Google Summer of Code 2019

creative commons

CC Vocabulary and CC Chooser

- A cohesive design system to unite the web facing Creative Commons
- A new Creative Commons license chooser built using the design system

Mentors: Hugo Solar, Timid Robot Zehta & Jane Park, Breno Ferreira

---

me, at a glance

basic information

Name: Dhruv Bhanushali
Gender: Male (he/him/his)
Nationality: India 🇮🇳 | IST (UTC + 5:30)
Résumé: PDF

educational information

Institute: IIT, Roorkee
Degree: B. Tech.
Major: Engineering Physics
Graduation: 2019 (this year)

contact information

Email address: dhruv_b@live.com | dhruv.b.1996@gmail.com
Phone number: +91 8171615670
Emergency phone: +91 9322136002
Website: https://dhruvkb.github.io

social media

GitHub: dhruvkb
LinkedIn: dhruvkb
Twitter: dhruvkb_

me, slightly detailed

I am Dhruv Bhanushali, a student of Engineering Physics at the Indian Institute of Technology at Roorkee in Uttarakhand, India. My love for computers began at a very early age and since the ninth grade, I have been keenly developing software for all platforms and across a host of languages, frameworks and tools.

After being introduced to Python in my freshman year, it has quickly taken up a special position amongst my favourite languages for its elegance alongside JavaScript for its ubiquity and Java for its entrenched OOP principles. I prefer to work on web technologies using Python on the backend (and, for a lack of choice more than anything else, JavaScript on the frontend), although I am more than open to mixing it up once in a while.

I spent the majority of my junior year creating and working on the Omniport project, which has since been open-sourced (more on that in projects), including its infrastructure, core apps and services. The source code of the project is available on GitHub and the documentation for the same, also largely written by me, is hosted on ReadTheDocs. I would encourage you to give it a read just to get a feel of the kind of quality you can expect from me; *it is some of the best-written code and documentation of my life*. After significant completion in the Omniport project, I realised the importance of open-source software that empowered me to develop the platform and decided to give back to the community and what better way to do it than GSoC. So I set out to apply to the GSoC programme this year.

I am also a prolific member and the chief technology coordinator of the Information Management Group at IIT Roorkee, which is a group of passionate developers and designers from IIT Roorkee, brought together by one common goal. At IMG, I got a chance not only to develop software that reaches thousands of people but also to configure our own servers and deploy our apps into production. I learned how to solve issues efficiently and also how to collaborate better with my team (more on that in experience). I even learned a lot of time management skills and the ability to deliver, punctually, under pressure.

development skills

programming languages, frameworks and tools

Python	:	Django, Django REST Framework, Jupyter, iPython, NumPy, SymPy
JavaScript/Node.js	:	React, Vue, Express, Babel, Webpack, Jest
TypeScript	:	Angular
Java	:	JavaFX, Android, Ant
Kotlin	:	Android, Maven
reStructuredText	:	Sphinx

Apart from these I also have a working knowledge of PHP, FORTRAN, C++ and Golang.

software packages and environment

Containerisation	:	Docker, Docker Compose
Configuration	:	Ansible, Puppet
Security	:	UFW, iptables
Reliability	:	Travis, Sentry
Queuing	:	Celery, RabbitMQ
Databases	:	SQL (MySQL, PostgreSQL) + NoSQL (Mongo, Redis)
OS	:	Fedora 29
Editor	:	Visual Studio Code, Vim
IDE	:	JetBrains PyCharm, JetBrains WebStorm
VCS	:	Git
Design	:	Figma, Adobe Illustrator, Photoshop, After Effects

As you can see, apart from development, I do have a bit of *informal* design experience and practice as well, which will come in handy in certain aspects of the project (see mockups).

experience

Information Management Group

Information Management Group, usually initialised as IMG, is a student group at IIT Roorkee, founded in 2001, which aims to enrich the lives of the students and faculty of the institute by weaving technology into their day to day lives. Chasing this goal, IMG developed, maintains and upgrades the web portal of the institute (formerly Channel i, now Omniport) and the IIT Roorkee website. We also manage the servers on which these services are run and constantly strive to ensure uptime and optimal functioning of these services.

In the tenure of one year as the chief technology co-ordinator at IMG, I accomplished the following:

- Pivoted the role of IMG in the campus from an app development group to a platform development group.
- Created Omniport, the rewritten open-source portal of IIT Roorkee replacing Channel i.
- Initiated, lead and oversaw a total overhaul of IMG's products to the latest software stack and technology.
- Mentored and tutored a team of over 50 developers and designers, at both the macro and micro levels.
- Shortened development cycles at IMG by up to 50% with a modular, containerised and isolated approach.
- Revolutionised the popular `<img>` of IMG leading to heightened public awareness and brand reputation.
- Increased the IMG's outreach via social media and cultivated a brand-conscious mindset about our products.
- Developed a culture of open-source at IMG and set the institute record of 30 repos open-sourced in one day.

All my technical skills like software development, server configuration and deployment skills, and a number of soft skills such as collaboration, communication and conversation, can be traced back to my time at IMG.

projects

Some of my contributions to open-source, *which spark joy*, are listed here.

