

Project website - Feature #35

Create MVP website for the project

07/01/2021 01:09 PM - koszko

Status:	New	Start date:	07/01/2021
Priority:	High	Due date:	
Assignee:	Ogitnick	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description			
We need better online presence.			
This is also going to become a separate project at this tracker at some point.			
It already happened			

History

#1 - 07/29/2021 03:33 PM - koszko

☐☐

When renewing the VPS I used the opportunity and coded the fix for credit card payment confirmation page of my bank. I wanted to additionally remove the red ugly javascript notice on that page but for some reason this didn't work properly. I renewed my VPS but needed to do some more experiments with removing the js notice and decided to do a dummy purchase I would cancel afterwards. I went with a purchase of `hachette-hydrilla.org` domain (all better ones were taken and I didn't care because I was going to cancel the purchase anyway). Suddenly, the purchase completed without my bank page showing up ;_;

As a result of this accident we have a `hachette-hydrilla.org` domain for 1 year

#2 - 07/30/2021 11:24 AM - jahoti

I suppose that's one less problem we have to solve! In all seriousness, it's not a bad choice; the only problem is that you're now out of pocket, of course. How much did you pay, and do you need reimbursement?

#3 - 07/30/2021 12:20 PM - koszko

the only problem is that you're now out of pocket, of course. How much did you pay, and do you need reimbursement?

It's ok. Domains are cheap (compared to VPS or commercial SSL certs). I just updated the application with regard to domain price (btw, the domain now seems cheaper than it showed to me when I checked the price before).

Btw, just to be sure, by including "Tasks list" in the attachment you mean moving the list from below "Current intended uses:"? It seems we could also move the budget table to that file. And in the application itself we could then just say the money is going to fund the "infrastructure and development of Hachette and Hydrilla". Does that seem good?

#4 - 07/30/2021 10:59 PM - jahoti

It's ok. Domains are cheap (compared to VPS or commercial SSL certs). I just updated the application with regard to domain price (btw, the domain now seems cheaper than it showed to me when I checked the price before).

I just realized I was thinking of the price for a VPS (thank God you weren't up to renewing that :). Even better that it was cheaper!

Btw, just to be sure, by including "Tasks list" in the attachment you mean moving the list from below "Current intended uses:"? It seems we could also move the budget table to that file. And in the application itself we could then just say the money is going to fund the "infrastructure and development of Hachette and Hydrilla". Does that seem good?

Exactly, and now that you mention it moving the budget table too would be a great idea! However, as the application is meant to be "self-contained", the statement in there should probably stay as is (minus table). In fact, it might even be useful to add a few "stretch goals" to that statement with some concrete examples of what more we could achieve, if there's any in particular that stand out (it's probably best I don't try that, however, given that my "estimation" for how long would be a good start was 12 months :).

#5 - 07/31/2021 03:16 AM - jahoti

- Assignee set to Ogitnick

Descriptive, not prescriptive; feel free to change this assignment.

#6 - 07/31/2021 11:32 AM - koszko

- Project changed from Haketilo to Project website

- Description updated

#7 - 11/25/2021 12:13 AM - Ogitnick

- Subject changed from create a website for the project to Create MVP website for the project

#8 - 11/25/2021 09:17 AM - koszko

Btw, we now have to decide whether the website should be part of Hydrilla-served content or a separate project.

My plan for Hydrilla is to have it serve both the JSON API (which Haketilo uses) and HTML pages which humans can browse. The API idea is already described on [Hydrilla Wiki](#) and mostly implemented as a Flask app in [this git repo](#). For the HTML part, consider what is currently served at <https://hachette-hydrilla.org/>. That page lists currently available javascript replacements and is generated using a shell script. Ultimately, we'd like an equivalent of that (and also a separate page with details of each replacement) to be served by Hydrilla software itself.

So, should the website itself also be served by Hydrilla?

Pros:

- We'd just have to maintain 1 Flask app and nothing besides it. No copy-pasting of code, no doubling of packaging work.

