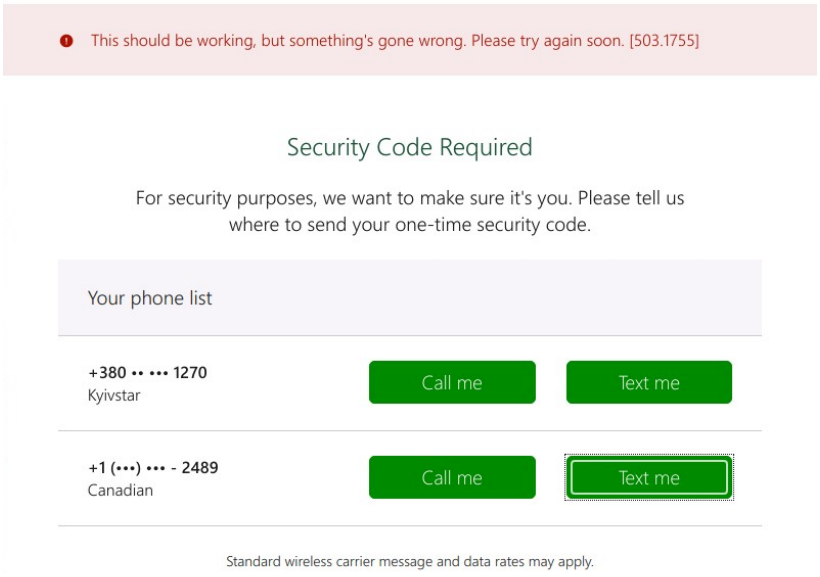
https://github.com/search



GitHub's search provides helpful user interfaces by placing suggestions under the search bar, thus limiting unnecessary typing. For each search result, it provides information about the author and name of the repository, as well as whether or not the repository is public, which may help to narrow down the search.

BAD EXAMPLE

https://authentication.td.com

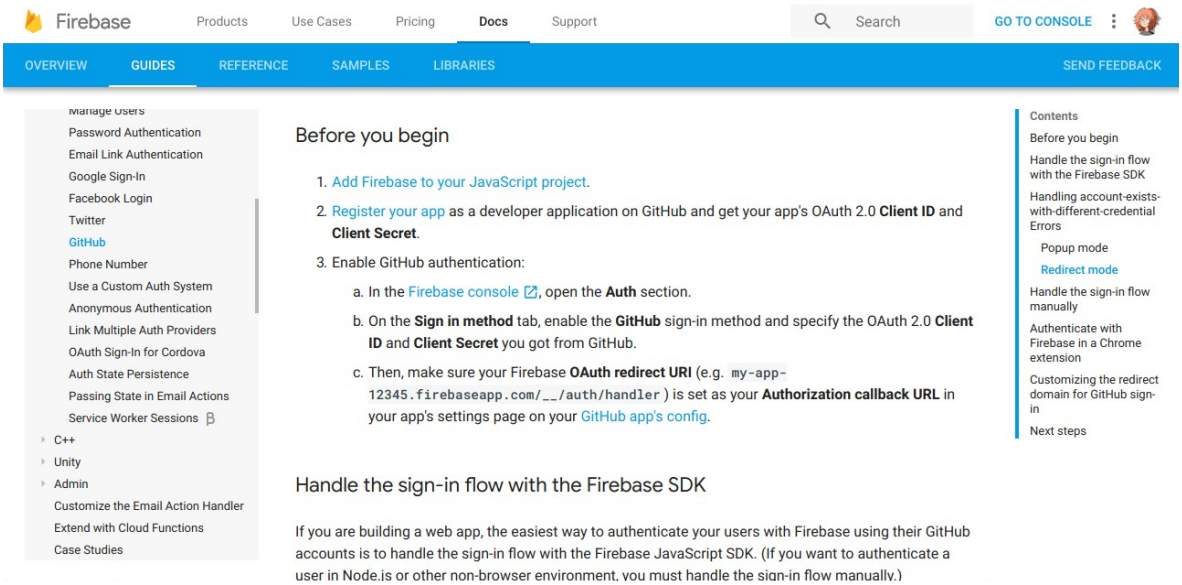


This example is bad for displaying unfriendly error messages. Aggressive red color and an error code will definitely scare the user away. The workaround for this may be to ask the user to send feedback or ask permission to send logging data in user friendly prompt, after explaining what went wrong and why.