Omniport

Technologies: Docker, Django, Django REST framework, React, Redux
Links: Infrastructure-as-code, backend codebase, frontend codebase, documentation

Omniport is a platform for institutes to customise and use as their web-portal. Omniport provides a modern suite of services and apps, all of which can be customised as per the needs and wishes of an institute, and a platform for individual developers to develop their own apps for their own institute and for the world. Omniport also allows third-party developers to tie-in their apps with the portal using OAuth2.

My experience at IMG led me to come up with the idea of Omniport, which was the answer to a number of significant existential questions facing IMG in the coming years. These included the following:

- How could we, as a group, explore diverse fields without having to give up our responsibilities towards our existing services? *Omniport allows us to share control with the students, freeing time and energy.*
- How could we bring the institute together and involve the students themselves in building a portal for their own convenience? *Omniport allows developers from the institute to work with IMG.*
- How could we enable other institutes to benefit from our years of experience and also benefit from their contributions to our portal? *Omniport is generic to be able to serve as many institutes as possible.*

Omniport is built using the latest technologies such as Docker on the infrastructure, socket and HTTP/2 support on the backend and a progressive web app on the frontend. All of Omniport's code is held to the highest standards of code-quality and is continuously updated to make use of the latest additions to both languages used, Python and JavaScript. My contributions to Omniport as an open-source project include reviewing PRs and patches submitted by our users and developers.

Repository	Commits	Additions	Deletions
Omniport Docker	142 commits	4,557 ++	2,788 --
Omniport backend	340 commits	30,674 ++	17,065 --
Omniport frontend	19 commits	1,832 ++	786 --
Omniport docs	43 commits	5,214 ++	1062 --

I am the creator of, and the top contributor to, the project's backend and Docker. I also made significant contributions to the documentation and frontend of the project. I also have a major role in over 30 sub-repositories and am actively working on adding new features and rectifying bugs across the project.

Rethink!

Technologies: Django
Links: Codebase

I developed a time and resource management and logging application for Rethink!, the tinkering lab of IIT Roorkee. The app manages the allocation of machine time as well as consumables with expenses to students availing the services. I also deployed the app on their in-house servers, integrating with the aforementioned intranet portal, Channel i via OAuth so as to be seamless for users in the campus to be able to sign in.

Repository	Commits	Additions	Deletions
Rethink!	216 commits	39,334 ++	17,519 --

Creative Commons catalogue

Technologies: Django, Vue
My contributions: backend, frontend

To familiarise myself with the contribution and review process at Creative Commons, I contributed to two repositories pertaining to the upcoming CC Search, which is still in beta as of this writing. I made a number of pull requests to and discovered a number of issues (some of which I fixed myself via PRs) in these repositories with invaluable input from Alden Page, Kriti Godey, Jane Park and Breno Ferreira.

api (cccatalog-api)

PR	Issues	Title
#260	#223, #225	Porting ingestion server to Pipenv
#242	#234, #235	Update the polaroid watermarking process with bug fixes *(merged without changes in #252)*
#225	#223	Requirements version pinning
#224	#221	Add instructions regarding vm.max_map_count to ensure ElasticSearch starts up correctly

Issue	Title
#256	The image search API returns one less image per page given the pagesize
#247	Watermark view cannot handle images that do not have EXIF data *(suggested fix incorporated in #252)*
#223	Inconsistent pinning of packages in requirements.txt
#222	Admin site unreachable because of missing packages
#221	Documentation missing an instruction, causing ES not to startup

frontend (cccatalog-frontend)

PR	Issues	Title
#260	#254	Implement OpenSearch for browser integration
#237	-	Ellipsise overflowing titles and show the full title on hover
#225	#103	Masonry layout
#221	#104	Add social media icons in the right sidebar
#219	#181	Add a check on creator URL and conditionally render anchor tags
#217	#106	Add option to 'Copy Rich text' in attributions

Issue	Title
#250	Image download button animation not right
#246	Images in 'Top Picks' are poorly cropped *(suggested fix incorporated in #225)*
#245	The top categories section on the homepage does not scale with width

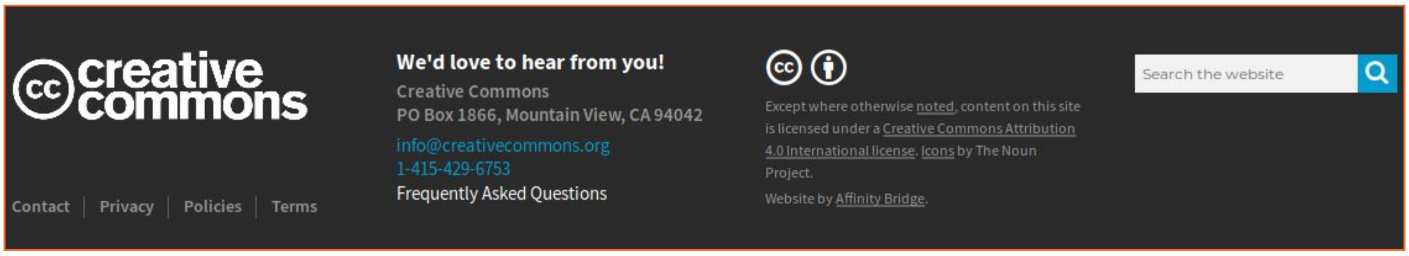
I'm quite pleased with some of these PRs that made a significant improvement to CC Search and taught me a lot about the project, the software and tools involved, and about the review and merge process at Creative Commons.