- It'd be easier to integrate both for fancy features, e.g. having Hydrilla main site always display a random package from the repo.

Cons:

- We'd be forced to use the same technology that is employed in Hydrilla - Flask.
- At some point we're likely to have to maintain multiple versions of the JSON API for compatibility reasons. The server would need to be running multiple versions of Hydrilla and having the website included would make setup of this a bit more complex.

I am still not sure what is the best approach, so please share your thoughts. Regardless of what we choose, I think it'd be good to give both Hydrilla and the website the same look&feel

#9 - 11/30/2021 04:28 AM - jahoti

Regardless of what we choose, I think it'd be good to give both Hydrilla and the website the same look&feel

I fully agree.

As for serving the website via Hydrilla, that in my opinion hinges on what exactly the website will involve. If, except for a few special pages, it can be made to work almost perfectly within the constraints of what site-resource pages do, I think it would be silly not to; if there is little crossover, however, then it will be to no-one's benefit. That will ultimately depend on what Nick intends for the website.

(That said, optional modifications using JavaScript would still be possible- and a good excuse to use Haketilo!)

#10 - 12/02/2021 08:55 AM - Ogitnick

For the [landing page](#) I'd like to have:

- the Haketilo logo
- a download button for mozilla/chrome browsers
- a section explaining the motivation and purpose of Haketilo
- a list of features
- a link to the haketilo and hydrilla repositories
- a link for reporting bugs. For now, this could just be a mailto URI
- an image of the UI
- nlnet donor logos near the bottom to add legitimacy to haketilo/hydrilla
- links to the wikis
- copyright information at the bottom

The existing wiki adequately separates [user concerns](#) from [dev concerns](#) so for the MVP it makes sense to copy the wiki to the website verbatim.

The website should reflect the good web design principles we want others to follow. We don't need JS and we just need non-bloated CSS. It only makes sense to serve the website with Hydrilla since we want Hydrilla serving HTML docs anyway. Not doing so would cause duplicate work.

Since Hydrilla is exclusively a developer thing, it should be included in the website's wiki, but I think it probably shouldn't be promoted equally alongside Haketilo on the landing page. I think that might cause confusion for users since Hydrilla isn't strictly relevant for them. It will still be referenced of course. Do either of you disagree?

I think we should avoid links to websites with proprietary JS / bad web design in general. At least we should put a warning. All webpages should have mobile support.

Probably easiest to start by working on the landing page and go from there.

#11 - 12/02/2021 09:21 AM - kozzko

It only makes sense to serve the website with Hydrilla since we want Hydrilla serving HTML docs anyway. Not doing so would cause duplicate work.

So it seems we agree to have them both together. I'll adjust the MoU draft to reflect that. I also added kozzko and nick branches to Python Hydrilla [git repo](#).

Since Hydrilla is exclusively a developer thing, [...] I think it probably shouldn't be promoted equally alongside Haketilo [...]

Agreed

I think we should avoid links to websites with proprietary JS / bad web design in general. At least we should put a warning.

Within reason ;) As you surely realize, even this issue tracker is not perfect as Debian's packaging of Redmine doesn't indicate licenses in all places :/

All webpages should have mobile support.

Agreed

Probably easiest to start by working on the landing page and go from there.

Seems reasonable. Just don't take on the dynamic pages with served items lists/details until we [settle on the new schema](#) (feedback/ideas wanted, btw). And good luck :)

#12 - 12/03/2021 12:49 AM - jahoti

Since Hydrilla is exclusively a developer thing, it should be included in the website's wiki, but I think it probably shouldn't be promoted equally alongside Haketilo on the landing page. I think that might cause confusion for users since Hydrilla isn't strictly relevant for them. It will still be referenced of course. Do either of you disagree?

Not at all. In fact, at risk of introducing additional complexities, I think it might be a good time to establish an idea of "the project" as separate from its software tools. A forum post is imminent :).

Everything else seems perfect as-is; the link policy sounds as if it permits unlabelled free JS anyway, which averts any issues with the bug tracker (not that it requires JS to view in any case).