

Google: Usage & Reuse Metrics Dashboard for CC Search

Project Synopsis

I will design & build a user-friendly dashboard which will use google analytic tools to understand user's data and help Creative Commons, their current and potential partners. Creative Commons and their partners can use the dashboard to easily find relative information about their users' behavior. The data can also help their partners better understand how much impact their presence is having on the creative common's ecosystem and identify contents that are least consumed by the users. The dashboard will use visual representation in order to show the pattern on a weekly, monthly and yearly basis. All these features aim to help Creative Commons and make them attract potential partners in a better way.

Project Goals

- Understand which data best represent the overall picture of client behavior
- Build a structural database in order to store data
- Design a user-friendly & aesthetically pleasing analytics UI dashboard.
- Implement an analytics UI dashboard
- Use google analytic to perform analytics of the data in order to glean insights
- Filter out data for catalog partners in order to help them understand how much impact their presence is making.
- Build a recommendation system [If time permits]

The Problem

- Lack of comprehensive understanding of user behavior due to data being scattered all over the places
- Catalog partners do not know how much impact their presence is making in the CC search ecosystem.
- Due to lack of proper analytics for previous partners CC is unable to share impact with potential partners which is hindering the company's ability to gain a lot of potential partners.

Project-Flow Breakdown

1. Deep understanding of the data.
2. Mockup of the analytics UI dashboard
3. Structure of the database
4. Wireframe of the back-end implementation of the dashboard

What do the data represent?

- Page view: The number of times a page load.
- Unique page view: Only counts for the first time a user browses each page.
- Sessions: The entire time someone browses a website.
- Average time on site: The average amount of time a user spends on the website.
- Average pages per session: The number of pages a user visits during each session
- Unique visitor: A person that visits a website at least once during the reporting period.
- Bounce rate: The percentage of visitors who exits the website after only viewing one page
- Conversion rate: Percentage of website visitors that complete desired actions
- % Exit: Percentage in which the current page is the last accessed page before leaving the website.
- Returning vs new visitors: Negative value represents loyal followers whereas positive value represents something needs to change in order to encourage visitors to come back.

Mockup of the analytics UI dashboard

The dashboard will have two different perspectives which will look every similar, however, different metrics being displayed on each.

1. Admin view: The admin will see a bird-eye view of the entire picture. In the admin view, the metrics will be the average of all the metrics calculated for the individual partner's table. The idea is to use the admin dashboard to give Creative Commons a macroscopic view of how their content is being consumed by the visitors and gauge the performance of their business model. This admin view can also be used to attract potential partners as any potential partner would want to know the performance of the company over a period.
2. Partner view: This dashboard would be tailored towards an individual partner's need as the partner would most likely want to know how much impact their presence is making on the Creative Commons ecosystem and find out any performance pitfalls which they will be able to use to further improve how their contents are viewed in the Creative Commons search engine.