FORMAT TEXT PROPERLY

GOOD EXAMPLE

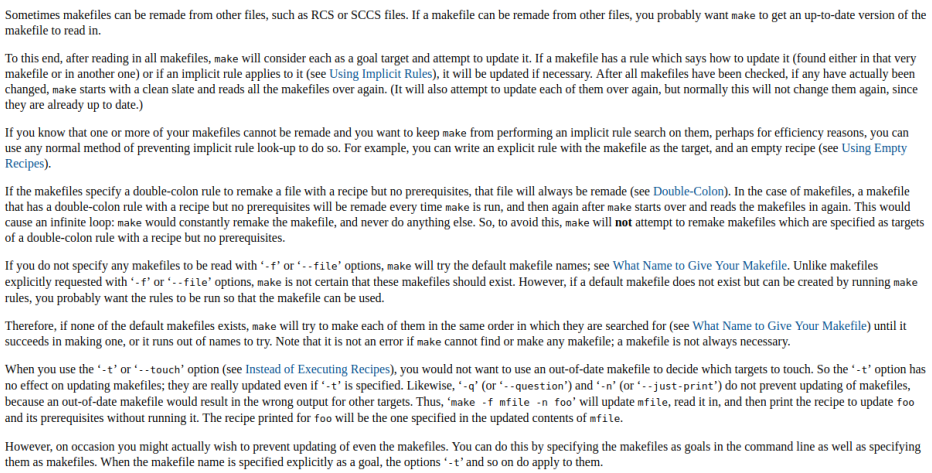
https://firebase.google.com/docs/auth/web/github-auth



Text formatting in this documentation article is good. Headings are used, that provide the main goal of a paragraph. Paragraphs are short, which is also good for users that scan and not read. List structure is used for better structure. Bold text is used to emphasize important objects. Code segments and links are also styled to match user's expectations. All this provides good user experience while reading the article.

BAD EXAMPLE

https://www.gnu.org/software/make/manual/make.html



Sometimes makefiles can be remade from other files, such as RCS or SCCS files. If a makefile can be remade from other files, you probably want `make` to get an up-to-date version of the makefile to read in.

To this end, after reading in all makefiles, `make` will consider each as a goal target and attempt to update it. If a makefile has a rule which says how to update it (found either in that very makefile or in another one) or if an implicit rule applies to it (see Using Implicit Rules), it will be updated if necessary. After all makefiles have been checked, if any have actually been changed, `make` starts with a clean slate and reads all the makefiles over again. (It will also attempt to update each of them over again, but normally this will not change them again, since they are already up to date.)

If you know that one or more of your makefiles cannot be remade and you want to keep `make` from performing an implicit rule search on them, perhaps for efficiency reasons, you can use any normal method of preventing implicit rule look-up to do so. For example, you can write an explicit rule with the makefile as the target, and an empty recipe (see Using Empty Recipes).

If the makefiles specify a double-colon rule to remake a file with a recipe but no prerequisites, that file will always be remade (see Double-Colon). In the case of makefiles, a makefile that has a double-colon rule with a recipe but no prerequisites will be remade every time `make` is run, and then again after `make` starts over and reads the makefiles in again. This would cause an infinite loop: `make` would constantly remake the makefile, and never do anything else. So, to avoid this, `make` will **not** attempt to remake makefiles which are specified as targets of a double-colon rule with a recipe but no prerequisites.

If you do not specify any makefiles to be read with '`-f`' or '`--file`' options, `make` will try the default makefile names; see What Name to Give Your Makefile. Unlike makefiles explicitly requested with '`-f`' or '`--file`' options, `make` is not certain that these makefiles should exist. However, if a default makefile does not exist but can be created by running `make` rules, you probably want the rules to be run so that the makefile can be used.