Even in the concluding weeks of GSoC and post that, I will keep contributing to CC Search to smoothen the transition of CC Search to CC Vocabulary (which as described below in etymology, is the tentative name of the platform in the GSoC proposal).

introduction

the problems

Observe the following two footers from two different Creative Commons sites. Both contain the same content but look worlds apart. There is even an open issue to make the latter look like the former!


creative commons

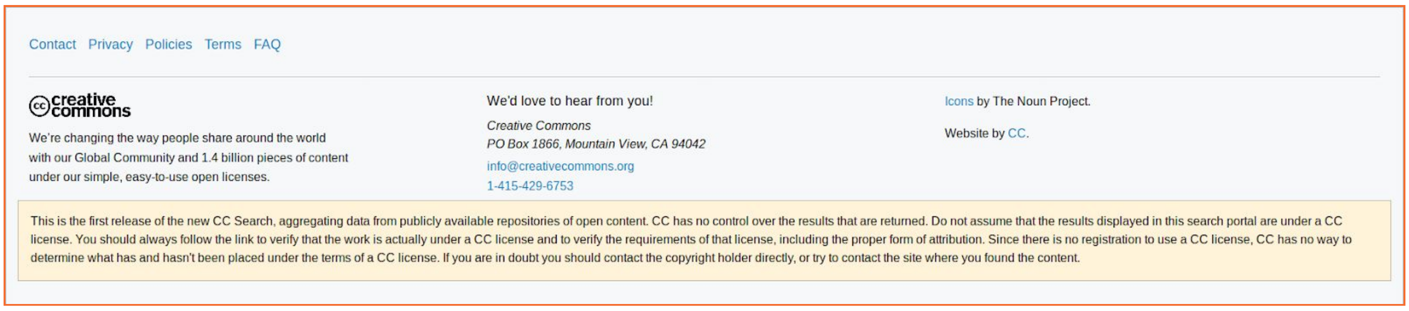
Contact | Privacy | Policies | Terms

We'd love to hear from you!
Creative Commons
PO Box 1866, Mountain View, CA 94042
info@creativecommons.org
1-415-429-6753
Frequently Asked Questions

Except where otherwise noted, content on this site is licensed under a Creative Commons Attribution 4.0 International license. Icons by The Noun Project.
Website by Affinity Bridge.

Search the website

This is the footer on the CC homepage.


Contact Privacy Policies Terms FAQ

creative commons
We're changing the way people share around the world with our Global Community and 1.4 billion pieces of content under our simple, easy-to-use open licenses.

We'd love to hear from you!
Creative Commons
PO Box 1866, Mountain View, CA 94042
info@creativecommons.org
1-415-429-6753

Icons by The Noun Project.
Website by CC.

This is the first release of the new CC Search, aggregating data from publicly available repositories of open content. CC has no control over the results that are returned. Do not assume that the results displayed in this search portal are under a CC license. You should always follow the link to verify that the work is actually under a CC license and to verify the requirements of that license, including the proper form of attribution. Since there is no registration to use a CC license, CC has no way to determine what has and hasn't been placed under the terms of a CC license. If you are in doubt you should contact the copyright holder directly, or try to contact the site where you found the content.

This is the footer on the CC Search homepage.

This is but one example of a much more pervasive and deep issue that in my opinion affects Creative Commons services at the present moment. **And with Creative Commons looking to expand, the number of services and projects under the CC banner are only bound to increase.** With that increase would come a huge disparity between various services. This disparity would affect how various components look, feel and work and would affect a large number of on-screen elements from individual buttons to entire webpages.

However, as of now, Creative Commons has no cohesive design system. *This project aims to change that.*

Another problem this project aims to address is the usability of CC Chooser. Findings from the Discovery phase of CC Usability state, *verbatim*:

> “
> - People understand that CC stands for free content sharing, but the nuances of the specific licenses are lost on them — including experts and longtime CC users.
> - People are motivated to license their work under CC, but have a hard time figuring out how to do it.

These issues are linked with the CC Chooser, a relatively simple application for determining the license that creations should be licensed under and then provides license imagery and insignia. **However, to a first time user, the chooser is quite unfriendly and unhelpful, and to put it dramatically, even discouraging.** It does not explain the license types, aspects of a license or guide the user to the right license with a helpful wizard. It doesn't even inform the user of various CC licenses (I had to discover them one by one by tweaking my choices). Utilising mockups by various UCB students linked to on the Ideas page, one can very easily come up with a much better, more helpful and inviting CC Chooser.