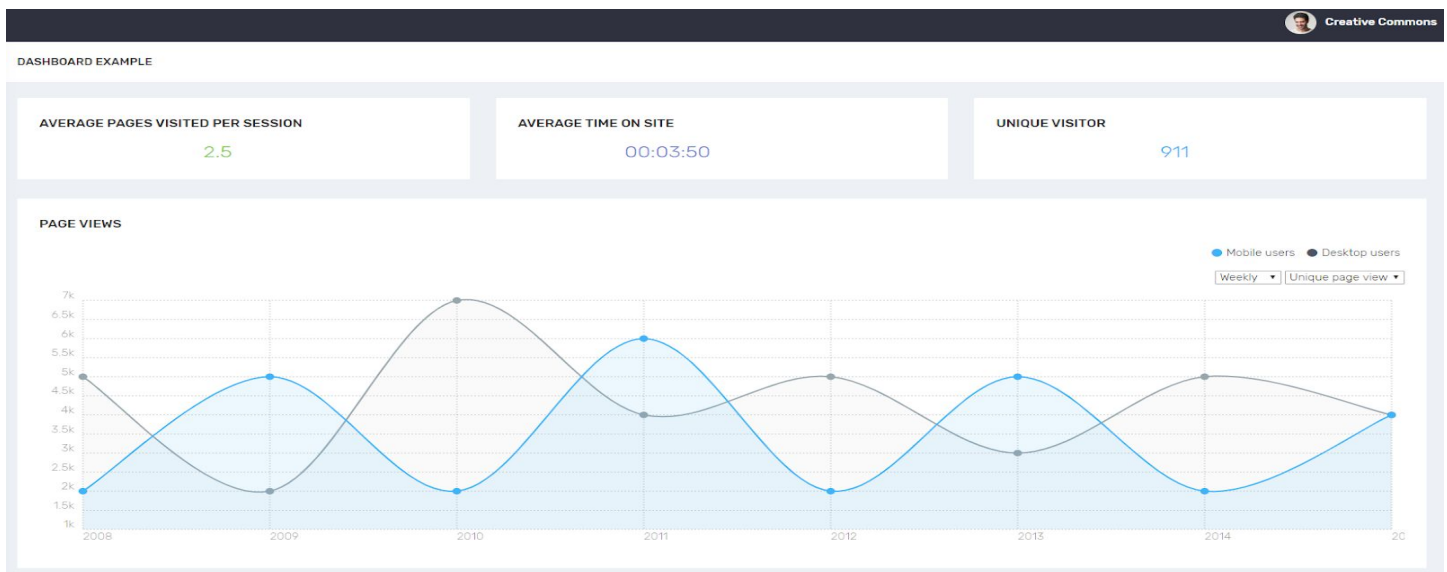
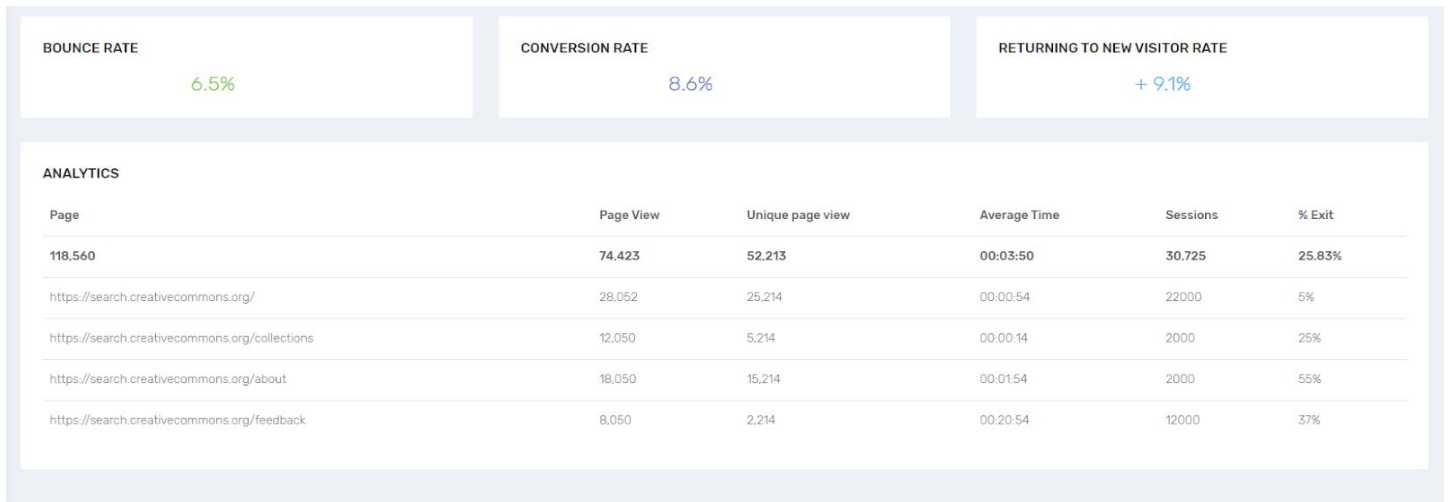