Therefore, if none of the default makefiles exists, `make` will try to make each of them in the same order in which they are searched for (see What Name to Give Your Makefile) until it succeeds in making one, or it runs out of names to try. Note that it is not an error if `make` cannot find or make any makefile; a makefile is not always necessary.

When you use the '`-t`' or '`--touch`' option (see Instead of Executing Recipes), you would not want to use an out-of-date makefile to decide which targets to touch. So the '`-t`' option has no effect on updating makefiles; they are really updated even if '`-t`' is specified. Likewise, '`-q`' (or '`--question`') and '`-n`' (or '`--just-print`') do not prevent updating of makefiles, because an out-of-date makefile would result in the wrong output for other targets. Thus, '`make -f mfile -n foo`' will update `mfile`, read it in, and then print the recipe to update `foo` and its prerequisites without running it. The recipe printed for `foo` will be the one specified in the updated contents of `mfile`.

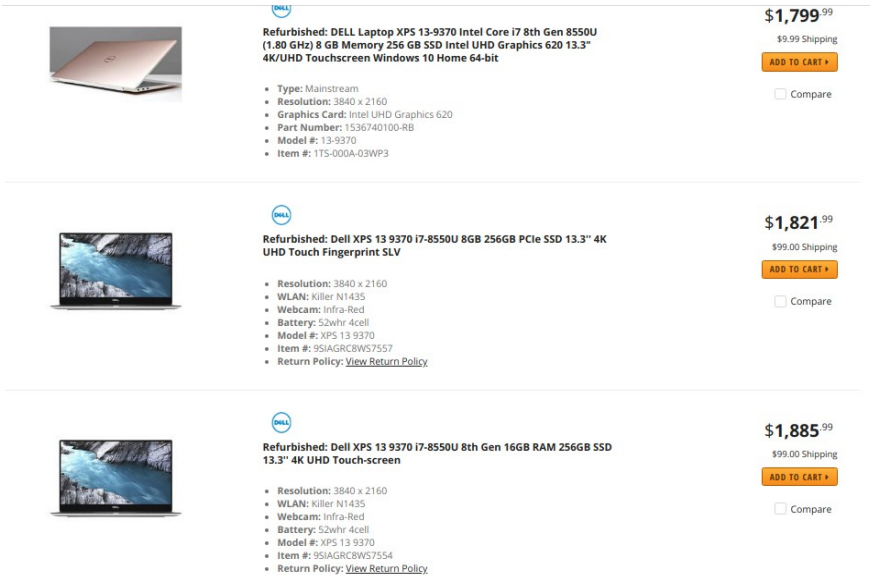
However, on occasion you might actually wish to prevent updating of even the makefiles. You can do this by specifying the makefiles as goals in the command line as well as specifying them as makefiles. When the makefile name is specified explicitly as a goal, the options '`-t`' and so on do apply to them.

This documentation article is a classic bad example, referred to as “Wall of text”. A large amount of mostly unformatted text, no headings, no bullet points. User will have a hard time finding what he wants on this website, if not leaves right away.

EASILY IDENTIFIED AREAS

GOOD EXAMPLE

https://www.newegg.ca



Refurbished: DELL Laptop XPS 13-9370 Intel Core i7 8th Gen 8550U (1.80 GHz) 8 GB Memory 256 GB SSD Intel UHD Graphics 620 13.3" 4K/UHD Touchscreen Windows 10 Home 64-bit

- **Type:** Mainstream
- **Resolution:** 3840 x 2160
- **Graphics Card:** Intel UHD Graphics 620
- **Part Number:** 1536740100-RB
- **Model #:** 13-9370
- **Item #:** 1TS-000A-03WP3

$1,799.99
$9.99 Shipping
ADD TO CART
Compare

Refurbished: Dell XPS 13 9370 i7-8550U 8GB 256GB PCIe SSD 13.3" 4K UHD Touch Fingerprint SLV

- **Resolution:** 3840 x 2160
- **WLAN:** Killer N1435
- **Webcam:** Infra-Red
- **Battery:** 52whr 4cell
- **Model #:** XPS 13 9370
- **Item #:** 9SIAGRC8WS7557
- **Return Policy:** View Return Policy