CC Chooser is in a state that has a lot to gain from an overhaul. *This project aims to achieve that.*

the solution: part 1

All companies that own a web presence have some form of a design language, because of a number of significant advantages described further on, namely:

- Google has *Material* Design
- Microsoft has *Fluent* Design
- Bloomberg BNA has the *Fish Tank* Design
- Apple has its *Human Interface* Guidelines

While Creative Commons too has a design language (easily identified by the bold use of typography and poppy colours), it is just not as concrete as those above and does not have a set of components that are distinct to call its own. I propose a solution that solves the problem via a multi-pronged approach, including

- a library with a collection of UI components that are flexible but familiar in look and feel
- highly interoperable, a seamless UI can be built using the components themselves
- a view that allows users to be able to experiment with the components in a variety of scenarios
- maximum possible a11y, i18n and L10n compliance

This project I'd call **CC Vocabulary**, undertaken under the mentorship of **Hugo Solar** and Breno Ferreira.

CC Vocabulary stems from my own experiences from working on Creative Commons' ambitious search and catalogue project, confirmed and refined by conversations with Hugo Solar, Kriti Godey, Timid Robot Zehta and Breno Ferreira, all very experienced developers at Creative Commons.

CC Vocabulary is a collection of UI components that

1. unify all CC services into a cohesive experience and appearance
2. cover as many use-cases and scenarios as possible out-of-the-box
3. are flexible to accommodate a certain degree of variation
4. ensure accessibility, internationalisation and localisation to a large extent
5. enable a clear separation between generic components and very app specific components
6. come packaged as a simple to use `npm` package for ease of use
7. make it easier for developers to make a seamless experience across the Creative Commons suite

As metaphorical icing on the cake, CC Vocabulary

1. provides Storybooks to make it easy to visualise and test-drive components
2. will also decrease the barrier of entry and increase the productivity in frontend-development
3. will take the effort out of developing the frontend of a new service by enabling remixing components

etymology

The name CC Vocabulary stems from the need for a common tongue spoken and understood by both frontend developers and designers to be able to communicate effectively and collaborate productively. The new components devised in the project would be added to the vocabulary of both parties and would be the backbone of a language in which CC expresses its identity across projects and the web.

If you're not sold on the project name yet, let's just say it's a working title.

the solution: part 2

Once a component system has been devised, it is only as useful as the products that use it. For a meaningful way to revamp one of CC's applications, we turn to the CC Chooser, a relatively simple but massively important tool of Creative Commons that is beginning to age and could benefit very well from a remake in a modern developer-friendly and user-friendly avatar.

The solution is to revamp the chooser to

- be explanatory, informative and responsive to new creators
- not slow down experienced creators
- allow people to understand the various parts and aspects of a license
- provide badges for creators to put on their content, sites and profiles
- showcase the capabilities of the CC Vocabulary while maintaining subtlety and minimalism

This project already has a name, **CC Chooser**, revamping which will be undertaken under the mentorship of **Timid Robot Zehta** and Breno Ferreira. Unofficially, Jane Park will also be mentoring the project.

CC Chooser is the primary tool of Creative Commons and is the first point of contact with creators. It is, in fact, the first interactive part of the CC website that one reaches when exploring the homepage.

CC Chooser, after the revamp, will

1. acts as the single point of contact for all licensing needs of creators
2. will inform and enable creators to make educated choices behind their licences
3. will lead to increased awareness of Creative Commons licenses
4. provide all sorts of labels, badges, insignia and logos for creators to show pride and associate with CC

As metaphorical icing on the cake, CC Chooser

1. gets revamped to make use of the new CC Vocabulary components
2. will be the second Vue-based modern, progressive web app in the CC suite
3. act as the model frontend-application for further frontend development at Creative Commons

Those are all the goals being pursued by this project.

ideas addressed from the list

This project, although initiated as an idea not in the list, covers or overlaps with a number of them, such as

- New educational tool for CC licenses (link)
- Reward and delight users of CC licenses (link)
- Contact content creators easily (link)

The details of the overlaps will be covered later on in this proposal.

research/references

A list of web services and documents that I reviewed when researching for other implementations that we could inherit code and/or learnings from (these are suggested readings from the ideas page itself):

Findings from the Discovery phase of CC Usability

Insights 1, 2 and 3 specifically address the issues presented above and in turn and are redressed by the ideas presented in the proposal.

CC usability prototypes

A number of different mockups and prototypes of how Creative Commons' tools can be made better and intuitive. They also contain Jane's feedback that has been worked into my own mockups.

Reward and delight users of CC licenses

Some reward and delight ideas could be baked into the CC Vocabulary making it seamless to add them to other CC projects.

CC projects which I contributed to or studied for the project:

CC Search

Over my time contributing to CC Search, I did experience a feeling like we were developing for the here-and-now, rather than building reusable components that could serve future projects.

CC Chooser

I chose to start with CC Chooser as a starting point for two reasons, one being that it is quite outdated so would stand most to benefit from a do-over and the other being that it was another idea on the list so this proposal would kill two birds with one stone (forgive the unnecessary violence in the metaphor).

Badges are very popular among creators, based on my experience with developers and as one myself. I studied the offerings of two quite famous badge sites:

For The Badge

This site provides funny badges (with no meaningful information) that developers can inject into the `README.md` files of their repos. Its popularity is a sign that people love badges for the badges' sake.