Figure: Dashboard Example view 1



2020 © Creative commons example dashboard

Figure: Dashboard Example view 2

Implementation details

Database design:

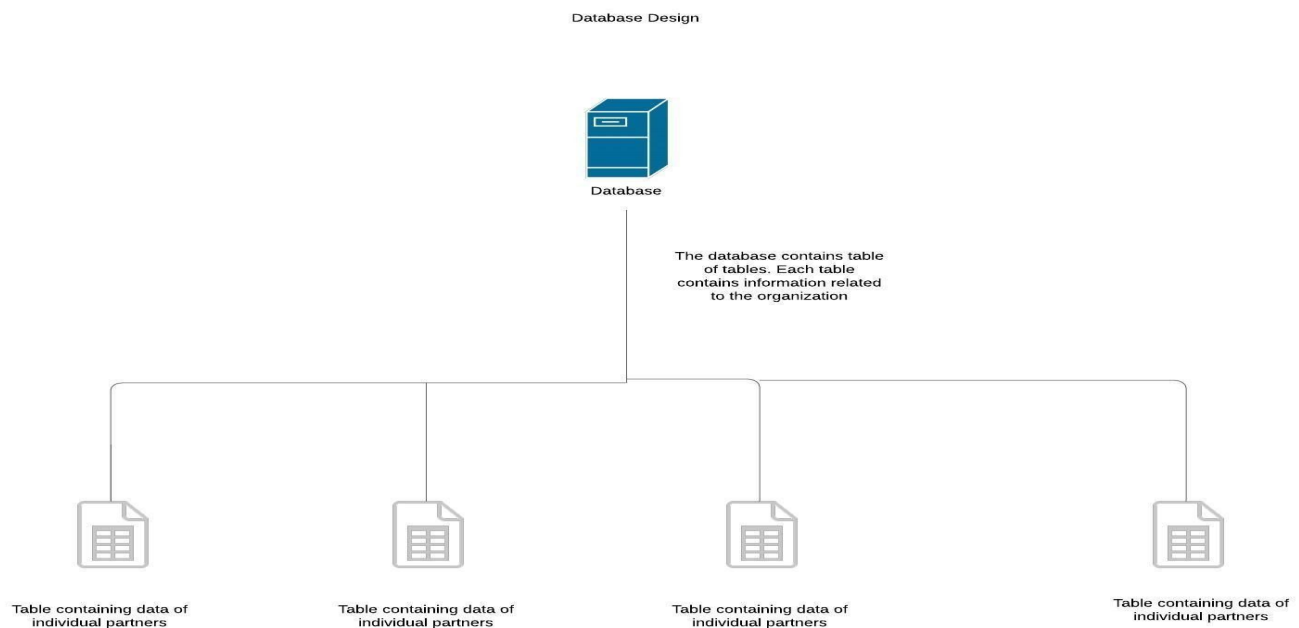


Figure: Database architecture

1. create a database in PostgreSQL: CREATE DATABASE analytics
2. connect to the database: \c analytics
3. Execute the following SQL code to implement a table of tables & individual table schema.

The database design consists of creating a database which contains a table of tables. The central table contains references to the individual partner's table and the individual partner's table contains all the data that can help the partner to better understand the behavior of users who have used their services. This database design is very neat allowing the use of dynamic SQL commands while allowing the programming a lot of flexibility.