$1,821.99
$99.00 Shipping
ADD TO CART
Compare

Refurbished: Dell XPS 13 9370 i7-8550U 8th Gen 16GB RAM 256GB SSD 13.3" 4K UHD Touch-screen

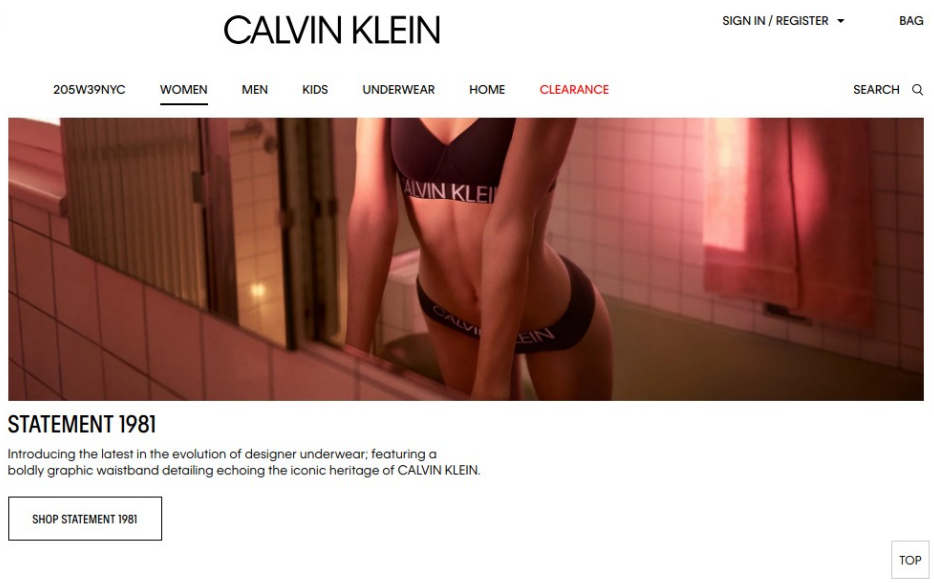
- **Resolution:** 3840 x 2160
- **WLAN:** Killer N1435
- **Webcam:** Infra-Red
- **Battery:** 52whr 4cell
- **Model #:** XPS 13 9370
- **Item #:** 9SIAGRC8WS7554
- **Return Policy:** View Return Policy

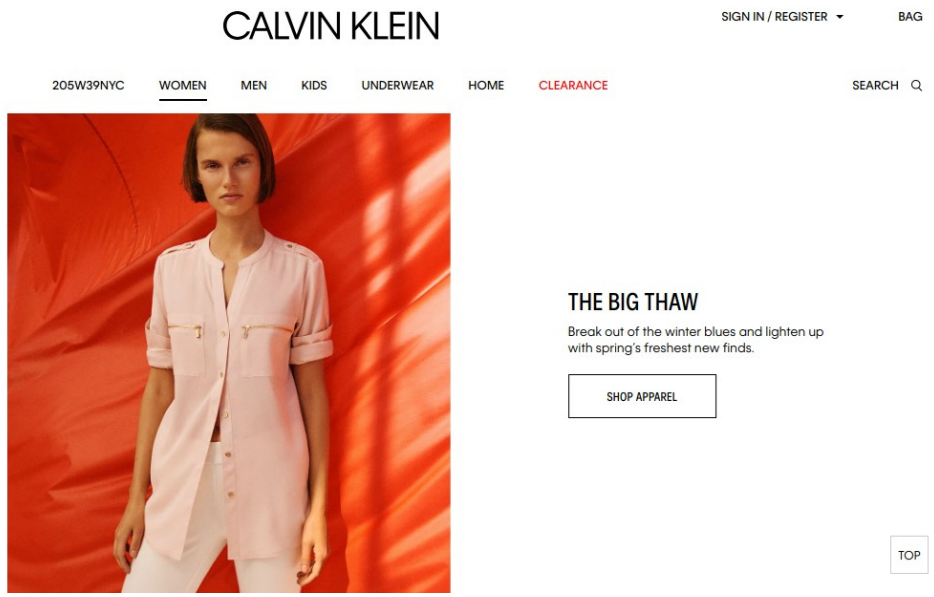
$1,885.99
$99.00 Shipping
ADD TO CART
Compare