Shields.IO

Shields is a similar site that provides a lot of pre-made and custom badges, but meaningful ones as compared to the above service.

CC License badges

A site that provides a large number of insignia, that could be coupled into CC Chooser and made slightly more intuitive to use as compared to a file browser in a web browser.

I found a number of design languages and frameworks, the most notable of which are listed here. They can be used to inherit insights and learnings:

Formula One

This is the design library used in Omniport, based on Semantic UI's React-based version. The introduction of this design framework eliminated a lot of friction and repetition in IMG's codebase and made Omniport development up to twice as fast.

Semantic UI (React)

One of my favourite design systems, provides clean components with ample documentation.

Ant design

Provides one of the largest collection of components that I have seen so far.

Other than these, we also many more from the likes of Google, Salesforce and Twitter. There is an abundance of design systems but we need our own because of Creative Commons' distinctively unique visual style. This is a non-exhaustive list of my sources behind analysing the feasibility, utility and need for the project.

mockups

disclaimer

Do note that these are very initial stage designs and are meant to change with feedback and reviews. Also, this is my first time putting my limited design knowledge into practicality, that too designing such a high-impact application and hence these designs might require many more iterations to reach production-level quality.

As a feel of what CC Vocabulary components could look like, I designed some quintessential ones. The mockup components and screens have been designed on Figma, which is a tool more suited to the task as compared to a document. I'll be thankful for any feedback or ideas to improve these designs.

This Figma document linked here shall be a living document that will be fine-tuned and refined as the components are being built. It will, in the end, be handed over to Creative Commons to share with all designers enabling them to prototype with components as they are supposed to look in real life.

navigation

CC Vocabulary

Consider a developer and designer duo, named **Alice** and **Bob** respectively for the sake of a time-honoured tradition.

Alice wants to develop a new CC service. It's an attribution locator say, for example, that takes the unique ID of some CC-licensed content and returns all the information about the content, its creator and ready-to-use attribution (such a tool could actually be in development in parallel with the project, so it's unfortunate this didn't happen sooner).

design

Bob would be designing the screens for the tool. He would obviously like the experience to be consistent with the rest of the CC experience, for the users of this tool.

1. Bob plans the flow of the tool and designs the user experience.
   He needs to worry only about the big picture because the minutiae of every component's experience have already been taken care of.
2. Bob starts the designing stage in Figma.
   He can very easily make use of the readymade component palette provided by Creative Commons. His design mockups would look very close to the actual designs. He can even add more frequently used components to the designs.
3. He does not have to confine himself to a very limited set of components.
   He can iterate faster because the library provides him with a lot of variations and also provides him with the mockups of all variations of a component.
4. After a few iterations, he sends the results to Alice.

development

Alice would be developing the frontend for the tool. She would also like the development experience to be fast, efficient and skip bikeshedding.

1. Alice can very easily install the CC Vocabulary package.
   In a couple of commands, she can get the components and start using them in her application. It is all bundled in a convenient package.
2. She can focus on building the application and the interaction between these components.
   This is possible only if she can leave the finer implementation of every component to the library. The components do allow her to be customised as per Bob's designs.
3. She can even contribute back to the library if some of her use-cases are left out.
   She can add new components to the library, for example, say an ID field that validates CC IDs via async API calls or add new functionality to existing components. All other developers can benefit from her contributions.
4. After a few iterations, her tool is ready to be deployed.

CC Chooser

While we're doing this, consider a content creator, more specifically a music composer, named **Carol**.

She might want to license her new creation, a music track, under a CC license so that her music can be remixed and shared with attribution back to her. She might want to license her work into the public domain to see what people make of it without restrictions of any sort whatsoever.

licensing

1. Carol visits the license chooser from the CC homepage.
   The very first choice she sees is whether she knows her way around or would prefer to be guided.
   This is how we allow experienced creators to be able to obtain their license as quickly as possible.
2. Carol chooses to get a helping hand.
   The first choice is between CC licenses and the public domain.
   The tool explains the differences and she can make her choice.
   A. Carol chooses to license under CC as she wants the attribution and certain restrictions on her work.
      ✓ All CC licenses include attribution anyways. (`BY`)
      ✓ The tool then informs Carol about adaptations and asks for her take on it. (`-/ND/SA`)
      ✓ The tool then informs Carol about commercial use and asks for her take on it. (`-/NC`)
   B. Alice chooses to license under the public domain.
      She wants to relinquish copyright of her work.
      ✓ The software informs Carol about the public domain and asks her of her purpose behind the visit:
         - Place a piece of work in the worldwide public domain (`CC0`)
         - Label a piece of work already in the worldwide public domain (`PDM`)
3. Based on the choices, she gets recommended one of the various licenses/marks (she can't choose PDM).
   She is guided at every step of the way at every decision she makes.
   She can easily go back and change any of her choices.
4. Keeping most other things unchanged, she is finally matched with some injectable HTML and some images that she can place in her work or on her webpages.

technology

Frontend in CSS:

SASS compiled to standard CSS

Frontend in **JavaScript**:

Minified DOM manipulation based JS *(for certain components that may need it)*
Vue, Vuex *(when using a shadow DOM framework, to remain consistent with other projects of CC)*

The goal of the component set is to be the set of building blocks that developers of CC projects can use to develop their frontend using a very modular approach. The component set aims to be exhaustive yet not limiting in nature.