- ON UPDATE CASCADE ON DELETE CASCADE: Any UPDATE **OR** DELETE to the parent, will result in automatic changes to the child.

```
CREATE TABLE Table0 (
tbl_id serial PRIMARY KEY,
sch text,
name text
);
```

```
CREATE TABLE Table1 (
    Table1 serial PRIMARY KEY,
    Page view INT,
    Unique page view INT,
    Sessions INT,
    Unique visitor INT,
    URL CHARACTER (30),
    Average pages per session DECIMAL (5, 2),
    Average time on site CHARACTER (30),
    Bounce rate INT,
    Conversion rate INT,
    % Exit INT,
    Returning vs new visitors INT,
    tbl_id integer REFERENCES Table0 (tbl_id) ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
Postgresql=# \dt+ Table0
Postgresql=# \dt+ Table1
```

Back-end:

I will be implementing several functions in order to construct the back end of the dashboard. The description of what each of this function will do and how it will be built is described below.

1. **Make_call**: This function will use python's request library and Google's Google Analytics Embed API to get data, for which I will need to acquire an API_KEY from Google. This function gets called for each individual partner in the creative common's ecosystem and populates them to a global array of HashMaps, where the name of the HashMap is the name of the partner. The global HashMap on the other hand is populated using a helper method called **hit_api** which populates the HashMap with all the relevant information.
2. **Hit_api**: This helper function stays inside the scope of the **make_call** function and takes in the individual partners json data as the parameters. It then filters out the relevant fields that are identified earlier and populates the HashMap. In the individual rows of the HashMap, the key is the name of the metric and the value is a list of tuples where each tuple has data along with the timestamps. The idea is to cache the data locally in order to avoid repeated number of calls to the Google API thus improving the efficiency and running time of the functions.
3. **Visual_graph**: This function takes in two parameters the type of data that needs to be displayed and the timeframe of the data to be shown. This function uses calls to a JavaScript library called **chart.js** to make responsive visual graphs which runs in the Django web application.

Example code:

```

class Page_view(models.Model):
    views = models.PositiveIntegerField()

class Unique_Page_view (models.Model):
    views = models.PositiveIntegerField()

from django.shortcuts import render
from mysite.core.models import Page_view

def line_graph (request):
    years = []
    data = []

    queryset = Page_view.objects.order_by('-page_views')[:5]
    for view in queryset:
        years.append(view.time)
        data.append(view.values)

    return render(request, 'line_graph.html', {
        'years': years,
        'values': data,
    })

```

4. Metrics_displayer: This function takes a time frame as the only parameter. The function first sorts the values in HashMap in ascending order and appends data to a temporary array that is found within the certain time frame. The average values of all the data from the temporary variable is then dynamically updated in the individual metric divisions in the index.html page.
5. Central_controller_adminView: This function is the central controller for admin view, the idea is to use a loop to call the individual functions and appends their value in a temporary array and then calculate their average before calling the Metrics_displayer, Analytics_table_population and Visual_graph functions. This function also uses Event Listeners to understand which type of data and time the user wants and pass them as parameters of the above three functions.
6. Central_controller_PartnetView: This function is the central controller for the partner view but unlike the controller this function doesn't create a temporary array in order to calculate the running average. This function simply calls the Metrics_displayer, Analytics_table_population and Visual_graph functions. This function also uses Event Listeners to understand which type of data and time the user wants and pass them as parameters of the above three functions.
7. Analytics_table_population: This function is used to populate data to their respective fields in the HTML table created in the index.html page. This function takes in the metric data as inputs and selects their individual fields in the HTML table using the ID attribute, and then appends their values. This function also keeps track of the global running total for each metric which can be used by the Analytics_table_running_total method.
8. Analytics_table_running_total: This function is used to populate the running total for each field in the HTML table created in the index.html page. This function takes in the globally calculated running total metric data as inputs and selects their individual fields in the HTML table using the ID attribute, and then appends their values.
9. Make_table_for_new_partners: This function first connects to the postgresql server, finds all the items in the global array and creates a table for each HashMap as a child-parent relation. This function executes the schema1 SQL code to carry out this functionality.
10. Populate_table_for_individual_partners: This function first connects to the postgresql server, finds all the