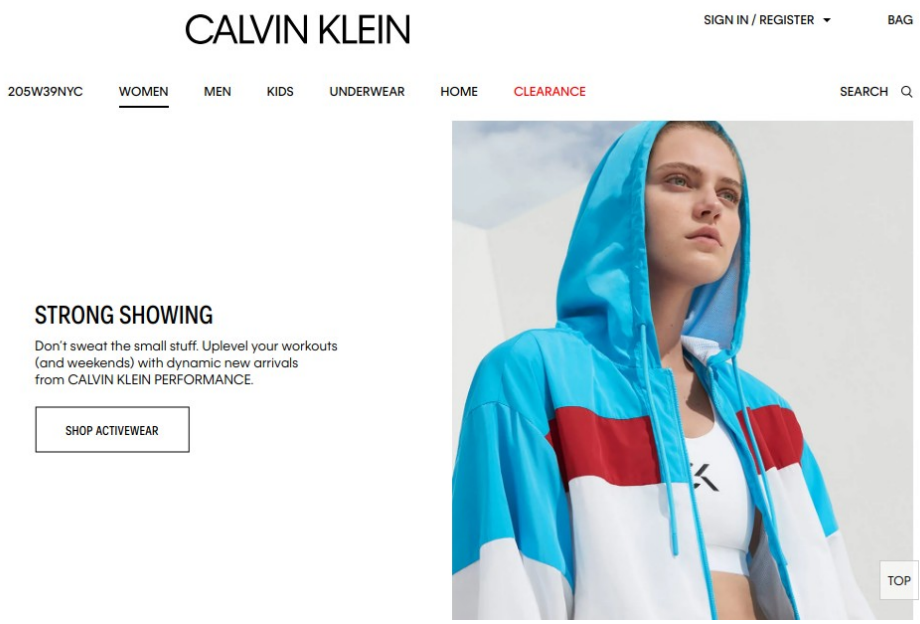
Newegg shopping website provides perfectly identifiable areas for their shopping items list. On the left of each entry there is a picture, on the middle there are logo, item name and bullet point description, and on the left there is price and “Add to cart” button. This pattern is always the same, no matter what you are shopping for and thus provides users with convenient interface.

BAD EXAMPLE

https://www.calvinklein.us/en/womens-clothing







Calvin Klein's shopping website, on the other hand, is much less convenient. Here, on one page we can see three different arrangements of picture, label and button. User will not be used to one particular design and often will be confused with each new arrangement. At least, relative position of button and label are always the same.