To be compatible with all existing services, the library would be distributed as vanilla minified CSS, and also as a Vue component collection that itself uses this CSS. This would enable CC sites to utilise the library just as well as any other CC applications.

It would include components from the very basic cards to the very diverse buttons. It would contain components that are frequently used such as the CC header and footer and also components that are not so frequently used such as CC insignia. Each component would also come with a placeholder, a simpler component that indicates loading. All components would also include colour options as it is quite evident that the CC visual style is marked with pop colours.

All components would undergo accessibility checks. They would be developed keeping in mind RTL layouts as well. This would ensure internationalisation and localisation in foreign languages, making it easier to develop CC services in multiple languages, furthering the proliferation of CC licenses worldwide.

Finally, to answer the question, *"Why Vue?"* we must look to CC Search. Since the frontend of CC Search is in Vue, and it is by far CC's most ambitious open-source software project (according to me at least), CC Vocabulary would make the most sense to be a set of Vue components. We could refactor CC Search to use CC Vocabulary and we could increase interoperability with other CC software projects in the future. With that in mind, the new CC Chooser will also be written in Vue.

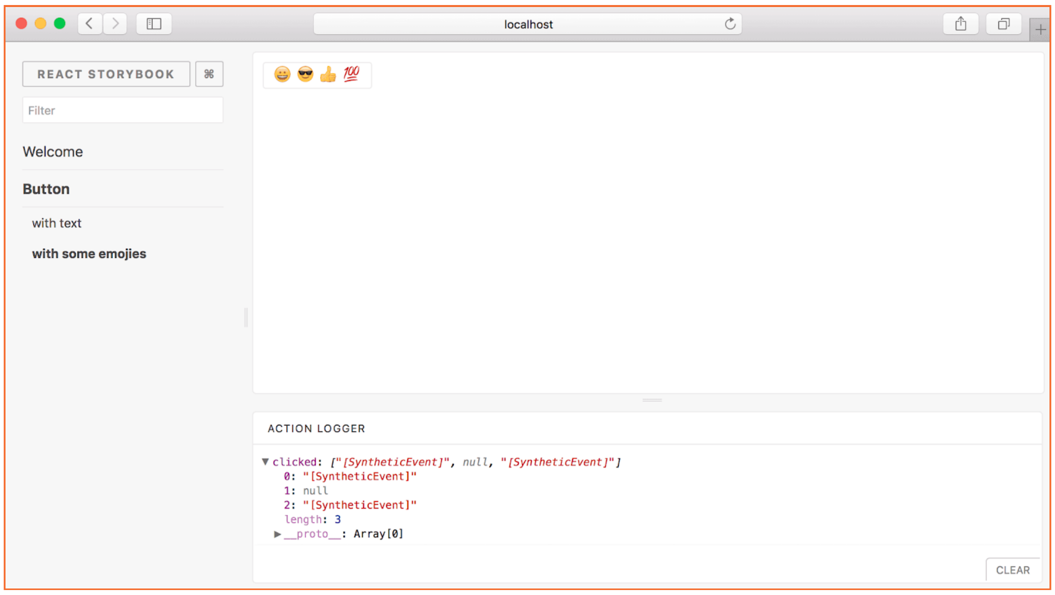
feature highlights

storybooks

Storybook is an open-source JavaScript tool that is compatible with all major frontend frameworks such as React, Angular and Vue.

Storybooks are testing grounds for components where one can adjust parameters passed to the components and see their state in real-time. Storybooks make prototyping components much easier because now instead of imagining what the component would look like and how it would behave, developers can just experiment and determine these details. Also, storybooks make for a wonderful showcase to demonstrate the components.

I'd be remiss not to include a screenshot of the wonderful tool.



```
▼clicked: ["[SyntheticEvent]", null, "[SyntheticEvent]"]
   0: "[SyntheticEvent]"
   1: null
   2: "[SyntheticEvent]"
   length: 3
  ▶__proto__: Array[0]
```

A storybook with a Button component

In fact, Storybooks are such fun that just for the sake of being able to test components using them, I insisted on having Vue components in the project.

badges

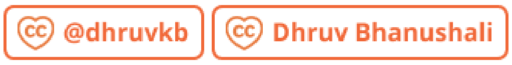
Badges are a way for users to show their pride and they act as links to their profiles.

A typical attribution badge from Unsplash
Shows the brand as well as the creator name

A typical `README.md` badge from Shields.IO
Shows live build status from a source like Travis

As you can see, the badge is a powerful item for conveying two units of information in a very compact form factor and providing creators with badges would no doubt increase the brand recognition of CC and the proliferation of content in the wild. Also, the badge would be a symbol of pride and a source of delight for creators (the **Reward and delight users of CC licenses** project).

I'd be remiss not to include a couple of early prototypes.