HashMaps being created globally, and then executes the schema2 SQL code to append necessary information into the database for storage purposes. This function also establishes the parent-child relationship and the name of the table to be created is the same as the name of the HashMap

Research & References

<https://www.searchenginejournal.com/content-marketing-kpis/user-engagement-metrics/>
<https://digital.gov/pdf/sessions-unique-page-views.pdf>
<https://www.invespcro.com/blog/impact-of-page-on-site-conversion-rate/>
<https://www.impactbnd.com/blog/marketing-metrics-on-google-analytics>
<https://medium.com/@jonolave/users-visits-or-page-views-why-the-unit-of-analysis-matters-for-your-news-site-d99446bffb66>
<https://www.dummies.com/business/start-a-business/small-business-marketing/how-to-use-google-analytics-to-measure-unique-visitors-to-your-website/>
<https://stackoverflow.com/questions/8162597/creating-a-table-of-tables-in-postgresql-or-achieving-similar-functionality>
<https://www.chartjs.org/>
<https://developers.google.com/analytics/devguides/reporting/embed/v1/>
<https://ga-dev-tools.appspot.com/embed-api/third-party-visualizations/>
<https://simpleisbetterthancomplex.com/tutorial/2020/01/19/how-to-use-chart-js-with-django.html>
<https://stackoverflow.com/questions/29720449/get-json-from-google-analytics>

Timeline & Deliverables

Timeline	Deliverables
Pre-Internship	Get to know all the members of the creative cloud team, my mentors and familiarize myself with the creative common's web application & architecture.
Week 1	Collect feedback from team-members, rewrite implementation details if needed and complete the design of the front-end dashboard.
Week 2 - 3.5	Complete the implementation of the dashboard and report to the team about progress & feedbacks
Week 3.5-5	Complete the database architecture and report to the team about progress & feedbacks
Week 6	Test out the features that have already been implemented to check if they are functioning properly, talk about possible problems faced and rewrite implementation design if needed.
Week 7-12	Complete the full implementation of the back-end functionality and report to the team about progress & feedback every other week.
Week 13	Test functionality, write code documentation & fix bugs if necessary.
Post-Internship	Present the project, ask feedback from the team to help me identify my areas of weakness and also how I can help the year beyond the summer.

Questionnaire

Why are you interested in working with Creative Commons, out of all the mentoring organizations available?

Creative Commons is the only non-profit open-source mentoring organization in this year's Google Summer of code. The company goal is to give every person and organization in the world a free, simple, and standardized way to grant copyright permissions to creative and academic works. As well as access to build upon the work of others is something I believe in. When I was just learning how to code, I would read and try to replicate other's work that is open source. This helped me a lot to develop strong problem skills.

Why are you interested in working on this project in particular?

Out of all the projects that are listed on the project ideas for Creative Commons. I felt this project suits me the most given my background in Math & Computer Science. I will also be gaining real work experience architecting the dashboard, which is a very challenging task, but it will make me a better developer. I am a team player and given my soft-skills I believe I will be able to make an instant impact on the company by helping them better understand their user and attract a lot of more potential partners into their ecosystem.

Why are you well-suited to take on this project?

I have over a year of experience in building web applications in Python and JavaScript. I have also earned the best web design award at Rice University's hackathon. During my time as an intern at Smooth Fusion I had to learn about their platform, Site infinity, and familiarize myself with C#, .NET, Harvest in order to solve several of their data driven problems like calculating developer efficiency, if the developer's times are not in sync and also writing a custom mailer in Node.JS. I also work in a 36-hour project at Princeton University where I had to split a string into several action works and construct a parse tree like data structure to help automate the workflow. I believe I have a strong understanding of web technologies & data science. But what makes me the perfect fit for this project is my strong mathematical background which is needed to understand the data and identify which data will give us a better picture. I have taken a graduate level Data Science course as a Junior and have also worked at many machine learning projects at various hackathons. I believe my background combined with my willingness to learn make the perfect candidate for this project.