EASY AND UNCLUTTERED NAVIGATION

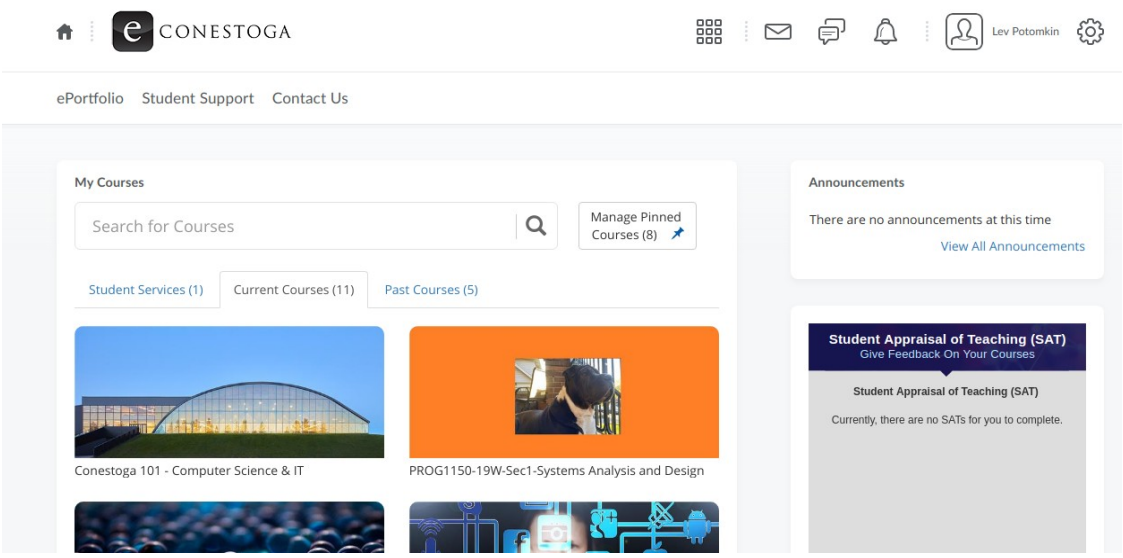
GOOD EXAMPLE

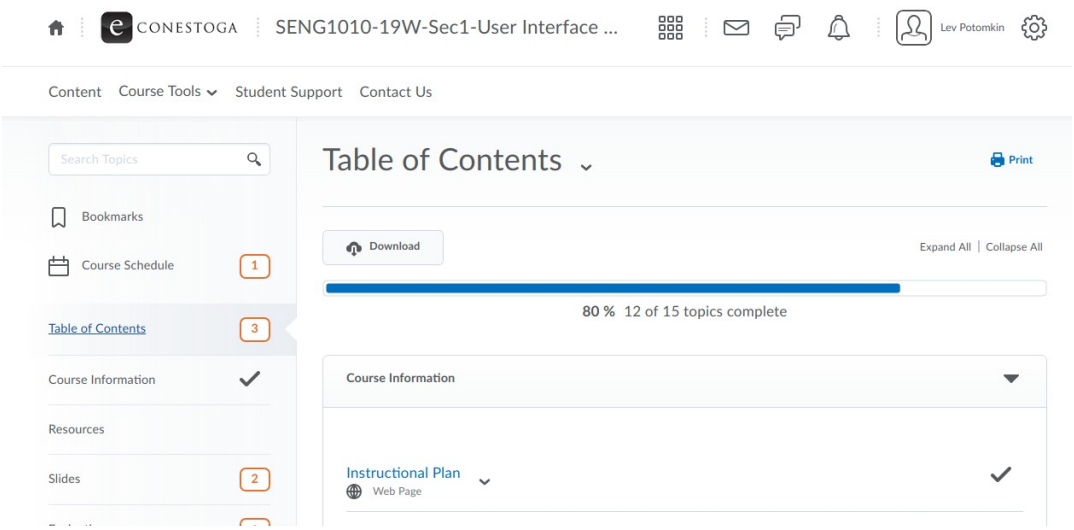
https://docs.microsoft.com/en-us/windows/desktop/

Microsoft documentation website provides easy navigation with:
1. Breadcrumbs, to return easily for pages of higher hierarchy
2. Navigation toolbar on the left provides quick access to any topic in particular
3. Tabs on the top panel give access to main pages of all sections of the website
4. Use of metaphors and signifiers simplifies navigation and makes interface more intuitive