A couple of very early impressions of a what a CC badge could look like.

timeline

This is the timeline that I will stick to when working on CC Vocabulary and CC Chooser before, during and after GSoC 2019. This timeline includes periods before the programme begins and well after it ends.

pre-GSoC period | community bonding

Interact with the mentors of the project and set up feedback loops.
Continue to refine the plans for the project in consultation with the mentors.
Learn Vue which I am relatively inexperienced in.
Start working on the project, which could easily benefit from a head-start before the GSoC period.
Get involved with the community which, after all, is what this period is for.

phase 1: CC Vocabulary initialisation

week 1-2 | May 27 - June 10

- Setup the **infrastructure** for the project, involving Docker, scripts and code structure.
- Setup the project tracking tooling such as Trello or GitHub projects.
- **Initialise the Vue and Storybook project** from within Docker.

week 3-4 | June 11 - June 24

- Design and develop the components for the **header and footer** which are the least variable in the ecosystem.
- Get reviews and refine them according to the review in order to reduce iterations on future components.
- Create stories for **all use-cases** of the components.
- This fortnight would get me in **the flow**, for maximum productivity in writing future components.

phase 2: CC Vocabulary components

week 5-6 | June 25 - July 8

- Design and develop the component **button** which is the most diverse component in any design system on account of the sheer number of knobs that can be turned on the component.
- Design and develop the component **card** which also encompasses segment, container and divider.
- *If time permits:*
  Design and develop the component **modal** in several variations such as confirmation, alert and notification.
- Create stories for **all variations** of these components.

week 7-8 | July 9 - July 22

- Design and develop the components for **various form elements** such as input fields for various data types and checkboxes, radio buttons and dropdowns.
- Design and develop the component **insignia**, that includes both license logos as well as CC icons.
- *If time permits:*
  Design and develop the component **message** that includes both in-situ messages and toast messages.
- Create stories for **all variations** of these components.

phase 3: CC Chooser revamp

week 9-10 | July 23 - August 5

Create the flow for creators who **are proficient** in the process of licensing their creations.
Since this flow involves very few differences from the current version, this is also the period where most of the element common to both flows would be covered.

week 11-12 | August 6 - August 19

Create the flow for creators who **are new and require help** in the process of licensing their creations.
This flow involves a major overhaul of the licensing tool but is made easier with shared components from the proficient flow.

week 13 | August 20 - August 26

Buffer period to catch up on any pending work, else:

- Make bug fixes and code quality refactorings.
- Add any small features that might've been overlooked.
- Add easter eggs as a part of the **Reward and delight users of CC licenses** project.

post-GSoC period | finishing touches

Continue to work on the project, finishing items that have been put on an if condition in the last two weeks.

Improve the codebase in aspects that will only come to light when integrations are getting merged. Write Sphinx and Swagger documentation for the project. Possibly, even enter the project into next year's Google Season of Docs. 😉

future | long-term commitment

Wordpress-compatibility will be a major part of the long-term goals of the project. Since a lot of CC sites are built with WordPress, writing a new WordPress theme whose components utilise the CC Vocabulary theme would be a very crucial follow-up to the project. I do have some WordPress theming experience so this is more of an exercise as compared to a learning experience. Since the project is already too vast to be able to incorporate the WordPress project into the timeline, I will have to put it for later, which is not a way to skimp on it at all. I promise to take this up after GSoC.

In the near future, I will continue to add components to the library as three months are clearly not enough to write all the components that CC projects require. In the distant future, I would love to work with Creative Commons as they explore a new age of multimedia creation and sharing.

I'd also love to mentor newcomer contributors to the project in any way I can.

motivation

My primary motivation to apply for my first GSoC is to get started with open-source in a way I have yet to experience. I have utilised a lot of open-source technologies in my projects and have developed and open-sourced a lot of software. But I haven't had a chance to work on software that has **a tangible effect on millions of people** around the globe. The prospect of developing software actually capable of making a difference means a lot to me and I would very much like to be able to give back to the community in some way.

I have used CC-licensed media from Wikimedia Commons extensively in the past and had a faint idea of what Creative Commons licenses (specifically CC0 and CC-BY) stood for. I did however not know about Creative Common's open-source software projects. When the list of participating organisations for GSoC was announced, Creative Commons had a familiar name and a familiar technology stack. I decided to contribute to it as a starting point in open-source. I have been contributing to Creative Commons since then and my journey so far with has been a great one. I am fortunate to have met this vibrant community of so many motivated developers and creators. Imbibing such culture into my own projects is a challenge I want to take on.

Every pull request I have made to Creative Commons has taught me something new and interesting be it Vue, Jest, Travis, Pillow, ElasticSearch or OpenSearch, which was only possible because of the constant presence and feedback on issues and pull requests. Apart from development, I got to learn more about how creative licenses work and even got to learn about EXIF data in photography! Even before GSoC has begun, I have learned so much, which only makes me look forward to this endeavour and the learnings it brings.

continued participation

I am very new to Creative Commons but the rate of growth in my own knowledge and confidence as a developer that I have experienced here is enough to keep driving me to contribute more and more to the projects in the future. I have seen that my contributions to Creative Commons make me a better developer!