Do you have any other commitments during the internship period? Provide dates, such as holidays, when you will not be available.

No, I am not taking summer classes. I will be available for a minimum 40 hours per week.

If your native language is not English, are you comfortable working closely with a mentor in English?

Yes

Have you collaborated on a project remotely before?

Yes. I have competed at HackTech at CalTech virtually building a computer vision assistive technology application in Python.

Do you plan to keep contributing to Creative Commons after the internship period is complete?

Yes, even after Google code of Summer I want to be involved in the community and hopefully return next year as a Creative Commons mentor.

General Information

About Me

My name is Mostofa Adib Shakib, I am an international student from Dhaka, Bangladesh. I am a Presidential Scholar earning dual degrees in Computer Science & Mathematics with a minor in Electrical Engineering at Texas Tech University. I moved to the US in August 2017 and over the past 2.5 years, I have worked in four different firms as a Software Engineer Intern. I have won the best web design award at HackRice at Rice University in 2018 and won HackPrinceton's title's sponsor at Princeton University in 2019. I love to learn, be challenged, and I love to work in teams. I love to solve data driven problems where you need to think out of the box to understand another human's behavior and how large-scale applications are distributed to millions of people around the globe. I Co-Founded RaiderHacks out of my interest in competitive programming. This is Texas Tech University's competitive programming team. RaiderHacks (<https://techconnect.dsa.ttu.edu/organization/raiderhacks>) competes at various programming competitions like Google code Jam, collegiate hackathons, datathons, and ICPC. I also love to spend my weekend getaways hacking away at different hackathons while making lifelong friends or writing blogs (<https://mynameisadib.herokuapp.com/blog>)

Basic Information

Name & Contact Information			
Full name*	Mostofa Adib Shakib	Email address*	adibshakib@gmail.com
Gender/Pronouns*	Male	City, Country*	Lubbock, United States
Slack username*	adibshakib	Phone number*	
Postal address*		Emergency contact information*	
Websites & URLs			
GitHub*	https://github.com/mostofashakib	Homepage	https://mynameisadib.herokuapp.com/
Twitter		LinkedIn	https://www.linkedin.com/in/adibshakib/
Other code or work samples	https://devpost.com/MostofaAdibShakib		

Education			
Institution	Texas Tech University	Degree	Dual Bachelor of Science
Major	Computer Science & Mathematics	Graduation Year	May 2021
Courses Taken	Data Structures & Algorithms, Data Science, Operating Systems, Intro. To Proof, Statistics & Probability, Linear Algebra, Object-Oriented Programming, Computer Architecture, Advanced Calculus 2, Differential Equations 2, Linear System Analysis		

Skills

Skill name	Proficiency (1-5)	Where/how you've used this skill
Python	4	Over 3 prior internships, hackathons, academic
HTML/CSS	4	Over 4 prior internships & hackathons
PostgreSQL	3	1 prior internship & hackathons
Git	4	Over 4 prior internships & hackathons
JavaScript	4	Over 4 prior internships & hackathons
Web design	4	Over 4 prior internships & hackathons

Experience and Contributions

Brief Description	Relevant Links	Additional Notes
Smooth Fusion		Worked with data in order to develop scripts to automate the task of generating developer efficiency stats and emailing to the individual developers. To check if the employee's times are in sync in both Harvest & Jira and email the employee if they are not in sync.
HackRice	https://team-red-raider-website.herokuapp.com/	Won best web design award for building a website

NextGen Code		Worked with several client-face applications in React, JavaScript and fixed several bugs including dashboards
HackPrinceton	https://devpost.com/software/auto-flow	Worked extensively with Python and back-end development. Also worked with several microservices like Twilio
TamuHacks	https://team-red-raider.herokuapp.com/	Designed as well as developed the web application. The web application uses a database to store data.