BAD EXAMPLE

https://conestoga.desire2learn.com/



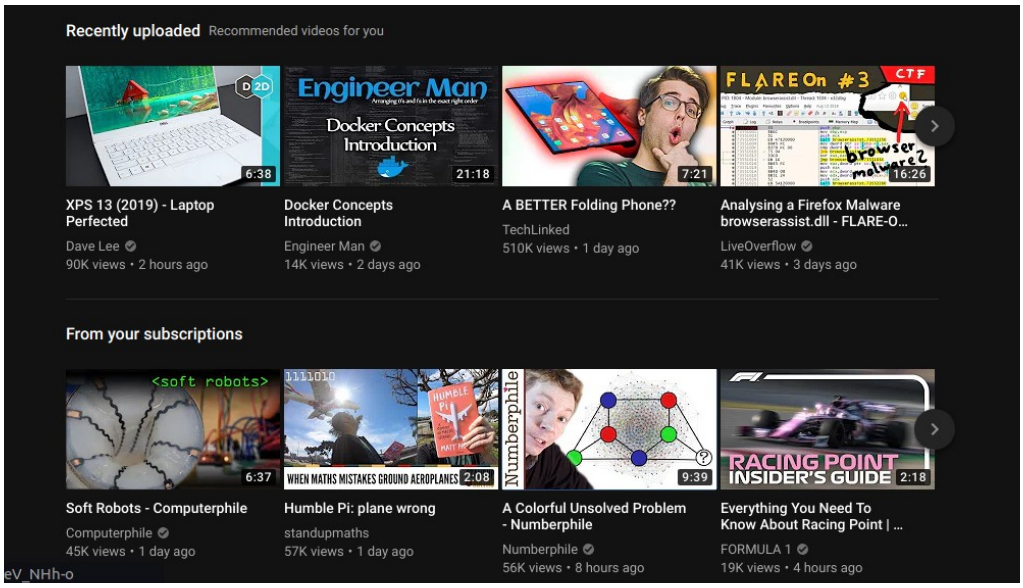


Navigation in eConestoga is not the best. To get to the slides or quizzes of a certain course, user has to make 3 clicks, whereas it could have been done by one. No breadcrumbs makes it hard to go back to previous pages. Unintuitive links to home page and course page (home icon vs. course title on top) make in less convenient and hard to find. Navigation is distributed across many pages which makes it harder to go to the desired page quickly. Also, plenty of unused/unimportant features are beside commonly used, which gives a feeling of cluttered user interface.

CREATE VISUAL HIERARCHIES

GOOD EXAMPLE

https://youtube.com



YouTube’s main page is a good example of visual hierarchies. Page is organized in rows that display videos by topic or channel. Rows are organized in blocks, each block being dedicated to a particular video. Blocks, in turn, display video’s thumbnail, title and description within its own area, thus providing intuitive and well-understood hierarchy.

BAD EXAMPLE

https://codeforces.com/contests

Codeforces Round #541 (Div. 2) Enter » Virtual participation »	MikeMirzayanov Sehnsucht Sender V--gLaSsH0ldEr593--V VFeafanov kun_ ch_egor grphil voidmax	Feb/23/2019 05:20UTC-5	02:00	Final standings	x8568
Microsoft Q# Coding Contest - Winter 2019 - Warmup Enter » Virtual participation »	Nickolas	Feb/22/2019 12:00UTC-5	3:00:00	Final standings	x6001
Codeforces Round #540 (Div. 3) Enter » Virtual participation »	MikeMirzayanov PikMike Vovuh	Feb/19/2019 09:35UTC-5	02:15	Final standings	x10104
Educational Codeforces Round 60 (Rated for Div. 2) Enter » Virtual participation »	BledDest PikMike Reziba Roms Vovuh adedalic	Feb/18/2019 10:40UTC-5	02:00	Final standings	x9902
Codeforces Round #539 (Div. 1) Enter » Virtual participation »	aleex markysha xolm	Feb/16/2019 11:35UTC-5	02:30	Final standings	x922
Codeforces Round #539 (Div. 2) Enter » Virtual participation »	aleex markysha xolm	Feb/16/2019 11:35UTC-5	02:30	Final standings	x9434
Codeforces Round #538 (Div. 2) Enter » Virtual participation »	Akikaze neko_nyaa xuanquang1999	Feb/10/2019 09:05UTC-5	02:00	Final standings	x10109
Codeforces Global Round 1 Enter » Virtual participation »	GreenGrape KAN Nebuchadnezzar V--gLaSsH0ldEr593--V _kun_ aitch grphil simonlindholm	Feb/07/2019 08:35UTC-5	02:00	Final standings	x11769

List of Codeforces contests could do better with visual hierarchies that just a table. Although each row represent one contest with columns being information about it, such as title, authors, date, etc., this information is scattered across the table and does not contain enough visual subdivisions.

EVALUATION

Item	Grade
Complete Set of Techniques Identified	/ 5
Well Chosen Set of “Good” Examples	/ 5
Well Chosen Set of “Bad” Examples	/ 5
Discussion of “Good” Examples	/ 5
Discussion of “Bad” Examples	/ 5
Subtotal	/ 25
Penalties (if applicable)	
Total	/ 25