The bonding with the wonderful, vibrant and diverse community, the guidance of the mentors and the constant experience of learning and building an impactful application would motivate me to work here well past the timeline of GSoC for the months and years to come. I am not sure how open Creative Commons is to outside mentors, as I haven't met any so far, but if you are I'd love to even mentor new contributors to the project in the future.

availability

I have a final exam that spans the last week of April and the first week of May. I will have to divert my focus from the project towards the exams. That is my only engagement that overlaps with the GSoC period.

My vacations start from the 7th of May (although my college membership extends till the 30th of June) and since this is my final year there is no tuition to get back to after the summer. Beginning there, I do not have any time-consuming engagements (even if I did, coding is how I unwind) and will easily be able to devote **40-50 hours per week** to the project. I have very successfully worked on and deployed Omniport *as a student*, so I can assure you that I will have no trouble managing time.

status updates

I shall keep my status posted to my mentors and other concerned parties no less frequently than on a weekly basis so that there is transparency in the adherence to the schedule described in the timeline. I am, quite literally, always available for communication via the channels mentioned in the section me, at a glance.

me and the commons

why me?

Not intending to toot my own horn, but I am

- skilled with years of experience in diverse fields. I have working knowledge of all (and in-depth of most) of the technologies the project requires
- committed to my field with an unwavering attention to detail. I have a passion to produce high quality software that is a joy to use and to develop further, documentation included
- driven to give my 100% to any and every project I participate in. If a project aligns with my ideology, which this does, I put my heart into it, working non-stop and without distractions

You can see my development skills, my experience and my projects.

In all fairness, I *might not* be the best person to do this work. That is for you to decide. All I can say is this: if you can find a contributor more dedicated, experienced, committed and interested than I am, it would be understandable to choose their proposal for GSoC over me, and I would be fine with that. Meanwhile, I'll try to make finding such a person difficult by upping my own dedication, experience and commitment and interest.

why Creative Commons?

As I mentioned in the section motivation, working on Creative Commons projects gives me a chance to learn something new every day and that is a feeling that I have never experienced before in my time as a single developer. The time I have spent here has been the most learning-packed and exciting of my life. I found myself checking GitHub hourly for reviews on my PRs and constantly going through documentation of libraries I had never heard of before. This is a feeling I do not want to give up so soon.

This may sound cliché but I have also come to appreciate the role of CC licenses especially CC0 and CC-BY in the world of creative content. As a person who appreciates remix culture (especially in music and code), it gives me a feeling of pride to work with an organisation that develops licenses that make it possible. In many ways Creative Commons fosters open-creativity much like GSoC fosters open-source and it would be a privilege to be able to contribute to both causes with one project.

why CC Vocabulary and CC Chooser?

As for the project, I have experienced first-hand how important it is for frontend developers to have a consistent system of components before developing an application. At IMG we used to develop apps independently with designers and developers working on one project to be isolated from the designers and developers working on another. Because of this divide, we ended up duplicating a lot of code across projects and ended up with no cohesive experience in the apps. Formula One, our frontend component framework, was designed to eliminate this problem. I am convinced that **CC Vocabulary would have a similar effect on frontend development at Creative Commons** with every new project and every new web page.

As for CC Chooser, I know from experience that it is the first point of contact for a creator wishing to license creations into the Commons. **A revamped CC Chooser would provide a great experience for a new creator**, both in terms of understanding the license being chosen and getting help where needed, and would go a long way in increasing the proliferation of CC licenses making it an ideal first candidate for receiving the benefits provided by CC Vocabulary components.

I would also be partially working on other insights from the CC usability survey making the commons a much more future-looking organisation. Seeing as this is a very groundwork-laying project, it is inevitable that I would have to collaborate with other developers to integrate the system into their own code.

questionnaire

A lot of these questions have been answered in detail in various sections above, so I'll just link to them here.

Why are you the best person to do this work?

Kindly refer to the why me? section.

Why are you interested in Creative Commons and this project in particular?

Kindly refer to the why Creative Commons? and why CC Vocabulary and CC Chooser? sections.

What programming languages, version control systems, and other developer tools are you familiar with?

Kindly refer to the development skills section.

Do you have any other commitments during the GSoC period?

Kindly refer to the availability section.

Are you comfortable working closely with a mentor in English?

My native language is not English, it is Hindi. However, I am more than comfortable to converse with the mentors in English. I am fluent in reading, speaking, writing and comprehending English, at instances more so than I am in Hindi. This proposal should be a testament to that.

Have you worked remotely before?

No, I have not. But I do have some experience in coordinating work across geographies because the kind of work IMG does requires one to be able to solve issues, on production, from anywhere, at any time.

I am also familiar with video conferencing tools like Skype and Hangouts and team collaboration software like Trello and GitHub Projects so working remotely should not pose any difficulty. I also am at my most productive late at night, so my work hours would largely coincide with the work hours of mentors' west of the Atlantic.

It's a win-win situation.

thanks

This proposal was made possible because of many reviews and constant feedback from Kriti Godey, and mentors Hugo Solar and Timid Robot Zehta. I am thankful to them for their wonderful feedback and constructive criticism. I would also like to thank Alden Page and Breno Ferreira for their help in getting me started contributing to CC Search.

licenses

*~~~ fin. ~~~*